International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 10 Issue 7 July - 2026, Pages: 202-209

Supervised Deep Learning for Real-Time Big Data Streams in IoT Networks: A Review of Latency-Accuracy Trade-offs

Virendra Tank

Shri Mahaveer College, Jaipur, Rajasthan, India
vtank87@gmail.com 0009-0003-9126-0982

Abstract: *The rapid expansion of Internet of Things (IoT) networks has produced continuous, high-velocity data streams that demand supervised deep learning models capable of making accurate predictions within strict real-time latency budgets. Unlike static big-data analytics, real-time IoT streaming applications – such as autonomous navigation, video surveillance, and industrial anomaly detection – must jointly optimize two often conflicting objectives: minimizing end-to-end latency and maximizing predictive accuracy. This review paper systematically examines the latency-accuracy trade-off in supervised deep learning for real-time big data streams in IoT networks. It surveys four major families of optimization strategies: model compression (pruning, quantization, and knowledge distillation), dynamic and early-exit inference architectures, edge-cloud cooperative offloading, and streaming-aware evaluation and forecasting methods. Representative studies are analyzed and compared in terms of latency reduction, accuracy impact, and suitability for different IoT application domains. The paper further discusses open challenges, including the mismatch between offline accuracy metrics and real-world streaming performance, resource constraints on edge hardware, and network variability. It concludes with future research directions such as streaming-native model architectures, reinforcement-learning-based adaptive offloading, and standardized streaming-accuracy benchmarks for IoT deep learning systems.*

Keywords: Deep Learning; Real-Time Systems; Big Data Streams; Internet of Things; Latency-Accuracy Trade-off; Edge Computing; Model Compression; Streaming Perception.

1. Introduction

The Internet of Things (IoT) has transformed physical environments into continuously sensed and monitored systems, generating big data streams characterized by high volume and, critically, high velocity. Many of the most valuable IoT applications – autonomous vehicles, industrial safety monitoring, smart surveillance, and robotics – are not merely big-data problems but real-time problems, where predictions must be delivered within strict latency budgets to remain useful [1]. A traffic-hazard detection made a second too late, or an industrial fault alert delivered after damage has occurred, provides little practical value regardless of its offline accuracy.

Supervised deep learning models, particularly convolutional and recurrent architectures, have become the default approach for extracting predictive value from such streams due to their superior accuracy on perception and classification tasks [1]. However, achieving high accuracy typically requires deeper, more computationally expensive networks, which directly conflicts with the low-latency requirements of real-time IoT deployments. This tension – the latency-accuracy trade-off – is the central problem addressed in this review.

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 10 Issue 7 July - 2026, Pages: 202-209

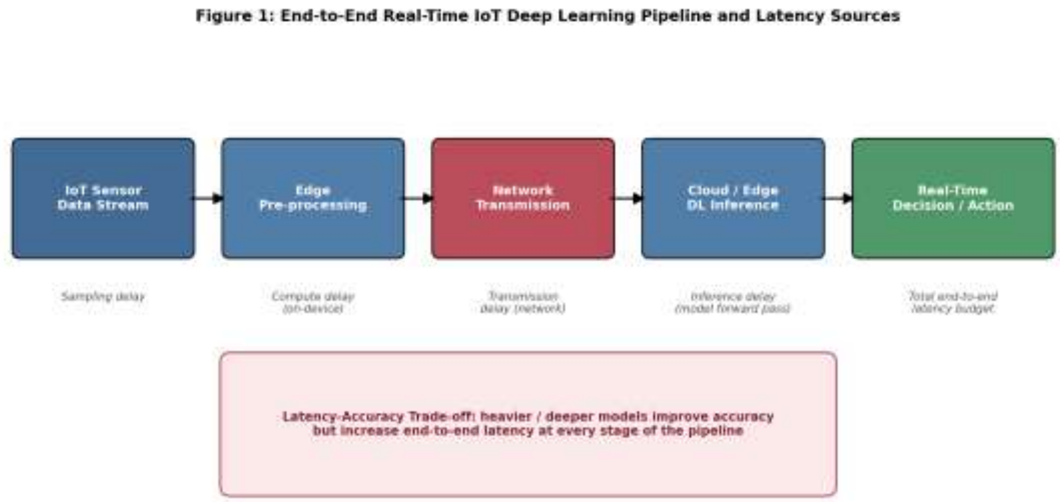

Figure 1: End-to-End Real-Time IoT Deep Learning Pipeline and Latency Sources

As illustrated in Figure 1, latency in a real-time IoT deep learning pipeline accumulates across multiple stages: on-device sampling and pre-processing, network transmission, model inference, and action generation. Optimizing only the inference stage in isolation, without considering the full pipeline, often yields limited real-world benefit. This review therefore surveys techniques spanning the entire pipeline: model-level compression, architecture-level dynamic inference, system-level edge-cloud offloading, and evaluation-level streaming-aware metrics.

The remainder of this paper is organized as follows. Section 2 introduces background concepts. Section 3 describes the review methodology. Section 4 surveys latency-accuracy optimization techniques in detail. Section 5 discusses applications across IoT domains. Section 6 presents a comparative analysis of results reported in the literature. Section 7 discusses open challenges, Section 8 outlines future research directions, and Section 9 concludes the paper.

2. Background and Related Concepts

2.1 Deep Learning for Big Data Streams

Deep learning architectures such as convolutional neural networks (CNNs) and recurrent networks including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models are widely used to process the spatial and temporal patterns present in IoT sensor streams [4]–[7]. More recently, attention-based Transformer architectures have also been applied to sequential streaming data due to their ability to model long-range temporal dependencies [8].

2.2 Real-Time Systems and IoT Big Data

Real-time IoT systems are distinguished from conventional big data analytics by the presence of a hard or soft latency deadline: predictions delivered after this deadline lose value, sometimes catastrophically, as in autonomous driving or industrial safety applications [1]. IoT big data is generated continuously by heterogeneous sensors and must often be processed close to the data source – at the network edge – to avoid the transmission delays associated with centralized cloud processing [9].

2.3 The Latency-Accuracy Trade-off

The latency-accuracy trade-off refers to the empirical observation that, for a fixed hardware budget, increasing model accuracy (through greater depth, width, or input resolution) generally increases inference latency, and vice versa [17]. Standard offline evaluation protocols, which assume the input world is static while the model computes, do not capture this trade-off realistically: in a true streaming setting, the world continues to change while inference is in progress, meaning that by the time a prediction is produced, it may already be outdated [19]. This insight motivates streaming-aware evaluation metrics discussed later in this review.

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 10 Issue 7 July - 2026, Pages: 202-209

3. Review Methodology

This review follows a structured literature synthesis approach. Relevant studies were identified through keyword searches across IEEE Xplore, the ACM Digital Library, SpringerLink, ScienceDirect, and arXiv using combinations of the terms “real-time deep learning,” “streaming data,” “latency-accuracy trade-off,” “Internet of Things,” “edge inference,” and “model compression.” Studies published between 2012 and 2026, written in English, and either peer-reviewed or hosted as citable preprints on arXiv were considered eligible.

Titles and abstracts were first screened to remove duplicates and studies unrelated to real-time or latency-constrained supervised deep learning in IoT or closely related streaming/edge-computing contexts. Full texts of the remaining candidates were then reviewed against inclusion criteria requiring a clear methodological contribution to model compression, dynamic inference, edge-cloud offloading, or streaming-aware evaluation, or a direct application to real-time IoT big data streams. This process yielded 25 core studies, which were thematically grouped by technique family for qualitative and comparative synthesis, as illustrated in Figure 2.

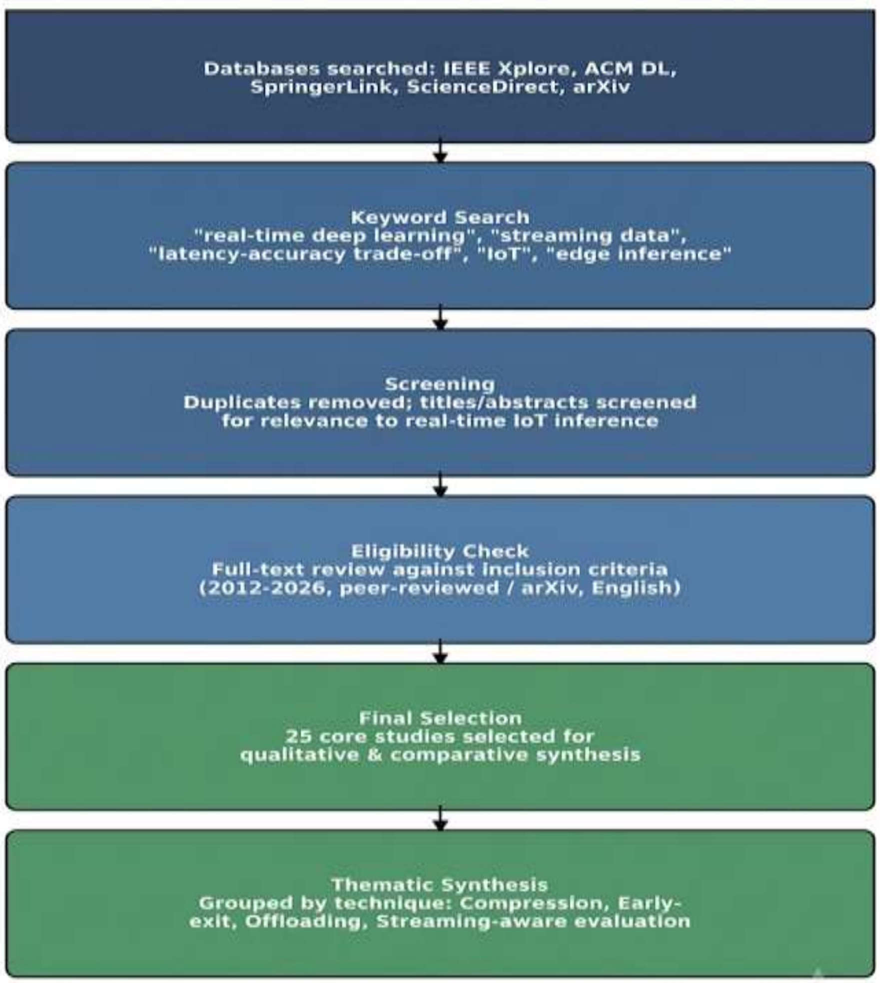

Figure 2: Review Methodology Workflow

4. Latency-Accuracy Optimization Techniques

Figure 3 presents a taxonomy of the latency-accuracy optimization techniques surveyed in this paper, organized into four major families that are discussed in the following subsections.

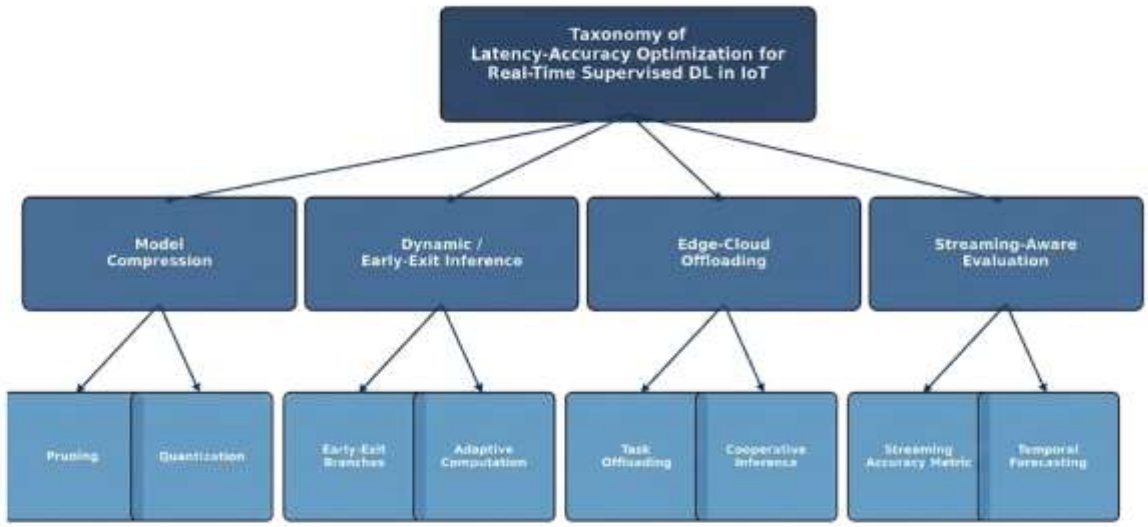

Figure 3: Taxonomy of Latency-Accuracy Optimization Techniques for Real-Time IoT Deep Learning

4.1 Model Compression: Pruning, Quantization, and Distillation

Model compression reduces the computational footprint of a trained network without redesigning its overall architecture. Pruning removes redundant weights, filters, or channels that contribute little to prediction accuracy, while quantization reduces numerical precision – for example, from 32-bit floating point to 8-bit integer representations – to speed up computation and reduce memory bandwidth [13], [15]. The classic Deep Compression pipeline combined pruning, trained quantization, and Huffman coding to shrink network storage by up to 35-49x without accuracy loss [13]. Knowledge distillation, introduced by Hinton et al., trains a smaller 'student' network to reproduce the soft output distribution of a larger 'teacher' network, enabling competitive accuracy at a fraction of the computational cost [14]. In IoT settings, communication-aware compression frameworks such as Network of Neural Networks (NoNN) partition a large teacher model into multiple lightweight student modules distributed across memory-constrained IoT devices, achieving up to 24x memory footprint reduction and up to 33x latency reduction relative to state-of-the-art baselines [16].

4.2 Dynamic and Early-Exit Inference

Dynamic inference architectures adapt the amount of computation performed per input based on its difficulty, rather than applying a fixed computational path to every sample. BranchyNet pioneered this idea by attaching auxiliary classifier branches at intermediate network layers: samples that can be classified with sufficient confidence exit early, avoiding the cost of deeper layers, while only genuinely difficult samples proceed through the full network [12]. This approach both reduces average inference latency and, in some cases, improves accuracy by preventing overfitting on easy examples. Such architectures are particularly well suited to IoT streaming scenarios where input difficulty varies significantly over time, for example between routine sensor readings and rare anomalous events.

4.3 Edge-Cloud Cooperative Offloading

Edge-cloud offloading strategies dynamically partition inference workloads between resource-constrained edge devices and more powerful cloud servers to balance latency, bandwidth cost, and accuracy. The SurveilEdge system exemplifies this approach for large-scale surveillance video streams, using an intelligent task allocator that balances load across edge and cloud resources; compared to cloud-only processing, it achieves up to 5.4x faster query response times and up to 7x lower bandwidth cost, while compared to edge-only processing it improves query accuracy by up to 43.9% [18]. More broadly, model offloading frameworks combine model compression, model distillation, and transmission compression to jointly optimize inference latency, data privacy, and monetary cost across the edge-cloud continuum.

4.4 Streaming-Aware Evaluation and Forecasting

A key insight from recent research is that standard offline accuracy metrics do not reflect true real-time performance, because they implicitly assume the world is frozen while the model computes. The streaming perception framework addresses this by evaluating a model's output against the state of the world at every time instant, coherently integrating latency and accuracy into a single 'streaming accuracy' metric [19]. This reveals that there exists an optimal 'sweet spot' along the latency-accuracy Pareto curve, and that asynchronous tracking and future forecasting can naturally emerge as strategies for maintaining accuracy under strict latency budgets [19], [20]. Reinforcement-learning-based adaptive streaming perception further extends this idea by dynamically selecting scale and skipping strategies to optimize streaming accuracy under varying compute availability [21].

5. Applications in Real-Time IoT Domains

5.1 Autonomous Vehicles and Robotics

Autonomous driving perception systems must process multi-modal sensor streams (cameras, LIDAR, radar) and make control decisions within milliseconds, since a delayed detection is functionally equivalent to a missed one [1]. Multi-resolution end-to-end networks that adapt input resolution to the available compute and latency budget have demonstrated consistent safety improvements over fixed-resolution baselines in urban driving benchmarks such as CARLA [17]. Streaming perception challenges specifically targeting autonomous driving have further validated that models optimized for streaming accuracy – rather than offline accuracy alone – achieve materially better real-world performance [22].

5.2 Smart City Video Surveillance

Large-scale camera networks in smart cities generate massive, continuous video streams that must be queried and analyzed in real time for applications such as public safety and traffic management. Collaborative cloud-edge systems like SurveilEdge demonstrate that intelligently partitioning CNN training and inference workloads across edge devices and the cloud can simultaneously reduce latency and bandwidth costs while improving accuracy relative to single-tier solutions [18].

5.3 Industrial IoT and Anomaly Detection

Industrial IoT deployments rely on continuous monitoring of machinery and processes, where anomalies or faults must be detected with minimal delay to prevent damage or safety incidents. Early-exit and dynamic inference architectures are particularly valuable here, since the vast majority of sensor readings represent normal operating conditions that can be classified quickly, reserving deeper computation for the rare, ambiguous, or anomalous cases [12].

5.4 Wearable and Healthcare IoT

Wearable health-monitoring devices operate under severe energy and compute constraints while requiring timely detection of critical physiological events. Model compression techniques – pruning, quantization, and knowledge distillation – enable deep learning models to run directly on such resource-constrained hardware, reducing both latency and energy consumption while preserving diagnostic accuracy [13]–[16].

6. Results and Comparative Analysis

Table 1 summarizes the core mechanism, typical latency reduction, accuracy impact, best-fit IoT use case, and key limitation of each latency-accuracy optimization technique family surveyed in this review, synthesized from the reviewed literature.

Technique	Core Mechanism	Typical Latency Reduction	Typical Accuracy Impact	Best-Fit IoT Use Case	Key Limitation
Pruning + Quantization	Removes redundant weights/filters; reduces numerical precision (e.g., 32-bit to 8-bit)	Up to 40% inference latency reduction at 70% compression [15]	Accuracy maintained between 91.8%-94.5% post-quantization [15]	On-device sensor classification, wearable health monitors	Aggressive compression can degrade accuracy on complex tasks

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 10 Issue 7 July - 2026, Pages: 202-209

Technique	Core Mechanism	Typical Latency Reduction	Typical Accuracy Impact	Best-Fit IoT Use Case	Key Limitation
Knowledge Distillation	Trains compact 'student' model to mimic larger 'teacher' model outputs	Up to 12x latency reduction via smaller student networks [16]	Near-teacher accuracy retained with far fewer parameters [14]	Federated edge inference across IoT gateways	Requires access to (or proxy for) a well-trained teacher model
Early-Exit / Dynamic Inference	Attaches auxiliary classifiers at intermediate layers; easy samples exit early	Significant average latency reduction for 'easy' input samples [12]	Comparable or improved accuracy vs. fixed-depth models [12]	Real-time anomaly detection, video-based surveillance IoT	Exit-threshold tuning is task-specific; hard samples still costly
Edge-Cloud Cooperative Offloading	Dynamically partitions/distributes inference between edge devices and cloud	Up to 5.4x faster query response vs. cloud-only [18]	Up to 43.9% accuracy improvement vs. edge-only [18]	Smart city video analytics, distributed IoT surveillance	Depends on network conditions; communication overhead
Streaming-Aware Evaluation & Forecasting	Evaluates/forecasts model output against a continuously evolving world state rather than static frames	Identifies the latency 'sweet spot' on the Pareto curve [19]	Corrects for accuracy loss due to world-state changes during processing [19]	Autonomous driving perception, real-time robotics IoT	Requires redesigning evaluation pipelines and forecasting modules

Table 1: Comparative Analysis of Latency-Accuracy Optimization Techniques for Real-Time IoT Deep Learning

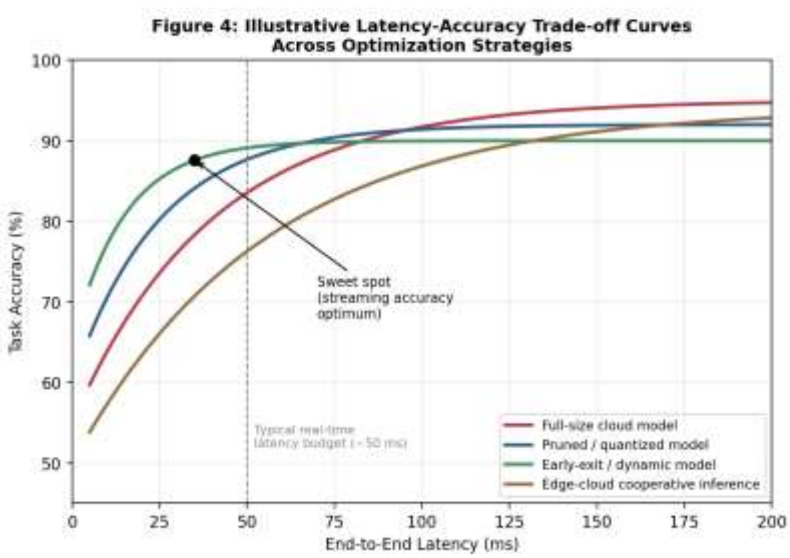

Figure 4: Illustrative Latency-Accuracy Trade-off Curves Across Optimization Strategies

Figure 4 conceptually illustrates the trade-off curves reported across the reviewed studies. Full-size cloud models achieve the highest asymptotic accuracy but require the largest latency budgets to approach it. Pruned and quantized models shift the curve toward lower latency at a modest accuracy cost. Early-exit and dynamic inference models achieve the steepest early gains, reaching near-optimal accuracy at very low latency for the majority of 'easy' inputs. Edge-cloud cooperative inference occupies an intermediate position, trading some latency for substantially improved accuracy relative to edge-only processing. Across the reviewed literature, no single technique dominates in all conditions: the optimal choice depends on the specific latency budget, hardware constraints, and accuracy requirements of the target IoT application [12], [15], [16], [18], [19].

7. Challenges and Open Issues

- Offline-online evaluation mismatch: Standard accuracy metrics evaluated on static datasets do not reflect real-world streaming performance, where the environment continues to change during inference [19].
- Resource-constrained edge hardware: Many IoT devices have severe memory, compute, and energy constraints that limit the depth of compression or the sophistication of dynamic inference strategies that can be deployed locally [13], [16].
- Network variability: Edge-cloud offloading strategies depend on network bandwidth and latency, which can be highly variable in real-world IoT deployments, complicating consistent real-time guarantees [18].
- Threshold and policy tuning: Early-exit confidence thresholds and offloading policies are often tuned per task and dataset, limiting generalizability across different IoT applications [12].
- Safety-critical accountability: In domains such as autonomous driving, accuracy losses introduced by latency-driven optimizations carry direct safety implications, requiring rigorous streaming-aware validation before deployment [17], [22].

8. Future Work

- Streaming-native model architectures designed from the outset around streaming accuracy rather than adapted post-hoc from offline-trained models [19], [20].
- Reinforcement-learning-based adaptive offloading and scaling policies that dynamically balance latency and accuracy under real-time resource availability [21].
- Standardized streaming-accuracy benchmarks for IoT deep learning, enabling fairer comparison of compression, early-exit, and offloading techniques across application domains [19], [22].
- Joint optimization across the full pipeline – sensing, compression, transmission, and inference – rather than optimizing individual stages in isolation.
- Energy-aware latency-accuracy optimization for battery-powered IoT and wearable devices, extending current compression techniques with explicit energy budgets [13], [16].

9. Conclusion

This review has examined the latency-accuracy trade-off central to deploying supervised deep learning models on real-time big data streams in IoT networks. Four major optimization families were surveyed: model compression, dynamic and early-exit inference, edge-cloud cooperative offloading, and streaming-aware evaluation and forecasting. Across autonomous vehicles, smart city surveillance, industrial IoT, and wearable healthcare applications, the reviewed literature consistently shows that no single technique dominates; rather, the optimal strategy depends on the specific latency budget, hardware constraints, and safety requirements of the target application. A recurring theme is the need to move beyond static, offline accuracy evaluation toward streaming-aware metrics that reflect the true real-time performance of deployed systems. Future research combining streaming-native architectures, adaptive reinforcement-learning-based offloading, and standardized streaming benchmarks holds strong potential to make latency-accuracy optimization systematic and reliable across real-world IoT deployments.

References

[1] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923–2960, 2018.

[2] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," Journal of Big Data, vol. 2, no. 1, pp. 1–21, 2015.

[3] Q. Zhang, L. T. Yang, and Z. Chen, "A survey on deep learning for big data," Information Fusion, vol. 42, pp. 146–157, 2018.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2012, pp. 1097–1105.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in Proc. EMNLP, 2014, pp. 1724–1734.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[10] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paradigms, architectures, and challenges," Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, 2019.

[11] D. Xu, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Architectures, challenges, and applications," arXiv:2003.12172, 2020.

[12] S. Teerapittayanon, B. McDanel, and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in Proc. IEEE International Conference on Pattern Recognition (ICPR), 2016, pp. 2464–2469.

[13] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," arXiv:1510.00149, 2016.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[15] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE CVPR, 2018, pp. 2704–2713.

[16] K. Bhardwaj, C. Lin, A. Sartor, and R. Marculescu, "Memory- and communication-aware model compression for distributed deep learning inference on IoT," ACM Transactions on Embedded Computing Systems, arXiv:1907.11804, 2019.

[17] Q. Weng and H. Yun, "Multi-resolution end-to-end deep neural network for optimizing latency-accuracy tradeoff in autonomous driving," arXiv:2605.29138, 2026.

[18] S. Wang, S. Yang, and C. Zhao, "SurveilEdge: Real-time video query based on collaborative cloud-edge deep learning," in Proc. IEEE INFOCOM, 2020, pp. 2519–2528.

[19] M. Li, Y.-X. Wang, and D. Ramanan, "Towards streaming perception," in Proc. European Conference on Computer Vision (ECCV), 2020, pp. 473–488.

[20] C. Thavamani, M. Li, N. Cebron, and D. Ramanan, "FOVEA: Foveated image magnification for autonomous navigation," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 15539–15548.

[21] A. Ghosh, A. Nambi, A. Singh, H. YVS, and T. Ganu, "Adaptive streaming perception using deep reinforcement learning," arXiv:2106.05665, 2021.

[22] S. Zhang, L. Song, S. Liu, Z. Ge, Z. Li, X. He, and J. Sun, "Workshop on autonomous driving at CVPR 2021: Technical report for streaming perception challenge," arXiv:2108.04230, 2021.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in Proc. NeurIPS, 2014, pp. 2672–2680.

[24] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," IEEE Computational Intelligence Magazine, vol. 10, no. 4, pp. 12–25, 2015.

[25] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346–2363, 2019.