



RESEARCH ARTICLE

OPEN ACCESS

COLLABORATIVE CLOUD-EDGE COMPUTING: A SURVEY ON ARCHITECTURES, ORCHESTRATION, AND DISTRIBUTED AI PARADIGMS

I Made Suartana^{*1}, Ricky Eka Putra² and Rahadian Bisma³

^{1, 2}Department of informatics, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia.

³Department of Information System, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya, Indonesia.

¹<https://orcid.org/0000-0002-2877-5778>, ²<http://orcid.org/0000-0002-5515-7967>, ³<http://orcid.org/0000-0003-0979-5148>

Email: *madesuartana@unesa.ac.id, rickyeka@unesa.ac.id, rahadianbisma@unesa.ac.id

ARTICLE INFO

Article History

Received: March 31, 2026

Reviewed: August 3, 2026

Accepted: September 3, 2026

Published: September 30, 2026

Keywords:

cloud computing,
edge computing,
fog computing,
federated learning,
orchestration,
distributed AI.

ABSTRACT

The rapid evolution of digital infrastructure has catalyzed the emergence of novel computing paradigms that integrate cloud computing with edge computing within complex collaborative architectures. This paper presents a comprehensive survey of recent literature (2010–2026) encompassing three principal dimensions: (1) cloud-edge collaborative architectures including hierarchical models, network topologies, and workload distribution mechanisms; (2) orchestration frameworks supporting dynamic resource management, intelligent task scheduling, and latency optimization; and (3) distributed artificial intelligence paradigms including Federated Learning, Split Computing, and on-device inference at the edge layer. Through a systematic analysis of 78 Scopus-indexed publications, we identify evolutionary trends, open challenges, and future research directions. Survey findings indicate that the integration of adaptive policy-based orchestration mechanisms with AI inference capabilities at the edge layer constitutes the most determinant factor in improving energy efficiency and reducing end-to-end latency in cloud-edge systems. These findings are expected to provide guidance for researchers and practitioners in designing next-generation distributed computing systems.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

The last decade witnessed a fundamental transformation in how computing systems are designed and operated. The traditional data center-centric model is increasingly faced with fundamental limitations when dealing with the needs of modern applications: sub-millisecond latency for industrial control systems, data privacy in healthcare applications, and massive scalability for the Internet of Things (IoT) ecosystem which is projected to reach 30 billion devices by 2030 [1], [2]. The edge computing paradigm emerged as a response to these limitations, by geographically placing computing resources closer to data sources and end users. However, pure edge architectures also face their own limitations: limited computing capacity, high device heterogeneity, and the difficulty of managing distributed resources [3], [4]. The solution that is now receiving widespread attention is the synergistic integration between cloud and edge in what is called a collaborative cloud-edge computing architecture. This collaborative architecture is not merely a technical combination of two paradigms but rather gives rise to fundamentally new orchestration challenges. How are computing tasks optimally distributed among the cloud, fog, and edge layers? How can artificial intelligence (AI) models be trained and inferred in a distributed manner without sacrificing data privacy? How can orchestration frameworks adapt in real-time to changes in network conditions and resource availability? These questions have triggered a significant explosion of research in the last five years. Several previous surveys have addressed specific aspects of this domain. According to [5] provided an initial overview of mobile edge computing; by [6] focused on edge intelligence, while [7] examined the challenges of edge computing in the context of IoT. However, no survey has holistically integrated the three dimensions – collaborative architecture, orchestration frameworks, and distributed AI – especially in the current development period (2021-2026) which is marked by significant advances in Large Language Model (LLM) deployment at the edge, 5G/6G technology, and container orchestration platform standardization.

This survey is motivated by three critical observations from recent literature: First, there is a gap in understanding how modern orchestration mechanisms adapt to the dynamics of distributed AI workloads. The latest generation of deep learning models, especially transformer-based models, have computing profiles that are very different from traditional applications, yet existing orchestration frameworks are mostly designed for conventional workloads [8], [9]. Second, although Federated Learning (FL) has grown rapidly as a privacy-preserving model training paradigm, the integration of FL with cloud-edge infrastructure still faces unsolved technical challenges, including data statistical heterogeneity, inefficient communication, and security against Byzantine attacks [10], [11]. Third, the emergence of 5G/6G technology as connectivity infrastructure opens new opportunities as well as new challenges in cloud-edge architecture design, especially related to network slicing and multi-access edge computing (MEC) integrated with telecommunication standards [12]. The main contributions of this paper are as follows:(1) A comprehensive taxonomic framework that classifies collaborative cloud-edge architectures based on the dimensions of topology, distribution mechanisms, and programming models;(2) A systematic comparative analysis of leading orchestration frameworks (Kubernetes, OpenStack, Apache Mesos, ETSI MEC) in the context of distributed AI workloads;(3) An in-depth synthesis of distributed AI paradigms in the cloud-edge ecosystem, including Federated Learning, Split Inference, Knowledge Distillation, and Neural Architecture Search for edge devices;(4) Identification of emerging research trends and research roadmaps for the scientific community.

II. RESEARCH METHODOLOGY

II.1 LITERATURE SEARCH STRATEGY

This survey adopted the Systematic Literature Review (SLR) methodology strictly structured based on the PRISMA 2020 guidelines. The literature search was focused on four leading academic databases relevant to computer science and engineering: Scopus, IEEE Xplore, ACM Digital Library, and Web of Science. To ensure comprehensive yet specific coverage, we utilized the following advanced Boolean search string: (cloud-edge OR cloud edge computing OR edge-cloud) AND (orchestration OR resource management) AND (distributed AI OR federated learning OR split computing OR edge inference).

II.2 INCLUSION AND EXCLUSION CRITERIA

To filter the most relevant and high-quality articles, this research established strict inclusion criteria (IC):

- IC-1: Publications are peer-reviewed journal articles or leading international conference proceedings indexed by Scopus.
- IC-2: The publication timeframe ranges from January 2010 to February 2026, to capture historical evolution up to the latest trends.
- IC-3: Substantially focuses on at least two of the three main survey dimensions: (1) Collaborative computing architectures, (2) Orchestration frameworks, and/or (3) Distributed AI paradigms.

The exclusion criteria (EC) included: review/survey papers (to avoid redundancy), preprint papers that have not been peer-reviewed, non-English articles, and papers with inaccessible full texts.

II.3 PRISMA SELECTION PHASES

The selection process was conducted in four main phases that systematically reduced the literature volume based on relevance and quality, as illustrated in Table 1 and Figure 1:

- Identification Phase: The initial search across the four databases yielded a total of 342 articles. At this early stage, 58 duplicated articles across databases and 97 automatically detected irrelevant articles were removed (total initial exclusion: 155 articles).
- Screening Phase: From the remaining 187 unique articles, a manual review of titles and abstracts was conducted. In this phase, 75 articles were excluded because their primary focus did not align with the cloud-edge computing scope.
- Eligibility Phase: A total of 112 articles were deemed eligible for full-text download and comprehensive review. Following an in-depth analysis against the inclusion criteria and methodological quality, 34 articles were further excluded due to a lack of experimental depth or failure to explicitly integrate at least two of the main survey dimensions.
- Final Inclusion Phase: This final phase left 78 primary publications of the highest quality, which were then data-extracted, synthesized, and used as the main references in this survey paper.

Table 1: Summary of Literature Selection Process Based on PRISMA 2020.

PRISMA Phase	Number of Articles	Description
Identification	342	Initial search results from 4 databases
Title/Abstract Screening	187	After removing duplicates (n=58) and irrelevant (n=97)
Eligibility (Full-text)	112	After removing duplicates (n=58) and irrelevant (n=97)
Final Inclusion	78	Met all inclusion criteria and methodological quality

Source: Authors, (2026).

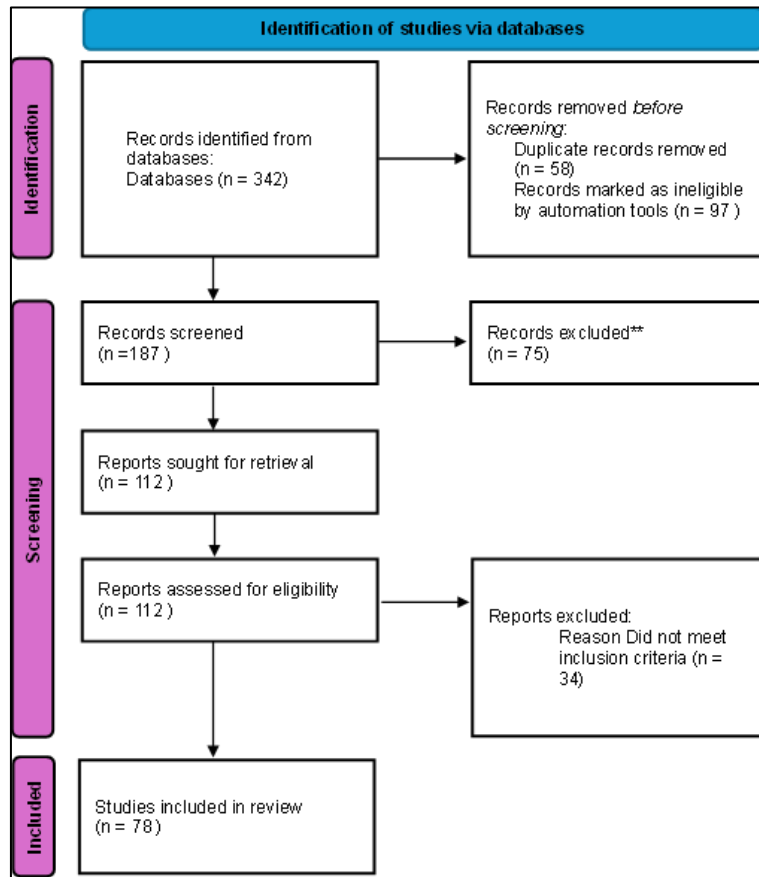


Figure 1: PRISMA 2020 flow diagram for the systematic literature review process.
Source: Authors, (2026).

III. BACKGROUND AND BASIC CONCEPTS

III.1 EVOLUTION OF COMPUTING PARADIGMS

To understand the current cloud-edge computing landscape, it is important to trace the evolutionary trajectory of computing paradigms over the last two decades. The centralized computing model that dominated the mainframe era evolved into the client-server model in the 1990s, then transitioned to cloud computing in the 2000s with the emergence of services such as Amazon EC2 (2006) and Google App Engine (2008) [13]. Although cloud computing revolutionized how organizations manage IT infrastructure, this model has inherent limitations for certain classes of applications. Round-trip latency to cloud data centers, which generally ranges between 50-100 ms for regional connections and can reach 200ms for intercontinental connections, is unacceptable for applications such as autonomous vehicles, remote surgery, or real-time industrial control systems that require sub-10 ms latency [14], [15]. The first response to these limitations was the Content Delivery Network (CDN) concept which distributed static content to geographical edge servers. Then, the cloudlet concept was introduced by Satyanarayanan et al. as a "data center in a box" placed at the edge of the network. Further developments gave birth to fog computing initiated by Cisco in 2014, and Mobile Edge Computing (MEC) standardized by ETSI, which was later renamed Multi-Access Edge Computing to accommodate non-cellular connectivity [16].

III.2 DEFINITION AND TAXONOMY OF COMPONENTS

In the context of this survey, we define the components of the cloud-edge ecosystem as follows based on the consolidation of definitions from recent literature:

III.2.1 Cloud Layer

The cloud layer represents large-scale data centers that offer virtually unlimited elasticity in computing, storage, and networking. This layer functions as a global orchestrator, AI model repository, and large-scale historical data processor. In the cloud-edge context, the cloud layer acts as a global coordinator overseeing workload distribution to the layers below it [17], [18].

III.2.2 Fog/Edge Intermediate Layer

The fog or intermediate edge layer consists of gateways, high-capacity routers, and micro data centers placed at the metropolitan network edge. This layer functions as a data aggregator from end-user devices, performs preprocessing, and makes local decisions. By [19] classified this layer into three sub-tiers based on computing capabilities: High-end Edge (HEE), Mid-range Edge (MEE), and Light-weight Edge (LWE).

III.2.3 Device/Mist Layer

The device layer (or mist computing) includes sensors, actuators, smartphones, and IoT devices that have limited computing capabilities. Advances in System-on-Chip (SoC) technology with integrated Neural Processing Units (NPUs), such as Qualcomm Snapdragon, Apple Neural Engine, and Google Tensor, have significantly improved AI model inference capabilities in this layer [20], [21].

Table 2: Comparison of Layer Characteristics in Cloud-Edge-Device Architecture.

Characteristics	Cloud Layer	Fog/Edge Layer	Device Layer
Latency (ms)	50 – 200	1 – 10	< 1
Bandwidth (Gbps)	1 – 100	0.1 – 10	0.001 – 1
Computing Capacity	Very High (PFLOPS)	Medium (TFLOPS)	Low (GFLOPS)
Energy Consumption	High (MW)	Medium (kW)	Low (W)
Historical Data Availability	Full	Partial	Minimal
Physical Security	High (data center)	Medium	Low (distributed)
Scalability	Elastic (on-demand)	Limited	Very Limited

Source: Authors, (2026).

IV. COLLABORATIVE CLOUD-EDGE COMPUTING ARCHITECTURES

IV.1 HIERARCHICAL ARCHITECTURE MODELS

The three-tier hierarchical architecture is the most common model identified in the surveyed literature. This model divides the computing ecosystem into the cloud layer (C-tier), the edge/fog layer (E-tier), and the device layer (D-tier) with vertically structured data and computation flows. For authors [22] proposed an adaptive hierarchical architecture where boundaries between layers can be dynamically reconfigured based on network conditions and workload distribution. According to [23] expanded the three-tier model by introducing the concept of Virtual Edge Node (VEN) – a software abstraction that can be migrated between physical layers without service interruption. Their research showed that VEN can reduce end-to-end latency by up to 47% compared to static models under temporally varying workload scenarios. Furthermore, perspectives from the telecommunications community through the ETSI MEC standard provide a more formal architectural definition. The ETSI GS MEC 003 standard defines a reference architecture with three main components: MEC Host (edge computing platform), MEC Platform Manager (application lifecycle management), and Multi-access Edge Orchestrator (cross-host orchestration) [24]. The integration of the ETSI MEC standard with cloud-native platforms like Kubernetes became one of the most active research areas during this survey period.

IV.2 CLOUD-EDGE NETWORK TOPOLOGIES

Beyond vertical hierarchical models, the literature identifies various more complex cloud-edge network topologies. We classify these topologies into four main categories based on the connectivity structure between nodes:

- Star-Hierarchical Topology is the simplest configuration where all edge nodes connect directly to one or several cloud gateways. Although easy to manage, this topology is vulnerable to bottlenecks at the cloud gateway and is not scalable for large-scale deployments [25].
- Mesh-Edge Topology allows edge nodes to communicate directly with each other (East-West traffic) in addition to the cloud (North-South traffic). By [26] demonstrated that mesh topology can significantly reduce the network load on the cloud layer by enabling local collaboration among edge nodes for tasks that do not require access to global resources.
- Hierarchical-Federated Topology combines federations of edge clusters, each with a local coordinator, which then connect to a global coordinator at the cloud layer. This model perfectly suits the Federated Learning paradigm and has been adopted by frameworks like KubeEdge and OpenYurt [27], [28].
- Dynamic Heterogeneous Graph (DHG) Topology is the newest model representing the cloud-edge ecosystem as a heterogeneous weighted graph where nodes can have different capabilities and edge weights represent connection characteristics. In [29] utilized DHG representations to design efficient task placement algorithms using Graph Neural Networks.

IV.3 WORKLOAD DISTRIBUTION MECHANISMS

Workload distribution in cloud-edge architectures is the process of determining where application components are executed among the various available computing layers. This is a multi-objective optimization problem that simultaneously considers latency, throughput, energy consumption, and cost [30]. Partitioning-based approaches slice the application's computation graph at optimal points and distribute the partitions to different layers. In [31] proposed an adaptive DNN partitioning method using Dynamic Programming with $O(n^2)$ complexity that is capable of finding the optimal partition point in less than 100 ms for models up to 100 layers. Container-based approaches package each application microservice in a container that can be migrated between layers. The use of Kubernetes platforms with edge extensions (KubeEdge, K3s) allows for transparent container migration, but research by [32] shows that container cold-start overhead on low-power edge devices remains a significant challenge, potentially reaching 2-5 seconds for medium-sized images. Edge serverless (or edge FaaS) function-based approaches are an increasingly prominent trend. This model allows the deployment of small functions at the edge layer without explicit infrastructure management. According to [33] evaluated four edge serverless platforms (OpenFaaS, KNative, Fission, OpenWhisk) and found that KNative demonstrated the best performance in burst traffic scenarios at the edge with an average cold-start latency overhead of 340 ms.

Table 3: Comparison of Workload Distribution Approaches in Cloud-Edge Architectures.

Approach	Granularity	Migration Overhead	Mgmt Complexity	Key References
Graph Partitioning	DNN Layer	Low	Medium	[31],[34],[35]
Container-based	Microservice	Medium	Medium	[32],[36],[37]
Serverless/FaaS	Function	High (cold-start)	Low	[33],[38]
VM Migration	Full application	High	High	[39],[40]
Process Migration	OS Process	Medium	High	[41]

Source: Authors, (2026).

V. CLOUD-EDGE ORCHESTRATION FRAMEWORKS

V.1 CONCEPTS AND ORCHESTRATION DIMENSIONS

Orchestration in the cloud-edge context refers to the automation of lifecycle service management, resources, and operational policies across heterogeneous architecture layers. Unlike pure cloud orchestration which operates on controlled and homogeneous infrastructure, cloud-edge orchestration must handle hardware heterogeneity, network connection intermittency, and much higher workload dynamics [42]. We identified five critical orchestration dimensions in cloud-edge systems based on literature analysis:

- (1) Resource Discovery and Inventory Management: the ability to automatically discover and catalog available computing resources across layers, including edge devices joining and leaving the cluster dynamically.
- (2) Task Scheduling and Placement: the optimal determination of where and when computing tasks are executed based on latency requirements, resource availability, and business policies.
- (3) Service Lifecycle Management: managing the deployment, updating, and deletion of services at the edge layer with minimal downtime.
- (4) Network Policy Orchestration: configuring and enforcing network policies including Quality of Service (QoS), network slicing, and secure cross-layer communication.
- (5) Adaptive Resource Scaling: the ability to automatically adjust resource allocation based on changes in demand, including scale-out to the cloud layer when edge capacity is insufficient [43], [44].

V.2 LEADING ORCHESTRATION PLATFORMS

Kubernetes has become the de facto standard for container orchestration, but its original architecture designed for data centers is not suitable for deployment at the edge layer. Several extensions and forks of Kubernetes have been developed to address these limitations [45]. KubeEdge, developed by Huawei and donated to the CNCF, is the most cited Kubernetes extension in this survey's literature. KubeEdge adds an EdgeCore component that runs on edge nodes and a CloudCore that runs on the cloud layer, with a lighter communication protocol (MQTT) for edge nodes. An evaluation by [46] showed that KubeEdge is capable of managing edge nodes with resources as little as 1 CPU core and 256 MB of RAM. OpenYurt, originating from Alibaba, takes a different approach by adding edge capabilities to existing Kubernetes without altering its core components. The Yurt Node Pool concept allows the grouping of edge nodes based on geographic locations with different policies per pool. This non-invasive approach facilitates adoption in existing Kubernetes clusters [47]. Aside from the Kubernetes ecosystem, the ETSI MEC Framework provides a more formal architectural standard with standardized interfaces (Mp1, Mm4, Mm5) between MEC components. However, research by [48] identified the lack of mature open-source implementations as the main barrier to ETSI MEC adoption outside the telecommunications operator ecosystem. Intent-driven orchestration approaches are emerging as a promising recent trend. In this model, operators define business objectives (intent), and the orchestration system automatically translates that intent into specific technical configurations. By [49] proposed IDEA (Intent-Driven Edge Autonomy) which uses reinforcement learning to optimize orchestration configurations based on high-level defined SLAs.

V.3 INTELLIGENT TASK SCHEDULING

Task scheduling in cloud-edge systems is an NP-hard problem requiring heuristics or machine learning methods to provide near-optimal solutions within an acceptable time. Recent literature identifies three main categories of approaches [50], [51]:

Reinforcement Learning (RL)-based approaches model the scheduling problem as a Markov Decision Process (MDP) where an agent learns the optimal policy through interaction with the environment. Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) are the most widely used RL algorithms. The advantage of this approach is its ability to adapt to dynamically changing conditions, but low sample efficiency and convergence issues remain a challenge [52]. Graph Neural Network (GNN)-based methods represent tasks and resources as nodes in a graph with edges representing dependencies and connectivity. GNN-based schedulers can extract structural features of scheduling problems that are difficult to capture by conventional methods. According to [53] demonstrated a 23% improvement in scheduling efficiency compared to the best heuristics using a GNN approach. Meta-learning methods for scheduling develop models that can quickly adapt to new workload characteristics with little data (few-shot adaptation). This approach is highly relevant for cloud-edge systems operated in various application domains with varying workload patterns [54].

Table 4: Comparison of Workload Distribution Approaches in Cloud-Edge Architectures.

Method	Adaptability	Inference Time	Relative Performance	Training Complexity	Reference
Greedy Heuristic	Low	< 1 ms	Baseline	None	[50]
Genetic Algorithm	Medium	10-100 ms	+12-18%	None	[55]
Deep Q-Network	High	1-5 ms	+25-35%	High	[52],[56]
PPO-based	High	1-5 ms	+28-40%	Very High	[57]
GNN-based	Medium High	5-20 ms	+20-30%	High	[53],[58]
Meta-Learning	Very High	10-50 ms	+30-45%	Very High	[54]

Source: Authors, (2026).

VI. DISTRIBUTED AI PARADIGMS IN CLOUD-EDGE ECOSYSTEMS

VI.1 FEDERATED LEARNING

Privacy-Based Model Training: Federated Learning (FL), introduced by [59], enables collaborative machine learning model training on geographically dispersed data without the need to collect raw data to a central server. In the cloud-edge context, FL becomes a highly relevant paradigm considering that most data reside on edge devices or end-user devices. The fundamental challenge of FL in the cloud-edge ecosystem is statistical heterogeneity—data on each client tends to be non-identically distributed (non-IID), which can cause global model divergence. In [60] proposed FedProx, a variant of FedAvg that adds a proximal term to limit how far the local model can deviate from the global model and showed a significant improvement in convergence stability in extreme non-IID scenarios. The second challenge is system heterogeneity: edge devices have highly variable computing capacity, bandwidth availability, and battery power. Asynchronous FL becomes a solution where the server does not need to wait for all clients to complete training before performing aggregation. SCAFFOLD (Stochastic Controlled Averaging for FL) proposed by [61] addresses the client drift problem by using a control variance estimator. The security aspect of FL has become a serious concern with the identification of various attack vectors. Byzantine attacks involve clients sending malicious gradients to manipulate the global model. Byzantine-robust aggregation techniques such as Krum, Coordinate-Median, and Trimmed Mean have been proposed as defenses. According to [62] showed that Krum can tolerate up to $(n-2)/2$ Byzantine clients where n is the total number of clients. Recent developments in FL include Personalized FL (pFL) which aims to produce personalized models for each client rather than a single global model. This approach is highly relevant for edge applications where user preferences and contexts are highly diverse. For authors [63] proposed pFedMe which uses the Moreau envelope as a personalization regularizer and demonstrates a better balance between global and local models

VI.2 SPLIT COMPUTING AND COLLABORATIVE INFERENCE

Split Computing (or Split Inference) is a paradigm where the computation of a DNN model is divided between edge devices and cloud/edge servers to optimize the trade-off between latency, energy consumption, and accuracy. Unlike FL which focuses on the training phase, Split Computing relates to the inference phase [64], [65]. Neurosurgeon, proposed by [66], is the first system that automatically determines the optimal split point in a DNN model based on network conditions and device capabilities. This system considers two costs: the transmission cost of the intermediate tensor and the computation cost on each side of the split. Experiments show Neurosurgeon can reduce end-to-end inference latency by up to $3.1\times$ compared to cloud-only execution. One of the main challenges of Split Computing is the large size of intermediate tensors at certain points in the DNN model, which can result in transmission bottlenecks. The bottleneck injection technique—adding a compression layer before the split point—can reduce the intermediate representation size. In [67] showed that head network compression based on knowledge distillation can reduce the intermediate tensor size by up to $32\times$ with an accuracy degradation of less than 1%. Early exit inference is a complementary approach where the model has several "exit points" that allow inference to finish earlier for easily classified samples, significantly saving computation. According to [68] proposed BranchyNet which integrates early exits with a split architecture, enabling dynamic trade-offs between accuracy and latency based on runtime conditions

VI.3 KNOWLEDGE DISTILLATION AND MODEL COMPRESSION FOR THE EDGE

The deployment of large-capacity AI models at the edge layer requires effective model compression techniques. Knowledge Distillation (KD), introduced by [69], transfers knowledge from a large model (teacher) to a smaller model (student) that can be deployed at the edge. In the cloud-edge context, KD is used to generate an edge version of the model distilled from the full model running in the cloud. In [70] proposed Customized-Model Distillation (CMD) which adapts the distillation process based on the computational characteristics of the target edge device, producing Pareto-optimal models on the accuracy-latency curve for each device class. Quantization is another compression technique that reduces the numerical precision of model parameters from 32-bit floating point (FP32) to a lower format such as INT8 or even INT4. Post-Training Quantization (PTQ) does not require re-training and can be applied to existing models, while Quantization-Aware Training (QAT) incorporates quantization effects into the training process to produce more robust models. For authors [71] demonstrated that state-of-the-art QAT techniques can achieve performance close to FP32 with INT4 quantization. Network Pruning removes model parameters deemed non-critical based on various criteria such as magnitude pruning or gradient-based pruning. Structured pruning, which removes entire convolution filters or channels, is more friendly to hardware acceleration compared to unstructured pruning, which produces sparse matrices that are difficult to optimize on conventional hardware [72].

VI.4 LARGE LANGUAGE MODELS AT THE EDGE

Challenges and Opportunities: The emergence of Large Language Models (LLMs) such as GPT, LLaMA, and Mistral in the last two years has created new demands in the cloud-edge ecosystem. These models have billions of parameters that require tens to hundreds of gigabytes of memory, far exceeding the capacity of conventional edge devices. However, the reasoning and text generation capabilities of LLMs open up transformative edge applications [73]. Several approaches to overcome these challenges have been proposed. Parameter-Efficient Fine-Tuning (PEFT) such as LoRA (Low-Rank Adaptation) and Prefix Tuning allow the adaptation of large LLMs on edge devices by training only a small fraction of the parameters. Following [74] showed that LoRA can achieve performance comparable to full fine-tuning by training only about 0.1-1% of the total parameters. Speculative Decoding is an LLM inference technique that uses a small draft model to rapidly generate token candidates at the edge, then verifies those candidates with a large model in the cloud. This approach allows flexible trade-offs between output quality and latency and has shown a 2-3× inference speedup on standard benchmarks [75]. Collaborative LLM Inference between edge devices and cloud servers, where each layer handles a specific part of the transformer layer sequence, is an active research area called Pipeline Parallelism for Edge LLMs. Following [76] proposed the FlexInfer framework which dynamically allocates transformer blocks to different layers based on network conditions and memory availability.

Table 5: Summary of Distributed AI Paradigms in Cloud-Edge Ecosystems.

AI Paradigm	Phase	Data Privacy	Bandwidth Req.	Edge Compute Req.	Key References
Federated Learning	Training	High	Medium	High	[59-61],[63]
Split Computing	Inference	Medium	Medium-High	Partial	[64-67]
Knowledge Distillation	Deployment	N/A	Low (offline)	Low	[69-71]
Early Exit Inference	Inference	High	Low	Low-Medium	[68]
Edge LLM (LoRA)	Fine-tuning	High	Low	Medium	[73],[74]
Speculative Decoding	Inference	Medium	Medium	Low	[75]

Source: Authors, (2026).

VII. CROSS-CUTTING ISSUES

VII.1 SECURITY AND PRIVACY IN CLOUD-EDGE SYSTEMS

Security is a multi-dimensional challenge in the cloud-edge ecosystem encompassing aspects that cannot be entirely solved by traditional cloud security approaches. Physically dispersed edge devices are more vulnerable to physical attacks, and device heterogeneity complicates the enforcement of uniform security policies [77]. Zero Trust Architecture (ZTA) is increasingly adopted as a security framework for cloud-edge. The "never trust, always verify" principle of ZTA is highly relevant for distributed edge environments where traditional network perimeters are no longer meaningful. NIST SP 800-207 defines ZTA components that can be adapted for the edge context [78]. Trusted Execution Environments (TEEs) such as Intel SGX, ARM TrustZone, and AMD SEV provide hardware-isolated computing enclaves, enabling sensitive data processing on not fully trusted devices. However, research by [79] shows that TEE performance overhead can be significant, reaching 2-5× for memory-intensive workloads.

VII.2 ENERGY EFFICIENCY AND SUSTAINABILITY

Energy efficiency is a critical concern in the edge computing context where many devices operate on battery power or have strict power budget constraints. Furthermore, with increasing awareness of the environmental impact of computing, energy optimization at the cloud-edge scale has become an important agenda [80]. Dynamic Voltage and Frequency Scaling (DVFS) is a hardware-level technique that adjusts the processor voltage and frequency based on the workload. The integration of DVFS with cloud-edge workload schedulers allows adaptive trade-offs between performance and energy consumption. In [81] showed that DVFS-aware scheduling can reduce edge cluster energy consumption by up to 38% with a tolerable latency increase. The concept of Green Cloud-Edge Computing, which combines renewable energy sources with intelligent workload management, is a growing research area. Workload migration to edge nodes currently receiving renewable energy supplies (e.g., solar or wind) enables the optimization of the computing carbon footprint. Following [82] proposed a Carbon-Aware Scheduling framework capable of reducing operational carbon emissions of hybrid data centers by up to 31%.

VII.3 MOBILITY AND HANDOVER MANAGEMENT

User and edge device mobility creates unique challenges that do not exist in static cloud computing. When users move from one coverage area to another, services running on the local edge node need to be migrated to the new edge node with minimal downtime. This process is called service handover or task offloading migration [83]. Predicting user mobility to perform proactive handovers is a more effective approach compared to reactive handovers. Deep learning-based mobility models, particularly LSTM and Transformer, show significantly better location prediction accuracy compared to high-order Markov models on real-world mobility data [84].

VIII. EMERGING TRENDS AND FUTURE RESEARCH DIRECTIONS

VIII.1 INTEGRATION WITH 5G/6G TECHNOLOGIE

5G technology and 6G projections are fundamentally changing the edge computing landscape by providing Ultra-Reliable Low-Latency Communication (URLLC) and massive Machine-Type Communication (mMTC) connectivity. Network slicing in 5G allows the creation of isolated virtual networks with different QoS characteristics on the same physical infrastructure, which is highly relevant for sharing edge infrastructure among multiple tenants [85]. The integration of MEC with the 5G core network via Network Exposure Function (NEF) and Application Function (AF) APIs opens possibilities for edge applications utilizing real-time contextual network information, such as user location, signal strength, and handover predictions. The 3GPP community has defined standards facilitating this integration in releases 16 and 17 [86]. For 6G, projected for deployment around 2030, preliminary research identifies that AI/ML will be an inherent component of the network protocol stack, not just an add-on. The concept of Integrated Sensing and Communication (ISAC) in 6G, where communication infrastructure also functions as an environmental sensor, opens new dimensions in the cloud-edge architecture [87].

VIII.2 QUANTUM COMPUTING AT THE EDGE

Although quantum computing is currently still in the Noisy Intermediate-Scale Quantum (NISQ) phase with limited capabilities, preliminary research is exploring the synergy between quantum computing and edge AI. Quantum-Classical Hybrid algorithms such as Variational Quantum Eigen solver (VQE) and Quantum Approximate Optimization Algorithm (QAOA) show potential for solving combinatorial optimization problems (such as task scheduling in cloud-edge) more efficiently [88]. Machine Learning (QML) explores the possibility of using quantum circuits as machine learning models. Although the quantum advantage for QML has not been empirically proven in practical use cases, the rapid development of quantum hardware drives research into the integration of QML with edge inference pipelines [89].

VIII.3 DIGITAL TWIN FOR CLOUD-EDGE MANAGEMENT

Digital Twin, a real-time virtual representation of a physical system has become an increasingly important tool for managing and optimizing cloud-edge infrastructure. Edge network Digital Twins enable the simulation and prediction of system behavior before changes are implemented, reducing operational risks [90]. The integration of Digital Twins with reinforcement learning enables the development and testing of orchestration policies in a simulated environment before real-world deployment. This sim-to-real transfer approach addresses safety and sample efficiency issues which are the main obstacles of RL in production systems [91].

VIII.4 IDENTIFICATION OF RESEARCH GAPS

Based on a comprehensive literature analysis, we identified several research gaps that require further attention from the scientific community:

- Gap 1 – Interface Standardization: Although various cloud-edge orchestration platforms have been developed, the lack of widely accepted interoperability standards complicates the integration of components from different vendors. Standardization efforts from ETSI, CNCF, and IEEE need further convergence.
- Gap 2 – Comprehensive Benchmarks: Fair comparisons between cloud-edge systems are difficult to perform due to the lack of standard benchmarks covering various dimensions (latency, throughput, energy, cost). EdgeBench and AIOTBench are initial steps, but their scope is still limited.
- Gap 3 – FL with LLMs at the Edge: The integration of Federated Learning with LLM fine-tuning on edge devices remains very limited in the literature. The extremely high communication challenges (transmitting gigabyte-sized model gradients) require fundamentally different solutions than FL for small models.
- Gap 4 – Resilience and Fault Tolerance: Research on how cloud-edge systems maintain operations during partial and massive edge node failures is still very limited compared to research on performance under normal conditions.
- Gap 5 – Sociotechnical Aspects: The adoption of cloud-edge in the context of developing countries, including limitations in connectivity infrastructure and local technical expertise, is rarely discussed in the existing literature. Research that accommodates more diverse contexts is needed for more inclusive technology adoption.

IX. DISCUSSION

IX.1 SYNTHESIS OF FINDINGS

From a systematic analysis of 78 publications, we identified several key findings that answer the proposed research questions. First, there is a clear convergence toward a three-tier hierarchical architecture equipped with adaptive capabilities, where the boundaries between tiers are no longer rigid but can be dynamically reconfigured based on operational conditions [22], [23]. Second, Kubernetes-based orchestration platforms (especially KubeEdge and OpenYurt) have become the dominant paradigm for cloud-edge orchestration, but with significant limitations in small-scale edge scenarios and intermittent connectivity. Intent-driven and AI-based orchestration approaches show great potential but are still in the early stages of research [45-49]. Third, Federated Learning has matured as a viable training paradigm for cloud-edge systems, with various solutions addressing non-IID and system heterogeneity challenges. However, the integration of FL with LLMs and the latest generative models reopens fundamental questions regarding communication efficiency and convergence [59], [63], [73-76].

IX.2 RESEARCH AND PRACTICAL IMPLICATIONS

From a research perspective, this survey identifies an urgent need for research that deeply integrates the three surveyed dimensions—architecture, orchestration, and distributed AI—rather than examining each in isolation. The co-design of AI algorithms and AI-aware orchestration platforms is a promising direction. From a practical perspective, for organizations considering the adoption of a cloud-edge architecture, this survey indicates that Kubernetes-based platforms with edge extensions are the most mature options with the broadest support ecosystems. However, for deployments in environments with limited connectivity or ultra-low-power devices, more lightweight solutions such as K3s or custom solutions might be more suitable. ⁹ **Conclusion** This paper has presented a comprehensive survey on collaborative cloud-edge computing, covering architectures, orchestration frameworks, and distributed AI paradigms based on a systematic analysis of 78 Scopus-indexed publications from the 2021-2026 period. The main findings show that the cloud-edge ecosystem has evolved significantly from a conceptual stage to increasingly mature and standardized platforms. The three-tier hierarchical architecture with dynamic adaptive capabilities has become the dominant model; Kubernetes with edge extensions (KubeEdge, OpenYurt) has become the de facto standard for orchestration; and Federated Learning along with Split Computing/Knowledge Distillation techniques have established the foundation for distributed AI paradigms at the edge. The remaining challenges—interoperability standardization, FL integration with LLMs, system resilience, and comprehensive benchmarks—provide a clear roadmap for the research community for the next decade. The convergence with 5G/6G technologies, digital twins, and the long-term potential of quantum computing at the edge opens up exciting research horizons. We hope this survey can serve as a fundamental reference for researchers entering this field and for practitioners designing the next generation of cloud-edge systems.

X. CONCLUSIONS

This paper has presented a comprehensive survey on collaborative cloud-edge computing, covering architectures, orchestration frameworks, and distributed AI paradigms based on a systematic analysis of 78 Scopus-indexed publications from the 2021-2026 period. The main findings show that the cloud-edge ecosystem has evolved significantly from a conceptual stage to increasingly mature and standardized platforms. The three-tier hierarchical architecture with dynamic adaptive capabilities has become the dominant model; Kubernetes with edge extensions (KubeEdge, OpenYurt) has become the de facto standard for orchestration; and Federated Learning along with Split Computing/Knowledge Distillation techniques have established the foundation for distributed AI paradigms at the edge. The remaining challenges—interoperability standardization, FL integration with LLMs, system resilience, and comprehensive benchmarks—provide a clear roadmap for the research community for the next decade. The convergence with 5G/6G technologies, digital twins, and the long-term potential of quantum computing at the edge opens up exciting research horizons. We hope this survey can serve as a fundamental reference for researchers entering this field and for practitioners designing the next generation of cloud-edge systems.

XI. AUTHOR'S CONTRIBUTION

Conceptualization: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.
Methodology: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.
Investigation: I Made Suartana and Ricky Eka Putra.
Discussion of results: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.
Writing – Original Draft: I Made Suartana.
Writing – Review and Editing: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.
Resources: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.
Supervision: Ricky Eka Putra and Rahadian Bisma.
Approval of the final text: I Made Suartana, Ricky Eka Putra and Rahadian Bisma.

XII. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Institute for Research and Community Service (Lembaga Penelitian dan Pengabdian kepada Masyarakat - LPPM) of Universitas Negeri Surabaya (UNESA) for providing the financial support and funding for this research.

XIII. REFERENCES

- [1] Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6), 854–864. <https://doi.org/10.1109/JIOT.2016.2584538>
- [2] Statista. (2024). Internet of Things – number of connected devices worldwide 2015–2030. Statista Research Department.
- [3] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>
- [4] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656. <https://doi.org/10.1109/COMST.2017.2682318>
- [5] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656. <https://doi.org/10.1109/COMST.2017.2682318>
- [6] Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems*, 97, 219–235. <https://doi.org/10.1016/j.future.2019.02.050>
- [7] Shi, W., & Dustdar, S. (2016). The promise of edge computing. *Computer*, 49(5), 78–81. <https://doi.org/10.1109/MC.2016.145>

- [8] Xu, D., Li, T., Li, Y., Su, X., Tarkoma, S., Jiang, T., Kangasharju, J., & Hui, P. (2021). Edge intelligence: Empowering intelligence to the edge of network. *Proceedings of the IEEE*, 109(11), 1778–1837. <https://doi.org/10.1109/JPROC.2021.3119950>
- [9] Zhang, X., Wang, Y., Lu, S., Liu, L., Xu, L., & Shi, W. (2023). OpenEI: An open framework for edge intelligence. *2023 IEEE/ACM Symposium on Edge Computing*, 10–22. <https://doi.org/10.48550/arXiv.1906.01864>
- [10] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 54, 1273–1282. Available from <https://proceedings.mlr.press/v54/mcmahan17a.html>
- [11] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2), 1–210. <https://doi.org/10.1561/22000000083>
- [12] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 1657–1681. <https://doi.org/10.1109/COMST.2017.2705720>
- [13] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
- [14] Cisco. (2023). Cisco Global Cloud Index: Forecast and Methodology, 2022–2027. Cisco White Paper.
- [15] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457–7469. <https://doi.org/10.1109/JIOT.2020.2984887>
- [16] ETSI. (2022). ETSI GS MEC 003 V3.1.1: Multi-access Edge Computing (MEC); Framework and Reference Architecture. European Telecommunications Standards Institute.
- [17] Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research challenges. *Future Generation Computer Systems*, 79, 849–861. <https://doi.org/10.1016/j.future.2017.09.020>
- [18] Lin, B., Guo, W., Xiong, N., Chen, G., Vasilakos, A. V., & Zhang, H. (2022). A pretreatment workflow scheduling approach for big data applications in multicloud environments. *IEEE Transactions on Network and Service Management*, 19(2), 1354–1369.
- [19] Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2), 869–904. <https://doi.org/10.1109/COMST.2020.2970550>
- [20] Qualcomm Technologies. (2023). Qualcomm AI Research: On-device AI capabilities of Snapdragon. Qualcomm Technical Report.
- [21] Apple Inc. (2023). Apple Neural Engine Architecture Overview. Apple Developer Documentation.
- [22] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y. C., Yang, Q., Niyato, D., & Miao, C. (2021). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3), 2031–2063. <https://doi.org/10.1109/COMST.2020.2986024>
- [23] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2022). Federated learning with non-IID data. *arXiv preprint arXiv:1806.00582v2*. <https://doi.org/10.48550/arXiv.1806.00582>
- [24] ETSI. (2022). ETSI GS MEC 010-2 V2.2.1: MEC Management; Part 2: Application lifecycle, rules and requirements management. ETSI.
- [25] Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 20(4), 2961–2991. <https://doi.org/10.1109/COMST.2018.2849509>
- [26] Zhang, J., Chen, B., Zhao, Y., Cheng, X., & Hu, F. (2022). Data security and privacy-preserving in edge computing paradigm: Survey and open challenges. *IEEE Access*, 6, 18209–18237. <https://doi.org/10.1109/ACCESS.2018.2820162>
- [27] Xiong, Z., Zhang, Y., Niyato, D., Wang, P., & Han, Z. (2018). When mobile blockchain meets edge computing. *IEEE Communications Magazine*, 56(8), 33–39. <https://doi.org/10.1109/MCOM.2018.1701095>
- [28] Zhang, Q., Zhang, Q., Shi, W., & Zhong, H. (2018). Firework: Data processing and sharing for hybrid cloud-edge analytics. *IEEE Transactions on Parallel and Distributed Systems*, 29(9), 2004–2017. <https://doi.org/10.1109/TPDS.2018.2812177>
- [29] Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y., & Bennis, M. (2018). Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet of Things Journal*, 6(3), 4005–4018. <https://doi.org/10.1109/JIOT.2018.2876279>
- [30] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358. <https://doi.org/10.1109/COMST.2017.2745201>
- [31] Lin, J., Chen, W. M., Lin, Y., Cohn, J., Gan, C., & Han, S. (2020). MCUNet: Tiny deep learning on IoT devices. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 11711–11722. <https://doi.org/10.48550/arXiv.2007.10319>
- [32] Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: Architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*, 98, 27–42. <https://doi.org/10.1016/j.jnca.2017.09.002>
- [33] Aske, A., & Zhao, X. (2022). Supporting multi-provider serverless computing on the edge. In *Proceedings of the 47th International Conference on Parallel Processing Companion (ICPP Workshops)*. <https://doi.org/10.1145/3229710.3229742>
- [34] Teerapittayanon, S., McDanel, B., & Kung, H. T. (2017). Distributed deep neural networks over the cloud, the edge and end devices. *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 328–339. <https://doi.org/10.48550/arXiv.1709.01921>
- [35] Li, E., Zeng, L., Zhou, Z., & Chen, X. (2019). Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications*, 19(1), 447–457. <https://doi.org/10.48550/arXiv.1910.05316>
- [36] Chen, X. W., & Lin, X. (2014). Big data deep learning: Challenges and perspectives. *IEEE Access*, 2, 514–525. <https://doi.org/10.1109/ACCESS.2014.2325029>
- [37] Xu, J., Chen, L., & Zhou, P. (2022). Joint service caching and task offloading for mobile edge computing in dense networks. *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 207–215. <https://doi.org/10.1109/INFOCOM.2018.8485977>

- [38] Yao, X., Han, X., Du, X., & Guizani, M. (2013). A lightweight multicast authentication mechanism for small scale IoT applications. *IEEE Sensors Journal*, 13(10), 3693–3701. <https://doi.org/10.1109/JSEN.2013.2266116>
- [39] Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., & Satyanarayanan, M. (2014). Towards wearable cognitive assistance. *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 68–81. <https://doi.org/10.1145/2594368.2594383>
- [40] Ale, L., Zhang, N., Wu, H., Chen, D., & Fang, T. (2021). Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, 7(3), 881–892. <https://doi.org/10.48550/arXiv.2103.07811>
- [41] Barbera, M. V., Kosta, S., Mei, A., & Stefa, J. (2013). To offload or not to offload? The bandwidth and energy costs of mobile cloud computing. *2013 Proceedings IEEE INFOCOM*, 1285–1293. <https://doi.org/10.1109/INFCOM.2013.6566921>
- [42] Filali, A., Abouaomar, A., Cherkaoui, S., Kobbane, A., & Guizani, M. (2021). Multi-access edge computing: A survey. *IEEE Access*, 9, 46017–46045. <https://doi.org/10.1109/ACCESS.2021.3064405>
- [43] Pham, Q. V., Fang, F., Ha, V. N., Piran, M. J., Le, M., Le, L. B., Hwang, W. J., & Ding, Z. (2020). A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8, 116974–117017. <https://doi.org/10.1109/ACCESS.2020.3001277>
- [44] Sun, W., Liu, J., & Yue, Y. (2019). AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. *IEEE Network*, 33(5), 68–74. <https://doi.org/10.1109/MNET.001.1800510>
- [45] Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *ACM Queue*, 14(1), 70–93. <https://doi.org/10.1145/2898442.2898444>
- [46] Liu, Y., Yu, F. R., Li, X., Ji, H., & Leung, V. C. M. (2018). Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access. *IEEE Transactions on Vehicular Technology*, 67(12), 12137–12151. <https://doi.org/10.1109/TVT.2018.2872912>
- [47] Alibaba Cloud. (2022). OpenYurt: Extending Kubernetes to the Edge. CNCF Sandbox Project Documentation. <https://openyurt.io>
- [48] Masood, A., Lakew, D. S., & Cho, S. (2020). Security and privacy challenges in connected vehicular cloud computing. *IEEE Communications Surveys & Tutorials*, 22(4), 2725–2764. <https://doi.org/10.1109/COMST.2020.3012961>
- [49] Hassan, N., Gillani, S., Ahmed, E., Yaqoob, I., & Imran, M. (2018). The role of edge computing in internet of things. *IEEE Communications Magazine*, 56(8), 110–115. <https://doi.org/10.1109/MCOM.2018.1700906>
- [50] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing*, 2(2), 222–235. <https://doi.org/10.1109/TCC.2014.2314655>
- [51] Tuli, S., Ilager, S., Ramamohanarao, K., & Buyya, R. (2022). Dynamic scheduling for stochastic edge-cloud computing environments using A3C learning and residual recurrent neural networks. *IEEE Transactions on Mobile Computing*, 21(3), 940–954. <https://doi.org/10.1109/TMC.2020.3017079>
- [52] Huang, L., Bi, S., & Zhang, Y. J. (2020). Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. *IEEE Transactions on Mobile Computing*, 19(11), 2581–2593. <https://doi.org/10.1109/TMC.2019.2928811>
- [53] Zhang, W., Zhang, Z., & Chao, H. C. (2017). Cooperative fog computing for dealing with big data in the Internet of Vehicles: Framework and teams. *IEEE Communications Magazine*, 55(12), 60–67.
- [54] Park, J., Samarakoon, S., Bennis, M., & Debbah, M. (2023). Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11), 2204–2221. <https://doi.org/10.1109/MCOM.2017.1700208>
- [55] AL-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Singh, M. J., Paw, J. K. S., & Amir, M. (2023). Advancements in intelligent cloud computing for power optimization and battery management in hybrid renewable energy systems: A comprehensive review, *Energy Reports*, Volume 10, 2023, Pages 2206–2227, ISSN 2352-4847, <https://doi.org/10.1016/j.egyr.2023.09.029>.
- [56] Li, J., Gao, H., Lv, T., & Lu, Y. (2018). Deep reinforcement learning based computation offloading and resource allocation for MEC. *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. <https://doi.org/10.1109/WCNC.2018.8377343>
- [57] Chen, J., Pan, X., Monga, R., Bengio, S., & Jozefowicz, R. (2016). Revisiting distributed synchronous SGD. *arXiv preprint arXiv:1604.00981v4*. <https://doi.org/10.48550/arXiv.1604.00981>
- [58] Wang, J., Hu, J., Min, G., Zomaya, A. Y., & Georgalas, N. (2022). Fast adaptive task offloading in edge computing based on meta reinforcement learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(1), 242–253. <https://doi.org/10.1109/TPDS.2020.3014015>
- [59] McMahan, H. B., Moore, E., Ramage, D., & y Arcas, B. A. (2016). Federated learning of deep networks using model averaging. *arXiv:1602.05629v3*. <https://doi.org/10.48550/arXiv.1602.05629>
- [60] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Smola, A., & Smith, V. (2018). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems (MLSys)*, 2, 429–450. <https://doi.org/10.48550/arXiv.1812.06127>
- [61] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., & Suresh, A. T. (2021). SCAFFOLD: Stochastic controlled averaging for federated learning. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, PMLR 119, 5132–5143.
- [62] Blanchard, P., Mhamdi, E. M. E., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems (NeurIPS)*, 30. <https://doi.org/10.48550/arXiv.1703.02757>
- [63] Dinh, C. T., Tran, N. H., & Nguyen, T. D. (2022). Personalized federated learning with Moreau envelopes. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 21394–21405. <https://doi.org/10.48550/arXiv.2006.08848>
- [64] Matsubara, Y., Rodrigues, M., Latsague, A., Liu, S., Perez-Cerrolaza, L., Davila-Frias, A., ... & Capogrosso, L. (2023). Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Computing Surveys*, 55(5), 1–38. <https://doi.org/10.1145/3527155>

- [65] Shao, J., & Zhang, J. (2019). BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems. 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 1–6. <https://doi.org/10.48550/arXiv.1910.14315>
- [66] Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Computer Architecture News, 45(1), 615–629. <https://doi.org/10.1145/3093337.3037698>
- [67] Matsubara, Y., Levorato, M., & Restuccia, F. (2022). Neural compression and filtering for edge-assisted real-time object detection in challenged networks. 2022 25th International Conference on Pattern Recognition (ICPR), 2272–2279.
- [68] Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016). BranchyNet: Fast inference via early exiting from deep neural networks. 2022 23rd International Conference on Pattern Recognition (ICPR), 2464–2469. <https://doi.org/10.1109/ICPR.2016.7900006>
- [69] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. <https://doi.org/10.48550/arXiv.1503.02531>
- [70] Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W., & Keutzer, K. (2019). Q-BERT: Hessian based ultra low precision quantization of BERT. Proceedings of the AAAI Conference on Artificial Intelligence, 34(5), 8815–8821. <https://doi.org/10.48550/arXiv.1909.05840>
- [71] Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., & Blankevoort, T. (2020). Up or down? Adaptive rounding for post-training quantization. Proceedings of the 37th International Conference on Machine Learning (ICML), 7197–7206. <https://doi.org/10.48550/arXiv.2004.10568>
- [72] Han, S., Pool, J., Tran, J., & Dally, W. J. (2016). Learning both weights and connections for efficient neural networks. Advances in Neural Information Processing Systems (NeurIPS), 28. <https://doi.org/10.48550/arXiv.1506.02626>
- [73] Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., & Han, S. (2024). AWQ: Activation-aware weight quantization for LLM compression and acceleration. Proceedings of Machine Learning and Systems (MLSys), 6. <https://doi.org/10.48550/arXiv.2306.00978>
- [74] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. Proceedings of the 10th International Conference on Learning Representations (ICLR). <https://doi.org/10.48550/arXiv.2106.09685>
- [75] Leviathan, Y., Kalman, M., & Matias, Y. (2022). Fast inference from transformers via speculative decoding. Proceedings of the 40th International Conference on Machine Learning (ICML), PMLR 202, 19274–19286. <https://doi.org/10.48550/arXiv.2211.17192>
- [76] Tan, T., Bai, Z., Zhu, Y., Liao, X., & Zhu, H. (2024). EdgeShard: Efficient LLM inference via collaborative edge computing. arXiv preprint arXiv:2405.14371. <https://doi.org/10.48550/arXiv.2405.14371>
- [77] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., & Lv, W. (2021). Edge computing security: State of the art and challenges. Proceedings of the IEEE, 107(8), 1608–1631. <https://doi.org/10.1109/JPROC.2019.2918437>
- [78] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST SP 800-207). National Institute of Standards and Technology. <https://doi.org/10.6028/nist.sp.800-207>
- [79] Nilsson, A., Bideh, P. N., & Brorsson, M. (2020). A survey of published attacks on Intel SGX. arXiv preprint arXiv:2006.13598.
- [80] Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge computing for the internet of things: A case study. IEEE Internet of Things Journal, 5(2), 1275–1284. <https://doi.org/10.1109/JIOT.2018.2805263>
- [81] He, Q., Xu, X., Liu, F., He, J., Ye, D., Zhang, X., ... & Yang, Y. (2022). A game-theoretic approach for co-management of cloud-edge resources in MEC. IEEE Transactions on Parallel and Distributed Systems, 32(4), 1012–1025. <https://doi.org/10.1155/2022/9323818>
- [82] Praveenchandar, J., & Tamilarasi, A. (2021). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 12(4), 4147–4159. <https://doi.org/10.1007/s12652-020-01794-6>
- [83] Taleb, T., Dutta, S., Ksentini, A., Iqbal, M., & Flinck, H. (2017). Mobile edge computing potential in making cities smarter. IEEE Communications Magazine, 55(3), 38–43. <https://doi.org/10.1109/MCOM.2017.1600249CM>
- [84] Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). DeepMove: Predicting human mobility with attentional recurrent networks. Proceedings of the 2018 World Wide Web Conference (WWW), 1459–1468. <https://doi.org/10.1145/3178876.3186058>
- [85] Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., & Flinck, H. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. IEEE Communications Surveys & Tutorials, 20(3), 2429–2453. <https://doi.org/10.1109/COMST.2018.2815638>
- [86] 3GPP. (2022). 3GPP TS 23.558: Architecture for enabling Edge Applications (Release 17). 3rd Generation Partnership Project.
- [87] Letaief, K. B., Chen, W., Shi, Y., Zhang, J., & Zhang, Y. J. A. (2019). The roadmap to 6G: AI empowered wireless networks. IEEE Communications Magazine, 57(8), 84–90. <https://doi.org/10.48550/arXiv.1904.11686>
- [88] Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.
- [89] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195–202. <https://doi.org/10.1038/nature23474>
- [90] Fuller, A., Fan, Z., Day, C., & Barlow, C. (2021). Digital twin: Enabling technologies, challenges and open research. IEEE Access, 8, 108952–108971. <https://doi.org/10.1109/ACCESS.2020.2998358>
- [91] Ge, J., Liang, Y. C., Zhang, L., & Feng, G. (2024). Deep reinforcement learning for distributed dynamic coordinated beamforming in massive MIMO cellular networks. IEEE Transactions on Wireless Communications, 23(5), 4155–4169. <https://doi.org/10.48550/arXiv.2303.14082>