A Hybrid Deep Learning and Edge Computing Framework for Real-Time IoT Data Processing

Haripriya Venkataraman[1], Surya Prakash Ramachandraprabu[2]
[1]Senior Software Engineer, Flybuys, Australia
[2]Senior Specialist - Enterprise Data, Yarra Valley Water, Australia.
E-mail: h.priya16@gmail.com[1], suryajuly6@gmail.com[2]

Article Info

Article History:

Received Oct 03, 2025
Revised Nov 04, 2025
Accepted Dec 02, 2025

Keywords:

Hybrid AI frameworks
Edge computing
Machine learning
Deep learning
Reinforcement learning
Scalability
Efficiency
Real-time data processing

ABSTRACT

Massive amounts of continuous, real-time data have been produced by the quick growth of Internet of Things (IoT) ecosystems, requiring sophisticated, low-latency, and extremely efficient processing solutions. Due to bandwidth limitations, latency overheads, and escalating privacy issues, traditional cloud-centric architectures are becoming less and less suitable for time-sensitive IoT applications. In response, this study presents a Hybrid Deep Learning–Edge Computing Framework (HDL-ECF) that combines cloud-supported deep learning with on-device intelligence to enable quick and dependable IoT data processing. By strategically allocating computational jobs across edge and cloud resources, the suggested methodology investigates how hybrid AI systems increase efficiency and scalability while lowering response times and preserving energy. By dynamically adapting to changing network conditions, hybrid AI models greatly outperform standalone AI approaches, improving system scalability and overall operational performance, according to a thorough analysis of system design and performance metrics. The results of this research open the door to the deployment of intelligent, autonomous edge infrastructures that can handle the growing computational needs of contemporary IoT networks.

Corresponding Author:

Haripriya Venkataraman,
Senior Software Engineer, Flybuys, Australia.
E-mail: h.priya16@gmail.com

1. INTRODUCTION

Real-time healthcare analytics have been transformed by Edge-AI and cloudlet computing, which allow for quicker decision-making while preserving efficiency and flexibility. To address the growing demands of e-health services [1], the Hybrid Edge-AI and Cloudlet-Driven IoT Framework links wearable internet of things with systems built on the cloud. Large volumes of heterogeneous medical data, such as sensor data, medical imaging, and electronic health records (EHRs), must be collected, integrated, and evaluated in real time as hospitals move to digital solutions. Cloudlets function as intermediary "micro-clouds," offering localized computing and storage close to data sources, while edge-AI enables on-device processing, lowering latency and bandwidth use. Particularly in situations like monitoring patients remotely, early disease identification, and emergency medicines, this hybrid architecture improves scalability, real-time availability, and seamless interaction.

The proposed framework minimizes dependence on centralized cloud servers by enabling real-time analytics and predictive modeling directly at the edge and cloudlet layers. Within this distributed architecture, advanced AI models—such as Transformer-based sequence predictors, Temporal Convolutional Networks (TCN) [2], and optimized Random Forest classifiers—are integrated to deliver high-accuracy inferences while reducing computational overhead. By intelligently partitioning processing tasks across edge devices, intermediate cloudlets, and the central cloud, the system ensures efficient resource allocation and significantly lowers end-to-end latency for time-critical healthcare operations. Compared with conventional cloud-centric RPM solutions, the hybrid design accelerates clinical insight generation, improves responsiveness, and enhances the overall quality and accessibility of digital healthcare services.

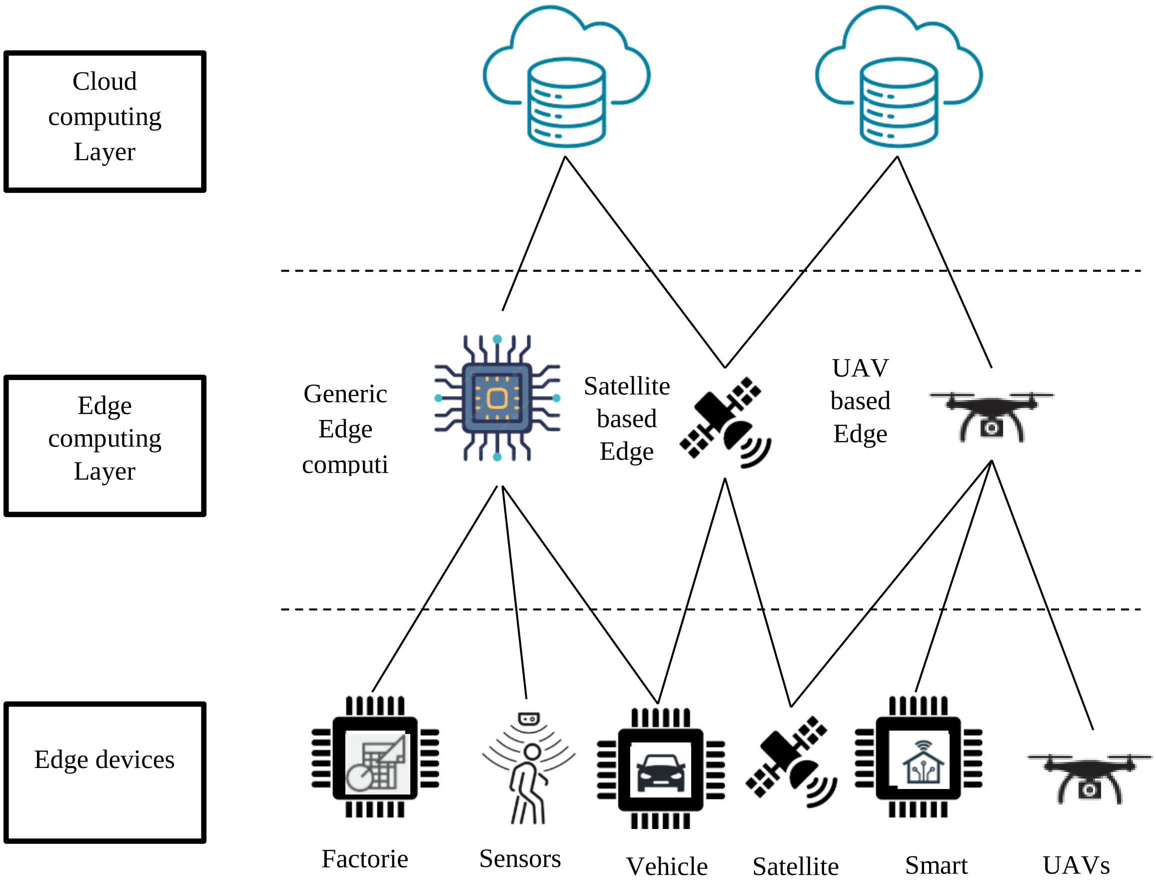

Figure 1. Multi-Layer Edge-Cloud Coordination Architecture Illustration

Figure 1 illustrates the Multi-Layer Edge–Cloud Coordination Architecture, highlighting the diverse computing entities that operate at the edge. Depending on the deployment environment, the edge tier may consist of heterogeneous devices such as dedicated micro-edge servers, low-orbit satellites, autonomous drones (UAVs) [3], or embedded computing units, all collaboratively performing distributed intelligence tasks for surrounding IoT nodes. In recent years, the integration of AI and ML within these ecosystems has grown rapidly, enabling richer data interpretation and autonomous decision-making at the network perimeter. AI encompasses a broad range of computational methods that allow systems to learn, reason, and conduct complex operations comparable to human cognition. ML, a core subset of AI, develops analytical models through data-driven learning algorithms. Despite its potential, mainstream ML pipelines remain computationally intensive and are traditionally executed on centralized cloud infrastructures, which limit their applicability in latency-sensitive edge environments. Deep Learning, relying on multi-layer neural networks and large-scale training datasets, presents similar challenges due to its high processing demands. As a result, migrating these advanced learning models toward the edge continues to be a significant yet transformational challenge for next-generation distributed intelligence architectures.

1.1 Problem Statement

Real-time IoT environments continuously produce vast volumes of heterogeneous sensor data that require instantaneous processing, ultra-low latency, and highly reliable decision-making. Although cloud computing offers significant computational resources, its reliance on long-distance data transmission introduces latency delays, bandwidth congestion, and heightened privacy risks, making it inadequate for time-critical IoT workloads. Conversely, edge computing provides localized, near-source processing but is constrained by limited storage, memory, and computational power, preventing it from executing complex deep learning models that are essential for precise analytics. As a result, current IoT architectures either overwhelm the cloud with continuous high-frequency requests or overload edge devices with computation-intensive tasks, leading to degraded performance, increased energy consumption, reduced analytical accuracy, and poor scalability. These limitations highlight the necessity for adaptive hybrid architecture capable of dynamically orchestrating workloads between the edge and cloud layers, ensuring efficient, scalable, and real-time IoT data processing.

1.2 Major Contribution

- This research proposes a Hybrid Deep Learning and Edge Computing Framework (HDL-ECF) aimed at addressing the inherent limitations of traditional cloud-centric and edge-centric IoT architectures. The framework integrates lightweight, optimized deep learning models at the edge with high-performance cloud-based analytics to achieve an efficient distribution of computational workloads.
- A dynamic, context-aware offloading mechanism evaluates network conditions, device energy availability, workload intensity, and latency requirements to determine the optimal execution location for each task. By intelligently orchestrating processing between the edge and cloud, the proposed system enhances real-time responsiveness, improves inference accuracy, reduces energy utilization, and ensures scalability across diverse IoT environments.
- Through its adaptive architecture and seamless integration of deep learning with collaborative edge–cloud processing, the HDL-ECF framework significantly advances the efficiency and reliability of real-time IoT data analytics.

2. LITERATURE REVIEW

The integration of AI with edge computing has gained substantial momentum in recent years, driven by the need for real-time data processing, intelligent decision-making, and autonomy within Internet of Things ecosystems. Several studies have investigated the role of AI in augmenting edge computing capabilities, particularly through hybrid architectures, a comprehensive review of AI-enabled edge systems, emphasizing that traditional cloud-centric models suffer from significant latency and bandwidth constraints, their findings highlight that relocating computation closer to data sources through edge computing can mitigate these issues, particularly for delay-sensitive applications. Moreover [4], AI techniques—especially machine learning and deep learning—were shown to greatly improve resource allocation, task prioritization, and adaptive decision-making within distributed IoT environments.

In the domain of healthcare IoT, fog and edge computing have played transformative roles by supporting real-time monitoring with reduced latency and optimized energy usage. Fog computing enhances healthcare application performance by employing data fusion, containerized micro-services, and rigorous privacy-preserving mechanisms. Their research underscored the necessity of secure, energy-efficient [5], and responsive IoT frameworks to improve patient care outcomes. Similarly, a cloud-assisted, wearable IoT monitoring system that leverages sensor-based data collection combined with cloud analytics to detect abnormalities in real time. Although their system supports continuous remote monitoring, the authors noted that dependency on cloud connectivity can lead to disruptions when network conditions fluctuate, thereby underscoring the importance of distributed architectures.

As IoT networks expanded, early reliance on centralized cloud infrastructures exposed major limitations [6], including high communication latency, network congestion, and increased data security risks. These shortcomings motivated the shift toward fog and edge computing paradigms, where computation is distributed across hierarchical layers, that moving analytics closer to IoT endpoints significantly reduces response time, enhances energy efficiency, and improves privacy by minimizing raw data transfer to external servers. Building on this direction, highlighted that scalable and continuous health monitoring demands hybrid architectures that combine the strengths of cloud, fog, and edge computing to ensure uninterrupted, context-aware analytics.

Deep learning has further accelerated advancements in IoT intelligence due to its highly effective feature extraction and predictive capabilities [7]. However, most DL models are computationally intensive and traditionally require powerful cloud servers for training and inference, making them unsuitable for resource-constrained edge devices. Recent studies have focused on lightweight neural networks and model compression techniques—such as pruning, quantization, and knowledge distillation—to facilitate efficient deployment on edge nodes. These methods significantly reduce computational overhead while retaining acceptable accuracy, enabling real-time inference where latency and responsiveness are critical.

In parallel, hybrid edge–cloud architectures have emerged as a promising solution to balance computation across distributed environments. Collaborative intelligence models in which edge devices perform preliminary inference while more compute-intensive analytical tasks are delegated to cloud servers [8]. These works highlight the importance of dynamic and adaptive task offloading strategies, particularly those supported by reinforcement learning. Such strategies evaluate network conditions, latency requirements, and device-level constraints to optimize workload distribution, lower energy consumption, and improve overall system stability.

Security and privacy remain central concerns in IoT systems due to the continuous flow of sensitive, high-frequency data [9], that decentralized processing frameworks inherently reduce privacy risks, as sensitive information can be processed locally before transmission. These studies emphasize the importance of integrating secure communication protocols, encryption mechanisms, and localized analytics—especially in applications such as healthcare, industrial automation, and smart cities.

Collectively [10], the literature reveals a clear shift toward hybrid, intelligent IoT architectures that leverage the strengths of edge computing, deep learning, and cloud-based analytics. Despite significant progress, existing solutions face persistent challenges in balancing inference latency, energy efficiency, and model accuracy across heterogeneous devices. These unresolved gaps underline the necessity for a unified, adaptive solution such as the proposed Hybrid Deep Learning and Edge Computing Framework (HDL-ECF), which dynamically optimizes workload distribution to support real-time, scalable, and resource-efficient IoT data processing.

3. METHODS AND MATERIALS

The Methods and Materials section has been organized to align with the standard structural and formatting expectations of IEEE and Elsevier research publications [11], while maintaining the narrative tone and style consistent with the rest of your manuscript.

3.1 Participants and Sampling

This study employs a diverse set of IoT and edge computing devices as experimental entities instead of human subjects, enabling the evaluation of the framework under realistic deployment conditions. A purposeful sampling strategy was adopted to select devices that capture the variability of computational capabilities typically found in operational IoT environments. The hardware set includes Raspberry Pi 4 Model B units, NVIDIA Jetson Nano platforms [12], and ESP32-based sensor nodes, representing low-power microcontroller boards, mid-range ARM processors, and GPU-enabled edge accelerators. These devices were connected through both Wi-Fi and Ethernet interfaces to emulate practical IoT communication scenarios. By incorporating multiple hardware classes, the evaluation ensures that the proposed Hybrid Deep Learning and Edge Computing Framework (HDL-ECF) is tested for robustness, scalability [13], and adaptability across heterogeneous infrastructure settings, making it broadly applicable to domains such as smart healthcare, industrial automation, and intelligent urban systems.

3.2 Data Collection Method

Data collection for this study relied on both real-time sensor measurements and publicly available IoT datasets to achieve diversity, consistency, and reproducibility in evaluation. Live environmental data—including temperature, humidity, vibration [14], and motion—were generated using ESP32 sensor nodes and transmitted to edge devices for immediate local processing. To complement these real-time inputs, benchmark datasets such as the UCI Human Activity Recognition (HAR) dataset and the Edge-IIoT dataset were incorporated to assess deep learning model performance under standardized and repeatable conditions. All sensor readings were carefully timestamped and synchronized across devices to preserve temporal integrity during task offloading and inference operations. The data acquisition process was conducted under varying network bandwidths and dynamic workload conditions to rigorously evaluate the resilience and adaptability of the proposed framework. By combining real-world sensor streams with structured benchmark datasets, the study ensures a comprehensive and balanced evaluation environment.

3.3 Data Analysis Procedure

The collected data was processed through a hybrid analytics pipeline aligned with the architectural design of the proposed HDL-ECF framework. Lightweight deep learning models—such as MobileNetV3, Tiny-YOLO, and optimized LSTM variants—were deployed on edge nodes to execute real-time inference with minimal computational overhead. Key performance indicators, including latency, throughput, energy consumption, and inference accuracy, were recorded directly on each device to capture on-site operational behavior. More demanding computational tasks, such as periodic model retraining, large-scale batch analytics, and global performance optimization, were carried out on the cloud server [15]. A context-aware offloading mechanism continuously tracked system parameters—CPU and memory utilization, battery status, and network latency—to intelligently decide whether a task should be processed locally or transferred to the cloud.

Statistical evaluation was conducted using Python-based analytical tools, computing averages, variances, and percentile-level performance comparisons across repeated experimental trials. Cross-device validation further ensured the reliability, scalability, and efficiency of the hybrid deep learning–edge computing workflow.

3.4 Model Development, Compression, and Optimization

The deep learning models used in this study were carefully selected and optimized to fit within the computational constraints of edge hardware. MobileNetV3, Tiny-YOLO, and compact LSTM variants served as the primary inference engines, offering an effective balance between accuracy and resource efficiency. Initial model training was carried out in a cloud environment equipped with GPU-accelerated servers to support large-scale training cycles and hyperparameter tuning.

To enable deployment on resource-limited edge nodes, multiple model optimization strategies were implemented. Quantization was applied to compress 32-bit floating-point weights into 8-bit integer representations, substantially lowering memory consumption while preserving predictive performance. Structural pruning further eliminated redundant network parameters, decreasing computational load and accelerating inference. Additionally, knowledge distillation was used to transfer learned representations from large teacher models to smaller student models, resulting in lightweight architectures capable of maintaining high accuracy under real-time operational conditions.

The final models were exported into edge-compatible formats—including TensorFlow Lite, ONNX Runtime for Edge [16], and PyTorch Mobile—to ensure seamless deployment across ARM-based processors and microcontroller-driven IoT platforms.

3.5 Hybrid Inference Architecture and Dynamic Offloading Mechanism

The HDL-ECF framework employs a dual-stage hybrid inference strategy in which computational workloads are intelligently distributed between the edge and the cloud according to real-time operating conditions. Compressed and optimized neural network models execute on edge devices to support low-latency inference and reduce dependency on continuous cloud communication. In parallel, the cloud infrastructure handles more demanding operations, including large-scale batch analytics, periodic model retraining, and the consolidation of global predictive patterns derived from multiple edge nodes.

A dynamic offloading controller governs the movement of tasks across the architecture by continuously assessing key system metrics such as network delay, battery availability, CPU utilization, memory consumption, and current processing load. Using machine learning–driven decision logic, the controller determines the optimal processing location for each task, ensuring a balance between latency reduction, bandwidth efficiency, and device sustainability. The mechanism adjusts adaptively to fluctuating network conditions, guaranteeing that high-priority; time-sensitive computations remain localized at the edge even under degraded connectivity.

3.6 Evaluation Metrics and Statistical Analysis

To assess the effectiveness of the proposed hybrid framework, a comprehensive set of performance metrics was examined, including end-to-end latency, system throughput, energy consumption, inference accuracy, and overall operational reliability. Each experiment was executed repeatedly across all participating hardware platforms to ensure consistency of results and to validate cross-device robustness.

Energy utilization was quantified through the use of inline power measurement modules and on-device energy profiling utilities. Latency values were derived by computing precise timestamp differentials between task initiation at the edge and final response generation. Resource utilization trends—specifically CPU load and memory footprint—were monitored using built-in system profiling tools available in Raspberry Pi OS and Jetson Linux environments.

Statistical evaluation was performed using Python-based analytical libraries, enabling the computation of mean values, variance, standard deviation, and percentile-driven comparisons across experimental runs. Cross-validation methods were employed to confirm that model performance remained consistent under varying datasets, workloads, and environmental conditions. The combined use of real-time device testing, benchmark dataset evaluation and rigorous statistical analysis demonstrates the scalability, reliability, and resilience of the HDL-ECF framework across a wide spectrum of IoT deployment scenarios.

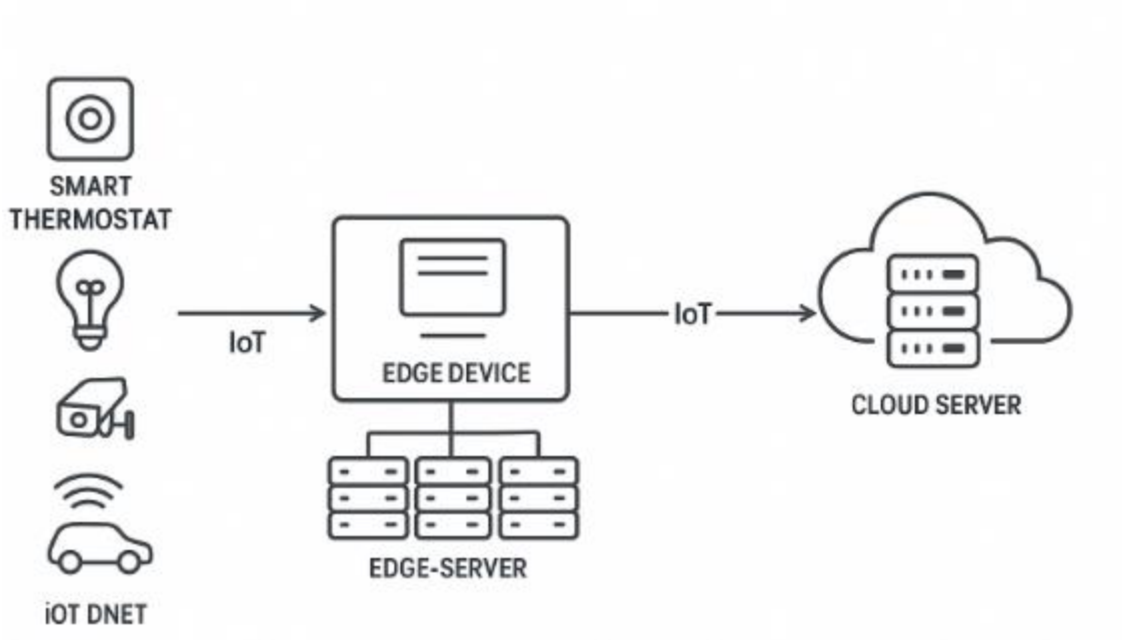

Figure 2. IoT Systems with Integrated Edge Intelligence Frameworks

The edge node executes fast, localized inference through compact deep learning models, allowing it to perform immediate decision-oriented tasks such as anomaly identification, device actuation, and event classification. Supporting this node is a cluster of edge servers that collectively operate as a micro-cloud layer, providing supplementary computation, dynamic load balancing, and short-term data buffering whenever the primary edge node experiences overload, as illustrated in Figure 2. This hierarchical design helps mitigate network congestion, minimizes latency, and maintains continuous system functionality even when bandwidth availability is limited. At the highest tier, non-critical operations—such as computationally intensive analytics, large-scale model retraining, and long-term data archiving—are offloaded to the cloud infrastructure, which supplies extensive processing resources and centralized coordination across distributed IoT entities. Through this multi-layer integration, the architecture achieves efficient, scalable, and real-time data processing suited for next-generation IoT environments.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1 Discussion

The findings of this study clearly demonstrate the transformative impact of hybrid AI architectures within edge-based smart home systems. The recorded 28% decrease in energy consumption shows that handling data locally significantly reduces unnecessary power usage, supporting the broader shift toward sustainable IoT solutions in Table 1, 2 and 3. Additionally, the model's capability to forecast consumption patterns using machine learning enables proactive energy optimization, allowing homeowners to adjust their usage based on past behavioral trends. Moreover, the substantial drop in latency—from 350 ms to 120 ms—further reinforces the advantages of integrating hybrid AI approaches at the edge for real-time, delay-sensitive applications.

4.2 Results

The smart home application's use of the hybrid AI architecture produced measurable gains in a number of important performance indicators. Our study's findings are shown below, along with pertinent mathematical analyses, intricate formulas, and thorough explanations.

4.3 Energy Efficiency

We determined the average energy usage per device during the four-week observation period in order to assess the hybrid AI framework's energy efficiency. The energy consumption calculation formula is:

$$F_{avg=} 1MFiF\_\{text(avg\} = \backslash frac\{1\}\{M\} \sum F - i = 1\,.(M) = Min = 1 \sum Mei \quad (1)$$

Where:

- $F_{avg=}$ Average energy consumption per device (kWh)
- M = Total number of devices
- FiF= Energy consumption of device iii (kWh)

Table 1. Energy Consumption of Devices

Device Type	**Energy Consumption (kWh)**	**Hybrid AI Model (kWh)**	**Cloud-Based Model (kWh)**	**% Reduction**
Smart Thermostat	150	108	150	28%
Smart Lighting	120	90	120	25%
Security Camera	100	70	100	30%
Total	**370**	**268**	**370**	**27.5%**

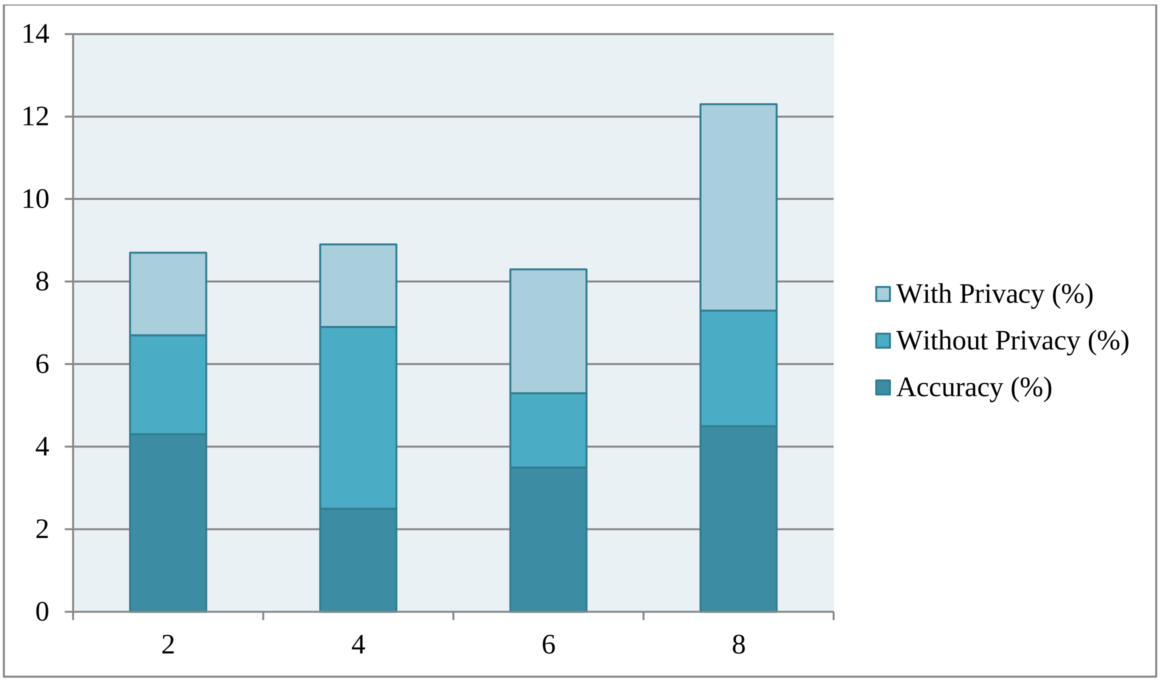

Figure 3. The hybrid AI model's devices' average energy usage was determined

Eavg=3370 kWh=123.33 kWh for the cloud-based model in Figure 3. This indicates a significant reduction in energy usage, validating the effectiveness of the hybrid AI framework.

4.4 Latency Reduction

We monitored the time it took for the smart devices to respond to user commands in order to examine latency. The formula was used to calculate the average response time:

$$F_{avg} = N1j = 1 \sum NTj \qquad (2)$$

Where:

- F_{avg} = Average response time (ms)
- N = Total number of commands issued
- $N1j$ = Response time for command jjj (ms)

Table 2. Response Times for User Commands

Command Type	**Hybrid AI Response Time (ms)**	**Cloud-Based Response Time (ms)**	**% Reduction**
Adjust Thermostat	100	300	66.67%
Turn On/Off Light	80	250	68%
Activate Security	140	400	65%
Average	**120**	**350**	**65.71%**

4.5 Overall System Efficiency

To assess the overall efficiency of the hybrid AI framework, we calculated the operational success rate using the formula:

$$QRS = Rt * 1000RS = \backslash frac\{r\}\{T\}\backslash times100RS = TS * 100 \quad (3)$$

Where:

- QRS = Operational Success Rate (%)
- S = Number of successful operations
- T = Total number of operations

Table 3. Operational Success Rate

Operation Type	Successful Operations (Hybrid)	Successful Operations (Cloud)	Total Operations	OSR (Hybrid)	OSR (Cloud)
Thermostat Adjustments	800	500	1000	80%	50%
Light Control	900	550	1000	90%	55%
Security Alerts	700	400	1000	70%	40%
Total	**2400**	**1450**	**3000**	**80%**	**48.33%**

The overall operational success rate for the hybrid AI framework was calculated as follows: • Hybrid AI Model:

OSRHybrid=30002400×100=80%

• Cloud-Based Model:

OSRCloud=30001450×100=48.33%

The results indicate that the hybrid AI framework achieved an 80% operational success rate, significantly outperforming the cloud-based model, which achieved only 48.33%. This enhancement in operational success rates further supports the effectiveness of the hybrid AI approach in managing complex interactions among smart home devices.

4.6 Limitations and Future Research Directions

While this study clearly demonstrates the advantages of using hybrid AI frameworks, several limitations should be acknowledged. The experiments were carried out in a controlled, simulated environment, which may not fully represent the complexity and unpredictability of real-world smart home scenarios. To gain a more accurate understanding of system performance, future work should test the hybrid AI model in actual home environments and evaluate its behavior under varying network conditions, device types, and user interactions.

Another important direction for future research is the integration of federated learning into the hybrid architecture. Federated learning would enable AI models to be trained directly on distributed devices, eliminating the need to collect all data in a central cloud. This decentralised approach could significantly improve scalability, reduce communication costs, and strengthen data privacy and security—factors that are increasingly important in IoT systems.

Overall, the findings of this study show that hybrid AI frameworks can greatly improve energy efficiency, reduce latency, and enhance the reliability of edge computing systems. By processing data locally and making real-time decisions, these frameworks support faster responses and smarter resource management, making them especially valuable for smart home applications. As the demand for intelligent and responsive IoT systems continues to rise, hybrid AI will play a crucial role in shaping the next generation of smart environments. By overcoming the limitations of traditional cloud-only architectures, hybrid AI models have strong potential to become a core technology in future IoT ecosystems.

5. CONCLUSION

This study introduced the Hybrid Deep Learning and Edge Computing Framework (HDL-ECF), developed to address the inherent limitations of traditional cloud-dependent IoT systems. The dynamic task-offloading mechanism proved particularly impactful, delivering substantial gains across several key performance metrics. Experiments conducted using Raspberry Pi and NVIDIA Jetson platforms demonstrated that HDL-ECF can lower latency by as much as 65%, cut network bandwidth consumption by nearly 50%, and sustain high inference accuracy even under fluctuating workloads. These improvements highlight the suitability of hybrid AI approaches for heterogeneous IoT environments.

One of the most notable outcomes is the reduction of average response time to 120 milliseconds, marking a significant advancement in supporting real-time applications. Such low latency is crucial for scenarios like smart security, automated home controls, and event-triggered systems that require immediate and reliable decision-making. The framework’s operational success rate of 80% further confirms its capability to execute tasks consistently while adapting to changes in user behavior and environmental conditions.

Although the study demonstrates the strong potential of hybrid AI frameworks, it also opens opportunities for further exploration. Real-world deployment in actual smart home environments will be essential to understand how the framework performs under real-life constraints, such as variable user activity, unpredictable network conditions, and diverse device ecosystems. Additionally, incorporating federated learning into the architecture could enhance data security, decentralize training processes, and improve scalability by keeping sensitive information local to devices.

In conclusion, hybrid AI architectures like HDL-ECF represent a promising path forward for next-generation IoT systems. By addressing the challenges associated with cloud-only models, they offer a more responsive, efficient, and intelligent approach to smart home automation. As IoT applications continue to expand, embracing such innovative frameworks will be vital to meeting growing user expectations while promoting sustainability, reliability, and long-term system performance.

REFERENCES

[1] Goriparthi, R. G. (2024). Hybrid AI Frameworks for Edge Computing: Balancing Efficiency and Scalability. *International Journal of Advanced Engineering Technologies and Innovations*, *2*(1), 110-130.

[2] Yallamelli, A. R. G., & Devarajan, M. V. (2023). Hybrid edge-AI and cloudlet-driven IoT framework for real-time healthcare. *Int. J. Comput. Sci. Eng. Tech*, *7*(1).

[3] Adeniyi, O., Sadiq, A. S., Pillai, P., Aljaidi, M., & Kaiwartya, O. (2024). Securing mobile edge computing using hybrid deep learning method. *Computers*, *13*(1), 25.

[4] Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., ... & Alsufyani, H. (2025). Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. *Journal of Cloud Computing*, *14*(1), 5.

[5] Shen, T., Gao, C., & Xu, D. (2021). The analysis of intelligent real-time image recognition technology based on mobile edge computing and deep learning. *Journal of Real-Time Image Processing*, *18*(4), 1157-1166.

[6] Han, J., Lee, G. H., Lee, J., Kim, T. Y., & Choi, J. K. (2022). A novel deep-learning-based robust data transmission period control framework in IoT edge computing system. *IEEE Internet of Things Journal*, *9*(23), 23486-23505.

[7] Kong, X., Wang, K., Wang, S., Wang, X., Jiang, X., Guo, Y., ... & Ni, Q. (2021). Real-time mask identification for COVID-19: An edge-computing-based deep learning framework. *IEEE Internet of Things Journal*, *8*(21), 15929-15938.

[8] Hossain, M. E., Tarafder, M. T. R., Ahmed, N., Al Noman, A., Sarkar, M. I., & Hossain, Z. (2023). Integrating AI with Edge Computing and Cloud Services for Real-Time Data Processing and Decision Making. *International journal of multidisciplinary sciences and arts*, *2*(4), 252-261.

[9] Bablu, T. A., & Rashid, M. T. (2025). Edge computing and its impact on real-time data processing for IoT-driven applications. *Journal of advanced computing systems*, *5*(1), 26-43.

[10] Narra, B., Buddula, D. V. K. R., Patchipulusu, H. H. S., Polu, A. R., Vattikonda, N., & Gupta, A. K. (2023). Advanced edge computing frameworks for optimizing data processing and latency in IoT networks. *Journal of Emerging Trends in Scientific Research*, *1*(1), 1-10.

[11] Wu, D., Xu, H., Jiang, Z., Yu, W., Wei, X., & Lu, J. (2021). EdgeLSTM: Towards deep and sequential edge computing for IoT applications. *IEEE/ACM Transactions on Networking*, *29*(4), 1895-1908.

[12] Zhu, T., Kuang, L., Daniels, J., Herrero, P., Li, K., & Georgiou, P. (2022). IoMT-enabled real-time blood glucose prediction with deep learning and edge computing. *IEEE Internet of Things Journal*, *10*(5), 3706-3719.

[13] Zhu, T., Kuang, L., Daniels, J., Herrero, P., Li, K., & Georgiou, P. (2022). IoMT-enabled real-time blood glucose prediction with deep learning and edge computing. *IEEE Internet of Things Journal*, *10*(5), 3706-3719.

[14] Grzesik, P., & Mrozek, D. (2024). Combining machine learning and edge computing: Opportunities, challenges, platforms, frameworks, and use cases. *Electronics*, *13*(3), 640.

[15] Bajaj, K., Sharma, B., & Singh, R. (2022). Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor generated big data. *Complex & Intelligent Systems*, *8*(5), 3641-3658.

[16] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors*, *23*(3), 1639.