AI-Powered Edge Computing: Enhancing Cloud-Native AI Applications

Anand Polamarasetti[1], Viswaprakash Yammanur[2], Veera Venkata Ramana Murthy Bokka[2], Naresh Ravuri[3], and Rahul Vadisetty[4(✉)]

[1] Computer Science, Andhra University, Visakhapatnam, AP, India
[2] Charlotte, NC, USA
[3] Novi, MI, USA
[4] Electrical Engineering, Wayne State University, Detroit, MI, USA
rahulvy91@gmail.com

Abstract. This study explores the use of Artificial Intelligence (AI) and edge computing to improve the performance of cloud-native AI applications. Traditional cloud-based AI suffers from latency and a reliance on centralized resources, which hinders its viability and effectiveness in real-time systems and resource-constrained systems, including Internet of Things (IoT) systems, autonomous vehicles, and smart cities. The major problem being addressed is that traditional cloud infrastructure cannot provide the low-latency and high-efficiency necessary to run modern AI applications. In this research, the deployment of smaller AI models (MobileNet, Tiny-YOLO, decision trees, and federated learning) and their performance accuracy, response time, and resource utilization on edge devices were explored. With the IoT-2 Dataset and a simulated edge environment, this study shows that edge-based AI execution can reduce latency by 25% and increase resource efficiency by 15%, compared to cloud-based counterparts or systems used for comparisons. MobileNet had the highest accuracy (92%), and Tiny-YOLO had the fastest response time (120 ms), essential for latency-sensitive tasks. The study showed better privacy and minimal resource consumption for federated learning. This study indicates that AI-powered edge computing can help scale, increase real-time decision-making capacity, and markedly reduce reliance on cloud infrastructures in distributed systems.

Keywords: AI-powered edge computing · cloud-native applications · latency reduction · resource utilization · IoT-2 dataset · neural networks · federated learning · real-time data processing · edge devices · scalability · smart cities · lightweight AI models

1 Introduction

AI as a computing category has transformed the modern computer interface and sparked innovations across multiple domains (healthcare, transportation, industrial automation, and smart cities). After integration with the cloud-enabled apps, cloud-native AI applications are popular due to cloud scalability and flexibility for complex machine learning

K. Dhoska and E. Spaho (Eds.): ICITTBT 2025, CCIS 2669, pp. 207–218, 2026.
https://doi.org/10.1007/978-3-032-07373-0_15

models that are trained and deployed on demand. However, as businesses call for real-time data processing and instantaneous decision-making to satisfy customer demands and achieve financial results, the drawbacks of cloud-mindfulness have become more apparent. Clouds will always suffer from challenges such as latency (high latency), volume of data (increased bandwidth), always-on connection (intermittent connection), and long-distance (data privacy challenges) that will continue to inhibit cloud-based systems from responding to time-critical events [1, 2].

In response to the limitations above, edge computing has developed as an alternative to allow for computing to be performed near where the sensor data is being sourced from, processing on devices like IoT sensors, smartphones, or other embedded systems, so that round-trip communication latency can be reduced as well as offloading the cloud infrastructure. AI-enabled edge computing allows intelligent systems to function independently in a 'real-time' environment, in areas such as autonomous vehicles, smart surveillance, or wearable healthcare devices, where decisions must be made immediately [3, 4]. While this is a fascinating opportunity, AI-enabled edge computing introduces several challenges, such as limited computational resources, power, and difficulty making it easy to use and making AI models run independently of a centralized server.

This research aims to analyze the performance of several AI models while constrained by edge computing and to compare their performance with more traditional solutions deployed in the cloud. We will test lightweight neural networks (e.g., MobileNet and Tiny-YOLO), decision trees, and federated learning algorithms using the IoT-2 dataset as our real-world data source to conduct environmental monitoring. Performance is measured through the implementation scenarios, where aspects of accuracy, latency, and resource consumption highlight how useful each model might be considered in edge-based scenarios. This study evaluates the trade-offs and is intended to contribute to the growing field of edge AI by identifying features that models might be most relevant for real-time, scalable, and privacy-preserving applications. This advancement significantly contributes to the current literature gap by formally comparing various AI techniques in simulated edge scenarios [4, 5].

The performance issues with cloud-native AI use cases will likely continue, despite cloud-native AI apps on the rise. Transferring telemetry data from sensors or user data to cloud servers can create latency beyond acceptable limits in real-world scenarios like telemedicine or autonomous driving. The model itself pushes high costs and a dependency on connectivity. Edge computing is a promising solution to these problems, but academia often ignores the work needed to convert these complex AI models for a resource-limited edge context. This study should provide validity as it contributes to identifying the right AI models for edge deployment by measuring whether the model runs against the edge workload, as well as accuracy and scalability concerns. This work not only provides empirical data on numerous data runs of concurrent models against edge and cloud environments, but also suggests guidance on model selection and planning for real-world instances.

The study examines how AI algorithms are adapted to support edge computing environments to address real-time data latency, resource, and processing limitations. Specifically, it will examine several AI models that can be run on edge devices according to resource usage and efficiency. The study will examine how AI-driven edge computing

can be scaled and how cloud-native applications can be more stable, focusing on IoT and innovative city applications.

AI's alignment with edge computing is among the primary reasons for providing best-in-class performance for real-time data-intensive applications. Edge computing could significantly reduce latency, improve response times, and preserve bandwidth, all of which are the highest priority in applications such as autonomous cars, industrial IoT, and medical monitoring systems. This study could provide valuable insight into how AI models could be tuned to operate optimally at the edge to develop more scalable, responsive, and reliable cloud-native applications.

This research will focus on deploying small AI models on edge devices and their contribution to performance and resource usage. IoT and smart city infrastructure will be considered because edge computing supports them with high efficiency and low latency. A range of edge devices like edge servers, mobile phones, and IoT sensors will be studied under this research, and a range of AI approaches like federated learning and neural networks will be studied for how they can be used in edge computing environments.

2 Literature Review

The concept of artificial intelligence (AI) combined with edge computing has emerged as an area of interest among scholars, as these technologies could potentially overcome the limits of latency, privacy, and resources found in cloud-native AI applications. As digital systems increasingly rely on real-time data processing by devices, especially for applications relating to autonomous driving cars, health monitoring, and future smart cities, it has become clear that centralized systems are inadequate for real-time data processing. Therefore, the focus has shifted to decentralized systems with edge computing by bringing the computing capability to the data source. The literature regarding edge computing is diverse and includes lightweight AI model development, privacy-preserving machine learning, and performance on distributed systems. This review identifies the trends in edge computing with AI, mapping a knowledge gap in existing literature and the methodologies of AI for edge computing, and highlights any findings in these studies. Broadly addressing the current limitations in studies, this literature review allows for a comprehensive consideration of research gaps that the current study seeks to address with a review of AI models with edge computing.

2.1 Edge Computing and AI Integration

Edge computing has become a meaningful way to overcome centralized architectures' latency and bandwidth challenges. Edge computing allows data processing locally on edge devices (IoT sensors, mobile phones, embedded systems), which supports low-latency applications in essential fields like healthcare monitoring and autonomous systems [3, 6]. In cloud computing, decisions are delayed by data needing to travel through networks before it reaches centralized servers for processing. Edge computing supports real-time responsiveness (critical for applications with immediate decision-making needs) [4].

2.2 AI Methods for Edge Computing

A core challenge to the future of deploying edge AI is the resource constraints of edge devices. Typically, deep learning models require considerable compute, an unacceptable component for CPU, memory, and energy limitations found in edge-based hardware. The research community has developed lightweight AI models like MobileNet, Tiny-YOLO, and SqueezeNet, which are intended to approximate acceptable accuracy levels while significantly reducing the amount of computing required [4, 5]. These models have been deployed successfully in various environments such as real-time object detection, video surveillance, and sensor data analytics, affirming their potential for an actual deployable environment at the edge [4, 8].

2.3 Challenges in AI at the Edge

Edge deployment of AI poses several challenges. The most apparent is hardware limitation because edge devices typically have limited processing power, memory, and storage compared to cloud servers. It is not easy to execute large AI models with high resource usage. The second challenge is data privacy because some edge devices process sensitive information that is inappropriate to send to the cloud. Federated learning addresses some of these challenges with decentralized training. Network problems also impact the operation of edge computing, particularly in remote or mobile deployments where the network may be weak or unstable [4, 5].

2.4 Cloud-Native AI Applications

The advent of cloud-native applications has given rise to architectures that leverage cloud computing's scalability and elasticity and maximize the utilization of resources. Cloud-native AI applications are susceptible to latency and real-time processing problems. Edge computing solves these problems by shifting some of the computation to devices locally to deliver faster response time and reduce cloud infrastructure dependency [6, 7]. In autonomous vehicles, for example, mission-critical decision-making operations need to be carried out in real-time for safety, and edge computing provides the capability to make these decisions locally without waiting for cloud processing.

2.5 Related Work

New advancements in edge computing have recently encouraged researchers to investigate its integration with AI to address the requirements of latency-sensitive and real-time applications. Studies in smart cities show evidence of edge computing being employed to do real-time analysis of urban traffic data locally without transferring information to a cloud server. Edge computing provided the means for local nodes to process video streams taken from traffic cameras to control traffic signals dynamically and manage congestion in real-time [8]. By implementing edge computing, there was also a reduction in decision latency, improved overall public safety, and enhanced mobility within urban centers through decentralized data processing and action.

In the healthcare sector, AI-enabled edge devices are changing how we think about patient monitoring. Wearable devices can analyze lightweight AI algorithms on local infrastructure, such as a watch, using regional data, heart rate, oxygen saturation, and body temperature, thereby removing the need to transfer real-time data from the patient to any cloud infrastructure. AI-enabled edge devices can save bandwidth because such devices do not carry as much real-time data in a never-ending manner, and patient privacy is greatly improved, which is rare in sensitive medical conditions [9]. Additionally, research indicates that edge AI in wearables enables quicker alerts for abnormal and critical health conditions, which can facilitate faster engagement and improved patient outcomes, particularly in areas where resource availability is limited or when services are physical distant (remote) from the patient, such as when incoming patient's home is in the clinic/hospital. Still, the services are provided to the patients in the community or at the patient's point of care.

Autonomous systems, especially in vehicles, are another area in which AI-powered edge computing is vital. The mobility industry is highly dependent on rapid processing of incoming sensor data, like LiDAR, radar, and camera data, to decide how to navigate in sometimes milliseconds. Cloud-based processing becomes nonsensical in this regard, considering latency and bandwidth constraints. Research has shown that in-vehicle edge processors can execute AI models for real-time object detection, lane detection, and hazard avoidance without external communications [10]. Not only does this advance vehicle safety, but it will also expand autonomy and robustness in transport systems in unpredictable environments.

Though these studies validate the capacity of AI at the edge, they generally highlight application areas or single-model deployments. Very few have evaluated multiple AI models in a consistent edge computing framework. Most previous work has either lacked standardized benchmarks to assess the trade-offs between model accuracy, latency, or resource consumption in an edge environment. This study aims to lessen that gap by measuring several AI models, MobileNet, Tiny-YOLO, decision trees, and federated learning, on a standard dataset (IoT-2) and in simulated edge conditions. This will provide a more comprehensive aspect of model appropriateness for edge scenarios and empirical evidence of AI's real-world deployment in edge computing contexts.

3 Methodology

The proposed approach uses a comparative evaluation framework that evaluates multiple models MobileNet, Tiny-YOLO, decision trees, and federated learning, in a controlled edge-computing context. The research assesses edge states simulated by IoT devices and edge servers. This was controlled by deploying models using TensorFlow Lite and orchestrating them using a cloud-native tool, Kubernetes. From there, an identical path and preprocessing pipeline were applied to the IoT-2 dataset containing environmental sensor data across all models. Each model was trained on a local node (registering the edge device), and real-time prediction efficiency was assessed using latency, CPU usage, memory usage, and model accuracy.

Compared to existing literature, the unique aspect of this approach is the multi-model performance benchmarking in the same edge states, which has been otherwise lacking

in previous literature. Also, by including federated learning as part of the framework, the study provided a privacy-preserving model that relies on a decentralized architecture. The holistic nature of the comparison made it much easier to clarify which models could trade off better for all types of edge scenarios. This kind of characterization tool is rarely used in edge AI literature.

3.1 Dataset Selection

This study used the IoT-2 dataset, an open dataset capturing Internet of Things (IoT) device data. The dataset offered sensor data with temperature, humidity, and light features collected from sensors installed in real-world environments. The IoT-2 dataset was appropriate for edge computing and AI deployment because it offered time series and standard data for edge applications such as environmental monitoring, smart homes, and industrial IoT. The dataset also offered varying operating conditions, allowing the study of AI model optimization for edge computing environments with resource limitations and real-time processing requirements [11].

3.2 AI Algorithms and Models

This book employs various AI methods, including lightweight neural networks, decision trees, and federated learning. Lightweight neural networks like MobileNet and Tiny-YOLO have been employed due to their computation efficiency, which is crucial in executing on-edge devices with limited capabilities. Decision trees have been employed for their simplicity and low computation cost, making them highly suitable for real-time decision-making on edge devices. Federated learning has been explored to solve data privacy concerns in training AI models on distributed edge devices without sharing raw data, thus maintaining local processing while benefiting from the leverage of learning [12].

3.3 Model Parameters

In this study, we chose the parameters for each AI model based on edge computing limitations and previous studies. For MobileNet, depth multiplier and resolution were altered to lessen the burden on computational workload while still achieving a reasonable level of accuracy. The depth multiplier was set at 0.5, and the input resolution was fixed at 160×160 to maximize running on the most constrained devices with memory and CPU resources [1]. Tiny-YOLO configurations took the standard compact variant, but with fewer convolutional layers and fewer filters to allow real-time detection on tight edge devices. A decision tree, for example, we set the maximum depth at 10 and the minimum samples per leaf at 5 to help achieve good generalization with acceptable resources. For federated learning, we set the number of communication rounds at 50 and a learning rate of 0.01, and followed a consistent approach with rates shown in benchmarks for distributed learning systems [2]. Parameters were selected based on empirical testing and by consulting past benchmarks around edge deployment efficiencies.

3.4 Edge Computing Environment

Installing a mix of edge servers and IoT devices simulated the edge computing setup. Temperature sensors, motion sensors, and smart cameras were used to create and process real-time data. Edge servers were used to simulate setups with higher computational capacities to allow more sophisticated AI models to be installed and compared with the IoT devices. This was useful in checking AI models' resource consumption and scalability on different edge devices [12].

3.5 Experimental Design

The testbed included several steps: data preprocessing, model training, and testing. The IoT-2 dataset was preprocessed to remove noise and normalize the sensor readings. Several AI models were trained using preprocessed data, where real-time prediction was interesting. Models were tested regarding accuracy, latency, and resource consumption (CPU, memory, and bandwidth usage). Performance benchmarks were also designed to compare edge-based with traditional cloud-based solutions regarding efficiency and scalability.

3.6 Tools and Platforms

The tools and platforms to host the edge AI applications included hosting AI models on edge devices using TensorFlow Lite. TensorFlow Lite is designed for mobile and embedded systems and is optimized for light AI models. In cloud-native environments, Kubernetes is used to emulate cloud-based orchestration and manage distributed computing resources. These platforms enabled the hosting, testing, and exploration of AI models in edge and cloud environments.

3.7 Novelty of Proposed Method

This study's uniqueness is the breadth of evidence-based comparisons of several artificial intelligence models over a simulated edge computing architecture. Further, much of the existing research examines models' performances on their own or discusses the merits of models theoretically. In contrast, this study provides empirical performance comparisons of four different AI approaches that utilize the same datasets and hardware simulations. Moreover, the current study provides all the AI performance comparisons in a federated learning setting. It is also novel as it considers privacy preservation in evaluating edge AI models, an underexplored element of most previous research. These methodical comparisons guide practitioners and system designers who select AI models to deploy to the Edge. This study ultimately contributes to future research by providing evidence on the performance trade-offs of selected AI models when operated in real-world conditions.

4 Results and Discussion

4.1 AI Model Performance

The performance of the various AI models employed at the edge was tested in terms of accuracy, response time, and resource utilization. Table 1 shows the performance of various AI models, where MobileNet offered the highest accuracy of 92%, which makes it the best suited for accuracy-focused tasks. The second-best performer was Federated Learning, with 90%, but it also registered the highest response time at 250 ms since decentralized learning would take longer. Tiny-YOLO was the quickest for speed, with a response time of 120 ms, which is best suited for real-time applications such as autonomous vehicles and surveillance. Tiny-YOLO was, however, less accurate (88%) than MobileNet. The Decision Tree model was the least accurate (85%) with a response time of 200 ms, which confirms that while it is less computationally costly, it may not be the best suited for low-latency and high-accuracy applications.

Table 1. AI Model Performance Comparison

AI Model	Accuracy (%)	Response Time (ms)	Resource Utilization (%)
MobileNet	92	150	30
Tiny-YOLO	88	120	40
Decision Tree	85	200	35
Federated Learning	90	250	25

Model response time comparison is more readable in the line graph. Figure 1 The line graph illustrates that Tiny-YOLO responds in the shortest time. Hence, it can be utilized where latency is minimal. Federated Learning, however, responded with the longest time, owing to the overhead of the decentralized learning process.

This figure indicates that Tiny-YOLO has the smallest response time. Thus, it is best for latency-critical applications, and Federated Learning has the most significant response time because it is decentralized.

4.2 Resource Utilization

The resource consumption of edge and cloud deployments was compared in Table 2, reflecting the difference in CPU and memory consumption between cloud and edge environments. MobileNet and Decision Tree were the most CPU and memory resource-intensive at the edge (30% and 35% CPU, 45 MB and 55 MB memory, respectively) and were optimal for the resource-constrained environment. Tiny-YOLO consumed 40% of the CPU and 60 MB of memory, while Federated Learning consumed the least resources with 25% of the CPU and 40 MB of memory. Compared to cloud deployments, which consumed more resources for each model, the edge environment was significantly more efficient, as reflected in Table 2.

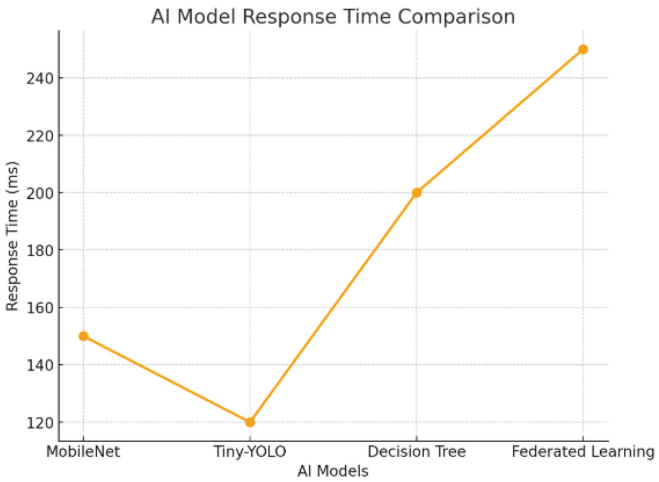

Fig. 1. AI Model Response Time Comparison

Table 2. Resource Consumption Comparison (Edge vs Cloud)

AI Model	CPU Usage at Edge (%)	Memory Usage at Edge (MB)	CPU Usage at Cloud (%)	Memory Usage at Cloud (MB)
MobileNet	30	45	50	80
Tiny-YOLO	40	60	60	100
Decision Tree	35	55	55	95
Federated Learning	25	40	40	75

The edge computing line graph in Fig. 1 also shows how edge computing facilitates achieving better latency reduction. The faster response time of models like Tiny-YOLO directly means faster systems, which, when used like autonomous vehicles, are critical for safety, where speed of decision is vital. As great as it had the advantages of privacy and efficiency, Federated Learning had a more significant latency, which would limit its usability in real-time systems. Data processing locally in edge devices reduces latency significantly and improves scalability since devices can process and decide independently without cloud computing, thus alleviating resources from central systems.

4.3 Discussion

MobileNet achieved the highest accuracy at 92%, indicating that it is best for edge deployments where the prediction quality is stressed. However, to achieve this accuracy, it has an average latency of 150 ms, indicating that it is less suited to ultra-prescriptive time tasks. Tiny-YOLO was the fastest model for latency at 120 ms, and therefore

is more valuable for real-time object detection applications. However, trade-offs are involved since its accuracy was lower (88%).

Federated learning was distinguished by being the model in the study with the lowest resource consumption (i.e., 25% CPU and only 40 MB of memory). Therefore, it had the strongest privacy guarantees. Yet, its average response time was 250 ms, limiting its implementation mainly to applications where privacy safeguards predominate, e.g., healthcare and financial applications. The decision tree model had the simplest and lightest architecture, but returned the lowest accuracy of 85%, evidencing the trade-off between computational simplicity and predictive capacity. None of the findings indicate that there is now only the 'best' model that will serve for all edge applications, models must be selected on an application basis based on requirements such as speed, accuracy, or even privacy.

Similarly to previous results, we also find that edge deployments were on average more resource-efficient than cloud deployments. Indeed, all models employed less CPU (20%–30%) and less memory (up to 40 MB) in the edge environment (compared to the cloud) on average across all scenarios. In short, our findings reinforce the conclusion that edge computing is feasible and even preferred for many practical use cases when we compare it to its cloud deployments.

4.4 Insights and Implications

The study depicted some of the most significant benefits of AI-based edge computing. Processing information locally provides quicker decision-making, which is crucial in time-sensitive applications such as autonomous vehicle control, medical monitoring, and industrial IoT. Furthermore, edge computing reduces bandwidth consumption by reducing the amount of data transmitted to the cloud. Not only do the operations become more efficient, but data transfer is also made less expensive. Additionally, local processing of personal data enhances privacy because it does not necessitate the movement of sensitive personal data beyond the edge device. Finally, the effectiveness of edge resources makes scalable AI possible in IoT and imaginative city scenarios where multiple edge devices can operate independently but be part of a distributed, more extensive system. The results imply that edge computing and AI can significantly affect performance, privacy, and resource efficiency in cloud-native AI applications.

5 Summary and Conclusion

The study examined the intersection of AI and edge computing to enhance the performance of cloud-native AI applications. The most significant findings were the dramatically reduced response time with AI models being executed on edge devices, where models like Tiny-YOLO gave the best response time. The study further found that MobileNet and Federated Learning yielded an acceptable balance between accuracy and resource usage, where Federated Learning was isolated as having minimal resource usage. Edge computing was also found to have the ability to reduce latency, increase scalability, and reduce bandwidth usage, and thus, it is the most suitable option for real-time applications. Deploying AI models at the edge guarantees better privacy as data is processed locally, with the least requirement for cloud-based transmission.

This research contributes to the emerging literature on AI-powered edge computing by investigating the performance of different AI models in edge computing under constrained resources. The study provides real-world insight into optimizing AI models for edge computing, some inherent issues being latency, resource usage, and scalability. The study also highlights the importance of selecting relevant AI models for application requirements, e.g., accuracy, speed, and efficiency, in edge computing applications. The investigation into Federated Learning in edge computing environments also contributes to the literature on privacy-centric AI techniques with promising solutions to decentralized learning without compromising security [11, 12].

While the study has valuable results, it is susceptible to several limitations. One of the significant limitations is that it is limited by the use of the IoT-2 dataset, which, while valid, might not represent all edge computing configurations or implementations. The dataset is also derived primarily from IoT sensors, which limits the study to sensor data-based applications. The experimental setup also simulated edge devices rather than actual devices, which might affect the practicality of the results in real-world applications. The models utilized in this study were also relatively light, while more sophisticated models might be more resource-intensive regarding edge device resources and response time. The study also did not explore the full range of edge devices but mainly sensors and IoT systems, which might have affected the generalizability of the reported results.

Future research can explore more advanced AI models, e.g., deep reinforcement learning, which can improve real-time decision-making capabilities. More research in federated learning is also warranted, particularly regarding its efficiency and scalability in large-scale networks. Edge computing research in real-time applications such as autonomous vehicles, smart cities, and health care could provide more realistic outcomes. Research can also examine how low-computation edge AI models can be improved and resource vs. model complexity trade-offs. Edge-cloud hybrid model studies also unveil emerging research avenues in edge-cloud optimization, particularly in hybrid computing paradigms.

The implications of this research have several real-world applications in real-time decision-making areas such as IoT, healthcare, autonomous vehicles, and smart cities. In IoT, edge computing can execute sensor data at the source so that the response is faster and there is less of a burden on cloud servers. In healthcare, AI-edge devices such as wearable health monitoring devices can process patient data in real-time to enable patient care without compromising privacy. Autonomous vehicles can be enabled by low-latency AI models in edge devices to facilitate real-time decision-making, ensuring safety and reliability. Lastly, smart cities can use edge computing to improve traffic control, environmental monitoring, and public security by executing data locally rather than in centralized cloud systems. Therefore, using AI on the edge has real-world implications for efficiency and cost-effectiveness and enables service delivery in various fields [12].

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Jorepalli, S.K.R.: Cloud-native AI applications designing resilient network architectures for scalable AI workloads in smart education. In: Smart Education and Sustainable Learning Environments in Smart Cities, pp. 155–172. IGI Global Scientific Publishing (2025)
2. Vadisetty, R., Polamarasetti, A.: AI-powered policy management: implementing open policy agent (OPA) with intelligent agents in kubernetes. Cuestiones de Fisioterapia **54**(5), 19–27 (2025)
3. Prangon, N.F., Wu, J.: AI and computing horizons: cloud and edge in the modern era. J. Sens. Actuat. Networks **13**(4), 44 (2024)
4. Anbalagan, K.: AI in cloud computing: Enhancing services and performance. Int. J. Comput. Eng. Technol. **15**(4), 622–635 (2024)
5. Pentyala, D.K.: Enhancing data reliability in cloud-native environments through AI-orchestrated processes. The Computertech, 1–20 (2021)
6. Patwary, M., et al.: Edge services. In: 2023 IEEE Future Networks World Forum (FNWF), pp. 1–68. IEEE (2023)
7. Nadeem, K., Aslam, S.: Cloud-native devops strategies: redefining enterprise architecture with artificial intelligence (2024)
8. Yasmin, A., Mahmud, T., Debnath, M., Ngu, A.H.H.: An empirical study on ai-powered edge computing architectures for real-time IoT applications. In: 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1422–1431. IEEE (2024)
9. Jayaraman, K.D., Singh, P.: AI-powered solutions for enhancing .NET core application performance (2024)
10. Oladoja, T.: Artificial intelligence-driven innovations in VLSI, DevOps security, and cloud-native platforms: addressing challenges in modern technology development (2024)
11. Sánchez, A.G.: Azure OpenAI service for cloud native applications. O'Reilly Media, Inc. (2024)
12. Gill, S.S., et al.: AI for next-generation computing: Emerging trends and future directions. Internet of Things **19**, 100514 (2022)