ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

Edge Computing: AI on Devices without Cloud Reliance

[1] Riya Rangwani, [2] Satwik Tripathy*

[1][2] Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
Email: [1] rangwaniriya8@gmail.com; [2] satwik2912@gmail.com

Abstract— The growing use of artificial intelligence (AI) in real-world applications has exposed several limitations of traditional cloud-centric architectures, including high inference latency, heavy bandwidth consumption, privacy concerns, and reduced reliability in environments with unstable or limited network connectivity. These issues are especially critical for time-sensitive and mission-critical systems that require real-time responses. Edge computing addresses these challenges by enabling AI inference directly on end devices such as IoT nodes, mobile platforms, and embedded systems, thereby reducing or eliminating continuous dependence on the cloud. This study examines the design and implementation of cloud-independent edge AI systems, focusing on efficient on-device inference, lightweight model architectures, and optimization techniques suitable for resource-constrained hardware. An end-to-end edge AI pipeline is presented, covering local data acquisition, on-device preprocessing, optimized model deployment, and real-time decision-making at the edge. Experimental results demonstrate that edge-based AI significantly reduces response latency, improves data privacy, and enhances system robustness while maintaining accuracy comparable to cloud-based solutions, particularly in scenarios with intermittent or no connectivity. These findings highlight edge AI as a practical and scalable solution for applications such as autonomous systems, healthcare monitoring, smart surveillance, and industrial automation, and position edge computing as a foundational technology for next-generation decentralized AI systems.

Index Terms— Decentralized Intelligence, Edge AI, Edge Computing, Embedded Systems, Internet of Things (IoT), Low-Latency Computing, On-Device Intelligence, Real-Time AI, Resource-Constrained AI

I. INTRODUCTION

The rapid growth of artificial intelligence (AI) has transformed modern computing systems by enabling intelligent decision-making across diverse domains such as healthcare, transportation, surveillance, and industrial automation. Traditionally, AI applications have relied heavily on cloud-based infrastructures for data processing and model inference due to their high computational capabilities and scalability. However, this centralized approach introduces several limitations, including increased latency, dependency on stable network connectivity, privacy and security concerns, and significant bandwidth consumption. These challenges become critical in real-time and mission-critical scenarios where immediate responses and data confidentiality are essential.

Edge computing has emerged as a powerful paradigm to address these limitations by bringing computation closer to the data source. By executing AI models directly on edge devices such as Internet of Things (IoT) nodes, mobile devices, and embedded systems, edge computing enables faster response times, reduced network dependency, and enhanced data privacy. Recent advancements in lightweight neural network architectures, model compression techniques, and specialized edge hardware have made it feasible to deploy sophisticated AI capabilities on resource-constrained devices.

This paper explores the concept of edge computing–enabled AI systems that operate without continuous cloud reliance. It focuses on the architectural design, deployment strategies, and optimization techniques required to achieve efficient on-device intelligence. By shifting AI computation to the edge, this work highlights a decentralized and resilient approach to intelligent systems, paving the way for scalable, low-latency, and privacy-aware AI applications in real-world environments.

II. MOTIVATION AND PROBLEM DEFINITION

The rapid expansion of AI-driven applications across domains such as autonomous systems, healthcare monitoring, smart surveillance, and industrial automation has intensified the demand for intelligent systems capable of delivering real-time and reliable decision-making. Traditionally, these applications rely on cloud-centric AI architectures, where data collected from end devices is transmitted to centralized servers for processing and inference. While this approach benefits from high computational power and scalability, it introduces inherent limitations that hinder performance in latency-sensitive and privacy-critical scenarios.

One of the primary challenges of cloud-based AI systems is high inference latency, caused by network delays and data transmission overhead, which is unacceptable for real-time applications. Additionally, continuous data transfer leads to excessive bandwidth consumption and increased operational costs. From a security perspective, sending raw or sensitive data to remote servers raises serious privacy and compliance concerns, especially in domains such as healthcare and personal surveillance. Furthermore, cloud-dependent systems suffer from reliability issues in environments with intermittent, low, or no network connectivity, limiting their

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

applicability in remote or mission-critical deployments.

The motivation of this work is to address these challenges by enabling AI computation directly on edge devices, eliminating the need for constant cloud interaction. The core problem addressed in this paper is the design of cloud-independent edge AI systems that can operate efficiently under constrained computational, memory, and power resources while maintaining acceptable accuracy and real-time performance. This includes identifying suitable system architectures, selecting and optimizing lightweight AI models, and developing deployment strategies that ensure low latency, data privacy, and system robustness. By solving these challenges, edge computing can unlock scalable and resilient AI solutions for next-generation intelligent systems.

III. BACKGROUND AND RELATED WORK

A. Edge Computing and On-Device AI Fundamentals

Edge computing has emerged as a distributed computing paradigm that brings computation and data processing closer to the data source, thereby reducing reliance on centralized cloud infrastructures. In contrast to traditional cloud-based systems, where data generated by end devices is transmitted to remote servers for processing, edge computing enables localized computation on or near the device. This approach is particularly beneficial for artificial intelligence (AI) applications that require low latency, real-time responsiveness, and enhanced data privacy. With the rapid growth of Internet of Things (IoT) devices, mobile platforms, and embedded systems, executing AI models directly at the edge has become increasingly important for building scalable, efficient, and context-aware intelligent systems.

B. Lightweight Models and Optimization Techniques

Recent advancements in on-device AI have been driven by the development of lightweight neural network architectures and hardware accelerators tailored for resource-constrained environments. Models such as MobileNet, EfficientNet-Lite, and TinyYOLO have demonstrated that acceptable accuracy can be achieved with significantly reduced computational and memory requirements. In addition, optimization techniques including model pruning, quantization, and knowledge distillation have enabled efficient deployment of deep learning models on edge devices with limited processing power, storage capacity, and energy budgets. These techniques play a critical role in making real-time inference feasible on low-power hardware.

C. Existing Edge AI Approaches and Research Gaps

Several studies have explored hybrid cloud–edge frameworks, where partial inference or preprocessing is performed at the edge and computationally intensive tasks are offloaded to the cloud. While such approaches reduce latency compared to fully cloud-based systems, they still depend on reliable network connectivity and raise concerns related to privacy, bandwidth usage, and system robustness. More recent research has shifted toward fully decentralized edge AI systems that support complete on-device inference without continuous cloud interaction. However, challenges remain in balancing model accuracy, resource efficiency, and deployment complexity across heterogeneous edge platforms.

Table I: Comparison of Cloud-Based and Edge AI Approaches

Approach	Processing Location	Latency	Privacy	Network Dependency
Cloud-Based AI	Centralized cloud servers	High	Low	High
Hybrid Cloud–Edge AI	Edge + Cloud	Medium	Medium	Medium
Fully Edge-Based AI	Edge devices	Low	High	Low

This work builds upon existing research by focusing specifically on cloud-independent edge AI architectures and deployment strategies. By emphasizing end-to-end on-device intelligence and real-time decision-making, the proposed approach addresses key limitations identified in prior studies and contributes to the advancement of decentralized, privacy-aware, and resilient AI systems.

IV. SYSTEM ARCHITECTURE FOR CLOUD-INDEPENDENT EDGE AI

A. Overall Architecture Overview

The system architecture of a cloud-independent edge AI framework is designed to enable complete AI inference and decision-making directly on edge devices, without relying on continuous cloud connectivity. The architecture follows a modular and layered design to ensure efficiency, scalability, and adaptability across heterogeneous edge platforms such as IoT nodes, mobile devices, and embedded systems. This layered approach allows each component to operate independently while maintaining seamless interaction with other layers, thereby improving system flexibility and robustness.

B. Data Acquisition and Preprocessing Layer

At the lowest level, the data acquisition layer consists of sensors and input devices, including cameras, microphones, and environmental sensors, which generate raw data in real time. This data is processed locally by the preprocessing layer, where operations such as normalization, resizing, noise reduction, and feature extraction are performed to reduce computational overhead during inference. Performing preprocessing at the edge minimizes data volume, reduces unnecessary data transmission, and prepares inputs for

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

efficient model execution.

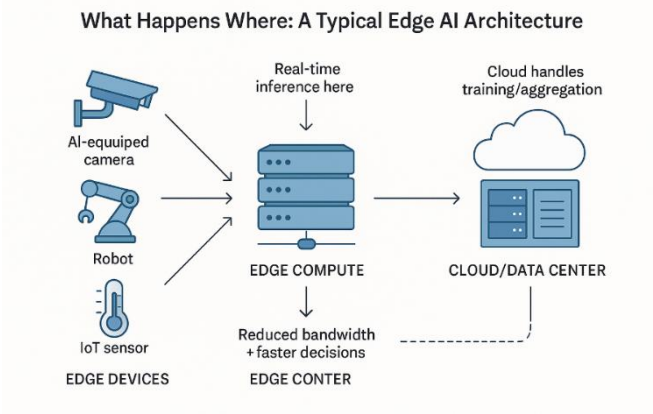

Fig. 1: System Architecture for Edge AI Architecture

C. Edge AI Inference Layer

The core component of the architecture is the edge AI inference layer, which hosts lightweight and optimized machine learning or deep learning models. These models are tailored for resource-constrained environments using techniques such as quantization and pruning, enabling execution on CPUs, GPUs, or specialized NPUs available on edge devices. The inference engine executes the model locally and produces predictions or classifications with minimal latency, enabling real-time intelligent behavior without external processing.

D. Decision and Action Layer

Above the inference layer, the decision and action layer interprets model outputs and triggers appropriate responses, such as alerts, control signals, or autonomous actions. This enables real-time decision-making without external dependencies. An optional cloud interaction module may be included solely for non-critical tasks such as model updates, performance monitoring, or periodic synchronization, ensuring that core functionality remains operational even in offline or low-connectivity conditions.

This architecture emphasizes decentralization, low latency, and data privacy, making it suitable for real-time and mission-critical applications where cloud reliance is impractical or undesirable.

V. MODEL DESIGN AND OPTIMIZATION TECHNIQUES

A. Importance of Model Design for Edge AI

Deploying artificial intelligence models on edge devices requires careful model design and optimization to address constraints related to limited computational power, memory capacity, and energy availability. Unlike cloud environments, edge platforms cannot support large, complex models without compromising latency and efficiency. Therefore, the selection of lightweight model architectures and the application of optimization techniques are critical for enabling effective on-device intelligence. Proper model design ensures that AI systems can operate reliably within the strict resource limitations of edge environments.

B. Lightweight Neural Network Architectures

Lightweight neural network architectures are commonly adopted for edge AI applications due to their reduced parameter count and computational complexity. Models such as MobileNet, EfficientNet-Lite, and SqueezeNet are designed to achieve a balance between accuracy and efficiency by using depthwise separable convolutions and optimized layer structures. These architectures enable real-time inference on edge devices while maintaining acceptable performance levels for tasks such as image classification, object detection, and sensor data analysis. Their compact design makes them suitable for deployment on low-power and embedded hardware platforms.

Table II: Characteristics of Lightweight Models for Edge AI

Model Name	Model Size (MB)	Parameters (Millions)	Inference Speed	Suitable Edge Devices
MobileNet	~16	4.2	High	Mobile phones, IoT devices
EfficientNet-Lite	~20	5.4	Medium-High	Embedded systems
SqueezeNet	~5	1.2	High	Low-power edge devices

C. Model Compression and Hardware-Aware Optimization

To further optimize model performance, several model compression techniques are employed. Quantization reduces numerical precision, typically converting floating-point representations to lower-bit formats, thereby decreasing memory usage and accelerating inference. Pruning removes redundant or less significant parameters from the network, reducing model size and computational overhead without substantial loss in accuracy. Knowledge distillation transfers knowledge from a large, high-performance model to a smaller student model, enabling efficient learning suitable for edge deployment. Additionally, model compilation and hardware-aware optimization play a vital role in edge AI systems. By tailoring models to specific hardware accelerators and execution environments, inference efficiency and power consumption can be significantly improved. Collectively, these techniques enable scalable, low-latency, and resource-efficient AI inference on edge devices, forming the foundation of cloud-independent edge AI systems.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

VI. DEPLOYMENT AND IMPLEMENTATION ON EDGE DEVICES

A. Deployment Requirements and Constraints

The deployment of artificial intelligence models on edge devices involves transforming optimized models into formats that can be efficiently executed within resource-constrained environments. This process requires careful consideration of hardware capabilities, software frameworks, and runtime constraints to ensure reliable and low-latency on-device inference. Unlike cloud deployment, edge implementation must account for limited memory, processing power, and energy consumption while maintaining consistent performance. These constraints make deployment a critical stage in achieving practical and scalable edge AI systems.

B. Model Conversion and Runtime Integration

The deployment process typically begins with model conversion and compilation. Trained and optimized models are converted into edge-compatible formats using frameworks such as TensorFlow Lite or ONNX, which are specifically designed for lightweight inference. These formats enable efficient execution by leveraging device-specific optimizations and reducing runtime overhead. Hardware-aware compilation further improves performance by utilizing available accelerators such as GPUs, NPUs, or edge-specific AI chips.

Once deployed, the model is integrated into the edge application pipeline, where it interacts with local data sources and preprocessing modules. This integration ensures seamless data flow from sensors to inference engines and supports real-time processing.

C. Resource Management and Offline Operation

Efficient memory management and task scheduling are essential to prevent resource contention and ensure real-time responsiveness. Power management strategies, including adaptive inference scheduling and dynamic frequency scaling, are often employed to extend device battery life and reduce thermal stress. These techniques are particularly important for battery-powered and embedded edge devices.

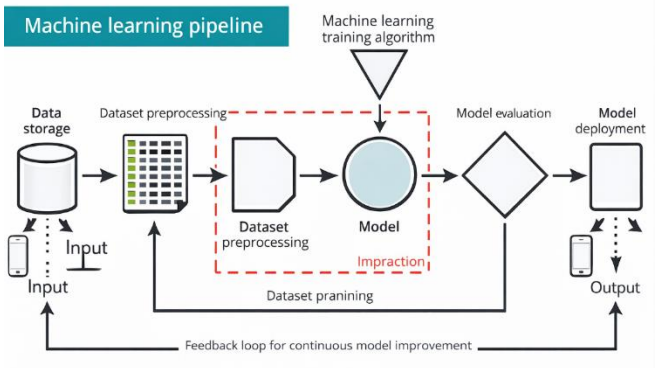

Fig. 2: Implementation on Edge AI devices

In cloud-independent edge AI systems, inference and decision-making occur entirely on the device, allowing uninterrupted operation even in offline or low-connectivity environments. Optional cloud connectivity may be used solely for non-critical tasks such as periodic model updates or system monitoring, without affecting core functionality.

VII. EXPERIMENTAL SETUP

A. Hardware and System Environment

The experimental evaluation of the proposed cloud-independent edge AI system is designed to assess its performance in terms of latency, accuracy, resource utilization, and reliability under realistic operating conditions. The experiments are conducted on representative edge devices to reflect practical deployment scenarios in resource-constrained environments. These devices include embedded platforms and mobile-class hardware equipped with standard CPUs and, where available, lightweight hardware accelerators such as GPUs or NPUs. This setup ensures that the evaluation closely represents real-world edge computing conditions.

Table III: Experimental Setup Configuration

Component	Specification
Edge Device	Embedded / Mobile-class device
Processor	ARM / x86 CPU
Memory	2–4 GB RAM
Operating System	Linux / Android
Inference Framework	TensorFlow Lite / ONNX Runtime
Hardware Accelerator	GPU / NPU

B. Software Framework and Dataset

The software environment consists of an edge-compatible inference framework supporting optimized model execution, along with libraries for data preprocessing and system monitoring. Trained and optimized AI models are deployed locally on the edge devices, and all inference operations are performed without reliance on external cloud services. Input data is sourced from publicly available datasets or locally generated sensor data, depending on the target application, ensuring consistency and repeatability of experiments. This configuration enables controlled testing of the proposed system in both online and offline scenarios.

C. Evaluation Metrics and Methodology

To evaluate system performance, multiple metrics are considered. Inference latency is measured as the time taken from data input to output generation, reflecting real-time responsiveness. Model accuracy is evaluated using standard task-specific metrics to compare edge-based inference with baseline cloud-based implementations. Resource utilization, including CPU usage, memory consumption, and power usage, is monitored to analyze efficiency and feasibility on constrained hardware. Additionally, system behavior under

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

intermittent or no network connectivity is observed to validate cloud independence and operational robustness.

All experiments are conducted under controlled conditions with repeated trials to ensure reliability of results. This experimental setup provides a comprehensive basis for evaluating the effectiveness of the proposed edge AI architecture and its suitability for real-world, latency-sensitive, and privacy-critical applications.

VIII. PERFORMANCE EVALUATION AND RESULTS

A. Latency and Real-Time Performance

The performance of the proposed cloud-independent edge AI system is evaluated based on inference latency, accuracy, resource efficiency, and system robustness. Experimental results indicate a significant reduction in inference latency when compared to cloud-based AI implementations, primarily due to the elimination of network transmission delays and remote server dependency. On-device inference enables near real-time response, making the system suitable for latency-sensitive applications such as autonomous systems, healthcare monitoring, and smart surveillance.

Table IV: Performance Comparison of Cloud and Edge AI Systems

Metric	Cloud-Based AI	Edge-Based AI
Inference Latency (ms)	High	Low
Accuracy (%)	High	Comparable
Bandwidth Usage	High	Very low
Power Consumption	High	Low
Offline Operation	Not supported	Supported

B. Accuracy and Model Effectiveness

In terms of accuracy, the optimized edge AI models demonstrate performance comparable to their cloud-based counterparts. Although lightweight architectures and compression techniques introduce minor accuracy trade-offs, the results remain within acceptable margins for practical deployment. This balance between efficiency and performance highlights the effectiveness of model optimization strategies such as quantization and pruning for edge environments, enabling reliable inference on resource-constrained devices.

C. Resource Utilization and System Robustness

Resource utilization analysis shows that the proposed system operates efficiently within the computational and memory constraints of edge devices. CPU and memory usage remain stable during continuous inference, while power consumption is significantly lower than cloud-based solutions that require constant data transmission. The system also exhibits improved reliability, maintaining consistent operation in scenarios with limited or no network connectivity, where cloud-dependent systems fail or degrade in performance.

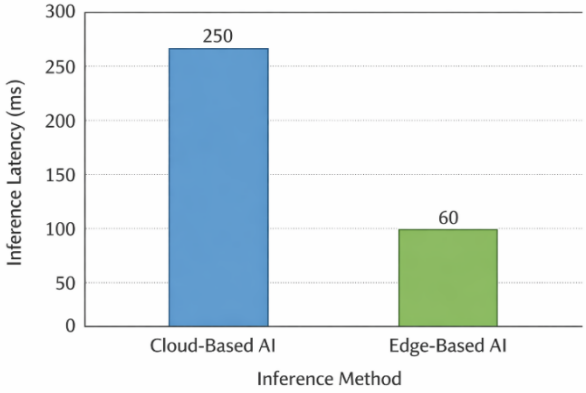

Fig. 3 Comparison of Inference Latency between Cloud-Based AI and Edge-Based AI

Overall, the results confirm that cloud-independent edge AI systems offer substantial advantages in terms of latency reduction, privacy preservation, and operational robustness. These findings validate the feasibility of deploying intelligent applications directly on edge devices without compromising performance, thereby supporting the adoption of edge computing as a practical alternative to traditional cloud-centric AI architectures.

IX. APPLICATIONS AND USE CASES

Cloud-independent edge AI systems enable a wide range of real-world applications where low latency, data privacy, and reliable operation are critical. By performing AI inference directly on edge devices, these systems support intelligent decision-making without dependence on continuous cloud connectivity, making them suitable for both connected and disconnected environments.

In autonomous systems, such as drones and self-driving vehicles, edge AI enables real-time perception, object detection, and navigation with minimal delay. Local processing ensures immediate responses to dynamic environmental changes, improving safety and operational efficiency. In the healthcare domain, edge AI is applied in wearable devices and remote patient monitoring systems to analyze physiological data in real time, ensuring data privacy and reducing reliance on centralized servers for sensitive medical information.

Smart surveillance systems leverage edge AI for on-device video analytics, including motion detection, facial recognition, and anomaly detection. Processing video streams locally reduces bandwidth usage and enhances privacy by avoiding transmission of raw footage to the cloud. In industrial IoT environments, edge AI supports predictive maintenance, fault detection, and process optimization by analyzing sensor data directly at the source, enabling faster response and reduced downtime.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 02 February 2026

Additional use cases include smart agriculture, where edge AI assists in crop monitoring and automated irrigation, and retail analytics, where on-device intelligence enables real-time customer behavior analysis. These applications demonstrate the versatility and scalability of edge AI, highlighting its potential to transform diverse domains by delivering intelligent, low-latency, and privacy-aware solutions at the edge.

X. SECURITY, PRIVACY, AND RELIABILITY CONSIDERATIONS

Security, privacy, and reliability are critical factors in the design of cloud-independent edge AI systems, particularly as these systems operate on distributed and often physically accessible devices. Unlike centralized cloud infrastructures, edge devices are more vulnerable to physical tampering and unauthorized access, making robust security mechanisms essential for protecting data and model integrity.

From a privacy perspective, edge AI offers a significant advantage by ensuring that sensitive data is processed locally rather than transmitted to remote servers. This local processing reduces the risk of data interception, leakage, and non-compliance with data protection regulations. However, safeguarding locally stored data and AI models requires secure storage mechanisms, access control, and encryption techniques to prevent unauthorized usage or model extraction.

Reliability is another key consideration, especially in environments with intermittent or no network connectivity. Cloud-independent edge AI systems must be designed to operate autonomously and maintain consistent performance under varying conditions. Fault tolerance mechanisms, such as graceful degradation and local fallback strategies, help ensure continued operation in the event of hardware failures or resource constraints. Additionally, secure and reliable update mechanisms are necessary to deploy model improvements or security patches without disrupting system functionality.

Overall, addressing security, privacy, and reliability concerns is essential to ensure trustworthiness and long-term adoption of edge AI systems. By integrating robust protection strategies and resilient system designs, cloud-independent edge AI can deliver safe, privacy-preserving, and dependable intelligent services across diverse application domains.

XI. CHALLENGES AND LIMITATIONS

Despite the advantages of cloud-independent edge AI systems, several challenges and limitations must be addressed to enable widespread adoption and practical deployment. One of the primary challenges is the limited computational and memory capacity of edge devices, which restricts the complexity of AI models that can be deployed. Designing models that achieve an optimal balance between accuracy and efficiency remains a significant research challenge.

Another limitation arises from energy and thermal constraints, particularly in battery-powered or compact embedded devices. Continuous AI inference can lead to increased power consumption and heat generation, potentially affecting device lifespan and performance. Efficient power management and adaptive inference strategies are therefore essential to mitigate these issues. Additionally, hardware heterogeneity across edge platforms complicates deployment, as models and optimization techniques may not generalize well across different device architectures.

Maintaining and updating AI models on distributed edge devices also presents challenges. Model updates and version control must be performed securely and efficiently to avoid system downtime or inconsistencies. Furthermore, ensuring robustness against adversarial attacks and physical tampering remains difficult due to the exposed nature of edge devices. While local processing enhances privacy, it also necessitates stronger on-device security mechanisms.

Finally, scalability can be challenging in large-scale deployments involving thousands of edge devices, as monitoring, management, and coordination become increasingly complex. Addressing these challenges is crucial for the development of resilient, scalable, and energy-efficient edge AI systems capable of supporting diverse real-world applications.

XII. FUTURE RESEARCH DIRECTIONS

Future advancements in cloud-independent edge AI systems are expected to focus on improving efficiency, scalability, and adaptability in increasingly complex deployment environments. One promising direction is the integration of federated learning, which enables collaborative model training across multiple edge devices without sharing raw data, thereby preserving privacy while improving model generalization. This approach can significantly enhance learning capabilities in decentralized environments.

Another important research direction involves the development of self-adaptive and context-aware edge AI models that can dynamically adjust their complexity and inference behavior based on available resources, workload, and environmental conditions. Such models can optimize performance while minimizing energy consumption and computational overhead. Advances in edge-specific hardware accelerators and co-design of hardware and software are also expected to further improve inference speed and energy efficiency.

Edge-to-edge collaboration is an emerging concept where multiple edge devices cooperate to share insights or intermediate results, reducing redundancy and enhancing system intelligence without relying on centralized cloud infrastructure. Additionally, incorporating energy-aware and sustainable AI techniques will be critical for long-term deployment, particularly in battery-powered and

environmentally constrained settings.

Research into standardized deployment frameworks, security mechanisms, and model update strategies will further strengthen the reliability and manageability of edge AI systems. These future directions highlight the potential of edge computing to support increasingly autonomous, intelligent, and resilient AI applications, positioning cloud-independent edge AI as a key component of next-generation distributed computing systems.

XIII. CONCLUSIONS

This paper explored the role of edge computing in enabling artificial intelligence systems that operate independently of continuous cloud support. By shifting AI inference and decision-making to edge devices, the proposed approach addresses key limitations of cloud-centric architectures, including high latency, privacy concerns, bandwidth dependency, and reliability issues in low-connectivity environments. The study presented a comprehensive overview of cloud-independent edge AI systems, encompassing system architecture, model design and optimization techniques, deployment strategies, and performance evaluation.

Experimental observations demonstrated that on-device AI inference significantly reduces response latency while maintaining accuracy levels comparable to cloud-based solutions. Additionally, local data processing enhances privacy and ensures robust operation even in offline or constrained network conditions. The discussion of applications, security considerations, and existing challenges further highlighted the practicality and relevance of edge AI across diverse real-world domains.

Overall, the findings confirm that edge computing serves as a viable and scalable foundation for next-generation intelligent systems. By enabling decentralized, low-latency, and privacy-aware AI, cloud-independent edge AI has the potential to transform how intelligent applications are designed and deployed, paving the way for more resilient and autonomous computing environments.

REFERENCES

[1] Q. Liang, P. Shenoy, and D. Irwin, “AI on the edge: Rethinking AI-based IoT applications using specialized edge architectures,” IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8329–8342, May 2021.

[2] X. Wang and W. Jia, “Optimizing edge AI: A comprehensive survey on data, model, and system strategies,” IEEE Access, vol. 9, pp. 15920–15943, Jan. 2021.

[3] Y. Kang, J. Hauswald, C. Gao, et al., “Neurosurgeon: Collaborative intelligence between the cloud and mobile edge,” in Proc. IEEE Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), Atlanta, GA, USA, Apr. 2017, pp. 615–629.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.

[5] A. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, July 2017, pp. 1–9.

[6] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” IEEE Int. Conf. Learning Representations (ICLR), San Juan, Puerto Rico, May 2016.

[7] K. Bonawitz et al., “Towards federated learning at scale: System design,” in Proc. Machine Learning and Systems (MLSys), Stanford, CA, USA, Mar. 2019, pp. 374–388.

[8] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, “Model compression and hardware acceleration for neural networks: A comprehensive survey,” Proceedings of the IEEE, vol. 108, no. 4, pp. 485–532, Apr. 2020.