International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 6, Nov-Dec-2026, ISSN (Online): 2395-566X

# An Overview on Edge Computing Research: A Comprehensive Analysis of Architecture, Task Offloading Paradigms, Security Models, and 5G/IoE Integration

**Gumatkar Shubham Baban[1], Prof. Sakshi S.S. [2] Prof. Nalawade T.S. [3] Prof. Maniyar A.A[4]**
Department of Artificial Intelligence & Machine Learning PPU Academic Year 2026–2027

**Abstract— The exponential expansion of the Internet of Everything (IoE) and high-throughput sensory streams has severely strained traditional centralized cloud computing paradigms. High propagation latencies, backhaul bandwidth saturation, and stringent data resi-dency regulations have necessitated a fundamental shift toward edge computing—a paradigm that decentralizes execution by placing computation, storage, and network-ing resources in physical proximity to end-terminal devices. This paper presents an extensive, multi-dimensional overview of edge computing research. We formalize the structural taxonomy comparing Multi-Access Edge Computing (MEC), Fog Computing, and Cloudlets. We systematically review 3-tier and microservice-driven ecosystems such as EdgeX Foundry and ECC 3.0. Furthermore, we develop rigorous mathemat-ical models for single-user and multi-user computational task offloading under joint latency-energy minimization constraints, non-convex Lyapunov optimization, and game-theoretic offloading frameworks. In addition, privacy-preserving vectors—including Attribute-Based Encryption (ABE), Fully Homomorphic Encryption (FHE), and dynamic access control—are thoroughly evaluated. Finally, we provide empirical performance benchmarks across 5G/6G radio access networks, computer vision telemetry, MEC video caching, and predictive maintenance in industrial automation.**

**Keywords: Edge Computing, Multi-Access Edge Computing (MEC), Task Offloading, Fog Computing, Attribute-Based Encryption, Lyapunov Optimization, 5G/6G Networks, Internet of Everything (IoE).**

## I. INTRODUCTION

Over the past decade, the proliferation of smart handheld devices, autonomous vehicles, industrial Internet of Things (IIoT) sensors, and interactive augmented reality (AR) platforms has transformed global data generation metrics. Annual data volume generated near the network boundary is projected to exceed hundreds of zettabytes.

Traditionally, application vendors relied heavily on public cloud infrastructure (e.g., AWS, Azure, Google Cloud Platform) to ingest, process, and store sensory data. Cloud computing provides virtually infinite elastic compute power and long-term analytical storage. However, offloading all raw device data directly to remote cloud data centers introduces critical systemic bottlenecks:

- Latency Instability and Round-Trip Delay: Real-time mission-critical applications—such as autonomous driving collision avoidance, remote telesurgery, and robotic control loops—require sub-10ms or even sub-1ms end-to-end latencies. Wide-Area Network (WAN) routing, queuing delays, and geographic distance to cloud data centers inherently prevent remote cloud setups from satisfying these real-time constraints.
- Backhaul Bandwidth Saturation: Transmitting continuous raw high-definition video streams from thousands of urban surveillance nodes or high-frequency telemetry from industrial turbine arrays overwhelms cellular core networks and backhaul link bandwidths.
- Privacy Vector Vulnerabilities and Data Sovereignty: Aggregating sensitive per-sonal data—such as medical telemetry, biometric authentication logs, or private home video feeds—in centralized cloud repositories dramatically increases the attack sur-face and risks violating strict compliance standards (e.g., GDPR, HIPAA).
- Edge computing addresses these foundational limitations by migrating computational tasks, data management functions, and intelligence algorithms from centralized cloud cores down to edge nodes positioned along the network edge (e.g., 5G gNodeB base sta-tions, local access points, industrial gateways, and regional micro-data centers).

## II. RESEARCH CONTRIBUTIONS

This comprehensive survey delivers the following contributions:

© 2026 IJSRET

- Architectural Taxonomy: A unified comparison and theoretical categorization of Cloud vs. Edge paradigms, detailing MEC, Fog, and Cloudlet continuum definitions.
- Framework Analysis: In-depth dissection of 3-Tier Edge architectures, ECC 3.0 ref-erence models, and open-source industrial platforms (EdgeX Foundry).
- Mathematical Offloading Formulations: Formal mathematical models for task par-titioning, static vs. dynamic offloading, non-convex optimization, and Lyapunov-based energy-delay trade-offs.
- Security and Privacy Taxonomies: Detailed evaluation of modern cryptographic trust models, including CP-ABE, FHE, and cross-domain authentication schemes in edge environments.
- Real-World Industrial Deployments: Comprehensive case studies on 5G MEC video caching, computer-vision railway safety monitoring, and predictive maintenance teleme-try.

## III. TAXONOMY AND PARADIGM COMPARISON

To understand the edge research landscape, it is essential to distinguish edge computing from complementary paradigms such as Cloud Computing, Fog Computing, and Cloudlets.

1. Cloud vs. Edge Computing Comparative Dynamics

While cloud computing focuses on global data aggregation, complex macro-model train-ing, and long-term historical storage, edge computing focuses on localized execution, im-mediate event-driven actuation, and bandwidth reduction.

Table 1: Systematic Comparison: Centralized Cloud Computing vs. Edge Computing

| Dimension | Centralized Cloud Comput-ing | Edge Computing |
|---|---|---|
| Node Location | Centralized hyper-scale data centers | Geographically distributed near terminals |
| Network Latency | High (50–250 ms) | |
| Hardware Elasticity | Virtually unlimited | Constrained |
| capacity | | CPU/GPU and bat- |
| Backhaul Impact | Massive core throughput re-quired | Minimal (local traffic filter-ing/offloading) |
| Context Awareness | Global scope, decoupled from location | Local spatial, temporal, and user context |
| Jitter & Reliability | Variable (subject to WAN con-gestion) | Highly predictable, resilient to offline modes |
| Data Privacy | Centralized | Local data isolation within trusted perimeter |
| vulnerability vec- | tor | |

Edge Paradigm Continuum: MEC, Fog, and Cloudlets

Edge research frequently intersects with several distinct architectural formulations:

**1. Multi-Access Edge Computing (MEC)**

Standardized by the European Telecommunications Standards Institute (ETSI), MEC collo-cates cloud IT capabilities directly within cellular Radio Access Networks (RAN). Operating at 5G base station sites (e.NodeB / gNodeB), MEC enables mobile operators to open their network edge to third-party software services, allowing real-time traffic redirection and ultra-low-latency processing.

**2. Fog Computing**

Originally conceptualized by Cisco and formalized by the OpenFog Consortium (now IEEE 1934 standard), Fog computing emphasizes a hierarchical, distributed computing layer spanning from physical sensors up to cloud servers. Fog nodes include network switches, routers, multi-service gateways, and programmable logic controllers (PLCs), making it ex-ceptionally well-suited for industrial automation and smart grid operations.

**3. Cloudlets**

Pioneered by Carnegie Mellon University, a Cloudlet is a ”data center in a box”—a small-scale, mobility-aware server cluster deployed one network hop away from mobile hard-ware over high-speed local wireless networks (Wi-Fi or enterprise LTE). Cloudlets primary serve user offloading applications, such as real-time language translation and wearable computer vision.

## IV. SYSTEM ARCHITECTURES AND REFERENCE ECOSYSTEMS

Modern edge deployments depend on standardized structural models to maintain inter-operability across multi-vendor hardware environments.

© 2026 IJSRET

**1. Three-Tier Federated Edge Architecture**

The foundational structural model of edge deployments is divided into three intercon-nected layers:

- Terminal Layer (Data Generation): Encompasses physical IoT devices, wearable sensors, IP cameras, connected vehicles, and industrial actuators. Devices in this layer generate heterogeneous data protocols (e.g., Modbus, BLE, MQTT, OPC UA) and often possess constrained compute power.
- Edge (Boundary) Layer (Real-Time Execution): Comprises base stations, edge gate-ways, local micro-servers, and smart routers. This layer executes localized data fil-tering, real-time inferencing, dynamic task scheduling, and state synchronization.
- Cloud Core Layer (Global Orchestration): Comprises public or private enterprise cloud data centers. The cloud layer receives pre-processed summaries from the edge, updates global AI models, and archives regulatory compliance records.

**2. Open-Source Industrial Reference Architecture: EdgeX Foundry**

In industrial and enterprise deployments, vendor neutrality is achieved using open-source microservice frameworks such as EdgeX Foundry. EdgeX provides a dual-transformation engine operating across four core service layers:

- Device Layer (Southbound Interface): Utilizes microservice Device Services to in-terface directly with physical sensors via native protocols (e.g., MQTT, Modbus, BAC-net, REST). It normalizes heterogeneous raw physical signals into standard JSON/CBOR event payloads.
- Core Services Layer: Consists of Core Data (event storage), Core Metadata (device registers), and Command (device control execution) modules. This layer acts as the local operational hub.
- Supporting Services Layer: Hosts local rules engines, alert mechanisms, scheduling modules, and local analytics microservices.
- Export Services Layer (Northbound Interface): Handles payload compression, lo-cal encryption, and distribution of refined metrics to cloud endpoints or enterprise ERP systems.

## V. COMPUTATION OFFLOADING FORMULATIONS

A fundamental problem in edge research is Computation Ofloading—determining whether a terminal device should process a workload locally or transfer it to an edge server over wireless link channels.

**1. Single-User Task Offloading Model**

Consider a terminal device with a computationally intensive task Ti defined by a tuple:

Tlocal = Ci
flocal

The energy consumption of local CPU execution is modeled as:

(2)

$$E_{local} = \kappa \cdot (f_{local})^2 \cdot C_i \quad (3)$$

where κ is the effective switched capacitance of the device CPU chip architecture.

**1. Edge Offloading Model**

Di Ci
Ttrans = , Tedge =

(5)

Rup fedge

where fedge is the CPU processing frequency allocated by the edge server node. Neglect-ing the small downlink result transmission time, total offloading delay Toff is:

Di Ci
Toff = Ttrans + Tedge = +

(6)

Rup fedge

The energy expended by the terminal device during offloading is strictly transmission power:

Eoff = Ptx · Ttrans = Ptx · Di (7)
Rup

**2. Joint Latency-Energy Optimization Formulation**

We define binary decision variable $\alpha \in \{0, 1\}$, where $\alpha = 1$ indicates local processing and $\alpha = 0$ indicates complete offloading. The system objective minimizes a weighted linear combination of time delay and energy consumption subject to latency deadlines:

min
α,flocal,Ptx

$$J = \alpha\,(w_t T_{local} + w_e E_{local}) + (1 - \alpha)\,(w_t T_{off} + w_e E_{off})$$

s.t. $C1 : \alpha T_{local} + (1 - \alpha) T_{off} \le \tau_{max}$ $C2 : 0 \le P_{tx} \le P_{max}$
$C3 : 0 \le f_{local} \le f_{max}$ $C4 : \alpha \in \{0, 1\}$ (8)

© 2026 IJSRET

where wt, we ∈ [0, 1] (wt + we = 1) represent application-specific weighting coefficients prioritizing latency vs. battery preservation.

**3. Multi-User Dynamic Task Scheduling and Game Theory**

In multi-user MEC scenarios where N terminal devices compete for finite computational resources across M edge base stations, individual offloading decisions create inter-channel interference and node contention.

This competition can be modeled as a Non-Cooperative Offloading Game:

G = (N , {Si}i∈N , {Ui}i∈N ) (9)

where N = {1, 2, . . . , N } is the set of devices, Si = {0, 1, . . . , M } is the strategy space of device i, and Ui(·) is the utility function representing cost reduction. A Nash Equilibrium strategy profile s∗ = (s∗, s∗, . . . , s∗ ) is achieved when no single device can unilaterally im-

1 2 N

prove its utility by changing its offloading decision:

Ui(s∗, s∗ ) ≥ Ui(si, s∗ ), ∀si ∈ Si, ∀i ∈ N (10)

i −i −i

**4. Lyapunov Stochastic Optimization for Energy-Constrained Edge Nodes**

In energy-harvesting edge systems where node arrival rates λ(t) and available power fluc-tuate stochastically, Lyapunov optimization enables dynamic task scheduling without re-quiring long-term distribution priors.

Let Q(t) denote the task queue back-log at time slot t. The virtual queue evolution equation is defined as:

Q(t + 1) = max [Q(t) − b(t), 0] + a(t) (11)

where a(t) is task arrival load and b(t) is processed task output.

The Lyapunov function L(Q(t)) measuring queue stability is defined as:

L(Q(t)) =

1 Q2(t) (12)

2

The conditional 1-step Lyapunov drift Δ(Q(t)) is given by:

Δ(Q(t)) = E [L(Q(t + 1)) − L(Q(t)) | Q(t)] (13)

To balance queue stability against energy cost minimization E(t), we construct the Drift-Plus-Penalty expression:

Δ(Q(t)) + V · E [E(t) | Q(t)] ≤ Bbound + V · E [E(t) | Q(t)] + Q(t)E [a(t) − b(t) | Q(t)] (14)

where V ≥ 0 is a tuning parameter controlling the trade-off between battery energy saving and queue backlog delay.

**2. Security, Privacy, and Trust Frameworks**

Decentralized edge architectures introduce multi-tenant computing across untrusted phys-ical locations, presenting severe security challenges.

**1. Cryptographic Data Confidentiality Vectors**

To process data securely at third-party edge nodes without exposing raw unencrypted payloads, advanced cryptographic methods are implemented:

**Ciphertext-Policy Attribute-Based Encryption (CP-ABE)**

In CP-ABE schemes, user data is encrypted under an access policy defined over logical at-tributes (e.g., (Role = Engineer AND Shift = Night) OR Manager). Edge nodes decrypt data payloads only if their credential attributes satisfy the access tree structure. Key dis-tribution protocols must maintain cryptographic secrecy during rapid handover events across base station boundaries.

**2. Fully Homomorphic Encryption (FHE)**

FHE allows edge nodes to perform mathematical computations directly on encrypted data without decrypting it first:

Dec (k, f (Enc(k, m1), Enc(k, m2))) = f (m1, m2) (15)

While computationally heavy, lightweight partial homomorphic schemes (PHE) are increas-ingly deployed at edge nodes to aggregate sensitive telemetry metrics (e.g., smart meter sums) securely.

**3. Identity Authentication and Trust Handovers**

When mobile units (e.g., connected vehicles) move between different edge cells, authenti-cating credentials via remote cloud servers introduces prohibitive authentication latency. Cross-domain Authentication, Group Key Management, and Public Key Infrastructure (PKI) models integrated with distributed ledger technology (Blockchain) establish localized trust verification within sub-5ms constraints.

**3. Industrial Applications and Performance Benchmarks**

To validate edge paradigms empirically, research deployments have been implemented across several high-impact enterprise sectors.

**1. 5G Multi-Access Edge Video Caching**

Content Distribution Networks (CDNs) deployed at cellular gNodeB sites cache popular media segments locally. By analyzing local user consumption patterns, predictive AI al-gorithms store high-frequency video chunks at the local edge, eliminating backhaul link congestion and buffering latency.

© 2026 IJSRET

Table 2: Taxonomy of Edge Security Vulnerabilities and Countermeasures

| Attack Surface Vector Description | Threat Primary | |
|---|---|---|
| **Rogue Node Insertion** edge node im- | Malicious personation sniffing local tasks | PKI Trust anchors and SGX Enclave Attestation |
| **Man-in-the-Middle (MitM)** during mobile | Interception RAN handover | Cross-domain AGKMS pro-tocol authentication Merkle tree cryptographic integrity audits Hardware-enforced Execu-tion Enclaves (TEE) K-Anonymity and Differen-tial Noise Injection |
| **Data Tampering** | Alteration of sensor state logs on local storage | |
| **Side-Channel Analysis** monitoring to ex- | Cache tract key materials | |
| **Location Tracking** location inference via | User offloading rates | |

**2. Computer-Vision Railway Safety Inspection**

Edge computer-vision nodes equipped with specialized neural network accelerators (e.g., TensorRT, Coral TPU) process continuous high-rate video feeds from track-side cameras. By running localized inference pipelines directly on the edge node, foreign object debris (FOD) on railway lines is detected in less than 8 ms, triggering automated alert protocols instantly.

**3. Predictive Maintenance in IIOT Smart Factories**

Industrial manufacturing plants deploy vibration, acoustic, and thermal sensors across robotic assembly arms. Edge gateways perform fast Fourier transform (FFT) signal anal-ysis locally to isolate anomalous harmonic frequencies. Operating locally avoids the con-tinuous streaming of raw, gigabyte-scale sensor logs to remote cloud servers, reserving WAN bandwidth strictly for verified anomaly alerts.

**4. Challenges and Open Research Directions**

Despite rapid theoretical and practical advances, several key research challenges remain open in the edge computing domain:

- Hardware and Software Heterogeneity: Edge nodes encompass highly disparate compute architectures (ARM64, x86-64, RISC-V, specialized ASICs). Creating unified containerization and micro-orchestration frameworks (e.g., lightweight K3s, WebAssem-bly runtimes) across non-uniform platforms remains an active research area.
- Serverless Edge Function Execution (FaaS): Adapting event-driven Serverless ar-chitectures to constrained edge nodes requires solving cold-start latency delays and maintaining light memory footprints.
- Distributed Edge Federated Learning: Training complex machine learning mod-els directly across millions of edge devices introduces significant challenges in non-IID (non-independently and identically distributed) data handling, gradient compres-sion, and communication-efficient consensus mechanisms.
- Dynamic Resource Allocation Under Mobility: High-speed user mobility (e.g., ve-hicular networks) requires seamless, ultra-fast task state migration algorithms to prevent service disruption during cell handovers.

## VI. CONCLUSION

Edge computing represents a critical structural shift in distributed system architecture, bridging the gap between resource-constrained physical devices and hyper-scale cloud data centers. By decentralizing computation, storage, and analytics to the network bound-ary, edge systems achieve ultra-low response latencies, preserve network backhaul band-width, and enhance data privacy.

This paper has presented a comprehensive analytical survey of the edge computing landscape. We formalized architectural taxonomy across MEC, Fog, and Cloudlet paradigms, detailed open microservice reference models such as EdgeX Foundry, derived mathemat-ical task-offloading and Lyapunov optimization models, analyzed modern cryptographic security structures, and evaluated empirical performance metrics across real-world 5G and IIoT applications. Continued research in serverless edge frameworks, federated edge intelligence, and standardized inter-node trust models will further accelerate the deploy-ment of intelligent, real-time cyber-physical systems.

© 2026 IJSRET

## REFERENCES

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016.
2. W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," Future Generation Computer Systems, vol. 97, pp. 219–235, Aug. 2019.
3. Y. Mao, C. Zhang, and K. B. Letaief, "A survey on mobile edge computing: The com-munication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.
4. M. Satyanarayanan, "The emergence of edge computing," IEEE Computer, vol. 50, no. 1, pp. 30–39, Jan. 2017.
5. N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450–465, Feb. 2018.
6. P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and com-putation offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, Thirdquarter 2017.
7. IEEE Architecture Working Group, "IEEE Standard for Architecture for Fog Computing in Enabled Internet of Things (IoT)," IEEE Std 1934-2018, pp. 1–142, Jul. 2018.
8. ETSI ISG MEC, "Mobile Edge Computing (MEC); Technical Requirements," ETSI GS MEC 002, V1.1.1, Mar. 2016.
9. R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
10. C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
11. L. Yang, J. Cao, G. Liang, and X. Han, "Cost-aware task offloading and resource alloca-tion in mobile edge computing," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 9, pp. 2158–2171, Sep. 2019.
12. Linux Foundation Edge, "EdgeX Foundry Architecture Framework Specification," LF Edge Technical Documentation, 2022.
13. R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," Future Generation Computer Systems, vol. 78, pp. 680–698, Jan. 2018.
14. K. Gai, M. Qiu, and H. Zhao, "Energy-aware task offloading using proactive caching in mobile edge computing," IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1143–1154, Nov. 2016.
15. X. Xu, X. Zhang, H. Khan, and L. Qi, "A survey on edge intelligence: Focus on AI model execution on edge nodes," IEEE Access, vol. 8, pp. 182400–182418, Oct. 2020.

© 2026 IJSRET