International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 8915-8921
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Seamless Integration of Edge-Cloud Computing and Distributed Artificial Intelligence: A Comprehensive Framework for Next-Generation Applications

Venkateswarlu Poka*

Microsoft, USA
* **Corresponding Author Email:** venkat.poka2@gmail.com- **ORCID:** 0000-0002-5447-7850

Article Info:

DOI: 10.22399/ijcesen.4330
Received : 29 September 2025
Revised : 01 November 2025
Accepted : 06 November 2025

Keywords

Edge-Cloud Synergy,
Distributed Machine Learning,
Federated Optimization,
Hierarchical Computing,
Privacy-Aware Systems

Abstract:

Integration of edge computing with cloud infrastructure and distributed artificial intelligence yields a revolutionary paradigm that meets computational needs across contemporary data-intensive applications. Hierarchical architectures for processing arise through synergy in the edge-cloud framework, where proximity-based processing is used for latency-critical operations in tandem with cloud processing for elastic analytics and storage. Real-time response becomes feasible for applications that include autonomous vehicles, smart cities, industrial automation, and healthcare informatics with this integration. Collaborative model training over geographically distributed settings becomes possible using distributed AI techniques, specifically federated learning and data parallelism, without compromising data privacy and reducing communication overhead. Great technical challenges face these frameworks, such as resource heterogeneity, communication bottlenecks, the requirement of fault tolerance, and security issues. System efficiency and model convergence are greatly enhanced using sophisticated optimization schemes using gradient compression, hierarchical aggregation, and adaptive resource allocation. Next-generation applications' foundation infrastructure arises from the integration of edge-cloud architectures with distributed intelligence mechanisms, at the same time fulfilling strict latency requirements, privacy protection, and computational scalability demands.

1. Introduction

Internet of Things (IoT) devices have spread very fast, and data-intensive applications have increased exponentially, which requires a basic rethinking of the computational architectures. Market research indicates staggering growth throughout the worldwide IoT market, with its connected IoT devices totaling about 12.3 billion active endpoints in 2021, a 9% rise from the preceding year, withstanding pandemic-led disruptions to global supply chains and manufacturing lines [1]. Predictions point towards this number increasing to 27 billion IoT devices that are connected by 2025, doubling the installed base in four years and translating into compound annual growth rates of over 18% across enterprise, industrial, and consumer sectors [1]. Gigantic amounts of diverse data streams result from this unprecedented proliferation of devices, including sensor telemetry, multimedia media, transactional data, and real-time monitoring data, all placing tremendous computational and bandwidth requirements on the available infrastructure.

Cloud-based paradigms historically provide unparalleled scalability and resource provisioning support via virtualization technologies and distributed data center infrastructures, but these environments increasingly fail to meet strict latency requirements and bandwidth restrictions of new real-time applications. Physical distance between centralized cloud data centers and end users or IoT devices places inherent architectural constraints, which are usually hundreds or thousands of kilometers, bringing in inherent delays controlled by light speed and routing complexities of networks [2]. Traditional cloud computing paradigms have round-trip latencies between 100 milliseconds and a few hundred milliseconds based on geographic location, network traffic, and routing optimization, making them inherently inappropriate for real-time responsiveness applications below 10 milliseconds [2]. Bandwidth demands for sending raw sensor information from billions of scattered IoT

endpoints to centralized cloud facilities outstrip current backbone network capacity, with estimates that it would take exabytes of daily bandwidth provisioning to send all data generated by the IoT to the cloud [2]. Purposing advanced artificial intelligence and machine learning models over geographically dispersed environments at the same time poses novel challenges in computational efficiency, data privacy, and system robustness, as training deep neural networks with tens or hundreds of millions of parameters requires tremendous computational resources while inference operations need to run with minimal latency.

Two key areas of research have emerged with this intersection of requirements: edge-cloud harmony and distributed AI workloads. Architectural unification of edge computing, where the processing of data happens near its origin, with traditional cloud infrastructure, is addressed via the former paradigm to support a hierarchical computational model that exploits the benefits from both sides. Maximizing deployment, training, and inference of AI models on distributed systems is the target of the latter paradigm, covering issues with data parallelism, communication overhead, and fault tolerance. Several application areas experience the importance of these areas of research, such as autonomous transportation systems, intelligent urban infrastructure, industrial automation, health informatics, and big data analytics. Each area has distinct technical needs that cannot be well supported by either edge or cloud computing alone, nor by centralized AI training practices. Integrated systems that balance edge-cloud structures with distributed AI functions thus constitute an essential frontier for computer science and engineering research.

2. Architectural Foundations of Edge-Cloud Integration

A hierarchical computational architecture constitutes the theoretical basis of edge-cloud synergy, partitioning processing tasks strategically along latency sensitivity, computational complexity, and data locality needs. Physical closeness to data sources and end users defines edge computing, which is superior in situations requiring sub-millisecond response time and lower network traffic. Evolution of computing from mainframes to personal computers to mobile computing has continued to follow alternating trends of centralization and decentralization, with edge computing being the current stage in this cyclical evolution spurred by mobile device proliferation and wide-area network latency constraints [3]. The velocity of light places fundamental limits, setting the theoretical minimum latency at around 1 millisecond per 100 kilometers of fiber-optic cable, which means applications demanding sub-10 millisecond response times cannot solely depend on faraway cloud data centers hundreds or thousands of kilometers from end users [3]. Cloud computing, conversely, provides virtually unlimited computational resources, sophisticated data analytics capabilities, and centralized management infrastructure, enabling economies of scale that deliver computing capacity at costs approximately one-tenth of traditional enterprise data centers through massive consolidation and resource pooling strategies [3].

Several architectural dimensions require careful consideration for integrating these complementary paradigms. Workload partitioning mechanisms need to dynamically distribute computation workloads between edge and cloud resources based on real-time evaluation of network conditions, computation workload, and application-specific needs. Processing resources are placed strategically near the network edge by mobile edge computing structures, normally at cellular base stations or access points, forming hierarchical frameworks where the edge layer processes latency-sensitive processing within 1-10 milliseconds and the cloud layer performs computationally intensive batch processing and long-term data storage [4]. It must be guaranteed across distributed storage systems through data synchronization protocols with low bandwidth usage and acceptable eventual consistency levels, with the protocols being optimized to function well over wireless links with intermittent throughput between several megabits per second and hundreds of megabits per second, depending on radio conditions and mobility patterns of the users [4]. Heterogeneous hardware platforms across edge devices, fog nodes, and cloud data centers must be coordinated through orchestration frameworks in terms of resource allocation, service deployment, and failure recovery.

Hierarchical processing pipelines that carry out early data filtering, aggregation, and preprocessing at the edge layer are essential to this architectural integration and then transfer cleaned datasets to cloud infrastructure for deeper analytics and long-term storage. Computational offloading situations exhibit specific efficacy by means of mobile edge computing deployments, wherein mobile devices equipped with small battery life and processing capacity can offload compute-intensive tasks to proximal edge servers to cut energy consumption on the mobile devices by 40% to 90% and lower application execution times by 30% to 80% concurrently compared to local execution [4]. The

network traffic is further minimized via this method, privacy is further supported by restricting egress data to only aggregated or anonymized data, and overall responsiveness of the system is further improved through localized decision-making features [4]. Different levels of connectivity need to be supported through the architecture such that edge devices can function independently when the network is disrupted, while synchronizing with cloud services upon re-establishment of connectivity, thus ensuring continuity of service even in the case of temporary or poor network conditions [3].

3. Distributed Artificial Intelligence: Computational Paradigms and Optimization Strategies

Workloads of artificial intelligence applied across distributed computing systems bring along a sophisticated optimization landscape that includes training efficiency, inference latency, communication overhead, and model accuracy. The whole datasets are brought together to one location using conventional centralized training methods, which become increasingly impractical given privacy regulations, bandwidth constraints, and sovereignty issues. These limitations are overcome through federated learning architectures by facilitating collaborative model training on millions of mobile devices or organizational silos without raw data centralization, with real-world implementations including 10 million to 100 million participating devices producing local model updates that get aggregated at coordination servers [5]. The efficiency in communication challenge is also especially critical if one takes into account the fact that normal deep neural network models have between 10 million and 100 million parameters, which for normal 32-bit floating-point representation means uploading requirements of 40 megabytes to 400 megabytes per training iteration, generating huge bandwidth usage that would be overwhelming in mobile networks if not sent compressed [5]. Distributed AI paradigms have thus become fundamental methodologies to contemporary machine learning use cases, requiring powerful communication reduction mechanisms to render federated training realistically feasible within resource-limited networks [5].

Parallel processing of distinct data subsets by identical model copies is the underlying methodology to distributed training, synchronizing gradient updates at regular intervals to ensure model consistency. Training throughput scaling is attained by contemporary distributed deep learning deployments, being close to linear in the case of small clusters to sub-linear with large-scale deployments, with efficiency factors usually in the range of 0.5 to 0.9 in scaling from single-node to 100-node deployments, meaning doubling computational resources doubles training speed by factors of 1.5 to 1.8 instead of the optimal factor of 2.0 [6]. Communication overhead is the main obstacle to ideal scaling, accounting for 20% to 80% of overall training time based on model size, network bandwidth, and synchronization policy, with communication-to-computation ratios rising significantly for large parameter models compared to computational complexity per example during training [6]. Volumes of communication can be compressed by factors from 10x to 1000x using state-of-the-art gradient compression methods with structured updates and sketching approaches preserving convergence properties at the same level as uncompressed training, with a reduction of bandwidth from 400 megabytes per iteration to 4 megabytes or even 400 kilobytes in the case of very sparse gradient representation [5]. Various computing nodes implement various parts of one model with model parallelism, providing an alternative solution for very large neural networks that cannot be accommodated in single-device memory limits, allowing for models with 1 billion to 175 billion parameters to be spread across multiple GPUs or computer nodes when single-device memory sizes of 16 to 80 gigabytes are inadequate [6].

Privacy-preserving distributed AI is also most promising in federated learning, which allows model training on decentralized data sources without raw data centralization. Federated optimization algorithms like Federated Averaging can realize model accuracies within 1% to 3% of centralized training baselines by minimizing communication needs by 10x to 100x through local computation methods that execute multiple gradient descent steps on each client before synchronization, as exemplified by empirical experiments [5]. Novel challenges about non-identically distributed data, communication efficiency, and protection from adversarial participants are introduced via federated learning, though with statistical heterogeneity among clients having the potential to necessitate 1.5x to 3x as many communication rounds as independent and identically distributed data environments in order to reach desired accuracy levels [5]. These constraints have started being met with newer developments in secure aggregation protocols, differential privacy mechanisms of privacy budgets epsilon = 0.1 to 10, and federated learning approaches that are tailored to individual users, making federated methods applicable to a

wider range of more sophisticated real-world applications [5][6].

4. Application Domains and Use Case Analysis

The practical implications of edge-cloud synergy and distributed AI manifest across diverse application domains, each presenting distinct technical requirements and performance constraints. Autonomous vehicle systems exemplify scenarios demanding ultra-low latency edge processing for immediate hazard detection and collision avoidance, while leveraging cloud resources for map updates, traffic pattern analysis, and fleet-wide learning. The emergence of Narrowband IoT (NB-IoT) technology as a low-power wide-area network standard has enabled massive IoT deployments supporting up to 50,000 to 100,000 devices per cell site, operating on 200 kilohertz bandwidth channels within existing LTE frequency bands and delivering theoretical peak data rates of approximately 250 kilobits per second for downlink and 20 kilobits per second for uplink communications [7]. This connectivity infrastructure supports diverse applications, including smart metering, asset tracking, and environmental monitoring, with devices capable of operating for 10 years or more on a single battery charge, consuming power levels below 1 milliwatt in idle mode, and transmitting data packets of 50 to 1,000 bytes at intervals ranging from minutes to hours [7]. The computational architecture must support real-time sensor fusion, object detection, and trajectory planning at the edge layer, supplemented by cloud-based services for high-definition map management and collective intelligence aggregation across vehicle fleets utilizing NB-IoT's extended coverage capabilities that penetrate buildings and underground locations with 20 decibel improved coverage compared to conventional cellular technologies [7].

Smart city infrastructure represents another compelling application domain, integrating thousands of IoT sensors monitoring traffic flow, air quality, energy consumption, and public safety. Comprehensive smart city architectures encompass six foundational pillars: smart governance facilitating citizen participation through digital platforms, smart economy promoting innovation and entrepreneurship, smart mobility optimizing transportation networks, smart environment monitoring air quality and energy consumption, smart people enhancing education and social inclusion, and smart living improving healthcare and safety services [8]. Implementation studies demonstrate that smart city deployments incorporating 10,000 to 50,000 interconnected sensors across urban areas of 100 to 500 square kilometers can reduce traffic congestion by 15% to 30%, decrease energy consumption by 20% to 40%, lower greenhouse gas emissions by 10% to 25%, and improve emergency response times by 20% to 35% through real-time data analytics and automated decision systems [8]. Edge computing nodes perform local analytics and immediate response actions, such as adaptive traffic signal control and anomaly detection, while cloud platforms aggregate city-wide data for long-term urban planning, resource optimization, and predictive modeling using machine learning algorithms trained on historical datasets spanning 6 to 36 months [8]. The distributed AI component enables federated learning across municipal boundaries, facilitating inter-city knowledge sharing while respecting data governance requirements and jurisdictional boundaries, with standardized protocols ensuring interoperability across heterogeneous sensor networks and communication technologies, including WiFi, ZigBee, cellular networks, and fiber-optic backbones [8].

Industrial automation and predictive maintenance applications illustrate the value proposition of distributed AI workloads in manufacturing environments. Sensor networks deployed across production facilities generate massive volumes of telemetry data requiring real-time analysis for quality control and equipment monitoring, with edge-based AI models detecting anomalous patterns indicative of impending equipment failures and triggering preventive maintenance actions with minimal latency [7]. Concurrently, cloud-based training pipelines continuously refine predictive models using historical data from multiple facilities, employing federated learning to preserve proprietary manufacturing process information while benefiting from collective intelligence. Healthcare informatics, particularly remote patient monitoring systems utilizing NB-IoT connectivity for wearable devices and implantable sensors, similarly leverage this architectural paradigm to balance real-time clinical decision support at the edge with population-level analytics and model refinement in the cloud [7][8].

5. Technical Challenges and Research Frontiers

Despite significant progress in both edge-cloud integration and distributed AI methodologies, numerous technical challenges persist, defining critical research frontiers for the coming years. Resource heterogeneity across edge devices,

ranging from resource-constrained IoT sensors to powerful edge servers, complicates workload placement and resource allocation decisions. Vehicular edge computing environments exemplify this heterogeneity, incorporating computational resources distributed across vehicles equipped with onboard processing units providing 10 to 100 gigaflops of computing power, roadside units deployed at intervals of 200 to 500 meters offering 100 to 1000 gigaflops capacity, and mobile edge computing servers co-located with base stations delivering 1 to 10 teraflops of processing capability [9]. The dynamic nature of vehicular networks, characterized by vehicle velocities ranging from 30 to 120 kilometers per hour and network topologies that change every 1 to 10 seconds as vehicles enter and exit communication range, necessitates rapid task offloading decisions that must be computed within 10 to 100 milliseconds to remain relevant [9]. Dynamic programming approaches and reinforcement learning techniques show promise for adaptive resource management, yet struggle with the computational overhead of continuous optimization in highly dynamic environments where the state space for optimal offloading decisions grows exponentially with the number of tasks and available edge nodes, often exceeding billions of possible configurations for systems with 10 to 100 concurrent tasks and 5 to 20 edge servers [9].

Communication efficiency remains a fundamental bottleneck in distributed AI systems, particularly as model complexity increases and network bandwidth becomes saturated. Federated learning deployments involving 100 to 10,000 participating clients face severe communication constraints, with typical mobile network uplink bandwidths of 1 to 10 megabits per second limiting the transmission of model updates containing 1 million to 100 million parameters to durations of 3 to 300 seconds per communication round [10]. Gradient compression techniques, including quantization reducing 32-bit floating-point representations to 8-bit or 4-bit integers, sparsification transmitting only gradient values exceeding threshold magnitudes representing 0.1% to 10% of total parameters, and low-rank approximation decomposing gradient matrices into products of lower-dimensional factors, offer partial solutions achieving compression ratios of 10x to 1000x [10]. However, these compression methods often introduce accuracy degradation of 1% to 5% on standard benchmark datasets, with the degradation severity depending on compression aggressiveness, model architecture complexity, and dataset characteristics [10]. Novel communication protocols leveraging hierarchical aggregation that reduces communication rounds by factors of 2x to 10x, peer-to-peer gradient exchange eliminating centralized bottlenecks, and adaptive communication scheduling that adjusts synchronization frequency from every iteration to every 5 to 50 iterations based on gradient convergence metrics represent active research areas seeking to minimize communication overhead while preserving convergence guarantees [10].

Fault tolerance and resilience pose additional challenges in distributed environments where node failures, network partitions, and Byzantine actors threaten system integrity. Federated learning systems with 1,000 to 100,000 edge participants experience client dropout rates of 10% to 50% per communication round due to device mobility, battery depletion, or network connectivity issues, requiring aggregation protocols that maintain convergence properties despite incomplete participation [10]. Traditional checkpoint-based recovery mechanisms introduce significant overhead and struggle with the scale of modern distributed systems. Emerging approaches based on coded computation, redundant gradient computation, and resilient aggregation protocols offer improved fault tolerance characteristics, though often at the cost of increased computational or communication overhead [9][10]. Security and privacy considerations permeate both edge-cloud architectures and distributed AI systems, with differential privacy mechanisms adding Gaussian or Laplacian noise calibrated to sensitivity parameters and privacy budgets epsilon ranging from 0.1 to 10, resulting in model accuracy reductions of 2% to 15% depending on the stringency of privacy requirements and the size of client datasets which may contain only 100 to 10,000 training samples per participant [10]. The development of efficient, scalable security and privacy mechanisms tailored to edge-cloud and distributed AI contexts represents an ongoing research priority [9][10].

Table 1: *IoT Device Proliferation and Cloud Computing Limitations [1][2]*

Aspect	IoT Ecosystem Characteristics	Cloud Computing Constraints
Device Growth	Exponential expansion of connected endpoints across industrial and consumer segments	Limited scalability for real-time applications requiring immediate response
Data Generation	Massive heterogeneous data streams	Insufficient bandwidth capacity for

	from sensors and multimedia sources	centralized data transmission
Latency Requirements	Sub-millisecond response demands for critical applications	Round-trip delays exceed acceptable thresholds for time-sensitive operations
Network Infrastructure	Distributed endpoints across geographic regions	Physical distance introduces inherent communication delays

Table 2: *Edge-Cloud Architectural Integration Principles [3][4]*

Component	Edge Computing Layer	Cloud Computing Layer	Integration Mechanism
Processing Location	Proximate to data sources and end users	Centralized data centers with distributed infrastructure	Hierarchical computational distribution
Latency Characteristics	Sub-millisecond to millisecond response times	Hundreds of milliseconds for remote operations	Workload partitioning based on time sensitivity
Resource Capacity	Limited computational power at the network periphery	Virtually unlimited processing and storage resources	Dynamic task allocation between tiers
Connectivity Model	Autonomous operation during network disruptions	Continuous connectivity requirement	Synchronization protocols for intermittent connectivity

Table 3: *Distributed AI Training Paradigms and Optimization [5][6]*

Training Approach	Federated Learning	Data Parallelism	Model Parallelism
Data Distribution	Decentralized across client devices	Partitioned subsets across worker nodes	Shared across computational nodes
Communication Pattern	Periodic model update aggregation	Synchronous gradient exchange	Layer-wise parameter distribution
Privacy Preservation	Raw data remains on local devices	Centralized dataset aggregation is required	Centralized training with distributed computation
Scalability Characteristics	Communication is constrained by upload bandwidth	Sub-linear scaling with cluster size	Memory-constrained by model architecture
Compression Techniques	Structured updates and gradient sketching	Quantization and sparsification methods	Activation checkpointing and layer partitioning

Table 4: *Application Domain Requirements and Implementations [7][8]*

Application Domain	Connectivity Technology	Edge Processing Requirements	Cloud Services Role
Autonomous Vehicles	Low-power wide-area networks with extended coverage	Real-time sensor fusion and trajectory planning	Map management and fleet intelligence aggregation
Smart Cities	Heterogeneous sensor networks with multiple protocols	Local analytics and immediate response actions	City-wide data aggregation and predictive modeling
Industrial Automation	Device connectivity with long battery life	Anomaly detection and quality control	Historical data analysis and model refinement
Healthcare Informatics	Wearable and implantable sensor connectivity	Real-time clinical decision support	Population-level analytics and epidemiological studies

6. Conclusions

The integration of edge-cloud computing architectures with distributed artificial intelligence workloads establishes a transformative computational paradigm addressing the fundamental limitations of traditional centralized systems. The hierarchical framework leverages edge computing's proximity advantages for latency-critical operations while exploiting cloud infrastructure's scalability for complex analytics and long-term storage. This architectural synthesis proves essential for emerging applications, including autonomous transportation, smart urban

infrastructure, industrial automation, and healthcare informatics, each demanding simultaneous satisfaction of real-time responsiveness, data privacy, and computational efficiency. Distributed AI methodologies, particularly federated learning and data parallelism, enable collaborative model training across geographically dispersed environments without centralizing sensitive data, achieving model accuracies comparable to centralized approaches while reducing communication overhead through advanced compression and aggregation techniques. Despite substantial advances, persistent challenges encompass resource heterogeneity across edge devices, communication bottlenecks in distributed training, fault tolerance in dynamic environments, and security mechanisms balancing privacy protection with model accuracy. Emerging solutions employing reinforcement learning for adaptive resource allocation, hierarchical gradient aggregation, coded computation for resilience, and differential privacy mechanisms demonstrate promising pathways toward addressing these limitations. The continued evolution of edge-cloud and distributed AI frameworks will fundamentally shape the computational infrastructure supporting data-intensive applications, with particular emphasis on developing efficient orchestration protocols, optimized communication strategies, and robust security mechanisms. As IoT device proliferation accelerates and AI model complexity increases, the importance of seamlessly integrated edge-cloud architectures coupled with distributed intelligence capabilities will intensify, establishing these domains as foundational pillars of modern computational systems. Future advancements will likely focus on unified frameworks that holistically address workload partitioning, cross-layer optimization, and automated adaptation to dynamic environmental conditions, ultimately enabling unprecedented levels of system performance, scalability, and reliability for next-generation applications.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Satyajit Sinha, "State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion," IoT Analytics, May 2021. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices-2021/

[2] Weisong Shi, et al., "Edge computing: Vision and challenges," IEEE, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7488250

[3] Mahadev Satyanarayanan, "The emergence of edge computing," ResearchGate, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7807196

[4] Arif Ahmed, Ejaz Ahmed, "A Survey on Mobile Edge Computing," ResearchGate, 2016. [Online]. Available: https://www.researchgate.net/publication/285765997_A_Survey_on_Mobile_Edge_Computing

[5] Jakub Konečný, et al., "Federated learning: Strategies for improving communication efficiency," arXiv, 2017. [Online]. Available: https://arxiv.org/abs/1610.05492

[6] Tal Ben-Nun, Torsten Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," ACM Digital Library, 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3320060

[7] Alexander S. Gillis, "What is narrowband IoT (NB-IoT)? Definition from TechTarget," TechTarget, 2025. [Online]. Available: https://www.techtarget.com/whatis/definition/narrowband-IoT-NB-IoT

[8] Andrea Zanella, et al., "Internet of Things for smart cities," IEEE, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6740844

[9] Lei Liu, et al., "Vehicular edge computing and networking: A survey," SpringerNature Link, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11036-020-01624-1

[10] Peter Kairouz, et al., "Advances and open problems in federated learning,"arxiv, 2021. [Online]. Available: https://arxiv.org/abs/1912.04977