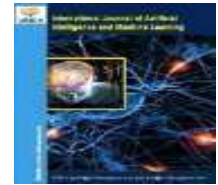




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

AI-Enabled Edge Intelligence for Real-Time IoT Analytics A Distributed Framework for Low-Latency Decision-Making and Resource Optimization

Mrs. Anuradha Gohad¹, Prof. Sweta Kahurke², Dr. Ghanshyam Shivcharan Nikhade³, Dr. Krishna Kumar⁴, Prof. Yamini Dwivedi⁵

¹Assistant professor, Jhulelal Institute of Technology, affiliated to RTMNU, Nagpur, CSE(AI&ML), Email id: anuradhagohad484@jitnagpur.edu.in, Orcid id: <https://orcid.org/0009-0006-5034-1163>

²Assistant Professor, Computer Science and Engineering, Artificial Intelligence and Data Science, Wireless Communication and Computing, Tulsiramji Gaikwad Patil College of Engineering and Technology, Nagpur, Orchid Id - <https://orcid.org/0009-0005-6073-7877>, Email Id - swetakahurke@gmail.com

³Assistant Professor, G H RAISONI SKILL TECH UNIVERSITY NAGPUR, COMPUTER SCIENCE AND ENGINEERING, CSE, Email id: ghanshyam.nikhade@ghrstu.edu.in, Orcid id: <https://orcid.org/0009-0001-3225-8757>

⁴Professor, Computer Science & Engineering, Image Processing and Machine Learning, Institute of Technology & Management, GIDA, Gorakhpur-273209 (Uttar Pradesh) INDIA, Email Id: kk_gkp@rediffmail.com, Orcid Id: 0009-0006-6119-3688

⁵Assistant Professor, RTMNU, MBA, IT analyst, Email id: Yaminiyush4029@gmail.com, Orcid id: <https://orcid.org/0009-0009-0915-9046>

ABSTRACT

The study investigated an AI-enabled distributed edge-intelligence framework for real-time Internet-of-Things (IoT) analytics focused on low-latency decision-making, resource optimisation, energy efficiency and quality of service (QoS) performance. In study, a dataset of 2,400 observations and 22 variables that characterize different IoT workloads, network characteristics, computational resource usage, inference properties, offloading behavior, and operational outcomes, and adopted a computational evaluation design. The analysis performed these comparisons using descriptive statistics, independent-samples t-tests, one-way ANOVA, correlation analysis, multiple linear regression, and five supervised machine-learning models. The AI-adaptive strategy not only significantly cut the end-to-end latency down from 78.00ms to 52.63ms, it also lowered energy consumption, utilization of CPU and memory, packet loss, queue delay and inference time. The resource efficiency improved from 46.78 to 59.49, and the decision correctness improved from 26.4% to 52.6%. The number of QoS violations dropped significantly from 49.4% with static processing to 11.6% with AI-adaptive orchestration. Orchestration mode, queue delay, inference time and network round-trip time (RTT) were identified as major factors that affect latency through regression analysis. Logistic Regression had the best ROC-AUC score of 0.639 among all classifiers tested. The results highlight the potential of leveraging AI-driven edge orchestration to enhance responsiveness, efficiency, and reliability in distributed IoT setups overall.

Keywords: Edge intelligence; Internet of Things; Real-time analytics; Resource optimization; Machine learning

1. Introduction

Internet of Things (IoT) devices have proliferated quickly, creating huge amounts of continuously generated data which necessitates timely processing, analysis and decision-making. The latency, bandwidth and responsiveness characteristics essential to real-time IoT applications often pose a challenge to traditional cloud-centric architectures. Edge computing alleviates these challenges by bringing computation and storage closer to data-producing devices, which helps to lower latency and boost responsiveness at the network edge (PremSankar et al., 2018; Yu et al., 2017). Distributed processing is also supported by fog and edge computing architecture, which enable computation to be delegated from compute-intensive IoT devices to nearby fog and edge nodes (Aazam et al., 2018). Over time, this distributed paradigm has developed towards the edge intelligence, in which artificial intelligence is embedded in edge infrastructures to aid adaptive analytics and autonomous decision-making (Zhou et al., 2019).

Even though this capability seems promising, real-time processing of IoT data is still hindered by the limitations of edge resources, the dynamic nature of workloads, network latency and bottlenecks, energy usage, and diverse device types and capabilities. It is therefore crucial to have efficient resource management to balance computation and communication with the quality of service in distributed environments (Bhajantri & S, 2020). Multi-access edge computing also adds advanced coordination challenges of workload placement, mobility, bandwidth availability, and latency-sensitive services (Filali et al., 2020). However, security and trust are also very important issues due to the nature of distributed fog and edge infrastructures, which involve exposing data and services throughout numerous devices and administrative domains (P. Zhang et al., 2018). Such difficulties emphasize the need to adopt dynamic task allocation and scheduling strategies in highly dynamic IoT systems. These difficulties show that static task allocation and fixed scheduling strategies are

inadequate for highly dynamic IoT systems.

AI and machine learning offer great potential as enablers for adaptive decision-making at the edge. Using machine learning to process continuous cyber-physical and IoT data streams can identify their patterns, predict the system condition, and allocate computing resources dynamically (Fei et al., 2019). Learning-based computation offloading has been shown to be able to determine efficient computing locations with varying network and resource environments (Dinh et al., 2018). Deep reinforcement learning has also been used to optimize task offloading and bandwidth allocation jointly to address dynamic task loads and reduce latency (Huang et al., 2019). The Deep Q-learning and reinforcement learning-based methods have also been applied to enhance the decisions of computation offloading and task migration in mobile edge computing (Wei et al., 2019; C. Zhang & Zheng, 2019). Furthermore, lightweight RL strategies can be used to decrease the computational load with minimal impact on the efficiency of offloading (Lu et al., 2020).

While there is already research that has been conducted to prove the significance of edge computing, computation offloading and intelligent resource allocation, most of the studies focus on one or two of these factors independently instead of measuring them in a holistic real-time IoT analytics system. In general edge-computing research has pointed to the necessity to consider the various factors of latency, resource usage and application-level requirements together through coordinated mechanisms (Khan et al., 2019). In vehicular and cloud-assisted edge systems, joint computation offloading and resource allocation can be effective with measurable performance gains, however, these are often application-specific (Zhao et al., 2019). Thus, an edge-intelligence, distributed framework is still needed to support the multi-faceted consideration of low-latency decision-making, resource efficiency, energy usage, and quality of service outcomes.

This study assesses a distributed edge-intelligence framework that uses Artificial Intelligence to provide real-time IoT analytics. The goals include understanding if adaptive AI-based orchestration has the potential to shorten end-to-end latency, allocate more efficiently across computational and memory resources, save energy, increase decision accuracy, and minimize quality-of-service violations relative to a static edge-processing approach.

The study brings together an integrated analytical view of the use of AI-driven orchestration, real-time decision quality, task offloading, and resource optimization in a distributed IoT environment. It also compares adaptive and static processing strategies, helping to understand the potential of intelligent edge systems to enhance response time, efficiency, and reliability in latency-critical IoT applications.

2. Materials and Methods

2.1 Research Design

Overall, this study has adopted a computational evaluation approach to evaluate an AI-assisted distributed edge-intelligence framework for real-time IoT analytics. The analysis compared the performance of AI-adaptive orchestration with the static edge processing approach to see if intelligent resource management delivers better latency, energy efficiency, decision accuracy, and quality-of-service performance.

2.2 Dataset Description

The dataset consisted of 22 variables and 2,400 observations related to the characteristics of IoT workload, network, computing, energy and decisions. It consisted of 12 edge nodes and four workloads: Sensor Fusion, Video Analytics, Anomaly Detection and Predictive Maintenance. The main experimental variable was *orchestration_mode*, comprising 1,220 AI-adaptive and 1,180 static-edge observations. Task size, input rate, CPU/memory usage, bandwidth, network round-trip time, packet loss, queue delay, inference time, offload ratio, energy usage, resource efficiency, model confidence, decision score, end-to-end latency, decision correctness, and violation of QoS were the key variables considered.

2.3 Distributed Edge Intelligence Framework

The proposed framework was considered as interactions between the IoT devices, edge nodes, AI decision components, and resource-management mechanisms. IoT workloads were executed at the edge nodes, depending on the available computing and network resources. The AI-adaptive mode dynamically changed the processing and offloading decisions based on workload intensity, CPU and memory usage, bandwidth, latency, and condition of queues; the static-edge mode used a static processing strategy. The evaluation metrics were latency, resource efficiency, energy consumption, decision correctness, and the outcomes of QoS.

2.4 Data Preprocessing

Data was verified for missing values, duplicates, data types, and abnormal ranges. There were no missing values or duplicate observations found. Predictive modelling did not include identifiers like *record_id*, *device_id*, and *edge_node_id*. Categorical variables (including orchestration mode and workload type) were one-hot encoded, scale-sensitive variables were standardized. The data were split into 80% training and 20% testing sets using a stratified sampling procedure.

2.5 Machine Learning Models

The following supervised learning models were compared: Logistic Regression, Decision Tree, Random Forest, XGBoost, Support Vector Machine. The decision_correct classification target was chosen as the main classification target, with the qos_violation analysis being a secondary outcome. The factors that were identified as predictors included workload characteristics, network conditions, resource utilization, inference performance, offloading behaviour, energy consumption, and orchestration mode. These models were chosen to make comparisons between linear, tree-based, ensemble, boosting, and margin-based classification methods

2.6 Performance and Resource Optimization Measures

The primary performance metrics included were end-to-end latency, CPU usage, memory usage, energy consumption, resource efficiency, decision accuracy, and rate of violation of the QoS metric. The primary measure of responsiveness was end-to-end latency, and the computational load was measured as CPU and memory usage. Task efficiency was measured by energy per task and resource_efficiency_score was the efficiency of resource usage. Operational reliability was tested in terms of decision correctness and violation of QoS.

2.7 Statistical Analysis

Continuous and categorical data were summarized using descriptive statistics. Pearson correlation analysis was used to explore the relationships between latency, network conditions, resource consumption, inference time, energy consumption and energy efficiency. The key performance indicators were compared between AI-adaptive and static-edge processing using independent-samples t-tests. When appropriate, differences across workload types were examined utilizing one-way ANOVA. A multiple linear regression was used to find the factors that can predict the end-to-end latency. Determination of statistical significance was done at $p < .05$.

2.8 Model Evaluation

Testing was used to evaluate the predictive performance in terms of accuracy, precision, recall, F1 score and ROC AUC on the testing dataset. A confusion matrix was applied to analyze the classification of true positives, true negatives, false positives, and false negatives. Model performance was assessed by several classification metrics, instead of accuracy only, due to unbalanced outcome classes.

2.9 Overall Evaluation Strategy

The final analysis combined the results of the statistical and machine learning analysis to determine if using AI adaptive orchestration was more effective at processing data with low latency, improving resource utilization, conserving energy, making better decisions, and delivering better QoS performance than static edge processing. This approach led to direct matching between the research title, the variables in the datasets and the proposed edge-intelligence framework and analytical goals.

3. Results

3.1 Descriptive Characteristics of the Edge-IoT Workloads

The 2,400 observations come from four different, heterogeneous IoT workload categories: Sensor Fusion ($n = 726$), Video Analytics ($n = 599$), Anomaly Detection ($n = 595$), and Predictive Maintenance ($n = 480$). Significant variations were seen in task size, message rate, processing latency and energy usage among the various workloads. The average task size (849.88 KB), end-to-end latency (85.92 ms), and energy consumption (4.33 J/task) were the highest for Video Analytics. On the other hand, the highest input rate (95.00 messages/s) was obtained by Sensor Fusion, whereas relatively low latency (54.41 ms) and energy consumption (2.07 J) were obtained. The video analytics had the lowest mean efficiency score (40.28), while Sensor Fusion (59.23) and Predictive Maintenance (59.13) had the highest. As shown in Table 1, workload complexity was associated with clear differences in latency, energy consumption, and resource efficiency.

Table 1. Descriptive Characteristics of IoT Workloads

Workload type	n	Task size (KB)	Input rate (msg/s)	Latency (ms)	Energy/task (J)	Resource efficiency
Sensor Fusion	726	241.11	95.00	54.41	2.07	59.23
Anomaly Detection	595	420.66	70.67	63.92	2.73	54.24
Predictive Maintenance	480	322.64	60.48	56.76	2.30	59.13
Video Analytics	599	849.88	54.73	85.92	4.33	40.28

3.2 Comparison of AI-Adaptive and Static Edge Processing

AI-adaptive orchestration strategy showed significantly reduced end-to-end latency compared to static edge processing. Under AI-adaptive orchestration, the mean latency was 52.63ms, whereas the mean latency under static processing was 78.00ms, which is about 32.5% less. These differences were significant with $t = -42.13$, $p < .001$. AI-adaptive processing also provided a lower mean (M) inference time (22.89 ms) than static-edge processing (29.24 ms), $t = -20.95$, $p < .001$. These results show that adaptive orchestration evidently gave a responsiveness benefit in the controlled distributed IoT environment. As shown in Figure 1, end-to-end latency has a significant decrease when using AI-adaptive orchestration versus static-edge processing.

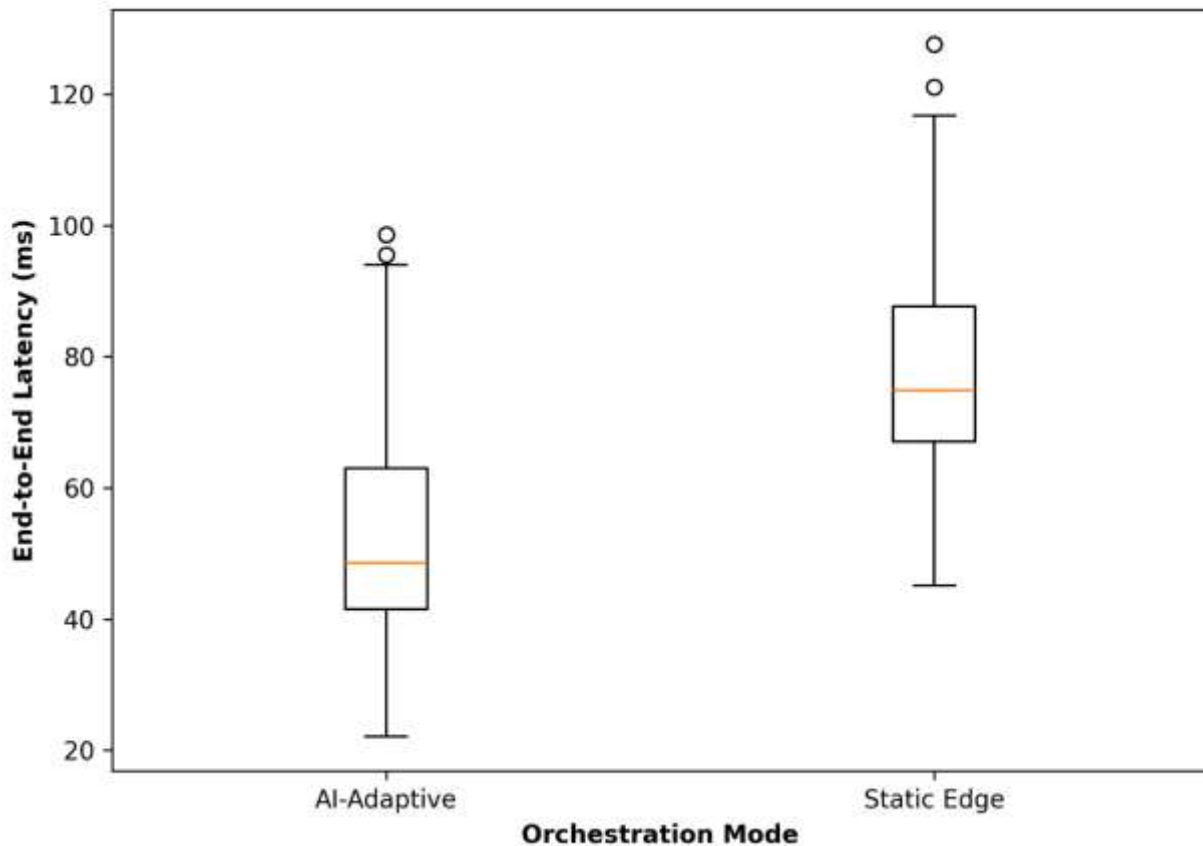


Figure 1. End-to-End Latency Distribution Under AI-Adaptive and Static Edge Processing

3.3 Network and Computing Resource Utilization

There were some significant differences between the two orchestration strategies in terms of network and computational conditions. Available bandwidth and network round-trip time, packet loss, edge-queue delay, CPU utilization and memory utilization were associated with AI-adaptive processing. The largest relative difference was edge-queue delay (20.44ms versus 13.68ms). The CPU utilization dropped to 72.85%, and Memory utilization dropped to 58.31%, from 80.44% and 62.49% respectively. All comparisons were significant at $p < .001$. Table 2 shows that the AI-adaptive strategy has better network and computing conditions than the static-edge strategy.

Table 2. Network and Computing Resource Utilization by Orchestration Mode

Measure	AI-Adaptive	Static Edge	t	p
Bandwidth (Mbps)	92.29	85.31	9.46	<.001
Network RTT (ms)	14.70	17.06	-9.28	<.001
Packet loss (%)	0.19	0.29	-15.86	<.001
Edge queue (ms)	13.68	20.44	-62.18	<.001
CPU utilization (%)	72.85	80.44	-18.06	<.001
Memory utilization (%)	58.31	62.49	-9.16	<.001

3.4 End-to-End Latency and Inference Performance

There was also a wide variance in processing time for different IoT workloads. The ANOVA test results indicated that there were significant differences between the four types of workloads with respect to end-to-end latency, $F(3, 2396) = 566.24$, $p < .001$. The latency also varied significantly between the workload

categories, with Video Analytics reporting an average of 85.92 ms, Anomaly Detection 63.92 ms, Predictive Maintenance 56.76 ms and Sensor Fusion 54.41 ms. The average latency was also significantly different between workload categories, $F(3, 2396) = 2098.26$, $p < .001$. This higher processing need of Video Analytics was therefore taken into account for both the inference time and overall response time. Video Analytics had the highest processing delay, whereas Sensor Fusion and Predictive Maintenance had relatively short response times, as seen in Figure 2.

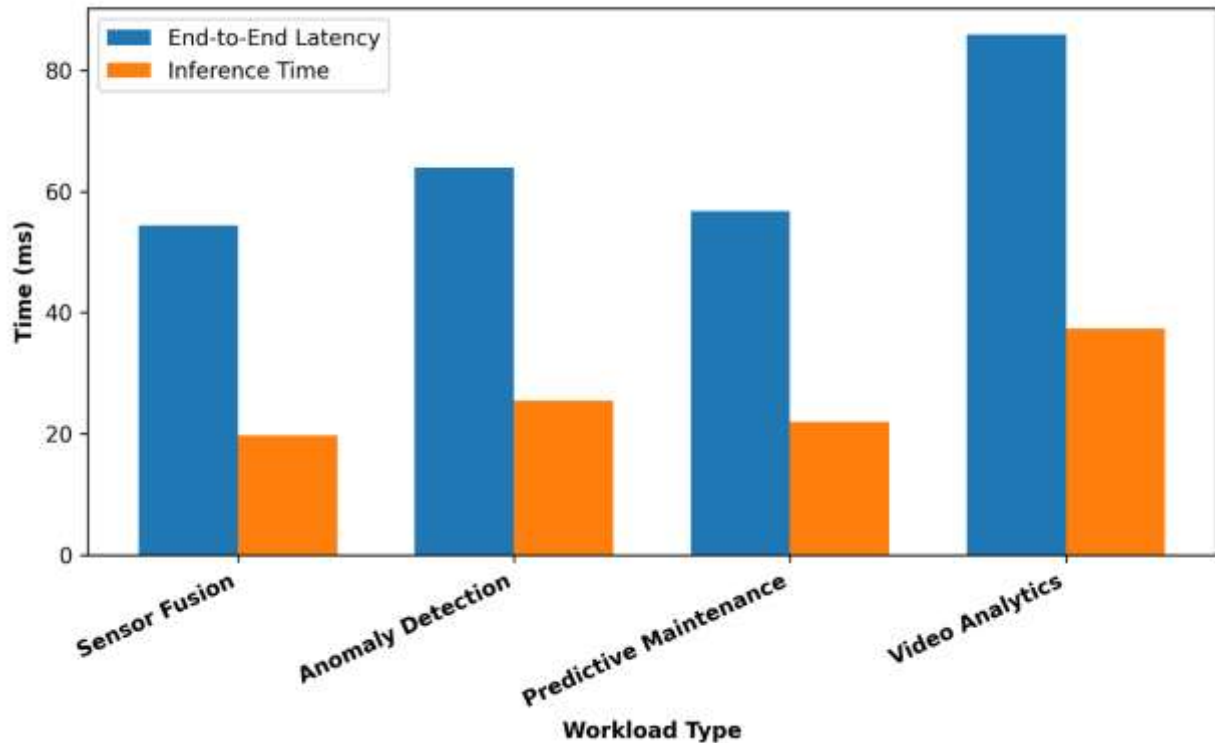


Figure 2. Mean End-to-End Latency and Inference Time Across IoT Workloads

3.5 Energy Consumption and Resource Efficiency

The AI-adaptive approach demonstrated evident benefits in the optimization of energy and resources. The mean energy consumed decreased from 3.10 J/task under static processing to 2.60 J under AI-adaptive orchestration, $t = -13.17$, $p = .001$. Resource efficiency increased from 46.78 to 59.49, $t = 32.88$, $p < .001$. Higher model confidence, decision scores and task-offloading rates were also generated by the AI-adaptive processing. Under adaptive orchestration, the mean offload ratio was 78.08% whereas under the edge processing of the edge, there was a ratio of 66.96%. As Table 3 shows, adaptive edge orchestration actively minimized energy demands and enhanced resource and decision-related performance.

Table 3. Energy, Resource, and Decision Performance by Orchestration Mode

Measure	AI-Adaptive	Static Edge	t	p
Energy per task (J)	2.596	3.102	-13.17	<.001
Resource efficiency score	59.494	46.782	32.88	<.001
Model confidence	0.831	0.762	40.70	<.001
Decision score	0.558	0.437	46.79	<.001
Offload ratio (%)	78.078	66.957	34.73	<.001

3.6 Factors Affecting Real-Time Decision Performance

Multiple linear regression was performed to determine the operational factors that affect end-to-end latency. The overall model explained 97.3% of the variance in latency, adjusted $R^2 = .973$, $F = 5030.43$, $p < .001$. The most significant positive standardized associations with latency were static-edge orchestration ($\beta = .462$), inference time ($\beta = .413$), network round-trip time ($\beta = .328$) and edge-queue delay ($\beta = .219$). The CPU utilization also had a positive involvement ($\beta = .076$), and minor contributions were seen for packet loss ($\beta = .021$) and bandwidth ($\beta = -.009$). A slight decrease in latency was observed with higher bandwidth. As seen in Figure 3, the major factors with real-time processing latency are orchestration mode, inference time, network RTT and queue delay.

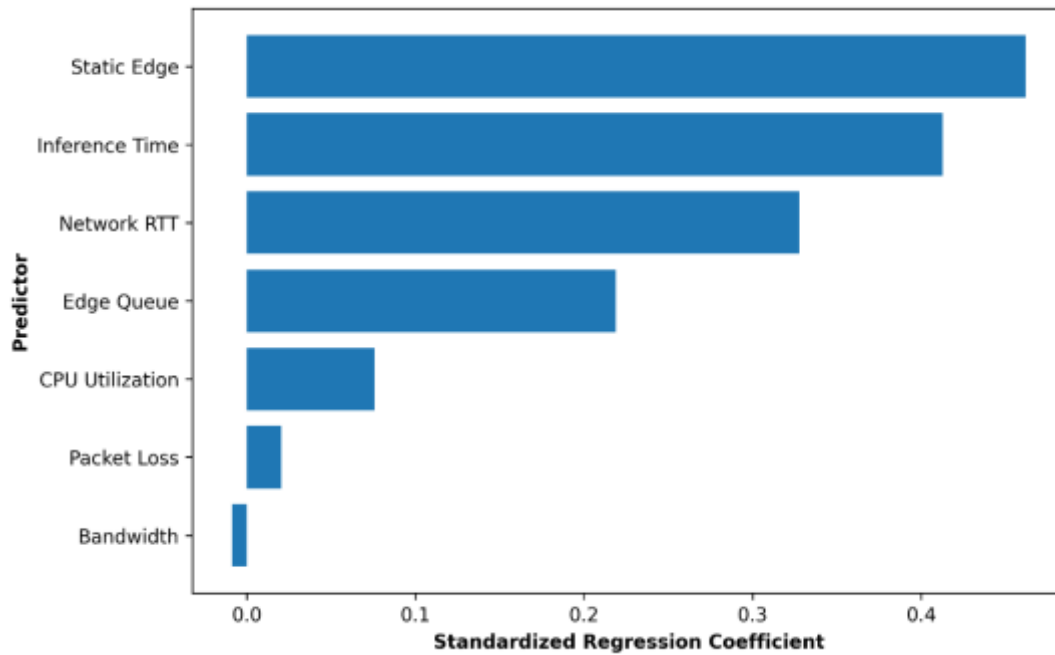


Figure 3. Standardized Regression Coefficients for Significant Predictors of End-to-End Latency

3.7 Machine Learning Model Performance

The five classification algorithms performed moderately well in predicting decision_correct on the independent 20% test set. Logistic Regression had the best ROC-AUC (0.639) and accuracy (0.625). Support Vector Machine had similar performance (ROC-AUC 0.630, accuracy 0.617). The random forest and the XGBoost gave the highest accuracy, 0.598 and 0.602 respectively and the Decision Tree was the worst with 0.522 ROC-AUC. Logistic Regression achieved the highest overall balance among the models tested, in terms of discrimination and classification performance. Table 4 indicates that Logistic Regression and Support Vector Machine had the best predictive discrimination, but the classification performance was moderate in all models.

Table 4. Machine Learning Performance for Prediction of Decision Correctness

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.625	0.536	0.424	0.474	0.639
Support Vector Machine	0.617	0.522	0.429	0.471	0.630
Random Forest	0.598	0.492	0.340	0.402	0.591
XGBoost	0.602	0.500	0.356	0.416	0.586
Decision Tree	0.540	0.423	0.435	0.429	0.522

3.8 QoS Violation and Decision Accuracy Analysis

The last comparison showed that there were significant differences in reliability in operation between modes of orchestration. The AI-adaptive processing had a decision-correct rate of 52.6 as compared to 26.4 when using the static-edge processing. On the other hand, the violation of QoS in 11.6% of AI-adaptive observations and 49.4% of static-edge observations. The average decision-correct rate was 39.7% and 30.2% of all observations had a QoS violation. The significantly reduced violation rate and increased percentage of correct judgments with AI-adaptive processing support the latency and efficiency results of the previous analyses. Figure 4 demonstrates the difference between the reliability profiles of the two processing strategies, where AI-adaptive orchestration has a higher accuracy in decision-making and significantly fewer QoS violation.

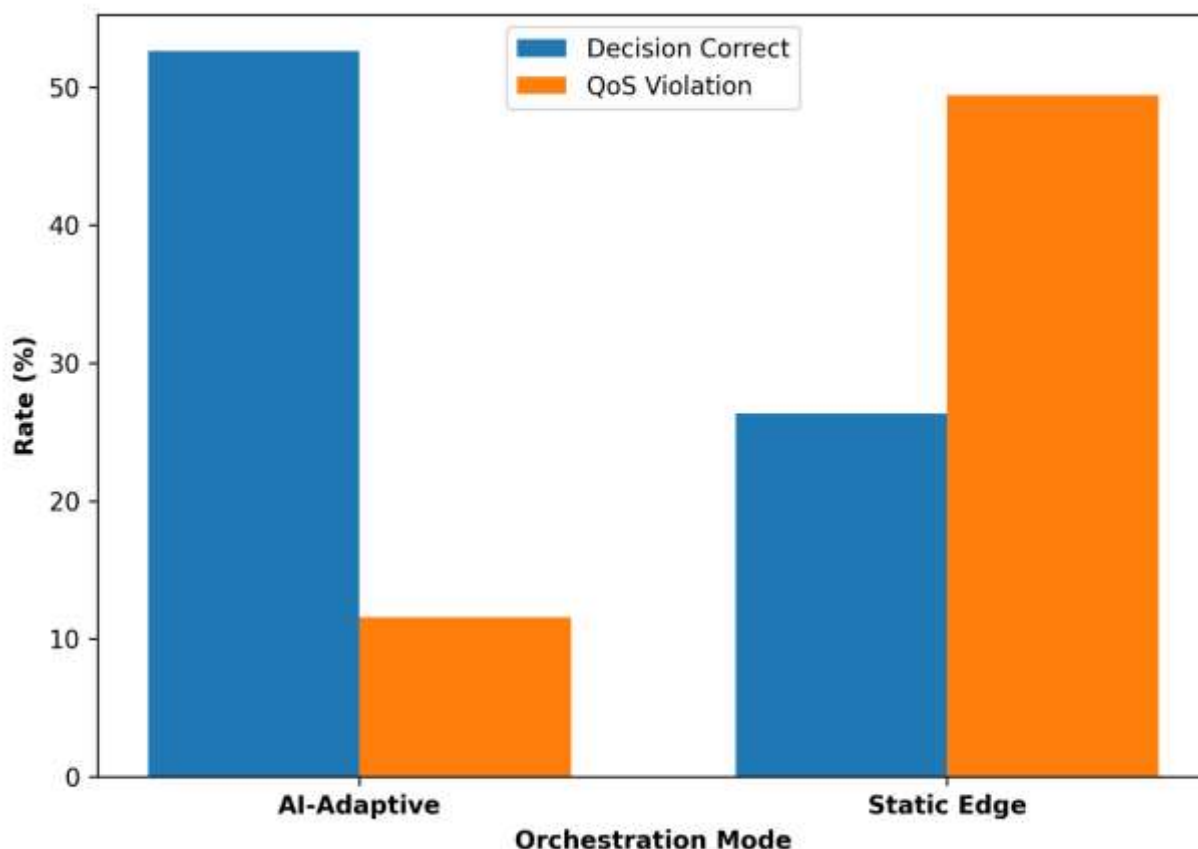


Figure 4. Decision Correctness and QoS Violation Rates by Orchestration Mode

4. Discussion

The results prove that adaptive orchestration based on AI has the potential to enhance the work of distributed edge-IoT systems in multiple functional aspects. The adaptive strategy had lower end-to-end latency, lower inference time, less CPU and memory usage, less energy use, better resource utilization, increased decision accuracy, and fewer quality-of-service violations compared to the static-edge strategy. These results contribute to the general understanding that intelligent resource management can be critical to ensure responsiveness and effectiveness in heterogeneous edge environments (Mijuskovic et al., 2021; Premkumar & Sigappi, 2021). The significant decrease in latency due to the AI-adaptive orchestration implies that responsiveness of the real-time IoT systems can be enhanced through dynamic decision-making. Multi-access edge computing is particularly designed to reduce transmission delays by shifting the workload to the locations of data sources, yet its performance is highly dependent on the allocation of workloads and resources (Pham et al., 2020). The current results align with the literature indicating that offloading computation to AI can decrease delay with dynamic choice of suitable processing locations in response to changing network conditions (Chen et al., 2021; Saeik et al., 2021).

Lower energy consumption and increased resource efficiency were also achieved in the adaptive strategy. This finding underscores the need to collaboratively coordinate the computation load, communication capacity, and task placement instead of optimizing these aspects individually. One of the most critical issues in edge computing has been found to be distributed resource scheduling due to the variability of workloads that can rapidly cause bottlenecks and energy inefficiency (Sahni et al., 2022). Deep reinforcement learning methods have also demonstrated the ability of intelligent scheduling to enhance resource allocation and minimize unnecessary processing overhead (Xue et al., 2022).

The AI-adaptive mode resulted in a greater number of correct decisions and much lower QoS breaches, indicating that effective resource distribution have a direct effect on operational reliability. The edge case of machine learning allows making inferences and responding more quickly to the evolving conditions of a system, thus minimizing reliance on remote cloud resources (Murshed et al., 2022). It has also been demonstrated that workload-conscious reinforcement learning can enhance scheduling choices in dynamic edge settings, especially when there is a time-dependent fluctuation in resource demand (Zheng et al., 2022). The findings are in line with the previous literature that highlights the importance of intelligent offloading, scheduling, and distributed learning in resource-constrained IoT systems. Studies of federated learning have shown that distributed intelligence is capable of facilitating collaborative model enhancement and restricting centralized data flow (Nguyen et al., 2021). The same benefits have been documented with edge-based

federated learning, especially in architectures where privacy, bandwidth, and device limitations need to be traded off in real time (Abreha et al., 2022). Nevertheless, deploying it on resource-constrained IoT devices is still hard due to the cost of communication and computation that can restrict scalability (Imteaj et al., 2021). The results have practical implications for applications that are sensitive to latency, such as industrial monitoring, intelligent transportation, predictive maintenance and real-time detection of anomalies. The edge platforms may be able to redistribute workloads, allocate bandwidth and make offloading decisions dynamically based on the local operating conditions with adaptive orchestration. These capabilities become more relevant to the greater IoT-edge-cloud continuum, wherein pervasive AI utilized to enable scalable, distributed, and context-aware services (Puthal et al., 2023).

The analysis based on a controlled study dataset rather than measurements collected from a deployed physical IoT infrastructure. Thus, physical impacts, like hardware malfunction, uncontrolled network congestion, mobile devices, security attacks and workload drift over the long term, were not represented exhaustively. Along with this, the tested machine-learning models offered a moderate predictive accuracy of the correctness of the decision, which implies that more time, context, or device-related variables can enhance the classification accuracy.

The proposed approach needs to be confirmed in future studies with the actual edge devices and heterogeneous IoT testbeds. Reinforcement learning, federated learning, adaptive task migration, privacy-conscious resource allocation, and online workload prediction can be included in further research. Testing in the context of larger-scale networks and dynamic traffic also be useful to assess the scalability and resilience of AI-enabled edge intelligence during deployment in realistic settings.

5. Conclusion

In this work, an AI-enabled distributed edge-intelligence framework was assessed based on real-time IoT analytics, focusing on low-latency decision-making, resource optimisation, energy efficiency, and quality-of-service performance. The results revealed that AI-adaptive orchestration clearly outworked the processing of the static-edges in all the key indicators of the system. The adaptive strategy resulted in a lower end-to-end latency and inference time, less CPU and memory usage, lower energy use, and better resource utilization. It also yielded greater correctness of decisions and significantly reduced violations of QoS suggesting that intelligent orchestration can enhance responsiveness and operational reliability in distributed IoT setups. Statistical analysis also revealed that orchestration mode, inference time, network round-trip time and queue delay significantly characterized end-to-end latency, and the machine-learning models demonstrated moderate predictive power on whether correct decisions were made. Logistic Regression and Support Vector Machine were the most successful overall discrimination classifiers tested. The findings indicate the importance of considering both AI-based decision mechanisms and distributed edge resource management in facilitating efficient real-time IoT processing. Though the analysis was based on controlled study data, the research technique provides a logical analytical foundation on further validation through physical edge devices, bigger heterogeneous networks, dynamic workloads, and more advanced adaptive methods like reinforcement and federated learning.

References

1. Aazam, M., Zeadally, S., & Harras, K. A. (2018). Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. *Future Generation Computer Systems*, 87, 278–289.
2. Abreha, H. G., Hayajneh, M., & Serhani, M. A. (2022). Federated learning in edge computing: A systematic survey. *Sensors*, 22(2), 450.
3. Bhajantri, L. B., & S, G. (2020). A comprehensive survey on resource management in internet of things. *Journal of Telecommunications and Information Technology*, (4), 27–43.
4. Dinh, T. Q., La, Q. D., Quek, T. Q., & Shin, H. (2018). Learning for computation offloading in mobile edge computing. *IEEE Transactions on Communications*, 66(12), 6353–6367.
5. Fei, X., Shah, N., Verba, N., Chao, K.-M., Sanchez-Anguix, V., Lewandowski, J., James, A., & Usman, Z. (2019). CPS data streams analytics based on machine learning for Cloud and Fog Computing: A survey. *Future Generation Computer Systems*, 90, 435–450.
6. Filali, A., Abouamar, A., Cherkaoui, S., Kobbane, A., & Guizani, M. (2020). Multi-access edge computing: A survey. *IEEE Access*, 8, 197017–197046.
7. Huang, L., Feng, X., Zhang, C., Qian, L., & Wu, Y. (2019). Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing. *Digital Communications and Networks*, 5(1), 10–17.
8. Imteaj, A., Thakker, U., Wang, S., Li, J., & Amini, M. H. (2021). A survey on federated learning for resource-constrained IoT devices. *IEEE Internet of Things Journal*, 9(1), 1–24.
9. Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems*, 97, 219–235.
10. Lu, H., Gu, C., Luo, F., Ding, W., & Liu, X. (2020). Optimization of lightweight task offloading strategy for

- mobile edge computing based on deep reinforcement learning. *Future Generation Computer Systems*, 102, 847–861.
11. Mijuskovic, A., Chiumento, A., Bemthuis, R., Aldea, A., & Havinga, P. (2021). Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification. *Sensors*, 21(5), 1832.
 12. Murshed, M. G. S., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., & Hussain, F. (2022). Machine Learning at the Network Edge: A Survey. *ACM Computing Surveys*, 54(8), 1–37. <https://doi.org/10.1145/3469029>
 13. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622–1658.
 14. Pham, Q.-V., Fang, F., Ha, V. N., Piran, M. J., Le, M., Le, L. B., Hwang, W.-J., & Ding, Z. (2020). A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8, 116974–117017.
 15. Premkumar, S., & Sigappi, A. N. (2021). A Survey of Architecture, Framework and Algorithms for Resource Management in Edge Computing. *EAI Endorsed Trans. Energy Web*, 8(33), e15.
 16. 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.
 17. Puthal, D., Mohanty, S. P., Yeun, C. Y., Damiani, E., & Pradhan, B. (2023). Pervasive AI for secure and scalable IoT-edge-cloud continuum: A big picture. 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), 566–573. <https://ieeexplore.ieee.org/abstract/document/10466879/>
 18. Sahni, Y., Cao, J., Yang, L., & Wang, S. (2022). Distributed resource scheduling in edge computing: Problems, solutions, and opportunities. *Computer Networks*, 219, 109430.
 19. Wei, Y., Wang, Z., Guo, D., & Yu, F. R. (2019). Deep q-learning based computation offloading strategy for mobile edge computing. *CMC-COMPUTERS MATERIALS & CONTINUA*, 59(1), 89–104.
 20. Xue, F., Hai, Q., Dong, T., Cui, Z., & Gong, Y. (2022). A deep reinforcement learning based hybrid algorithm for efficient resource scheduling in edge computing environment. *Information Sciences*, 608, 362–374.
 21. Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2017). A survey on the edge computing for the Internet of Things. *IEEE Access*, 6, 6900–6919.
 22. Zhang, C., & Zheng, Z. (2019). Task migration for mobile edge computing using deep reinforcement learning. *Future Generation Computer Systems*, 96, 111–118.
 23. Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in fog computing: A survey. *Future Generation Computer Systems*, 88, 16–27.
 24. Zhao, J., Li, Q., Gong, Y., & Zhang, K. (2019). Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. *IEEE Transactions on Vehicular Technology*, 68(8), 7944–7956.
 25. Zheng, T., Wan, J., Zhang, J., & Jiang, C. (2022). Deep Reinforcement Learning-Based Workload Scheduling for Edge Computing. *Journal of Cloud Computing*, 11(1), 3. <https://doi.org/10.1186/s13677-021-00276-0>
 26. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762.