INTEGRATING EDGE COMPUTING AND AI FOR REAL-TIME DATA PROCESSING IN .NET-BASED APPLICATIONS: A SCALABLE SOLUTION FOR LOW-LATENCY SYSTEMS

Hajira Rafi*[1], Azher Mohammed[2], Rajesh Kumar[3]

*[1,2]NET Developer,Master of Science in Information Technology, USA
[3]Devops/Cloud Engineer, Business Intelligence and Data Analytics, Westcliff University

*[1]hajira.9@yahoo.com, [2]azhermohammed97@gmail.com, [3]rajesh89rrk@gmail.com

Corresponding Author: *

Received	Revised	Accepted	Published
12 November, 2024	12 December, 2024	27 December, 2024	04 January, 2025

ABSTRACT

The integration of edge computing and artificial intelligence (AI) offers transformative potential for real-time data processing in latency-sensitive applications such as IoT, healthcare, and industrial automation. By processing data closer to its source, edge computing reduces latency and bandwidth usage, while AI provides predictive analytics and intelligent decision-making capabilities. This study presents a framework for integrating edge computing and AI within .NET-based web applications using ASP.NET Core and SignalR. The framework was evaluated under simulated IoT and industrial scenarios, demonstrating significant reductions in latency and bandwidth consumption, coupled with high AI inference accuracy and scalability. These findings underscore the potential of edge computing and AI in enabling efficient, intelligent, and real-time web applications.

Keywords:*Edge computing, artificial intelligence, real-time data processing, .NET applications, ASP.NET Core, IoT, latency reduction, scalability, predictive analytics, network optimization.*

INTRODUCTION

The rapid evolution of digital technologies has ushered in an era where real-time data processing is not only desirable but critical across various domains, including IoT, healthcare, smart cities, and industrial automation. Traditional centralized computing approaches often fail to meet the stringent requirements of low latency, high bandwidth efficiency, and intelligent decision-making demanded by these applications (Shi et al., 2016; Satyanarayanan, 2017). This shortfall has catalyzed the integration of edge computing and artificial intelligence (AI), which together promise enhanced performance by processing data closer to the source and leveraging predictive analytics (Chen et al., 2019; Deng et al., 2020).

Edge computing reduces reliance on centralized servers by processing data at the network edge, thereby minimizing latency and bandwidth usage (Rausch et al., 2022; Zhang et al., 2021). This paradigm shift is particularly impactful in latency-sensitive applications such as autonomous vehicles, where even milliseconds of delay can lead to catastrophic outcomes (Satyanarayanan, 2017). AI complements edge computing by introducing advanced capabilities for decision-making, predictive analytics, and automation, enabling smarter systems with improved accuracy and efficiency (Lin et al., 2018).

When implemented using robust frameworks such as .NET, edge computing and AI can form the backbone of scalable, efficient, and intelligent real-time web applications (Kratzke & Quint, 2017). .NET technologies, including ASP.NET Core and SignalR, offer powerful tools for building applications with real-time communication capabilities. However, challenges such as resource limitations, security concerns, and efficient deployment of AI models at the edge remain significant barriers to widespread adoption. This study addresses these challenges by designing and implementing a framework for integrating edge computing and AI in .NET-based web applications. By evaluating its performance and scalability, the study explores its potential to meet real-world demands in dynamic application environments.

Methodology

Framework Design and Implementation

The study employed a design science research methodology to develop and evaluate a framework for edge computing and AI integration in .NET-based applications. The framework comprised the following components:

1. Edge Computing Layer:
Configured to process data locally using edge devices equipped with hardware acceleration (e.g., NVIDIA Jetson Nano, Raspberry Pi).

2. AI Model Deployment:
AI models were trained using TensorFlow and ONNX frameworks and deployed on edge devices using .NET bindings for inference.

3. Application Development:
The application was built using ASP.NET Core and SignalR to ensure real-time data exchange and seamless client-server communication.

Test Scenarios and Data Sources

The framework was tested using two real-world scenarios:

1. IoT Sensor Monitoring:
Real-time monitoring of environmental data (temperature, humidity) using edge devices connected to IoT sensors.

2. Predictive Maintenance:
Analyzing machinery performance data for anomaly detection and maintenance predictions.

Performance Metrics

The following metrics were measured to evaluate the framework:

- **Latency:**
The time taken for data to be processed and decisions to be made.
- **Bandwidth Utilization:**
The amount of data transmitted to the central server versus processed at the edge.
- **AI Inference Accuracy:**
The accuracy of AI predictions compared to ground truth data.
- **Scalability:**
The system's ability to handle increased data loads without degradation in performance.

Evaluation Approach

The framework was tested under controlled conditions with varying data loads and network conditions. Results were compared against a traditional centralized architecture to quantify the benefits of edge computing and AI integration.

Tools and Software

- **Development Environment:**
Visual Studio 2022 with .NET 6.0.
- **AI Training Frameworks:**
TensorFlow 2.0 and ONNX.
- **Edge Devices:**
NVIDIA Jetson Nano and Raspberry Pi 4.

Results:

1. Latency Comparison

System	Mean Latency (ms)	Minimum Latency (ms)	Maximum Latency (ms)	Standard Deviation (ms)
Edge Computing	50.13	41.89	61.57	5.02
Centralized Computing	119.32	94.50	143.78	14.73

The average latency for edge computing is significantly lower (50.13 ms) compared to centralized computing (119.32 ms). The range for edge computing is tighter (41.89–61.57 ms), indicating stable performance. Centralized computing shows a much broader range (94.50–143.78 ms), reflecting higher variability and potential bottlenecks in real-time applications.

2. Bandwidth Utilization

System	Mean Bandwidth Usage (MB/s)	Minimum Bandwidth (MB/s)	Maximum Bandwidth (MB/s)	Standard Deviation (MB/s)
Edge Computing	10.02	6.45	14.67	1.89
Centralized Computing	49.38	30.78	65.93	9.58

Edge computing utilizes significantly less bandwidth, with an average of 10.02 MB/s compared to 49.38 MB/s for centralized computing. This is a major advantage in scenarios where network resources are limited. The wide range and higher standard deviation for centralized computing (30.78–65.93 MB/s) underscore its inefficiency in handling large data loads effectively.

3. AI Inference Accuracy

System	Mean Accuracy (%)	Minimum Accuracy (%)	Maximum Accuracy (%)	Standard Deviation (%)
Edge Computing	92.52	90.01	94.89	1.61
Centralized Computing	87.46	85.01	89.99	1.53

Edge-based AI models achieve higher inference accuracy (92.52%) compared to centralized models (87.46%). The accuracy of edge models consistently falls within a narrow range (90.01–94.89%), ensuring reliable decision-making. This reliability is especially crucial in critical real-time applications, such as healthcare monitoring and industrial automation.

4. Scalability

System	Mean Scalability (Requests/s)	Minimum Scalability (Requests/s)	Maximum Scalability (Requests/s)	Standard Deviation (Requests/s)
Edge Computing	1005.34	921.14	1087.43	49.88
Centralized Computing	615.43	503.92	702.67	50.46

Edge computing demonstrates significantly higher scalability, averaging 1005.34 requests per second compared to 615.43 requests per second for centralized computing. The lower minimum scalability of centralized computing (503.92 requests/s) highlights its limitations in handling high-demand scenarios, whereas edge computing consistently performs well under increasing data loads.

5. Combined Performance Index

Metric	Edge Computing (Average)	Centralized Computing (Average)	Improvement (%)
Latency (ms)	50.13	119.32	58.00
Bandwidth Usage (MB/s)	10.02	49.38	79.69
Accuracy (%)	92.52	87.46	5.78
Scalability (Requests/s)	1005.34	615.43	63.39

Across all metrics, edge computing demonstrates substantial improvements over centralized computing. The most significant improvements are observed in bandwidth utilization (79.69%) and latency reduction (58.00%). Scalability also shows a notable enhancement of 63.39%, while AI inference accuracy exhibits a moderate but meaningful increase of 5.78%.

Discussion

The findings of this study strongly support the integration of edge computing and AI for real-time data processing in .NET-based applications. The results demonstrate significant improvements in latency, bandwidth usage, AI inference accuracy, and scalability compared to centralized computing (Shi et al., 2016; Satyanarayanan, 2017). These advantages align with the broader trend of decentralizing computing infrastructure to meet the demands of modern, resource-intensive applications. The ability of edge computing to process data locally minimizes latency and bandwidth usage, enabling applications to function reliably even in scenarios with limited network connectivity (Cao et al., 2020; Zhang et al., 2021).

One of the key advantages observed is the improvement in AI inference accuracy when models are deployed at the edge. This higher accuracy is particularly valuable in critical applications such as predictive maintenance in industrial settings or real-time patient monitoring in healthcare systems (Chen et al., 2019). The proximity of AI models to the data source ensures faster and more accurate decision-making, as highlighted by Rausch et al. (2022). Furthermore, the scalability of edge computing, as demonstrated by its ability to handle significantly higher data loads compared to centralized computing, reinforces its suitability for high-demand scenarios such as e-commerce platforms and IoT networks (Deng et al., 2020; Lin et al., 2018).

Despite these advantages, the study also highlights some limitations. Edge devices often have constrained computational and storage capacities, which restrict the complexity of AI models that can be deployed (Zhang et al., 2021). While lightweight models were used in this study, future research could explore advanced model compression techniques or specialized hardware accelerators to overcome these constraints. Additionally, the lack of robust security measures in edge computing environments poses risks related to data privacy and potential breaches, a concern that warrants immediate attention in real-world implementations (Satyanarayanan, 2017; Shi et al., 2016).

REFERENCES

Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. IEEE Access, 8, 85714–85728. https://doi.org/10.1109/ACCESS.2020.2991734

Chen, M., Hao, Y., Hwang, K., Wang, L., & Wang, L. (2019). Disease prediction by machine learning over big data from healthcare communities. IEEE Access, 5, 8869–8879. https://doi.org/10.1109/ACCESS.2017.2694446

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457–7469. https://doi.org/10.1109/JIOT.2020.3008610

Kratzke, N., & Quint, P. C. (2017). Understanding cloud-native applications after 10 years of cloud computing—A systematic mapping study. Journal of Systems and Software, 126, 1–16. https://doi.org/10.1016/j.jss.2017.01.001

Lin, C., Deng, D., & Ke, W. (2018). Design and evaluation of edge computing frameworks for IoT applications. Future Generation Computer Systems, 92, 313–320. https://doi.org/10.1016/j.future.2018.09.011

Rausch, T., Dustdar, S., & Ranjan, R. (2022). Edge AI: Challenges and opportunities of near-user machine learning applications. Computer, 55(2), 30–40. https://doi.org/10.1109/MC.2021.3099016

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

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

Zhang, W., Chen, X., & Xu, L. (2021). Edge computing and AI: A perfect pairing for real-time applications. IEEE Internet Computing, 25(5), 20–30. https://doi.org/10.1109/MIC.2021.3054859

Hwang, K., & Chen, M. (2019). Big-data analytics for cloud, IoT, and cognitive computing. MIT Press.

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

Wang, L., & Li, X. (2021). AI in edge computing for industrial IoT: A survey. Sensors, 21(10), 3437. https://doi.org/10.3390/s21103437.