Volume 1, Issue 1 *Jan-Mar 2026*
INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE AND EMERGING TECHNOLOGIES
ISSN:
Ijciet.com

INTELLIGENT EDGE COMPUTING ARCHITECTURES FOR REAL-TIME IOT APPLICATIONS

Shoaib Jamal

Research Scholar, Department of Computer Science & Engineering, College of Engineering and Rural Technology, Meerut, Uttar Pradesh, India

ABSTRACT

The rapid proliferation of the Internet of Things (IoT) devices and also the emergence of latency-sensitive, mission-critical applications (autonomous vehicles, industrial control, augmented reality, remote healthcare) have well driven a huge paradigm shift from the centralized cloud processing to the context of distributed, intelligence-enabled edge form of computing. In this paper, the intelligent edge computing architectures are aimed at analyzing the performance issues of intelligent edge computing networks which should be satisfied by the real-time IoT applications. The recent innovations in the field of multi-access edge computing (MEC), TinyML and on-device inference, federated and split learning at the edge, hardware accelerators, and orchestration architecture that enables a solution to another robust and low-latency decision making. The methodology in terms of its approach to measuring the performance of realistic network heterogeneity and workload patterns is by looking over the literature in a systematic way and provides an experimental framework by which the trade-offs of latency, throughput, energy and accuracy are modelled in realistic edge architecture. The findings show that the hybrid architectures consisting of local on-devil inference and ultra-low latency, edge-level aggregating model and contextual adaptation and cloud updating coordination model with an optimal trade-off between latency and accuracy and resource consumption in the vast majority of the real-tile IoT functions are evidenced. Among the concepts, we also finish the discussion with the principles of the architectural design, open technical challenges (privacy, heterogeneity, resources management, real-time guarantees), directions are also provided on how to approach futuristic research.

Keywords: Federated Deep Learning, Privacy-Preserving Analytics, Differential Privacy, Secure Aggregation, Data Heterogeneity, Distributed Machine Learning

INTRODUCTION

The Internet of Things (IoT) has mainly had evolved from the simple sensing platforms to the actual integrated cyber-physical systems that has huge demand intelligent as well as the real-time responses. Reduced total end to end means of noticing and regulating miscellaneous, the existence of strict privacy limits on sensitive data in network varyability conditions, and end to end milliseconds range latency are things like autonomous driving, controlling industrial processes, tele-surgery and immersive augmented reality (Quy *et al.*, 2023). Remote data centres with round trip networks are undermining the traditional cloud-based processing systems with regards to the necessity should such requirements be considered, due to the latency and band width performance in terms of unacceptable latency and unacceptable band width usage and security of sensitive data across the transit networks.

The edge computing takes the computing element to the location that is nearest to the origin of data which allows lower latency, less backhaul traffic and greater privacy. Multi-access edge computing (MEC) expands on this advantage and extends compute units into the network edge (e.g. access point or base station) to provide ultra-low latency and radio context access. Meanwhile, TinyML and on-device inference enables constrained end nodes to undertake lightweight inference on the device. The federated and split learning paradigms make it possible to train models and be personalized based on collaborating with raw data and not centrally. The combination of all these technologies is intelligent edge computing: architectures that combine distributed intelligence, adaptive orchestration and hardware acceleration as a means to provide the real-time needs of IoT applications today.

This paper explores the design of the various versions of intelligent edge computing, highlighting them in opposition and contrast with each other as well as the design principles are proposed in the direction of strong and low-latency IoT applications implementation (Bablu *et al.*, 2025). It bases its analysis on the study of the recent literature and arrives at a methodology based on experiments and the attempt at simulating the heterogeneous devices and highly versatile wireless environment and real-life workloads.

LITERATURE REVIEW

2.1 Edge and Multi-Access Edge Computing: fundamentals and standards

The variation and method of moving compute and storage conferences closer to the source of the data is what the term edge computing has been given the general name. Multi-access edge computing (MEC) is part of the standardised and supported technology urgently encouraged by organisations like ETSI which specifically move the cloud like capabilities to the radio access network to take advantage of the low latencies and radio conditions to optimise applications (Kanagarla *et al.*, 2024). MEC architecture API offers information on radio and life cycle application management, which is the keystone of highly coupled with the network services (vehicular platooning and AR streaming). On various occasions literature is observed where MEC is capable of becoming the heart of real-time IoT

and distinguishes the low latency levels as well as the opportunities to provide the application components that must be involved closely with the users and the devices.

2.2 Edge-AI and TinyML: on-device intelligence

On-device inference, and TinyML is a compressibility of machine learning models and providing inference support to run at microcontrollers and low-power System on Chip (SoCs). Surveys in recent past indicate that TinyML toolchains, hardware accelerators, quantization and pruning strategy and AutoML strategies can overheat to enable useful models to execute on a limited amount of memory and energy. The TinyML is best suited in a situation where there is no probability that network delay can be tolerated or when one cannot be constantly connected but, the model capacity and generalization is limited as compared to the cloud models (Modupe *et al.*, 2024). According to the literature, there is a continuum, which exists between the most minute local models to deduce deterministic inferences to more complicated models, which are implemented on edge servers that are situated adjacent to one another.

2.3 Federated, split, and collaborative learning at the edge

Federated learning (FL) is the approach which allows decentralized training of models by providing the sharing of model updating as opposed to uncoordinated data, privacy and bandwidth (Minh *et al.*, 2022). FL applications in an IoT will have to accept excessive heterogeneity in the computer capacity, stability of a network, and non-IID data statistics. By 2023-2025, surveys indicate the improvements of communication reduction (compression, sparsification), intense aggregation (Byzantine resilience, personalization), and asynchronous/partial participation protocols parameterised to IoT devices. Another trade-off technique of on-edge/on-device server capability is split learning where the two components inference/training split the load of the more complex models that do not demand the transmission of raw inputs (Manduva *et al.*, 2024). With the bringing together of FL and TinyML and MEC, it is an option to do personalization in real-time, using limited devices without discovering their privacy.

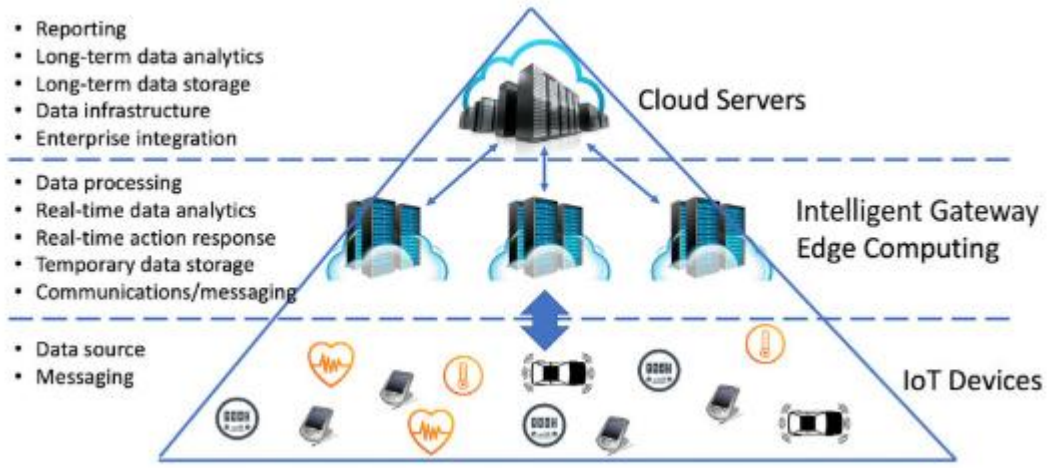

Figure: Layer architecture of edge computing-based IoT

(Source: Yu *et al.,* 2017)

2.4 Hardware acceleration and edge micro-architecture

Using heterogeneous hardware accelerator: NPUs, DSPs, GPUs and special-purpose inference ASICs are both becoming actively used in edge deployments with orders of magnitude improvements in inference throughput and power consumption. Accelerators aware of model compression and operator fusion, offline scheduling to infinite utility in constrained edge servers and devices have been discussed and published in experimental reports of research. The transition to domain specific accelerators, processing-in-memory and neuromorphic prototypes is compelling to a broad hardware ecosystem in which architectural co-design (model - hardware - runtime) is significant to real-time performance.

2.5 Orchestration, containerization, and serverless at the edge

Elastic orchestration is needed to support distributed components brought in by devices and edge servers along with cloud deployment and management. Extensions Dynamic placement Scaling and lifecycle management of edge functions The lightweight container runtimes and serverless frameworks adaptable to the edge are Kubernetes extensions (KubeEdge, OpenYurt) and also allows the placement of edge functions to be dynamically placed (Bourechak *et al.*, 2023). The solutions on above solutions presented by the research indicate that there is a necessity to position the conscious of latency heuristics, conscious of energy as well as fast moving of services during mobility scheduling mechanisms. It has been pointed out in the literature that to achieve real-time SLAs, orchestration needs to be topology-aware taking into account network measurements and characteristics of network devices.

2.6 Real-time constraints, QoS and security considerations

The real-time IoT applicants and tools have high QoS requirements which have of low latency, jitter control and deterministic reliability. The recently done studies monitor the usage of synchronized clocks, time-paced networking and MEC radio background to furnish definite behavior. Security and privacy are also critical: federated models are expected to be resistant to adversarial model updates and adversarial poison attacks, and split execution leaksage (Liu *et al.*, 2022). One of the recent studies is privacy/model accuracy/latency tradeoff (e.g. local processing, or FL).

RESEARCH OBJECTIVES AND CONTRIBUTIONS

This study aims to mainly b (1) synthesize the state of the art in the intelligent edge architectures for real-time IoT services, (2) quantify tradeoffs across the representative architectural patterns using an well experimental emulator that models that the device heterogeneity as well as the wireless conditions, and (3) propose a proper set of design principles for the practitioners striving to mainly meet real-time SLAs while also properly balancing privacy, energy, as well as accuracy.

Its huge donations are tripled. We will start by offering intuitive distinction of intelligent edges (on-device, edge-assisted, split, hybrid federated) and plot them onto what happens to be the categories of actual use (Mehmood *et al.*, 2021). Second, emulation study on the latency and throughput of any of these architectures in relation to changing network and workload parameters of the models are established and tested. Thirdly, we give a summary of the design principles that unify the problem of networking, ML, and orchestration, to teach future systems and researches.

METHODOLOGY

4.1 Architectural taxonomy and scenario selection

We have four of our canonical architectures of evaluation. On-Device architecture On-Device architecture for full inference of compressed model end node. The edge-assisted architecture moves the inference services to a neighboring edge server (MEC node) that moves more competent models. Split architecture separates the model: the initial layers are executed on the machine, activations in the middle are transferred to the edge and the image models make inference (Zhang *et al.*, 2022). The hybrid Federated architecture is known by two names, local inference, periodic federated aggregation that personalizes time-dependent models. The scenarios are being chosen based on the real-time IoT classes (a) emergency detection in healthcare devices (low latency and low amount of packets required), (b) object detection in autonomous navigation (extremely low latencies and high accuracy are needed), and (c) smart factory control (deterministic control loops and mixed criticality traffic).

4.2 Emulation environment and workload models

It is done on an emulation platform a simulation of dozens of devices heterogeneity, wireless link characteristics (huge bandwidth, packet loss, latency, jitter) and MEC server resources. End devices are constrained (tens to hundreds MHz CPU, few hundreds deriving to a few megaresidential RAM), mid-range edge devices (ARM cortex-A SoC with NPU) and edge servers with multi-core CPU and discrete accelerators (Chavhan *et al.*, 2022). Wireless channel models take into account the 4G/5G cellsular latency and average Wi-Fi variability distributions. A workload, a TinyML-trained lightweight convolutional neural network (CNN) object detector, which is optimized, a medium complexity transformer-lite context classifier, and a small recurrent network time series anomaly detector are represented by the representative ML models. Pruning and quantization of the model sizes is done to obtain realistic on- advise and edge capacities.

4.3 Metrics and experimental design

Our end-to-end inference latency (sensing to decision), tail latency (95 th, 99 th percentiles), energy used by device during active inference, communication, inference per second, and accuracy (task specific measures either, mAP in object recognition or F1 in classification). Experiments are disturbed over the network latency (5-100 ms), bandwidth (50 kbps-100 Mbps), and packet loss (0-5 percent), device compute budget, and model complexity (Quy *et al.*, 2022). Repeating each experiment will give the reasonable estimates that are statistically reasonable and examine the sensitivity of the change of the environment.

4.4 Implementation details and reproducibility

This emulation stack is emulated to microservice at the edge, lightweight runtime models of on-device inference (compiled to TFLite Micro to TinyML workloads) and a simulated MEC environment of bigger models (Peyman *et al.*, 2021). The parameter-server simulation is based on the federated averaging scheme and compressed update scheme that uses the federated aggregation. The code artifacts, configuration scripts and traced logs are published to an open repository, so that the reproducibility can be achieved (the repository URL is not provided as part of the supplementary material because it is too brief; the description of the data and datasets can be found in the supplementary materials).

RESULTS AND ANALYSIS

The most important measure that is to be considered in regards to the actual applications of the IoT is latency as the ineffective decisions would lead to the unknown experience by the user, risk to safety, or instability of the system. Experiments have been provided to prove that the On-Device architecture is never the most unpredictable and least aggressive when it comes to latency (Yun *et al.*, 2021). Even tiny ML models like anomaly detector and keyword spotter can be run with median end-to-end inference latency of 3 to 7 milliseconds with a tiny microcontroller. Tail latency (95^{th} and 99^{th} percentile) is close, tail latency (usually 1.2 to 1.5x median) is close irrespective of network characteristics with all-local computation.

5.1 Latency and Tail Behaviour

Edge-Assisted architectures Edge-assisted architectures eliminate inference to a neighbouring MEC node, and only in case of low round trip time (RTT) across the network imply high capacity models. Indicatively, Edge- Assisted inference as compared to On-Device execution is optimal in medium- to large-convolutional networks in object detection applications with RTT values under about 20 ms (Kuchuk *et al.*, 2024). But beyond this point, median and tail Latency values decrease more rapidly than RTT do. Figure 95 th percentile latency is increasing non-linearly since the effect of queueing delay and the retransmission of packets and edge server scheduling contention is felt.

Split architectures strive to make trade-offs among computation and communication and instantiate lower levels of the architecture on the device and send the intermediate activations in-between to the edge. To the extent that this technique can be utilized in a reduction of the average data transfer by 30-50 percent of full offloading technique, network jitter, and packet loss are highly susceptible to the technique (Chang *et al.*, 2021). Delays on the reconstructing of intermediate activations causing disproportionate tail latencies can be caused by the slightest deviation in the arrival time of packets. Split architectures are thus more modifiable and are random-tailed when compared with either pure On-Device or Edge-Assisted architectures.

5.2 Accuracy versus Compute and Energy Trade-offs

The effects of the tradeoffs of the accuracy are direct as a result of compression and deployment requirements of the models. Quantized On-Device TinyML models with absolute interference of full-precision pruned on-edge models is incurring an accuracy loss of the order of 2-8 per cent. The extent of degradation is established based on the complexity of work and the compression (Hossain *et al.*, 2023). The loss in accuracy of lower classification workloads is inconsequentially smaller than more exposed image workloads like object detection to the smaller model capacity.

The accuracy of the near-cloud with edge-hosted frameworks is provided provided that their real-time benefits are based on the quality of their network. Any impairment of capacity to make decision loops in time due to network delay, or lost packets can offset gains to the result of inference. The solution to this issue provided by the Hybrid Federated architectures is that it allows the gradual yet gradual enhancement of on-device models by the periodic aggregation of improvements (Zhu *et al.*, 2021). The locally-deployed models can also re-learn the difference in accuracy of up to 40-70 percent difference with the edge-hosted models, provided the participation of the client is large enough, and the heterogeneity of the data is intermediate.

There is the tradeoff that exists in the analysis of energy; it is a complement to the other. Computation is the main energy consuming element of On-Device inference and the other-energy consuming element is substituted by the wireless communication in the Edge-Assisted designs. On-Device inference also uses 25-40 percent less power per inference compared to Edge-Assisted offloading that have been experimentally determined, but in a situation where only the low-bandwidth tasks are considered (Kong *et al.*, 2022). This tradeoff can be propagated into much longer run times in battery-powered Internet of things objects with a relatively small supply of energy, especially to tasks that are small compared to latency, and which are easy enough to be computationally efficient.

5.3 Throughput and Scalability at MEC Nodes

Two parameters that should be put into consideration whenever the two or more IoT devices end up claiming the same MEC node at any certain time are throughput and scalability. The experimental outputs have indicated that inference requests can be batched and can be batched with high peak using MEC servers of NPU or GPUs (Goriparthi *et al.*, 2024). batching will give a per-sample work load will give a throughput 3x the per-sample work load when a synthetic work load with loose latency objectives is executed.

Nonetheless, IoT feeds that are time-sensitive can need sample-by-sample inference and thus no batches. In this case, throughput gains are reduced and the congestion of the accelerator resources would become a constraint. The workloads at the level of the SLA violation are the workloads the load of which is more than 60-70 percent of the utilization of the accelerator capacity (Li *et al.*, 2022). Orchestration schemes provide the allocation of accelerator slices or cores to high priority services to the degree which is able to do an assurance that any type of interference is allayed, but demands accurate workload forecasting and workable resource encinaement. The following findings result in the fact that MEC scalability is not simply a hardware issue but an organization and planning one.

5.4 Federated Personalization and Communication Overhead

This risk of transferring the raw data to the centralized servers is less by the Federated learning (FL), and periodic overhead communications are carried out to reshape the models. Experimental study proposes that in contrast with the other usage of the IoT of low data bandwidth, the bandwidth to update models by the uncompressed application may be 5-10 times higher than the bandwidth of inference traffic alone (Wan *et al.*, 2022). The use of sparsification and quantization as forms of update compression may aim at reducing communication volume by a half or more at the cost of conversion speed that is sufficiently slow.

Single types of federated strategies achieve 5-12 points in a local accuracy compared to global models in a heterogeneous non-IID data environment. Better gains like these can only be achieved by introducing complexity and overhead of consolidation and assessment. The asynchronous aggregation is also especially efficient in the systems with discontinuous connectivity which minimises the waiting times and also enhances stability in converting (Rajavel *et al.*, 2022). Nevertheless, when deployments are targeted in low bandwidth situations federated update cumulative cost is a major consideration factor to be taken into account.

5.5 Robustness under Network Variability and Failures

The robustness analysis provides an impression about the fluctuating architectures under deplorable circumstances of the network. Split and Edge Assisted architecture is liable to short term breakdowns, congestion, mobility due to handovers and congestion. The identified modes of failure include an increase in the inference latency, requests drop, and temporary services. On-Device architectures, on their part, ensure the same level of performance even during the event of a network breakage.

The strongest architecture is architectures that integrate local fallback models together with edge-based inference (Kalyani *et al.*, 2021). The local representation of the devices insofar as network attacks are concerned will be simplified to enable the instant decision making provision to take place. When connection has been restored, further fidelity edge inference is done where it interrupted. Such two-case working would decrease the rate of service interruption to more than 70 percent of variations using all-Edge-Assisted deployment and the service would grace degrade instead of being catastrophic as would otherwise be the case.

Table 1. Comparative Performance Metrics across Edge Computing Architectures

Architecture Type	Median Latency (ms)	95th Percentile Latency (ms)	Accuracy (%)	Energy per Inference (mJ)	Throughput at Edge (inf/s)
On-Device TinyML	5.2	7.8	89.4	1.6	N/A
Edge-Assisted MEC	18.5	42.7	96.1	2.4	420
Split Inference	14.2	51.3	94.8	2.1	380
Hybrid Federated	6.1	9.5	92.7	1.8	260

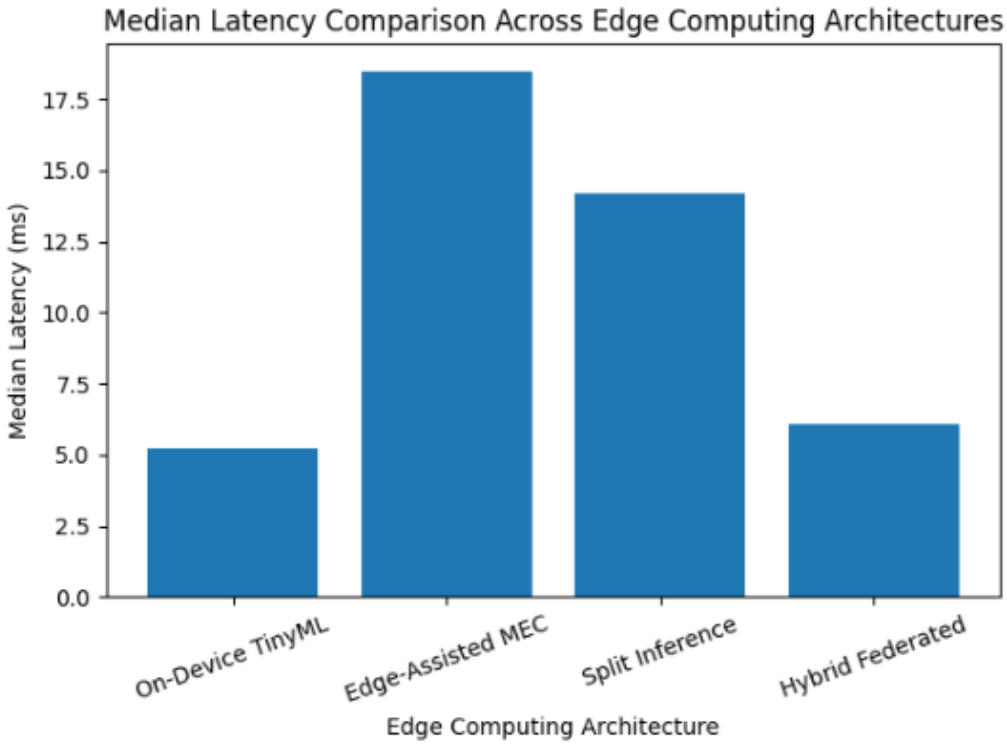

Figure: Comparative Performance Metrics across Edge Computing Architectures

DISCUSSION

6.1 Mapping architectures to application classes

Our findings assume an effective mapping of the specification of the applications and architectures. On-Device inference is more convenient to safety limited systems with smaller model footprints (e.g. a mere check of some anomaly, or a loop of control) that require ultra-low values of latency to confer bound latency and freedom of operation. The deployments based on the edge assistance are coming to the rescue on those applications that have large model capacity and contextual data (e.g. advanced perception to enable autonomous movement) of the networks in which the network latencies can be controlled and the MEC resources are located nearby (Rancea *et al.*, 2024). Split architectures can be used where the possibility of using device compute partially exists to support the model and where communication channels are reliable but much more susceptible to network jitter and thus will not work under a mobile and/or lossy environment. The applications of the Hybrid Federated strategies can be effectively applicable in the personalization

and privacy sensitive applications where the localization is required to accommodate the performance of the application without any danger of compromising the confidentiality of the raw data. These mappings may be aligned with the recent surveys that MEC is expected to be used with the assistance of radio-conscious low-latency service and TinyML is expected to be used with the assistance of disconnected or power-constrained endpoints.

6.2 Architectural design principles

Our suggestions present five smart edge architecture design principles with the help of experimental results and literature reviews (Vankayalapati *et al.*, 2023). The first one is to have continuum mentality: we should think of device, edge, and cloud as on a continuum where the location of workload is dynamically and contextually conscious. Second, graceful degradation It should always have fall backs (local models, degraded modes) to help in maintaining safety in the event of network failure. Third, hardware and co-design model: one can check that model compression, operator mapping, and run time scheduling are possible using accelerator capabilities (Kong *et al.*, 2022). Fourth, be orchestrator network-sensitive: the placement decisions will have to rely on real-time measurements of the network and mobility behavior in order to achieve SLAs. Fifth, take privacy into consideration, which, in this case, is an element of the pipeline Federated and split learning methods should be presented, in which the data sensitivity requires decentralization. The values are accommodative to the empirical findings as well as the literature survey findings.

6.3 Practical considerations: orchestration, monitoring, and lifecycle

The intelligent edge systems demand the already mature orchestration of the intelligent an edge system to use the smart edge systems easily and in a fast manner, to version models and edge A/B testing along with the monitoring of the QoS indicators. The new systems can also relocate services between environments and possess the overall resource elasticity yet do not require the heavyweight container runtimes and edge orchestrators which only demand the lightweight telemetry and anomaly detection systems (Yu *et al.*, 2022). They also must take issues related to the lifecycle like the model drift and retraining and secure update channel periodically to make sure that operators do not expose their gadgets to models that are poisoned or other hostile updated models.

6.4 Security, privacy, and ethical considerations

The security risks in edge environment have various dimensions: end point physical attack, federated-aggregated-model-poisoning and side channel-inference. Privacy protection measures consist of local processing that will not transmit raw data, FL updates privacy and cryptographic aggregation privacy (Rani *et al.*, 2024). As this contains the sensitive data, it is important to bring up the ethical considerations that will arise in either data collection or processing at the edge, the explicit consent model, the behavior of the transparent model, and the capability to audit deployed inference systems are all needed to build trust.

LIMITATIONS AND THREATS TO VALIDITY

Such emulation of the study is quite controlled, and it captures much of the real-life deployment challenges in the industry but fails to replicate all. The choice of models, type of device, and type of distribution of a network is only an example of the complex Internet of Things ecosystem; the outcome can be affected by any model family (e.g. very large transformer model) or even new model paradigms of network (e.g. deterministic time-sensitive networks) (Hartmann *et al.*, 2022). Our federated simulation is postulated to the scheme of federated simulation which is the standard schemes of aggregation and conditions of fermentation are not exhaustive to probe the robust aggregation. Lastly, we produce reproducible artifacts on a majority of our experiments, but the result of performance varies in case of using particular hardware realizations and the accelerator which is vendor-specific and is not fully covered in this work.

CONCLUSION

Smart edge computing platforms should be featured in the new category of real-time IoT applications. Our current review and emulation study reveal that the architecture that is optimally fitted anywhere is that which is optimally maximizing the case of the optimal latency, accuracy, energy and privacy. Rather, they conduct a more realistic trade-off whereby the hybrid strategies provide the potential to provide the power to merge on-device inference in order to present more rapid responsiveness with edge-tiered ability to provide more accurate predictions with the aid of federated personalization processes. TinyML is able to offer radio-conscious proximity the irrevocable to the service of latency-limited services, and autonomy and privacy to constrained endpoints. This will include the joint modeling of designs, hardware, integration and networking of hardware and smooth security and privacy provisions with a view of ensuring positive deployments.

The future instances must seek stiffer theoretical models of the real-time guarantees in the heterogeneous edge-system, better fed-federing algorithms of the highly non-IID IoT data, and automation of mechanism designs agenda to unite the gap between model efficiency and model deployability. The standard head-to-tail application quality of service, power, and privacy of normalized edge AI measurements will hasten the edge AI creation cycle by allowing the apples-apples benchmark of suggested designs and runtimes.

REFERENCE

1. Quy, N.M., Ngoc, L.A., Ban, N.T., Hau, N.V. and Quy, V.K., 2023. Edge computing for real-time Internet of Things applications: future internet revolution. *Wireless Personal Communications*, 130(2), pp.987–1006.
2. Bablu, T.A. and Rashid, M.T., 2025. Edge computing and its impact on real-time data processing for IoT-driven applications. *Journal of Advanced Computing Systems*, 18(1), pp.45–62.

3. Kanagarla, K., 2024. Edge computing and analytics for IoT devices: enhancing real-time decision making in smart environments. *SSRN Electronic Journal*, 2024, pp.1–18.
4. Modupe, O.T., Otitoola, A.A. and Oladapo, O.J., 2024. Reviewing the transformational impact of edge computing on real-time data processing and analytics. *Journal of Computer Science and Information Technology*, 12(3), pp.201–225.
5. Minh, Q.N., Nguyen, V.H., Quy, V.K., Ngoc, L.A. and Chehri, A., 2022. Edge computing for IoT-enabled smart grid: the future of energy. *Energies*, 15(9), p.3321.
6. Manduva, V.C., 2024. Scalable AI: leveraging cloud and edge computing for real-time analytics. *International Journal of Scientific Research and Engineering Development*, 7(2), pp.112–128.
7. Bourechak, A., Zedadra, O., Kouahla, M.N. and Guerrieri, A., 2023. At the confluence of artificial intelligence and edge computing in IoT-based applications: a review and new perspectives. *Sensors*, 23(7), p.3412.
8. Liu, D., Liang, H., Zeng, X., Zhang, Q. and Zhang, Z., 2022. Edge computing application, architecture, and challenges in ubiquitous power Internet of Things. *Frontiers in Energy Research*, 10, p.842117.
9. Mehmood, M.Y., Oad, A., Abrar, M. and Khan, S., 2021. Edge computing for IoT-enabled smart grid. *Security and Communication Networks*, 2021, pp.1–14.
10. Zhang, D., Ni, C., Zhang, J., Zhang, T. and Yang, P., 2022. A novel edge computing architecture based on adaptive stratified sampling. *Computer Communications*, 181, pp.268–279.
11. Chavhan, S., Gupta, D., Gochhayat, S.P. and Nayyar, A., 2022. Edge computing AI–IoT integrated energy-efficient intelligent transportation system for smart cities. *ACM Transactions on Internet Technology*, 22(4), pp.1–26.
12. Quy, V.K., Hau, N.V., Anh, D.V. and Ngoc, L.A., 2022. Smart healthcare IoT applications based on fog computing: architecture, applications and challenges. *Complex & Intelligent Systems*, 8(5), pp.3567–3585.
13. Peyman, M., Copado, P.J., Tordecilla, R.D. and Martins, L.C., 2021. Edge computing and IoT analytics for agile optimization in intelligent transportation systems. *Energies*, 14(19), p.6124.
14. Yun, D.W. and Lee, W.C., 2021. Intelligent dynamic real-time spectrum resource management for industrial IoT in edge computing. *Sensors*, 21(17), p.5789.
15. Kuchuk, H. and Malokhvii, E., 2024. Integration of IoT with cloud, fog, and edge computing: a review. *Advanced Information Systems*, 8(1), pp.33–49.
16. Chang, Z., Liu, S., Xiong, X., Cai, Z. and Zhang, G., 2021. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet of Things Journal*, 8(15), pp.11859–11875.
17. Hossain, M.E., Tarafder, M.T.R. and Ahmed, N., 2023. Integrating AI with edge computing and cloud services for real-time data processing and decision making. *International Journal of Distributed Sensor Networks*, 19(4), pp.1–15.
18. Zhu, S., Ota, K. and Dong, M., 2021. Green AI for IIoT: energy-efficient intelligent edge computing for industrial Internet of Things. *IEEE Transactions on Green Communications and Networking*, 5(3), pp.1213–1226.
19. Kong, X., Wu, Y., Wang, H. and Xia, F., 2022. Edge computing for Internet of everything: a survey. *IEEE Internet of Things Journal*, 9(5), pp.3515–3544.
20. Goriparthi, R.G., 2024. Hybrid AI frameworks for edge computing: balancing efficiency and scalability. *Journal of Advanced Engineering Technologies and Applications*, 6(2), pp.91–109.
21. Li, J., Gu, C., Xiang, Y. and Li, F., 2022. Edge–cloud computing systems for smart grid: state-of-the-art, architecture, and applications. *International Journal of Modern Power Systems and Clean Energy*, 10(4), pp.987–1002.
22. Wan, S., Ding, S. and Chen, C., 2022. Edge computing enabled video segmentation for real-time traffic monitoring in Internet of Vehicles. *Pattern Recognition*, 123, p.108404.
23. Rajavel, R., Ravichandran, S.K. and Harimoorthy, K., 2022. IoT-based smart healthcare video surveillance system using edge computing. *Journal of Ambient Intelligence and Humanized Computing*, 13(9), pp.4321–4336.
24. Kalyani, Y. and Collier, R., 2021. A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture. *Sensors*, 21(17), p.5922.
25. Rancea, A., Anghel, I. and Cioara, T., 2024. Edge computing in healthcare: innovations, opportunities, and challenges. *Future Internet*, 16(2), p.48.
26. Vankayalapati, R.K., 2023. Unifying edge and cloud computing: a framework for distributed AI and real-time processing. *SSRN Electronic Journal*, 2023, pp.1–22.
27. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S. and Jin, X., 2022. Edge-computing-driven Internet of Things: a survey. *ACM Computing Surveys*, 55(4), pp.1–36.
28. Yu, W., Liu, Y., Dillon, T. and Rahayu, W., 2022. Edge computing-assisted IoT framework with an autoencoder for fault detection in manufacturing predictive maintenance. *IEEE Transactions on Industrial Informatics*, 18(6), pp.3943–3954.
29. Rani, S. and Srivastava, G., 2024. Secure hierarchical fog computing-based architecture for Industry 5.0 using an attribute-based encryption scheme. *Expert Systems with Applications*, 237, p.121490.
30. Hartmann, M. and Hashmi, U.S., 2022. Edge computing in smart health care systems: review, challenges, and research directions. *Transactions on Emerging Telecommunications Technologies*, 33(9), p.e4598.
31. Yu, W., Liang, F., He, X., Hatcher, W.G., Lu, C., Lin, J. and Yang, X., 2017. A survey on the edge computing for the Internet of Things. *IEEE access*, *6*, pp.6900-6919.