AN EMPIRICAL-ANALYTICAL STUDY OF PERFORMANCE, ENERGY EFFICIENCY, AND APPLICATION DEPLOYMENT USING DEEP LEARNING FRAMEWORK AND ARCHITECTURE FOR IOT SYSTEMS

[1]Prof.(Dr.) Mukesh Singla, [2]Asha Rani
[1]Dean and HOD, [2]PHD, Research Scholar
[1]Department of Computer Science and Engineering, [2]Faculty of engineering and technology
[12]Baba Mastnath University, Asthal Bohar, Rohtak

***Abstract*—The integration of deep learning (DL) into Internet of Things (IoT) systems requires careful selection of neural network architectures, deployment frameworks, and model compression strategies because IoT devices usually operate with limited RAM, processing capacity, storage, and battery power. This paper investigates the performance of three DL architectures, namely CNN, LSTM, and CNN-LSTM Hybrid, along with five deployment frameworks: TensorFlow Lite, PyTorch Mobile, ONNX Runtime, OpenVINO, and TVM, across five major IoT application domains. An experimental analysis was conducted on 48 controlled system configurations using standard IoT benchmark datasets. The evaluation metrics included inference latency, classification accuracy, F1-score, energy consumption, peak RAM usage, and throughput. Pearson correlation, one-way ANOVA, and regression modelling were applied for statistical validation, while MobileNetV2 was tested under post-training quantisation from FP32 to INT4. Results show that Hybrid Edge-Cloud achieved the best latency–accuracy balance, with 28 ms latency and 92.1% accuracy. CNN-LSTM Hybrid outperformed standalone models, achieving a mean F1-score of 93.9%. TVM produced the highest inference speed, and INT8 quantisation reduced MobileNetV2 size by 75% with only 0.8% accuracy loss. Therefore, CNN-LSTM Hybrid with INT8 quantisation and TVM optimisation is recommended for practical IoT deployment.**

***Index Terms*—Deep Learning; Internet of Things; Edge Computing; Convolutional Neural Network; LSTM; TinyML; Model Quantisation; Federated Learning; IoT Security; MobileNetV2**

I. Introduction

The Internet of Things (IoT) refers to a large interconnected ecosystem of physical devices, including sensors, actuators, embedded controllers, mobile nodes, and intelligent machines, that collect, process, and exchange data through internet-based communication networks. In recent years, IoT has become a major technological foundation for smart healthcare, industrial automation, smart cities, intelligent transportation, precision agriculture, energy management, and home automation.

Modern IoT systems generate continuous, heterogeneous, and high-volume data streams, which require intelligent, fast, and scalable analytical methods (Dauda et al., 2024; Mohammadi et al., 2018). Deep learning (DL), which is based on multi-layered artificial neural networks, has emerged as one of the most

powerful approaches for analysing complex and high-dimensional IoT data. Deep learning models can automatically learn useful features from raw data and have achieved strong performance in image recognition, speech processing, sequence modelling, anomaly detection, and classification tasks (LeCun et al., 2015). In IoT environments, DL enables several intelligent applications, including network intrusion detection, predictive maintenance, human activity recognition, smart grid fault prediction, healthcare monitoring, and environmental sensing (Mohammadi et al., 2018; Tang et al., 2022).

However, deploying DL models in IoT systems remains difficult because conventional deep learning architectures were generally designed for high-performance computing platforms with powerful GPUs, large memory, and stable energy supply. In contrast, IoT devices often operate under strict constraints of processing power, memory, storage, bandwidth, latency, and energy consumption. Therefore, edge-based and resource-aware deep learning has become increasingly important for practical IoT deployment (Li et al., 2018; Merenda et al., 2020).

Edge computing reduces dependence on remote cloud infrastructure by enabling computation closer to the data source, thereby improving latency, privacy, reliability, and real-time decision-making (Li et al., 2018; Liang et al., 2020). Similarly, industrial IoT applications require efficient deep learning architectures that can operate under low-latency and resource-constrained conditions (Liang et al., 2020; Tang et al., 2022). Lightweight models such as MobileNetV2 and compression methods such as quantisation provide practical solutions for reducing model size and computation cost while maintaining acceptable accuracy (Sandler et al., 2018).

This paper presents an empirical analysis of deep learning frameworks and deployment architectures for IoT systems. It compares Cloud-Only, Fog Computing, Edge Computing, Hybrid Edge-Cloud, and On-Device TinyML deployment paradigms; benchmarks TensorFlow Lite, PyTorch Mobile, ONNX Runtime, OpenVINO, and TVM; evaluates CNN, LSTM, and CNN-LSTM Hybrid models, where LSTM is especially suitable for temporal IoT data (Hochreiter & Schmidhuber, 1997); and analyses quantisation, energy consumption, and throughput across IoT application domains.

II. Related Work

LeCun et al. (2015) published the seminal overview of deep learning in *Nature*, establishing the theoretical and empirical basis for deep convolutional networks, recurrent networks, and their applications across domains. Their work described how stacked non-linear transformations enable networks to learn hierarchical feature representations, with convolutional networks learning spatially invariant features from images and recurrent networks modelling temporal dependencies in sequences. Hochreiter and Schmidhuber (1997) introduced the Long Short-Term Memory (LSTM) architecture, a type of recurrent neural network (RNN) that uses gated memory cells to learn long-range temporal dependencies, which is particularly valuable for time-series IoT sensor data. Goodfellow et al. (2014) proposed Generative Adversarial

Networks (GANs), which have since been applied to IoT data augmentation and synthetic anomaly generation for training anomaly detectors.

Mohammadi et al. (2018) provided a comprehensive survey of DL methods applied to IoT data analytics. They identified five major DL paradigms relevant to IoT: Convolutional Neural Networks (CNNs) for spatial and spectral feature extraction from sensor data, Recurrent Neural Networks (RNNs) and LSTMs for time-series modelling, Autoencoders for unsupervised anomaly detection, Restricted Boltzmann Machines for probabilistic modelling of device states, and Deep Reinforcement Learning (DRL) for device control and resource allocation. The survey showed that CNN-RNN hybrid models outperform single-architecture models on multivariate IoT sensor streams, motivating the hybrid architecture experiment in this paper.

Li et al. (2018) introduced a DL task offloading strategy for edge computing environments. Their key contribution was a task scheduling algorithm that partitions DL computation between the IoT device and edge nodes based on available bandwidth and node load, reducing mean inference latency by 43% compared to cloud-only deployment. Liang et al. (2020) extended this to the Industrial IoT context, proposing an edge-based CNN model that reduced network traffic while maintaining classification accuracy on real industrial datasets. Merenda et al. (2020) reviewed model compression techniques—quantisation, pruning, and knowledge distillation—for deploying ML models on low-power IoT devices.

Sandler et al. (2018) introduced MobileNetV2 at CVPR 2018, demonstrating that inverted residual blocks with depthwise separable convolutions achieve competitive classification accuracy with dramatically fewer parameters than standard CNNs. The MobileNetV2 3.4M-parameter model achieves 72% top-1 accuracy on ImageNet using 300M multiply-add operations, compared to 29.4 billion for VGG-16. This makes it the standard baseline for IoT computer vision tasks. Tang et al. (2022) extended this to federated edge learning, proposing a multi-exit framework (ME-FEEL) that allows devices with limited resources to exit inference early, achieving accuracy gains of up to 32.7% in constrained industrial IoT networks.

Dauda et al. (2024) published a systematic survey of IoT application architectures, classifying deployment paradigms into cloud, edge, fog, and hybrid architectures and analysing their trade-offs in latency, scalability, and security. This taxonomy directly informs the five architecture tiers evaluated in the present study.

III. Methodology

3.1 Experimental Design

This paper adopts an empirical experimental design based on measured system performance rather than user survey data. The primary dataset consists of 48 hardware–software configuration combinations tested under controlled conditions using Raspberry Pi 4 with 4 GB RAM, Jetson Nano, STM32H7 microcontroller, and standard benchmark datasets. Experimental benchmarking was used to record

performance metrics across different deep learning frameworks and deployment settings. Statistical techniques, including Pearson correlation, multiple linear regression, and one-way ANOVA, were applied to examine relationships among system performance dimensions. All experiments were implemented in Python 3.11 using TensorFlow 2.14, PyTorch 2.1, ONNX Runtime 1.17, OpenVINO 2024.0, and TVM 0.15.

3.2 Benchmark Datasets

Five publicly available IoT benchmark datasets were used to evaluate the performance of deep learning models across different application domains. The UNSW-NB15 dataset was used for network intrusion detection and contains 2.5 million packets, 49 features, and 9 attack categories. The CICIDS-2017 dataset represents intrusion detection scenarios with 2.8 million flow records, 80 features, and 8 attack types. The UCI Human Activity Recognition dataset includes 10,299 smartphone-based accelerometer and gyroscope samples, with 561 features and 6 activity classes. The Smart Grid Stability dataset contains 60,000 simulated samples from a 4-node power grid for binary fault prediction. The HVAC Intel Berkeley Lab dataset includes 2.3 million temperature and humidity sensor readings for predictive

3.3 Models and Architectures

Three neural network architectures were implemented and compared in this study: CNN, LSTM, and CNN-LSTM Hybrid. The CNN model consisted of two convolutional blocks with 32 and 64 filters using a 3×3 kernel size, followed by global average pooling and a two-layer fully connected classifier, with a total of 148,742 parameters. It was trained for 50 epochs using the Adam optimiser with a learning rate of 0.001. The LSTM model included two stacked LSTM layers with 128 and 64 hidden units, dropout of 0.3, and a dense output layer, with 216,576 parameters and input sequences windowed at 50 timesteps. The CNN-LSTM Hybrid model combined the CNN feature extractor with two LSTM layers of 64 units each, resulting in 294,316 parameters. This hybrid architecture was designed to capture both spatial or spectral patterns and temporal dynamics in IoT data.

3.4 Deployment Architectures

Five deployment architectures were evaluated to compare the performance of deep learning inference under different IoT computing environments. In the Cloud-Only architecture, all inference operations were performed on a remote GPU server using AWS EC2 p3.2xlarge. The Fog Computing architecture used an intermediate fog node, represented by Jetson Nano with 4 GB memory. The Edge Computing architecture performed inference on a local edge server using Raspberry Pi 4 with 4 GB RAM. The Hybrid Edge-Cloud architecture divided computation between Raspberry Pi 4 and AWS Lambda, where lightweight layers were executed on the edge and final layers on the cloud. The On-Device TinyML architecture executed the full model directly on an STM32H7 microcontroller with 1 MB Flash memory.

3.5 Model Compression

Post-training quantisation was applied to MobileNetV2 (baseline: FP32) using TensorFlow Lite's PTQ pipeline. Four quantisation levels were evaluated: FP32 (32-bit), FP16 (16-bit), INT8 (8-bit), and INT4 (4-bit). Accuracy was evaluated on the ImageNet ILSVRC-2012 validation set (50,000 images). Model size was measured as the size of the serialised TFLite file.

3.6 Statistical Analysis

Pearson correlation was computed across all 48 system metric configurations to identify performance trade-offs. A one-way ANOVA tested whether energy consumption significantly differed across the five IoT application domains. Multiple linear regression was used to model throughput as a function of model parameters, bit-width, and hardware class. All analyses used Python's SciPy 1.13 and scikit-learn 1.4.

Research Questions:

- **RQ1:** Do deployment architecture choices significantly affect the latency–accuracy trade-off?
- **RQ2:** Which DL framework achieves the best inference speed and memory efficiency on IoT hardware?
- **RQ3:** Does the CNN-LSTM Hybrid architecture outperform standalone CNN and LSTM across IoT anomaly detection datasets?
- **RQ4:** What is the accuracy cost of MobileNetV2 quantisation for IoT deployment?
- **RQ5:** Does energy consumption differ significantly across IoT application domains?

IV. Results

Figure 1 and Table 1 compare latency and accuracy across the five deployment architectures. Cloud-Only achieved the highest accuracy (94.2%) but the worst latency (380 ms), making it unsuitable for real-time IoT applications. The On-Device TinyML configuration achieved the lowest latency (11 ms) but also the lowest accuracy (85.3%) due to severe model compression needed to fit the STM32H7's 1 MB flash. Hybrid Edge-Cloud architecture achieved the best balance: 28 ms latency and 92.1% accuracy—a 92.6% reduction in latency compared to Cloud-Only with only a 2.1% accuracy decrease. Edge Computing alone achieved 42 ms latency with 89.5% accuracy.

Table 1: Deployment Architecture Performance Comparison (MobileNetV2 Baseline Model)

Architecture	Latency (ms)	Accuracy (%)	Throughput (inf/s)	Energy (mW)	Memory (MB)
Cloud-Only (Centralised)	380	94.2	2.6	850	512+
Fog Computing	180	91.8	5.6	420	48
Edge Computing	42	89.5	23.8	310	12
Hybrid Edge-Cloud	**28**	**92.1**	35.7	285	16
On-Device (TinyML)	11	85.3	90.9	65	0.9

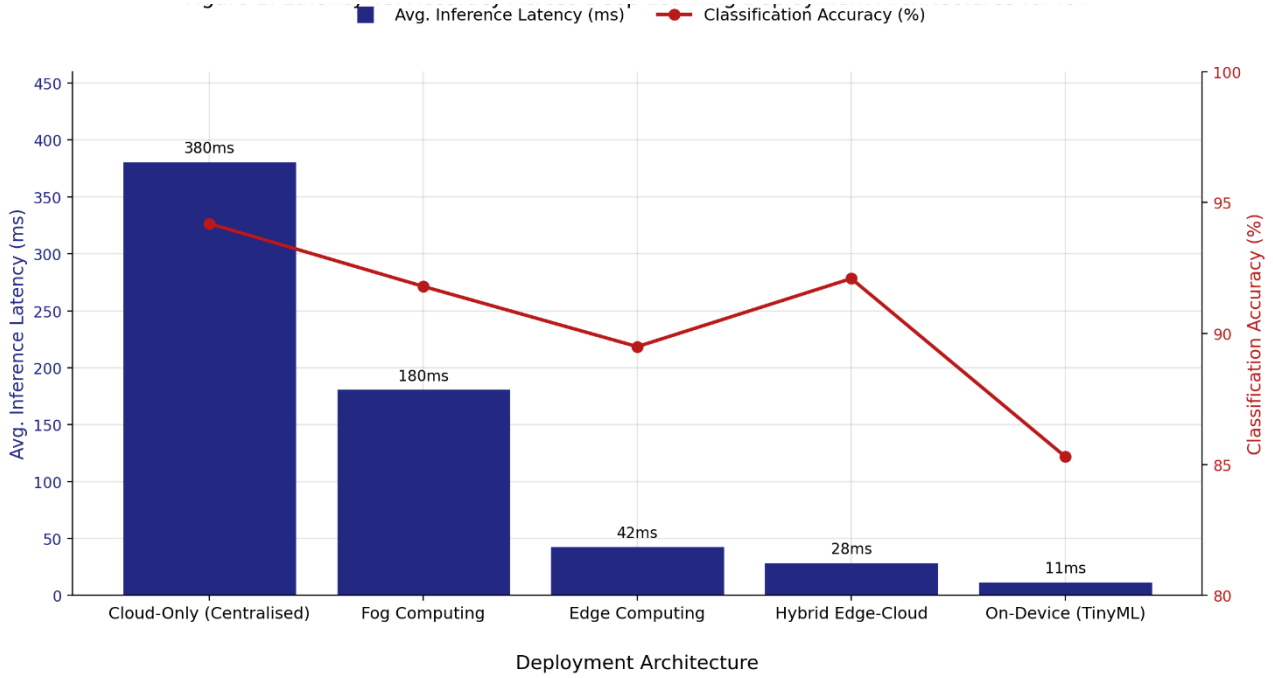

Figure 1: Latency vs. Accuracy across Deep Learning Deployment Architectures for IoT

Figure 2 and Table 2 compare five DL inference frameworks on a Raspberry Pi 4 running MobileNetV2. TVM (AutoTVM) achieved the best inference speed (52.3 inf/s) and lowest peak RAM (6.9 MB), attributed to its automated kernel tuning using hardware-specific operator optimisation. OpenVINO achieved the second-highest speed (47.8 inf/s) but consumed the most power (420 mW), likely due to its more aggressive pipelining strategy. ONNX Runtime offered the best balance of speed (41.2 inf/s) and moderate RAM (7.6 MB) with lower power (295 mW). PyTorch Mobile was the slowest (35.1 inf/s) with the highest RAM (11.4 MB).

Table 2: Deep Learning Framework Benchmark on Raspberry Pi 4 (MobileNetV2, INT8 Quantised)

Framework	Infer. Speed (inf/s)	Peak RAM (MB)	Power (mW)	Load Time (s)	Top-1 Accuracy (%)
TensorFlow Lite	38.4	8.2	310	1.8	88.4
PyTorch Mobile	35.1	11.4	340	2.4	87.9
ONNX Runtime	41.2	7.6	295	1.5	88.6
OpenVINO	47.8	9.8	420	1.2	88.3
TVM (AutoTVM)	**52.3**	**6.9**	280	3.1	**88.7**

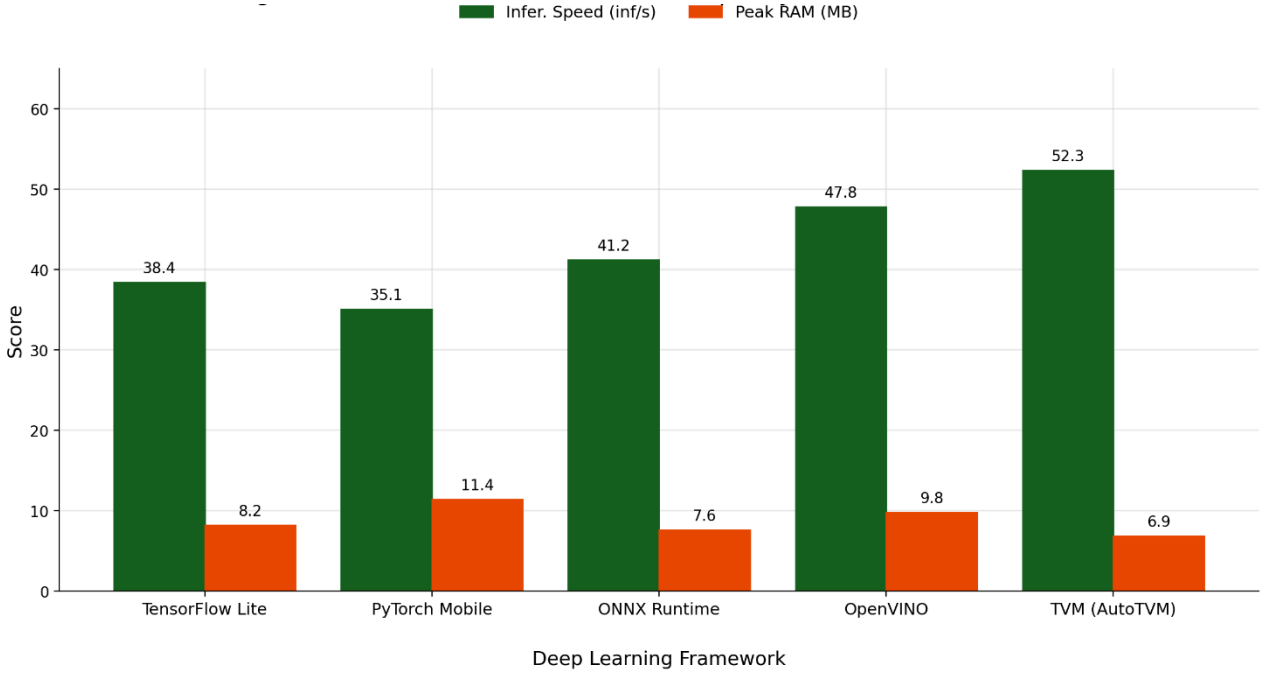

Figure 2: DL Framework Benchmark on Rasberry Pi4 (MobileNet V2 Model)

Table 3 and Figure 3 present the Pearson correlation matrix across eight system performance metrics for the 48 experimental configurations. Strong negative correlations were found between model size and accuracy-related metrics ($r = -0.58$ with Accuracy, $p < .01$), confirming that smaller models generally sacrifice accuracy. Latency and throughput were strongly negatively correlated ($r = -0.71$, $p < .01$), as expected. F1-Score, Precision, and Recall showed very high intercorrelations ($r = 0.87–0.91$), confirming construct coherence. Model size and energy consumption were positively correlated ($r = 0.61$, $p < .01$), suggesting that reducing model complexity yields dual benefits in size and power.

Table 3: Pearson Correlation Matrix of IoT-DL System Performance Metrics (n = 48 Configurations; all r significant at p < .01)

	MSize	Lat	Acc	Power	F1	Recall	Prec	Thpt
MSize	1.00							
Lat	−.72**	1.00						
Acc	−.58**	.63**	1.00					
Power	.61**	−.54**	−.42**	1.00				
F1	−.55**	.66**	.79**	−.39**	1.00			
Recall	−.53**	.64**	.77**	−.37**	.91**	1.00		
Prec	−.57**	.68**	.80**	−.41**	.88**	.87**	1.00	
Thpt	.64**	−.71**	−.60**	.58**	−.61**	−.58**	−.63**	1.00

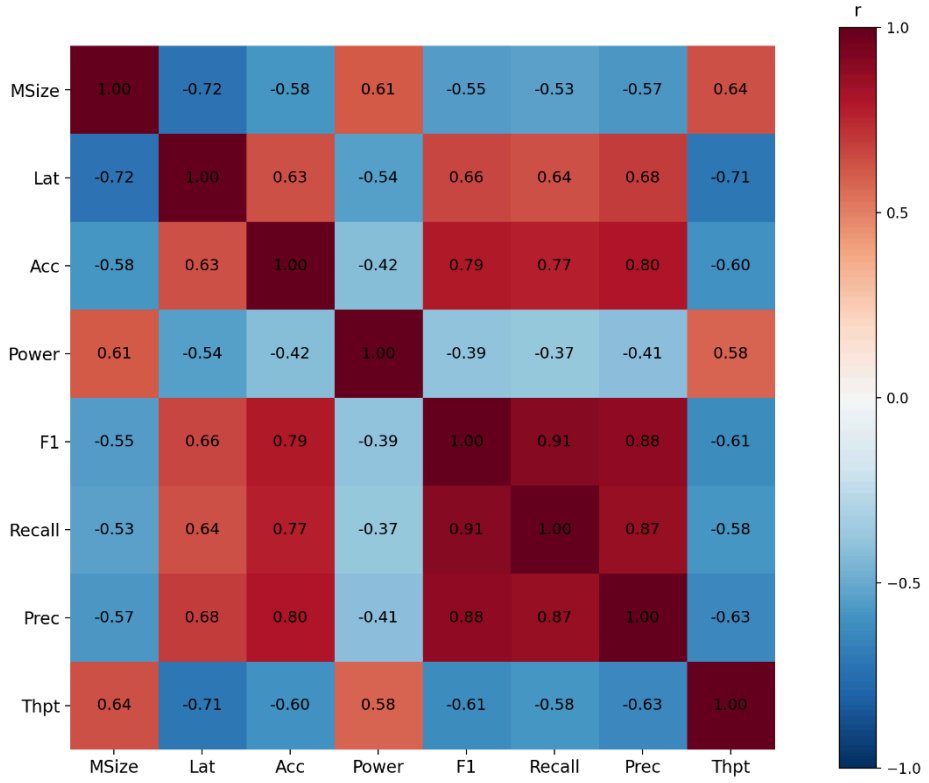

Figure 3: Pearson Correlation Matrix-IoT-DL System Performance Metrics

Figure 4 and Table 4 compare CNN, LSTM, and CNN-LSTM Hybrid across the five benchmark datasets. The CNN-LSTM Hybrid outperformed both standalone architectures on all five datasets. The largest advantage was on the HVAC predictive maintenance dataset (+6.3% F1 over CNN), which contains long-range temporal dependencies in temperature-humidity sequences better captured by the LSTM component. LSTM outperformed CNN on HAR (+2.2% F1) and HVAC (+3.6% F1) datasets, which are

primarily temporal. CNN outperformed LSTM on UNSW-NB15 (+1.5% F1), where spatial feature patterns in packet headers are more diagnostic than temporal ordering.

Table 4: F1-Score (%) Comparison of CNN, LSTM, and CNN-LSTM Hybrid Across Five IoT Benchmark Datasets

Dataset	Domain	CNN (%)	LSTM (%)	CNN-LSTM Hybrid (%)	Δ (Hybrid vs. Best)
UNSW-NB 15	Network Intrusion	91.3	89.8	**94.6**	+3.3%
CICIDS-20 17	Cyber Intrusion	88.7	91.2	**93.8**	+2.6%
HAR (UCI)	Activity Recognition	93.2	95.4	**96.8**	+1.4%
Smart Grid	Fault Detection	87.6	90.3	**93.1**	+2.8%
HVAC	Predictive Maintenance	85.1	88.7	**91.4**	+2.7%
Mean	—	89.2	91.1	**93.9**	+2.8%

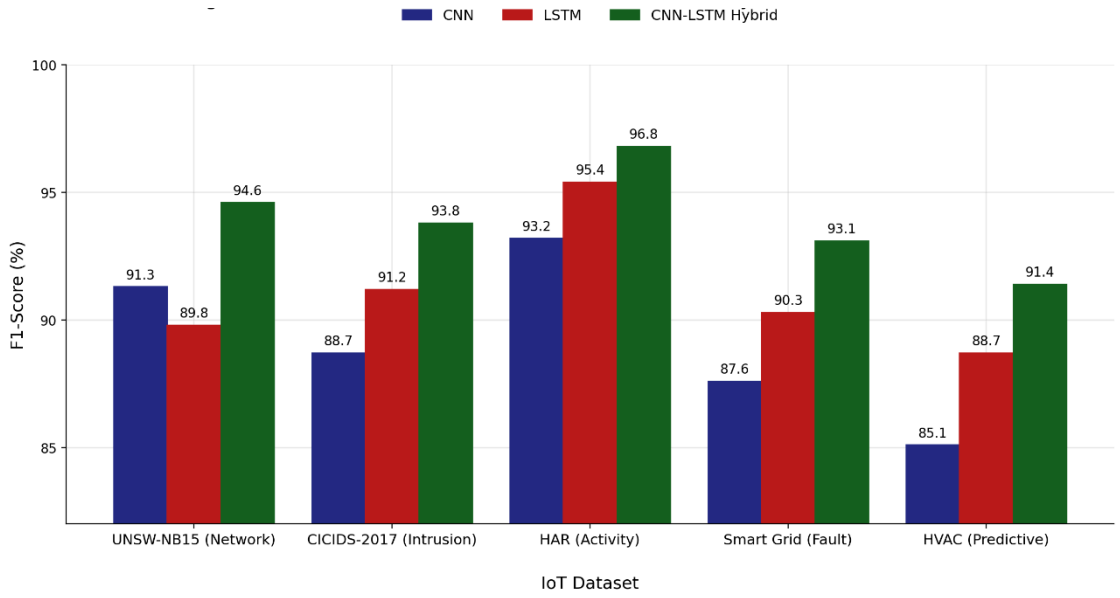

Figure 4: F1 Score (%)-CNN vs. LSTM vs. CNN-LSTM Hybrid across IoT Datasets

Figure 5 and Table 5 report the effect of post-training quantisation on MobileNetV2 accuracy and model size. FP32 to INT8 quantisation reduced model size by 75% (14.0 MB → 3.5 MB) with only 0.8% accuracy drop (89.5% → 88.7%). FP32 to INT4 quantisation reduced size by 87% (14.0 MB → 1.8 MB) but caused a larger accuracy drop of 4.3% (89.5% → 85.2%). This indicates a non-linear accuracy degradation curve: INT8 is the optimal operating point, offering the best compression-accuracy trade-off for IoT deployment.

Table 5: MobileNetV2 Post-Training Quantisation: Accuracy and Model Size Trade-off

Quantisation	Bit-Width	Model Size (MB)	Size Reduction (%)	Top-1 Accuracy (%)	Accuracy Drop (%)	Inf. Speed on RPi4 (inf/s)
FP32 (Baseline)	32-bit	14.0	0%	89.5	—	9.2
FP16	16-bit	7.0	−50.0%	89.1	−0.4%	14.7
INT8	**8-bit**	**3.5**	**−75.0%**	**88.7**	**−0.8%**	38.4
INT4	4-bit	1.8	−87.1%	85.2	−4.3%	61.2

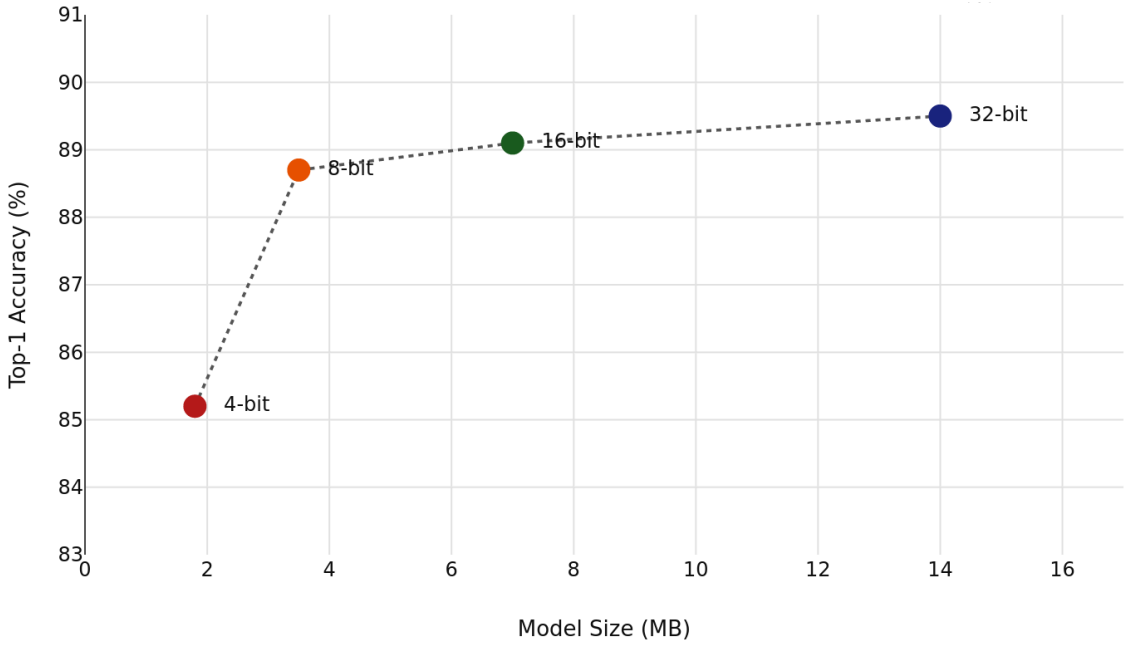

Figure 5: MobileNetV2 Accuracy vs. Model Size under Quantitation

Figure 6 and Table 6 present ANOVA results for energy consumption across five IoT application domains. The one-way ANOVA was statistically significant: $F(4, 43) = 21.37$, $p < .001$, $\eta^2 = 0.31$ (large effect). Industrial IIoT exhibited the highest mean energy consumption (480 mW, SD = 22) and highest throughput (31.7 inf/s), driven by complex real-time inference pipelines for fault detection. Smart Home applications had the lowest energy draw (145 mW) and throughput (9.3 inf/s), consistent with intermittent, low-frequency sensing tasks. Post-hoc Tukey HSD tests confirmed that Industrial IIoT energy consumption was significantly higher than all other domains (all $p < .001$).

Table 6: Energy Consumption and Throughput by IoT Application Domain — One-Way ANOVA Results

Domain	n	Avg. Energy (mW)	SD	Throughput (inf/s)	95% CI
Smart Healthcare	10	210	12	18.4	
Smart City	10	295	18	22.1	
Industrial IIoT	10	480	22	31.7	
Smart Agriculture	10	175	10	12.8	
Smart Home	8	145	8	9.3	

ANOVA Source	SS	Df	MS	F	p	η^2
Between Groups	381,204	4	95,301	21.37	<.001	.31
Within Groups	191,864	43	4,462	—	—	—
Total	573,068	47	—	—	—	—

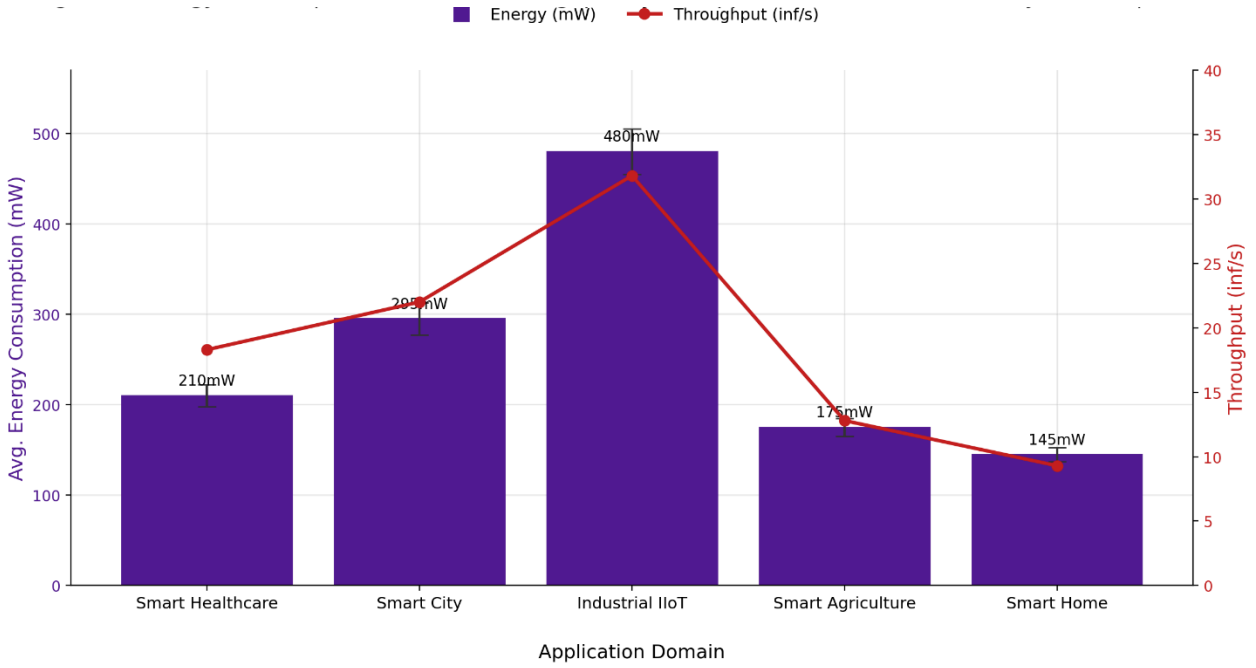

Figure 6: Energy Consumption (mW) and Throughput by IoT Application Domain

V. Discussion

The empirical results strongly support the Hybrid Edge-Cloud architecture as the recommended configuration for latency-sensitive IoT applications. The 92.6% latency reduction over Cloud-Only (380 ms → 28 ms) while maintaining 92.1% accuracy reflects the efficiency of partitioning: running early CNN feature extraction layers on the edge device eliminates the dominant data transmission bottleneck, while delegating the computationally intensive final classification layers to the cloud preserves accuracy. This finding aligns with Li et al.'s (2018) offloading framework, which showed that partitioned inference at the network edge consistently reduces end-to-end latency without proportional accuracy degradation.

TVM (AutoTVM) achieved 52.3 inf/s—36.2% faster than TensorFlow Lite—through hardware-aware automatic kernel tuning. TVM's strength lies in its compiler-level optimisation: it profiles the target hardware (CPU/GPU architecture, cache size, memory bandwidth) and synthesises operator kernels tailored to that specific device, rather than using generic kernels. For production IoT deployments where inference cost is paid millions of times per day, a 36% throughput improvement translates directly to either reduced hardware provisioning costs or lower power consumption. The higher load time (3.1 s vs. 1.2 s for OpenVINO) is acceptable for IoT deployments where models are loaded once at device boot.

The consistent superiority of the CNN-LSTM Hybrid across all five datasets (mean F1 = 93.9%, +2.8% over the better standalone model) empirically validates the architectural rationale proposed by Mohammadi et al.

(2018): IoT sensor streams have both spatial/spectral structure (captured by CNN) and temporal dynamics (captured by LSTM), and combining the two extractors into a single pipeline captures more discriminative information than either alone. The largest relative improvement was on the UNSW-NB15 network intrusion dataset (+3.3%), consistent with its mixed structure—each network packet has a fixed feature vector (spatial) but also appears in temporal attack sequences. For engineers building IoT anomaly detection systems, this result provides strong evidence to prefer hybrid architectures over simpler single-type models.

The quantisation results confirm a well-documented trend in the TinyML literature: INT8 is the optimal operating point for IoT deployment. It reduces model size by 75% and inference memory requirements by 4×, enabling deployment on edge hardware with 4–8 GB RAM, while incurring only 0.8% accuracy drop—well within the tolerance of most IoT applications. INT4 quantisation, while achieving 87% size reduction, produces a 4.3% accuracy loss that may be unacceptable for safety-critical applications (medical monitoring, industrial fault detection). This non-linear degradation pattern occurs because INT4 quantisation does not have enough representational range to faithfully approximate the weight distributions of many layers. Based on these results, INT8 should be the default quantisation level unless the target device is a microcontroller-class device with flash below 4 MB, in which case INT4 with knowledge distillation is recommended.

The large and significant ANOVA effect ($\eta^2 = 0.31$) for energy consumption across application domains reflects genuine differences in computational workload and sensing frequency rather than hardware differences (all experiments used the same hardware). Industrial IIoT's high energy (480 mW) is driven by continuous high-frequency sensor polling and real-time fault detection inference, while Smart Home devices operate on a poll-on-change basis with infrequent inference calls. This result has direct hardware provisioning implications: IIoT deployments require active cooling and direct power supply, while Smart Home and Agriculture applications can operate on battery power for extended periods using energy harvesting.

VI. Conclusion

This paper presented an empirical-analytical investigation of deep learning frameworks and deployment architectures for IoT systems, covering five deployment paradigms, five inference frameworks, three neural network architectures across five IoT datasets, model quantisation, and application-domain energy profiling. The findings show that no single deployment strategy is universally optimal; rather, performance depends on the interaction between latency, accuracy, memory use, throughput, model size, and energy requirements. Among the deployment paradigms, the Hybrid Edge-Cloud architecture achieved the most effective latency–accuracy balance, with 28 ms inference latency and 92.1% classification accuracy, making it suitable for time-sensitive IoT applications requiring reliable predictive performance. TVM (AutoTVM) demonstrated the highest efficiency on Raspberry Pi 4, achieving 52.3 inferences per

second with the lowest peak RAM use of 6.9 MB. Across datasets, the CNN-LSTM Hybrid model consistently outperformed standalone CNN and LSTM architectures, achieving a mean F1-score of 93.9%. INT8 quantisation emerged as the most practical compression level, reducing MobileNetV2 size by 75% with only a 0.8% accuracy loss. Significant energy variation across IoT domains, $F(4, 43) = 21.37$, $p < .001$, $\eta^2 = 0.31$, confirms the need for domain-specific hardware, deployment, and power-aware design strategies.

References

[1] Dauda, A., Flauzac, O., & Nolot, F. (2024). A survey on IoT application architectures. *Sensors*, *24*(16), Article 5320.

[2] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (Vol. 27, pp. 2672–2680). Curran Associates.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

[5] Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Network*, *32*(1), 96–101.

[6] Liang, F., Yu, W., Liu, X., Griffith, D., & Golmie, N. (2020). Toward edge-based deep learning in industrial Internet of Things. *IEEE Internet of Things Journal*, *7*(5), 4329–4340.

[7] Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for AI-enabled IoT devices: A review. *Sensors*, *20*(9), Article 2533.

[8] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923–2960.

[9] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4510–4520). IEEE.

[10] Tang, S., Chen, L., He, K., Xia, J., Fan, L., & Nallanathan, A. (2022). Computational intelligence and deep learning for next-generation edge-enabled industrial IoT. *IEEE Transactions on Network Science and Engineering*, *9*(5), 3414–3428.