

Article

Real-Time Service Migration in Edge Networks: A Survey

Yutong Zhang ^{1,2}, Ke Zhao ^{1,2}, Yihong Yang ^{1,2}  and Zhangbing Zhou ^{1,2,*}

¹ School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China; yutongzhang@email.cugb.edu.cn (Y.Z.); kezhaoyang@email.cugb.edu.cn (K.Z.); yihongyang@email.cugb.edu.cn (Y.Y.)

² Technology Innovation Center of Geoscience Knowledge and Intelligent Service, China University of Geosciences (Beijing), Beijing 100083, China

* Correspondence: zbzhou@cugb.edu.cn

Abstract

With the rapid proliferation of Internet of Things (IoT) devices and mobile applications and the growing demand for low-latency services, edge computing has emerged as a transformative paradigm that brings computation and storage closer to end users. However, the dynamic nature and limited resources of edge networks bring challenges such as load imbalance and high latency while satisfying user requests. Service migration, the dynamic redeployment of service instances across distributed edge nodes, has become a key enabler for solving these challenges and optimizing edge network characteristics. Moreover, the low-latency nature of edge computing requires that service migration strategies must be in real time in order to ensure latency requirements. Thus, this paper presents a systematic survey of real-time service migration in edge networks. Specifically, we first introduce four network architectures and four basic models for real-time service migration. We then summarize four research motivations for real-time service migration and the real-time guarantee introduced during the implementation of migration strategies. To support these motivations, we present key techniques for solving the task of real-time service migration and how these algorithms and models facilitate the real-time performance of migration. We also explore latency-sensitive application scenarios, such as smart cities, smart homes, and smart manufacturing, where real-time service migration plays a critical role in sustaining performance and adaptability under dynamic conditions. Finally, we summarize the key challenges and outline promising future research directions for real-time service migration. This survey aims to provide a structured and in-depth theoretical foundation to guide future research on real-time service migration in edge networks.

Keywords: real-time service migration; edge networks; migration motivation; key techniques; vision



Academic Editor: Annan M. Abu-Mahfouz

Received: 20 June 2025

Revised: 28 July 2025

Accepted: 29 July 2025

Published: 1 August 2025

Citation: Zhang, Y.; Zhao, K.;

Yang, Y.; Zhou, Z. Real-Time Service

Migration in Edge Networks:

A Survey. *J. Sens. Actuator Netw.* **2025**,

14, 79. [https://doi.org/10.3390/](https://doi.org/10.3390/jsan14040079)

[jsan14040079](https://doi.org/10.3390/jsan14040079)

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

([https://creativecommons.org/](https://creativecommons.org/licenses/by/4.0/)

[licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

1.1. From Cloud to Edge: The Evolutionary Foundation for Service Migration

The explosive growth of smart applications, such as autonomous driving, Augmented Reality (AR), industrial automation, and real-time video analytics, has introduced increasingly stringent requirements on service latency, reliability, energy efficiency, and adaptability. These applications typically operate in dynamic environments, interact with mobile users or devices, and often generate massive volumes of time-sensitive data. Initially, cloud computing emerged as a foundational infrastructure paradigm by providing centralized, elastic, and scalable computing and storage resources, which can effectively

address the limitations of resource-constrained end devices [1]. However, the centralized nature of cloud computing introduces significant latency due to long transmission paths, causes bandwidth bottlenecks under massive data streams, and raises privacy concerns due to remote data processing, all of which undermine its suitability for real-time and location-sensitive applications [2].

To overcome these limitations, edge computing has emerged as a decentralized paradigm that relocates computational and storage capabilities to the proximity of data sources and end users, typically at edge servers, access points, or base stations [3]. This architectural evolution significantly reduces end-to-end latency, alleviates backbone network congestion, and improves data privacy and availability by enabling localized processing. With the maturation of *Fifth-Generation* (5G) technology and the anticipated rollout of *Sixth-Generation* (6G) technologies, the proliferation of Mobile Edge Computing (MEC) platforms further strengthens the role of edge infrastructures in supporting ultra-low-latency, high-bandwidth, and mission-critical services [4,5]. Nevertheless, this architectural decentralization introduces new operational complexities. Edge environments are inherently heterogeneous, with edge nodes varying in computing capacity, energy profiles, connectivity, and geographic distribution, and they are highly dynamic due to fluctuating workloads, user mobility, and environmental uncertainty. To ensure consistent Quality of Service (QoS), edge service management should be both responsive and adaptive. Unlike cloud environments where services can remain relatively static, edge systems should support dynamic service migration: the timely and efficient relocation of running service instances across edge nodes in response to changes in workload distribution, resource availability, or user location [6,7]. Service migration, in this context, refers to the process of transferring service states, runtime contexts, and application logic from one edge node to another without significant service interruption. It plays a foundational role in maintaining system performance and reliability by enabling load balancing, congestion avoidance, failure recovery, and seamless service continuity for mobile users [8,9]. Typical triggers for service migration include resource over-utilization (e.g., CPU or memory saturation), user hand-offs between access points, energy-aware scheduling on battery-constrained devices, and anticipated latency violations due to changing network conditions [10]. Beyond continuity and responsiveness, service migration contributes to multiple system-wide optimization objectives. These include reducing service response times, improving resource utilization balance across nodes, prolonging device lifetime through thermal and energy-aware load redistribution, and supporting green computing goals by reducing redundant computation. Furthermore, by integrating with intelligent orchestration frameworks, such as those based on reinforcement learning (RL), metaheuristics, or federated control, service migration can be proactively triggered based on predictive models of future system states, enabling anticipatory adaptation to dynamic environments [11].

In summary, the transition from cloud to edge computing represents not only an architectural shift but also a paradigm change in how services are provisioned, managed, and optimized. In this new paradigm, real-time service migration emerges as a central mechanism for achieving agility, scalability, and resilience in edge networks, especially under latency-sensitive conditions. This paper presents a comprehensive survey of real-time service migration in edge environments, analyzing the driving factors, key challenges, enabling technologies, and representative solutions across diverse application domains.

1.2. Service Migration in Edge Networks

In recent years, service migration has emerged as a critical research focus in the context of edge networks, attracting increasing attention from both academia and industry. Unlike traditional cloud computing environments, where services are deployed and executed in

centralized, static data centers, edge environments are distributed, mobile, and resource-constrained, thus demanding more agile and real-time service adaptation. In particular, the need to maintain low latency and uninterrupted service continuity under dynamic conditions underscores the importance of real-time service migration. Before delving into the mechanisms and challenges, two fundamental questions must be addressed.

1.2.1. What Is Service Migration in Edge Networks?

Service migration in edge networks refers to the process of dynamically relocating service instances, along with their computational states and execution contexts, across heterogeneous edge nodes in response to changing network conditions, user mobility, resource availability, and application demands. Unlike traditional service configuration, which focuses on initial deployment and static orchestration, service migration emphasizes runtime adaptability and continuity, ensuring that services remain responsive, efficient, and reliable under dynamic edge environments. In this paper, “real-time” service migration is defined as the process of relocating service instances within a time frame that satisfies the latency requirements of the specific application or end users. This notion of “real-time” is application-specific and depends on the latency tolerance determined by the service context. For instance, latency-sensitive scenarios such as autonomous driving or real-time video streaming require timely migration to avoid perceptible service degradation or operational instability. The goal is to perform migration quickly enough to maintain continuous, responsive service delivery in edge environments. To fully understand service migration, it is essential to clarify several related concepts and components that collectively define its scope and mechanisms:

- **Service Resources:** These are the computing, storage, and networking assets distributed across the edge network that support service execution and migration. Unlike static allocation, service migration enables the temporal reallocation of these resources to optimize utilization and responsiveness. Typical resources include CPU/GPU cycles, memory, bandwidth, and cache, which are often subject to spatial and temporal constraints [12,13].
- **Service Instances:** These refer to the actual running units of services (e.g., containers, virtual machines (VMs), microservices) deployed on edge devices or servers. During migration, these instances are transferred—either via live migration, checkpoint restart, or state rehydration—to new nodes without disrupting service continuity [14,15].
- **Participants:** Multiple entities collaborate in service migration, including mobile end devices (e.g., smartphones, sensors, vehicles), edge servers (e.g., base stations, fog nodes), and cloud platforms. Coordination models may follow device–edge, edge–edge, or edge–cloud topologies. Effective migration depends on synchronization and negotiation among these participants [16,17].
- **Objectives:** Service migration aims to address several optimization goals, including minimizing response latency, avoiding overloaded nodes, maintaining service availability under user mobility, reducing energy consumption, and enhancing overall QoS. In latency-critical or mission-critical applications such as autonomous driving or remote healthcare, timely migration is key to meeting service-level agreements (SLAs) [18,19].
- **Actions:** The core operations involved in service migration include monitoring node loads and user positions, evaluating migration triggers (e.g., SLA violations, mobility prediction, energy thresholds), transferring the service state, and reinitializing execution at the target node. This also involves handling dependencies, preserving data integrity, and updating routing or session states [8,20].

- **Methodologies:** Various methodologies have been developed to enable intelligent and efficient service migration. These include heuristic and metaheuristic optimization, deep reinforcement learning for adaptive decision-making, and container-based orchestration technologies such as Kubernetes or KubeEdge. Moreover, distributed cooperative mechanisms, often supported by blockchain or federated learning, are used to ensure scalability and decentralization in multi-domain edge environments [19,21].

1.2.2. Why Is Service Migration Necessary in Edge Computing Environments?

As edge computing evolves to support vast numbers of heterogeneous and mobile devices, the demand for real-time, adaptive, and context-aware service provisioning has grown rapidly. Static service deployment strategies are often inadequate for addressing the dynamic nature of edge workloads and user mobility. Against this backdrop, real-time service migration emerges as a key enabler for maintaining continuous, efficient, and high-quality service delivery across distributed edge environments. Its importance can be viewed from the following three perspectives:

- **Users:** Edge networks connect billions of geographically distributed devices—including stationary sensors, mobile phones, autonomous vehicles, and drones—each with diverse and evolving QoS needs. For example, autonomous driving scenarios require ultra-low-latency access to decision-making services, while battery-powered *IoT* devices prioritize energy-efficient offloading and minimal data transmission. As users move across network boundaries, static placements can lead to service delays or disruptions. Real-time service migration supports location-aware, demand-responsive relocation of services, ensuring seamless and low-latency experiences even under dynamic mobility patterns [22,23].
- **Service Providers:** The edge ecosystem includes various commercial stakeholders such as network operators, infrastructure providers, and third-party application vendors. These providers should balance performance, cost, and resource constraints while delivering high service quality. Static placement of services often leads to unbalanced loads, underutilized resources, or *SLA* violations in hotspots. Real-time service migration enables dynamic reassignment of services based on current traffic, resource availability, and user distribution. This enhances efficiency, reduces energy costs, and aligns with revenue-driven strategies such as demand-aware scaling and pricing differentiation [24,25]. In today's *MEC* settings, AI-driven and cooperative migration frameworks have demonstrated success in reducing latency and increasing operational gains.
- **Edge Network Infrastructure:** Edge computing infrastructure is inherently decentralized and heterogeneous, often consisting of micro-data centers, access points, vehicular nodes, and even end devices with opportunistic computing capabilities. These resources exhibit varying computational power, energy availability, and connectivity quality. Many edge resources—such as parked autonomous vehicles or idle roadside units (*RSUs*)—remain underutilized unless they are actively integrated into the service ecosystem. Real-time service migration serves as the orchestrator, redistributing workloads across these fragmented units based on real-time availability. Through migration-aware scheduling, edge systems can proactively or reactively bypass overloaded nodes and utilize transient capacity, improving both efficiency and resilience across the network [26,27].

1.3. Contribution and Organization

This article provides a comprehensive survey of recent research on service migration in edge computing networks, with a particular focus on key migration motivations,

namely, overload mitigation and resource rebalancing, user mobility and location awareness, energy efficiency and device lifespan management, and latency optimization and QoS enhancement, as well as enabling techniques, application scenarios, and future challenges. To offer readers a structured overview of the field, this article provides contributions on five main aspects:

- **Architecture, Basic Model, Benchmark Datasets, and Open Platforms (Section 2):** We present four representative edge computing architectures, i.e., cloud–edge–end, edge–edge, cloud–edge fusion, and edge–device collaboration, and analyze their respective support for real-time service migration. We also introduce four analytical models (network model, latency model, energy consumption model, and utility model) that collectively provide a theoretical foundation for service migration decision-making and performance evaluation.
- **Migration Motivation (Section 3):** We identify and elaborate on four primary motivations for real-time service migration in edge environments: (i) overload mitigation and resource rebalancing; (ii) user mobility and location awareness; (iii) energy efficiency and device lifespan management; and (iv) latency optimization and QoS enhancement. These motivations reflect the practical challenges faced by edge systems and drive the design of adaptive migration strategies.
- **Key Techniques for Service Migration (Section 4):** We categorize and examine six mainstream technical approaches for enabling real-time service migration: (i) approximate algorithms, (ii) heuristic algorithms, (iii) game-theoretic models, (iv) reinforcement learning, (v) deep learning, and (vi) deep reinforcement learning. We analyze how each technique contributes to efficient migration decisions under constraints such as delay, energy, load imbalance, and mobility uncertainty.
- **Service Migration Application Scenarios (Section 5):** We explore the deployment and effectiveness of real-time service migration in four key application domains: smart cities, smart homes, smart manufacturing, and smart healthcare. These scenarios demonstrate how timely and adaptive migration supports system scalability, responsiveness, and contextual awareness in real-world edge environments.
- **Challenges and Future Directions (Section 6):** We highlight critical open issues: inaccurate or delayed migration decisions that limit service awareness; scheduling and coordination challenges in large-scale heterogeneous edge networks; lack of comprehensive security and privacy protection during service migration; lack of adaptive and context-aware autonomous migration mechanisms; challenges and opportunities of AI-driven service migration; migration in the age of 6G: ultra-dense and high-mobility networks; and sustainable and energy-aware service migration.

Although this survey does not follow the PRISMA guidelines for systematic reviews, we have curated a representative and focused set of recent studies. The selection was guided by three main principles: (i) technical relevance to real-time service migration in edge environments; (ii) publication quality, including peer-reviewed papers from high-impact venues; and (iii) empirical rigor, as demonstrated by the presence of evaluation, modeling, or implementation results. To facilitate a comprehensive understanding of this survey's structural framework, Figure 1 schematically illustrates its overall organization. To ensure the consistency and relevance of our survey, we adopted the following criteria for including or excluding specific service migration strategies:

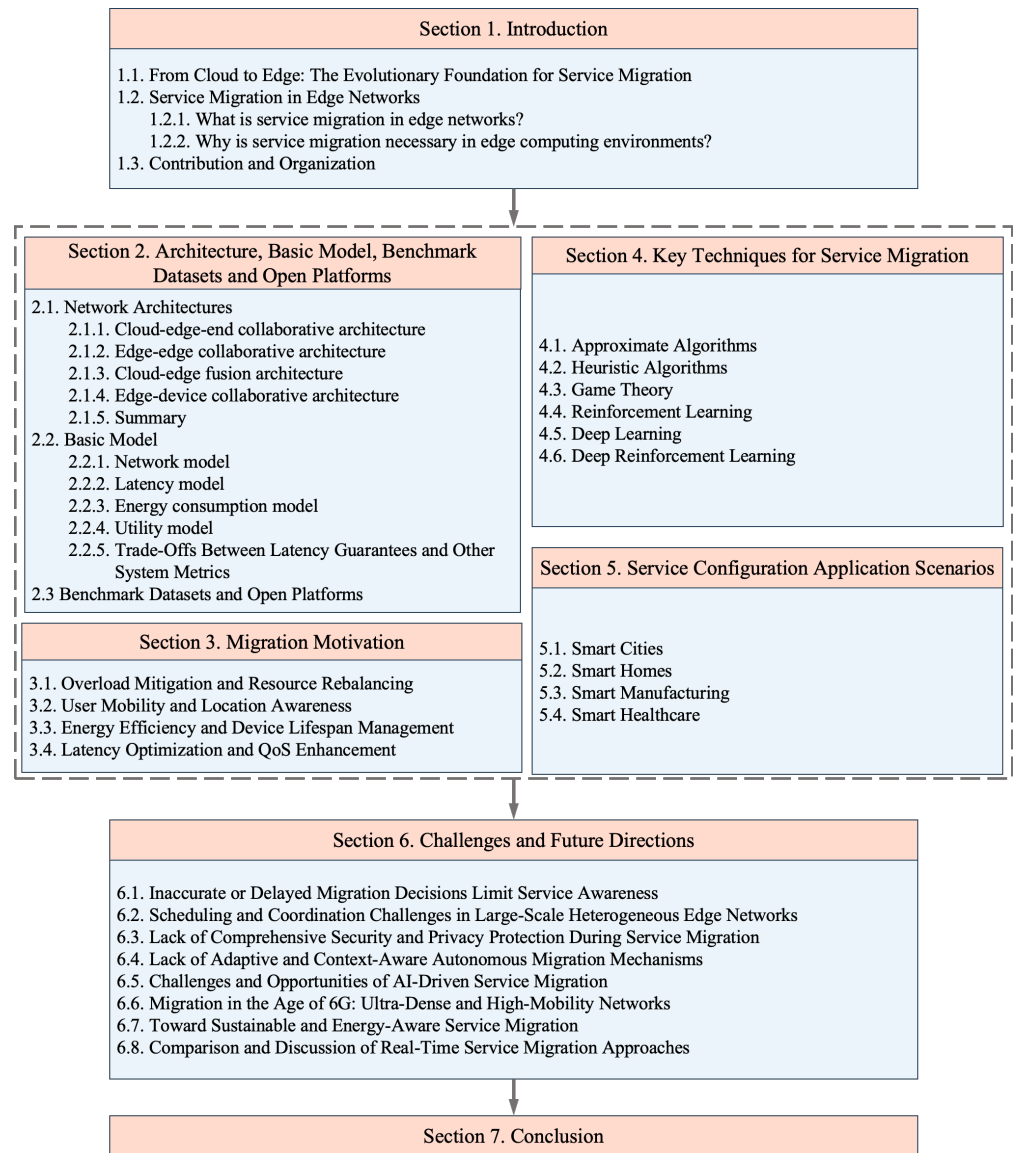


Figure 1. Road map of this survey.

- The strategy must support real-time or low-latency requirements, which are fundamental for delay-sensitive edge applications. Given our focus on real-time migration, this serves as the primary inclusion criterion.
- The strategy must be applicable to edge computing environments, such as *MEC*, *IoT*, and other distributed edge infrastructures, where computational offloading and proximity-aware scheduling are essential.
- The strategy should be compatible with the system models introduced in Section 2. This ensures comparability and analytical consistency across reviewed methods.
- The strategy must address practical challenges such as overload mitigation, user mobility support, energy efficiency, and QoS enhancement—all of which are elaborated in Section 3.
- The strategy should adopt one of the key technical approaches reviewed in Section 4, such as heuristic optimization, approximation methods, game-theoretic models, or reinforcement learning. We selected representative works from each category to reflect methodological diversity.
- The strategy should align with the architectural patterns outlined in Section 2.1. Only strategies where migration decisions and executions are primarily performed

at the edge layer are considered, excluding those solely based on centralized cloud-side control.

2. Architecture, Basic Model, Benchmark Datasets, and Open Platforms

2.1. Network Architectures

With the increasing demand for latency-sensitive, high-throughput, and dynamically changing service requirements in the IoT, traditional cloud-centric infrastructures face growing challenges. Edge computing has emerged as a key paradigm for addressing these issues by moving computation and data storage closer to the data source. In this context, service migration is not limited to static deployment but is evolving toward dynamic, context-aware, and migration-enabled frameworks. This section explores four representative architectures that underpin modern service migration systems, with a focus on how each supports efficient service migration.

2.1.1. Cloud–Edge–End Collaborative Architecture

This classical architecture partitions computing responsibilities across three layers: cloud, edge, and end devices (as illustrated in Figure 2). The cloud layer is responsible for macro-level orchestration, historical data analysis, and global decision-making. The edge layer enables near-source processing and real-time response, while the end layer, comprising sensors and mobile devices, conducts lightweight data acquisition and preliminary computation.

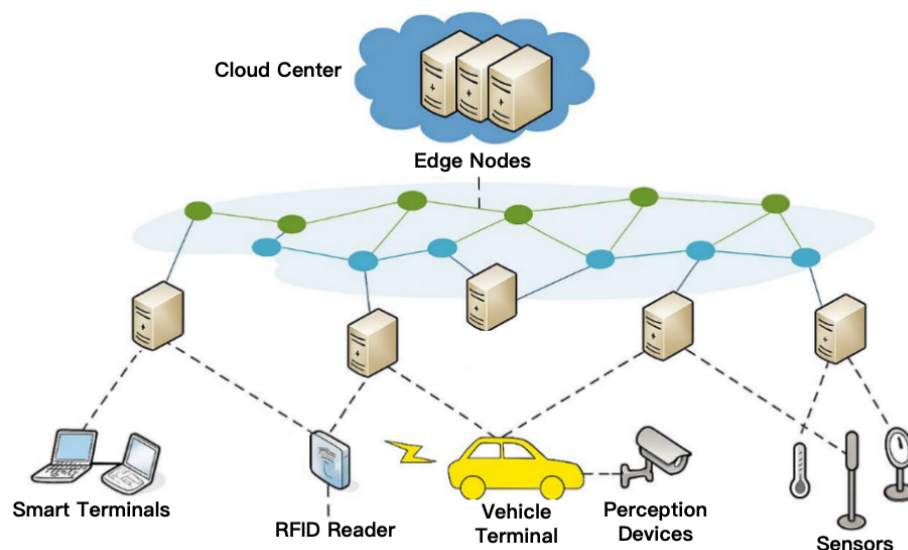


Figure 2. Cloud–edge–end collaborative architecture.

Service migration in this hierarchical model is typically performed from the cloud to the edge to reduce latency or from the edge back to the cloud to relieve local node overload. A notable feature is its support for vertical service migration, which facilitates adaptive offloading based on task priority, data size, or network conditions.

To support such vertical migration, containerization and virtualization are adopted to encapsulate services in lightweight, portable units. Policies based on latency thresholds, bandwidth usage, and node energy status determine when and where migrations occur.

2.1.2. Edge–Edge Collaborative Architecture

In edge–edge collaborative architectures, services are not solely coordinated via the cloud. Instead, multiple edge nodes form a lateral cooperative network. Such architecture is particularly advantageous for highly dynamic or mobile applications like vehicular

networks and *unmanned aerial vehicle (UAV)* systems, where continuous connectivity to the cloud is impractical.

Service migration in this framework occurs horizontally between the edge nodes. Migration decisions are made based on real-time load balancing, resource availability, and proximity to the data source. Predictive models based on user mobility patterns or traffic forecasts help optimize migration timing.

This lateral migration requires a decentralized orchestration strategy, commonly enabled by distributed *RL*, gossip protocols, or blockchain-based service registries. Inter-edge synchronization mechanisms are crucial to prevent service inconsistency or migration delays.

2.1.3. Cloud–Edge Fusion Architecture

The cloud–edge fusion architecture blurs the boundaries between the cloud and the edge by treating them as an integrated computing fabric. Services are dynamically decomposed, with components concurrently running in both layers depending on computational demands, latency constraints, and resource availability.

Service migration in this architecture is highly flexible and hybrid, supporting both vertical and horizontal movements. For instance, in a video analytics pipeline, preprocessing may occur at the edge, while deep feature extraction and long-term pattern analysis are migrated to the cloud.

Such collaborative service execution relies on fine-grained service decomposition, container orchestration platforms such as Kubernetes with edge extensions, and elastic scaling. Shared state management across layers ensures service integrity during migrations.

This architecture model not only enables cooperative service execution but also supports dynamic service migration across heterogeneous cloud–edge infrastructures. Several recent studies have proposed real-time service migration strategies tailored for hybrid cloud–edge environments. For example, some people introduced a data- and computation-intensive service adaptation method based on service migration, which utilizes a combination of greedy algorithms and the Non-dominated Sorting Genetic Algorithm II (*NSGA-II*) to optimize placement and reduce communication overhead during migration. Similarly, they proposed a game-theoretic distributed migration strategy for dynamic network reallocation, enabling lightweight containerized services to be adaptively reassigned based on temporal and spatial resource demands.

These approaches demonstrate how real-time service migration can be efficiently achieved within a cloud–edge fusion architecture, improving quality of service, reducing latency, and ensuring better resource utilization in dynamic and heterogeneous environments.

2.1.4. Edge–Device Collaborative Architecture

With the increasing computational power of terminal devices, the edge–device collaborative architecture has gained traction. In this model, end devices such as smartphones, *AR* glasses, or embedded sensors actively participate in service execution.

Service migration between edge nodes and end devices enables device-aware migration, a strategy where lightweight services or early-stage processing are pushed to capable devices. For instance, in mobile *AR* applications, scene recognition modules can be migrated to a user's smartphone to reduce network dependency.

Migration is guided by *RL* agents on the edge nodes, continuously evaluating device status like CPU load, battery level, and location. Services are often sandboxed and encrypted to protect privacy due to the semi-trusted nature of end devices.

2.1.5. Summary

The architectural diversity inherent in edge computing reflects the system-of-systems nature of *IoT* environments, where heterogeneous components should seamlessly interact across dynamic and distributed layers. As Fortino et al. [28] emphasize, the *IoT* can be conceptualized as an integrated system of systems, which demands robust methodologies and frameworks to coordinate services across multiple computing domains. The architectural approaches discussed above offer complementary strategies for addressing this complexity, supporting scalable, context-aware, and latency-sensitive service migration mechanisms. These four architectures represent complementary strategies in supporting service configuration and migration:

- **Cloud–Edge–End:** Global orchestration and stable vertical migration.
- **Edge–Edge:** High-frequency, localized lateral migration for dynamic contexts.
- **Cloud–Edge Fusion:** Fine-grained hybrid service placement and joint orchestration.
- **Edge–Device:** Maximizes responsiveness and privacy via terminal computation.

All these architectures aim to enhance *QoS*, reduce latency, and optimize resource utilization in heterogeneous *IoT*-driven edge networks.

2.2. Basic Model

The notations used in the paper and their descriptions are summarized in Table 1.

Table 1. Notations and descriptions.

Notation	Description
s_i	The i -th subtask of a service request
d_i	Data volume to be processed by subtask s_i
c_i	Computational demand of s_i , typically in CPU cycles
τ_i	Maximum response time constraint for subtask s_i
x_{ij}	Binary decision variable: 1 if s_i is assigned to node n_j and 0 otherwise
f_j	Computing capacity of node n_j , typically CPU processing speed
b_j	Bandwidth of node n_j
T_i^{comp}	Computation delay of s_i on node n_j , $T_i^{\text{comp}} = \frac{c_i}{f_j}$
T_i^{comm}	Communication delay of s_i between nodes, $T_i^{\text{comm}} = \frac{d_i}{b_j}$
T_i	Total delay of s_i , $T_i = T_i^{\text{comp}} + T_i^{\text{comm}}$
T_{total}	Total delay of all tasks, $T_{\text{total}} = \sum_{i=1}^n T_i$
$\bar{T}$	Average delay of all tasks
E_i^{comp}	Computation energy of s_i , $E_i^{\text{comp}} = \kappa(f_j)^2 \cdot c_i$
E_i^{comm}	Communication energy of s_i , $E_i^{\text{comm}} = P_{tx} \cdot \frac{d_i}{b_j}$
E_i	Total energy of s_i , $E_i = E_i^{\text{comp}} + E_i^{\text{comm}}$
E_{total}	Total energy of all tasks, $E_{\text{total}} = \sum_{i=1}^n E_i$
κ	Energy coefficient based on hardware characteristics
P_{tx}	Power consumption for data transmission per unit time
$\alpha_1, \alpha_2, \alpha_3$	Weights for components in the utility or <i>QoS</i> functions
α_4	Weight for energy consumption in the utility function
$\beta_1, \beta_2, \beta_3$	Weights for components in the quality of experience (<i>QoE</i>) function
A_i	Availability of task s_i
R_i	Reliability of task s_i
S_i	Stability score of task s_i
U_i	System responsiveness to the user request of task s_i

Table 1. Cont.

Notation	Description
Success Rate	Task success rate, $\frac{N_{\text{success}}}{N_{\text{total}}}$
N_{success}	Number of successfully completed tasks
N_{total}	Total number of tasks requested
R_j^{used}	Resources used on node j
R_j^{total}	Total available resources on node j
U_j	Resource utilization of node j , $\frac{R_j^{\text{used}}}{R_j^{\text{total}}}$
$\bar{U}$	Average resource utilization across all nodes
m	Total number of edge nodes in the system
U	Utility function value representing overall system performance

We elaborate on the basic models, including Network Model, Latency Model, Energy Consumption Model, and Utility Model, as shown in Figure 3.

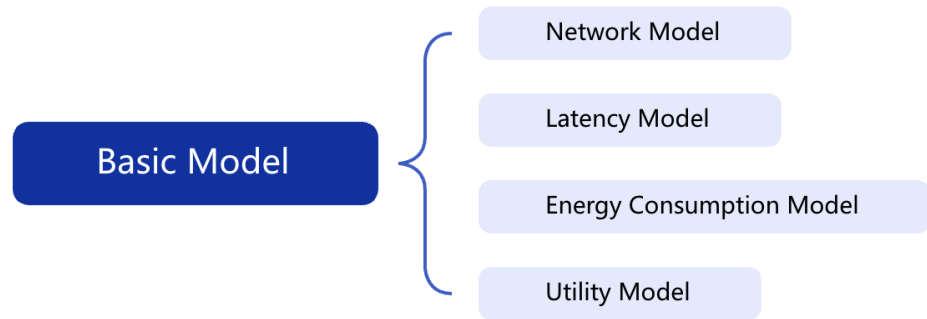


Figure 3. Basic model.

2.2.1. Network Model

In edge computing environments, efficient service configuration and scheduling are foundational to ensuring QoS under heterogeneous and resource-constrained conditions. The core objective of the network model is to determine how service tasks should be distributed across the device layer (e.g., mobile or IoT terminals), edge layer (e.g., MEC servers, base stations), and cloud layer so as to optimize system performance in terms of delay, energy, reliability, and resource utilization.

A complex service request R is typically decomposed into a set of interrelated subtasks $\{s_1, s_2, \dots, s_n\}$, each with distinct computational demands, data characteristics, and latency sensitivity. Each subtask s_i is characterized by three key attributes:

- d_i : Data volume required for processing or transmission.
- c_i : Computational demand (e.g., CPU cycles).
- τ_i : Latency constraint or deadline for subtask execution.

To formulate the assignment of tasks, we define a binary scheduling variable x_{ij} such that

$$x_{ij} = \begin{cases} 1, & \text{if subtask } s_i \text{ is assigned to node } n_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

This binary decision variable is used extensively in service deployment problems, where node n_j can be a device, edge node, or cloud server [29]. The goal is to ensure that each subtask is assigned to a computational node for which its resources and network connectivity are sufficient for meeting both its processing and communication demands.

Furthermore, in real-world service scenarios, such as multi-stage *IoT* data analytics, *AR*, or industrial control systems, subtasks are often not independent. Instead, they exhibit strict inter-task dependencies where a task can only start execution after the completion of one or more prerequisite tasks. These dependency relationships can be effectively captured by a *directed acyclic graph* (*DAG*), denoted as $G = (V, E)$, where each vertex $v_i \in V$ represents a subtask s_i , and each directed edge $e_{ij} \in E$ implies that subtask s_j depends on the output of s_i .

DAG-based task modeling has been widely adopted in edge computing, especially for complex workflows in video processing pipelines, smart healthcare monitoring, and vehicular networks. For instance, in a real-time video surveillance service, subtasks may include frame capture, preprocessing, object detection, and result aggregation, each sequentially dependent on prior stages. *DAG* scheduling ensures correctness in execution order while optimizing makespan or latency.

In this context, the network model must solve a joint problem of task-to-node assignment and inter-node scheduling under dependency constraints. This introduces several practical challenges:

- **Resource Heterogeneity:** Edge nodes vary in computing power, storage, and connectivity.
- **Mobility and Dynamism:** Task execution environments may change due to user mobility or fluctuating load.
- **Dependency-Aware Scheduling:** Inter-task communication delays must be considered in dependent subtasks.

To address these challenges, several task scheduling strategies have been proposed, including heuristic approaches (e.g., earliest finish time, minimum critical path), graph partitioning methods, and learning-based schedulers. For example, some people proposed a context-aware dynamic scheduling method for *DAG*-based service graphs in edge–cloud collaborative networks. And they further introduced a multi-level dependency inference mechanism that allows runtime reconfiguration of the *DAG* under uncertain task arrivals.

By integrating *DAG* models with real-time resource state and network metrics, the network model provides a formal basis for dynamic, distributed, and latency-aware task allocation. This modeling framework lays the foundation for subsequent latency, energy, and utility optimization discussed in the following subsections.

2.2.2. Latency Model

In edge computing environments, latency is a core performance metric that directly affects the responsiveness and reliability of time-sensitive applications such as autonomous driving, real-time video analytics, and industrial control. Accurately modeling latency is therefore essential for task scheduling, resource allocation, and service migration decisions.

The total latency T_i for executing a subtask s_i on node n_j is commonly decomposed into two additive components: computation latency and communication latency:

$$T_i^{\text{comp}} = \frac{c_i}{f_j}, \quad T_i^{\text{comm}} = \frac{d_i}{b_j} \quad (2)$$

$$T_i = T_i^{\text{comp}} + T_i^{\text{comm}} = \frac{c_i}{f_j} + \frac{d_i}{b_j} \quad (3)$$

Here, c_i is the computational requirement of task s_i , f_j is the processing capacity of node n_j , d_i is the volume of data to transmit, and b_j is the available bandwidth. These parameters are heterogeneous across nodes due to differing hardware and networking capabilities.

This latency decomposition is widely adopted in the mobile edge computing literature, and aligns with empirical observations in 5G-enabled *IoT* platforms. For instance, Li et al. [30] experimentally showed that computation delay is dominant for CPU-intensive

tasks, whereas communication latency becomes critical in bandwidth-constrained scenarios such as *UAV* swarms or remote sensing networks.

In practice, latency is not only a function of node capabilities but also influenced by dynamic factors such as background workload, link congestion, and user mobility. For tasks with dependencies (e.g., *DAG*-modeled workflows), the end-to-end delay should also account for waiting time and data transfer latency between predecessor and successor tasks:

$$T^{\text{total}} = \max_{\text{all paths in DAG}} \sum_{s_i \in \text{path}} T_i \quad (4)$$

This reflects the critical-path-based latency formulation adopted in delay-aware *DAG* scheduling algorithms. To optimize latency in such settings, researchers have developed multi-hop service placement models, predictive mobility-aware routing, and computation offloading strategies that proactively reassign tasks to nodes with lower execution and transmission latency.

By integrating real-time bandwidth sensing, queue length estimation, and edge load profiling, advanced latency models can support adaptive service migration. This is particularly important in applications requiring end-to-end latency guarantees (e.g., haptic control, telemedicine), where even minor fluctuations may violate *SLAs*.

2.2.3. Energy Consumption Model

In edge computing systems, energy efficiency is a critical design objective, especially under constraints imposed by battery-powered IoT devices and mobile edge nodes. Accurately modeling energy consumption is essential for optimizing task offloading, scheduling, and service migration decisions in real-world scenarios.

The total energy consumed by a task s_i executed on node n_j consists of two main components: computation energy and communication energy:

$$E_i^{\text{comp}} = \kappa_j \cdot f_j^2 \cdot c_i \quad (5)$$

$$E_i^{\text{comm}} = P_{tx,j} \cdot \frac{d_i}{b_j} \quad (6)$$

$$E_i = E_i^{\text{comp}} + E_i^{\text{comm}} \quad (7)$$

$$E_{\text{total}} = \sum_{i=1}^n E_i \quad (8)$$

In Equation, κ_j is the energy coefficient dependent on the hardware characteristics of node j , such as CPU types and cooling efficiencies. The term $f_j^2 \cdot c_i$ represents the dynamic power consumption of the processor when executing c_i CPU cycles at frequency f_j . This model is consistent with the power–frequency squared relationship derived from the *dynamic voltage and frequency scaling (DVFS)* principle, which is widely implemented in modern mobile and embedded processors [31].

In real deployments, edge servers, mobile devices, and cloud nodes exhibit heterogeneous κ_j values. For instance, Li et al. [29] observed that terminal nodes such as smartphones exhibit significantly higher energy cost per CPU cycle than fixed edge servers, which justifies prioritizing local execution only for low-complexity tasks.

In Equation (6), communication energy is modeled as a linear function of transmission time, which is determined by the data size d_i and bandwidth b_j . $P_{tx,j}$ is the average transmission power of node j , which may vary depending on its radio technology (e.g., Wi-Fi, 5G, LoRa). This model assumes a constant transmit power during data transfer, a simplification that aligns with most empirical studies on wireless edge networks.

Advanced models may further refine this by incorporating link quality indicators such as the signal-to-noise ratio (SNR) or *packet error rate* (PER). For example, adaptive modulation and coding schemes may adjust P_{tx} dynamically to trade off energy with reliability.

From a system-wide perspective, the total energy consumption E_{total} is the summation of energy consumed by all subtasks across all layers. Optimizing this metric is vital in applications such as environmental sensing or smart wearables, where energy conservation directly prolongs operational lifespan.

Recent studies have proposed multi-objective scheduling strategies that jointly minimize latency and energy, often expressed as the *energy-delay product* (EDP). They introduced a lightweight EDP-aware migration policy that achieved over 35% energy savings in UAV edge systems without compromising response time. Similarly, RL agents have been trained to minimize long-term energy costs under workload uncertainty.

In summary, energy modeling at the edge requires capturing the quadratic impact of CPU frequency on computation and the linear dependence of transmission energy on bandwidth. Such models lay the foundation for green computing strategies in resource-constrained edge environments.

2.2.4. Utility Model

In real-time edge computing systems, service orchestration often involves balancing multiple conflicting performance goals such as minimizing latency, reducing energy consumption, maximizing task success rate, and ensuring QoS. To address this, a utility function is typically defined to quantitatively assess overall system performance and guide decision-making.

The utility function can be expressed as a weighted combination of key metrics:

$$U = \alpha_1 \cdot QoS + \alpha_2 \cdot \text{Success Rate} - \alpha_3 \cdot \bar{T} - \alpha_4 \cdot E_{total} \quad (9)$$

where the following definitions are provided:

- QoS: An aggregate score based on service availability, reliability, and responsiveness.
- Success Rate: Ratio of completed to requested tasks.
- $\bar{T}$: Average task latency.
- E_{total} : Total energy consumption.
- $\alpha_1, \alpha_2, \alpha_3$, and α_4 : Scenario-specific weights.

This formulation allows system designers to trade off between performance metrics according to the application context. For instance, autonomous driving systems prioritize latency and reliability ($\alpha_3 \gg \alpha_4$), whereas smart home scenarios may prefer energy efficiency and cost ($\alpha_4 \gg \alpha_3$) [30].

Utility-based models form the foundation of many optimization frameworks, including the following:

- **Pareto-Based Multi-Objective Optimization:** Used when no single solution dominates all objectives (e.g., *NSGA-II*, *multi-objective evolutionary algorithms* (MOEAs)).
- **RL:** Agents learn optimal migration/scheduling policies by maximizing long-term utility over dynamic environments.
- **Heuristic/Metaheuristic Search:** Methods like genetic algorithms, simulated annealing, or ant colony optimization are used to explore the utility landscape efficiently.

In recent work, some people proposed a double-objective utility function considering EDP, while they employed a Deep Q-Network (DQN)-based learning agent to maximize service utility under dynamic task arrivals and user mobility. Such models have been successfully applied in smart city platforms, UAV swarms, and mobile health applications.

Therefore, the utility function serves not only as a performance metric but also as a core design principle for service migration algorithms, enabling the system to adaptively allocate resources under diverse operational constraints.

2.2.5. Trade-Offs Between Latency Guarantees and Other System Metrics

In real-time edge service migration, latency is often treated as a primary optimization objective due to the stringent response time requirements of applications such as autonomous driving and real-time video analytics. However, enforcing strict latency guarantees often introduces trade-offs with other critical system metrics, including energy consumption, operational cost, and reliability.

First, optimizing for minimal latency typically involves deploying services to nodes closest to end users or rapidly migrating services upon user mobility. This low-latency objective often results in increased service migration frequency, leading to higher energy consumption on edge nodes and mobile devices. Additionally, the need for more frequent computation offloading or redundant service instances to meet latency constraints increases bandwidth usage and processing overhead, ultimately raising infrastructure cost and system complexity.

Second, minimizing latency may conflict with energy-efficient scheduling strategies. For instance, offloading tasks to underutilized but distant nodes could save energy but induce higher delays. Conversely, using powerful nearby nodes ensures low latency but drains energy faster, reducing sustainability in resource-constrained environments.

Third, latency-optimized strategies may occasionally sacrifice reliability. Real-time migration that favors latency may not always consider node failure risks or resource instability, potentially leading to service interruptions.

Therefore, a balanced design is crucial. For latency-critical applications (e.g., autonomous vehicles), latency must be prioritized even at the cost of energy or cost efficiency. In contrast, for delay-tolerant scenarios (e.g., periodic data aggregation in smart homes), system designers may lean toward optimizing cost or energy while relaxing latency constraints.

A comprehensive service migration model should integrate latency as one of multiple objectives, with weights dynamically adjusted based on application requirements and environmental context. Some recent studies propose *RL* or Pareto-based multi-objective optimization frameworks to adaptively balance latency and other metrics in dynamic edge environments.

2.3. Benchmark Datasets and Open Platforms

To support reproducible experimentation and to bridge the gap between theoretical models and real-world deployment, several studies have introduced benchmark datasets and open platforms tailored for service migration in edge computing environments. These resources play a critical role in validating the effectiveness of migration policies under realistic mobility, network, and resource conditions.

Wang et al. [32] conducted one of the earliest comprehensive surveys on service migration in mobile edge computing, emphasizing the importance of leveraging empirical datasets, such as GPS traces from urban vehicles and mobile users, to evaluate latency and continuity under dynamic mobility patterns. Expanding on this idea, Wang et al. [33] formulated the service migration problem as a *Markov decision process (MDP)* and evaluated their solution using real-world taxi traces from San Francisco. Their experiments demonstrated how realistic mobility patterns can influence the decision boundaries of migration policies and highlighted the benefit of data-driven evaluation in capturing spatiotemporal user behaviors.

In a more recent study, Liu et al. [34] proposed a *deep reinforcement learning (DRL)*-based framework to jointly optimize service migration and resource allocation in edge-enabled *IoT* systems. The effectiveness of the approach was evaluated using real-world urban taxi mobility traces, demonstrating how realistic user mobility patterns impact the adaptability and efficiency of migration strategies. This work further highlights the importance of benchmark datasets in capturing the dynamic and context-sensitive nature of edge computing environments.

Yuan et al. [35] proposed a joint optimization framework for vehicular service migration and mobility control, and they used large-scale vehicular traffic simulations to emulate realistic road conditions and service demands. By incorporating routing constraints and edge resource limitations into their test scenarios, their work illustrated the complex interplay between migration strategies and user trajectory planning—offering a valuable benchmark for future research.

In terms of implementation, Chen et al. [36] introduced an edge cognitive computing architecture that includes a dynamic service migration mechanism based on user behavior cognition. They also developed an *edge cognitive computing (ECC)*-based test platform that supports real-time migration and load adaptation, enabling elastic and personalized service delivery. This practical system was evaluated using user interaction data and demonstrated measurable gains in latency reduction and energy efficiency—making it a rare example of an open experimental platform supporting real-world service migration scenarios.

Together, these works provide both methodological guidance and practical tools for the community, illustrating how real datasets, mobility traces, and cognitive platforms can be combined to develop, test, and benchmark service migration solutions under diverse and dynamic edge environments.

3. Migration Motivation

We elaborate on the key aspects of service migration motivation, including Overload Mitigation and Resource Rebalancing, User Mobility and Location Awareness, Energy Efficiency and Device Lifespan Management, and Latency Optimization and QoS Enhancement, as shown in Figure 4.

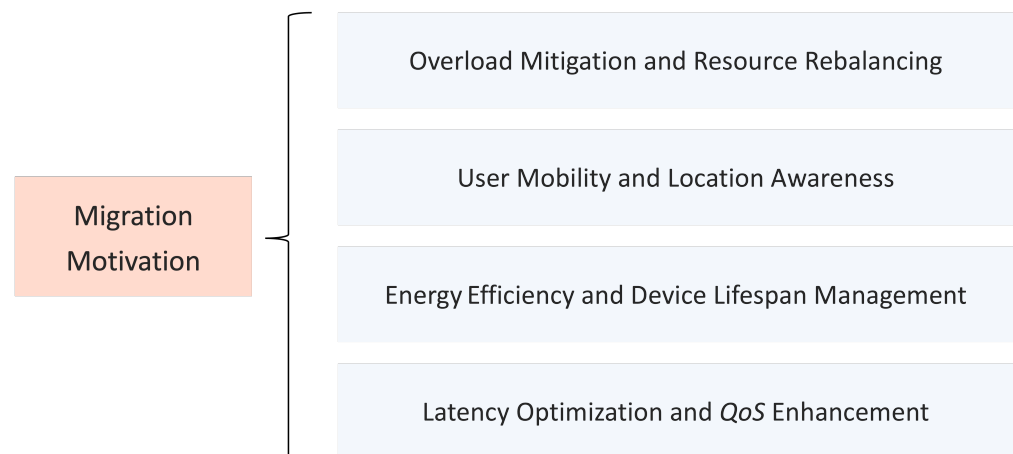


Figure 4. Service migration motivation.

3.1. Overload Mitigation and Resource Rebalancing

In edge computing networks, localized overload frequently arises due to highly dynamic user behavior, bursty traffic patterns, or spatially uneven service demands. Without timely mitigation, overloaded nodes suffer from degraded *QoS*, prolonged response times, and potential service failures. Service migration serves as an effective strategy to

redistribute workloads toward underutilized edge nodes, thereby maintaining system performance and improving user experience.

A broad body of research has explored overload-triggered service migration mechanisms. Gao et al. [37] proposed a time-aware model that migrates services based on real-time load and response delay to reduce latency. Liang et al. [38] introduced a joint optimization framework, leveraging game theory to dynamically schedule service migration and bandwidth allocation in multi-cell edge environments. Yuan et al. [35] extended this idea with an MDP-based model to determine optimal migration policies under multi-user interference and mobility uncertainty.

Container-level service migration has also received attention for its granularity and responsiveness. Yang et al. [39] designed a task-aware migration mechanism using checkpointed containers to minimize cold-start delays and enable load-sensitive relocation of microservices. Kaur et al. [40] provided a survey on container migration strategies across cloud, fog, and edge systems, highlighting how predictive load monitoring and adaptive migration policies can improve performance and energy efficiency.

In addition, Gao et al. [37] presented a time-segmented cloud–edge collaboration framework that dynamically reconfigures edge services across spatial and temporal dimensions to alleviate overload and improve resource balance in large-scale infrastructures. Wang et al. [33] analyzed service migration trends from a holistic perspective, identifying congestion mitigation as a primary driver and emphasizing the integration of AI-driven orchestration frameworks to enable proactive migration. Li et al. [41] emphasized the importance of hybrid orchestration mechanisms that combine centralized global controllers with distributed local agents to address overload scenarios. Federated learning (FL) has also emerged as a promising enabler. Caro et al. [42] developed a privacy-preserving federated migration framework that enables edge nodes to collaboratively learn overload handling policies without raw data exchange. Building upon these developments, Bozkaya-Aras [43] introduced a digital twin-assisted migration framework that replicates the real-time operational state of *IoT* edge systems. By incorporating predictive analytics with bipartite graph optimization, the framework proactively reallocates services to balance workloads, reduce energy consumption, and minimize latency. This approach demonstrates how real-time resource sensing and system-level modeling can significantly enhance overload mitigation strategies, particularly under dynamic and traffic-intensive conditions.

Overall, service migration for overload mitigation and resource rebalancing plays a crucial role in maintaining edge system stability, improving responsiveness, and optimizing resource utilization in dynamic environments. As edge infrastructures become increasingly heterogeneous and demand-intensive, intelligent and adaptive migration strategies will remain a core mechanism for balancing workloads and sustaining service quality.

3.2. User Mobility and Location Awareness

In MEC, user mobility significantly impacts the effectiveness of service delivery. As users traverse different network regions—such as vehicles moving between *RSUs*, pedestrians switching between access points, or drones navigating dynamic airspace—the static deployment of edge services often results in increased latency, service discontinuity, and degraded *QoE*. To address this, service migration driven by mobility and location awareness aims to dynamically adapt service placement in response to user movements, enabling persistent low-latency access and seamless user experience.

Recent research underscores the importance of mobility-aware service migration, particularly in vehicular networks and *IoT*-rich urban environments. Jia et al. [44] proposed a Lyapunov optimization-based MEC framework for Internet of Vehicles (*IoV*), where computing resources are dynamically allocated and services are migrated across *RSUs* to

minimize communication delay and handoff overhead. Labriji et al. [22] further designed a dynamic migration model tailored for vehicular MEC environments, where *RSU* loads and vehicle positions are continuously monitored to facilitate predictive disruption-free service handover.

To better align migration decisions with user behavior, trajectory prediction techniques are widely adopted. Maleki et al. [45] leveraged machine learning models to predict short-term user positions and proactively trigger migration events before users leave a given edge node's coverage, thus reducing service drop rates. Filiposka et al. [46] developed a real-time mobility-aware resource management scheme that adjusts task allocation based on both current locations and historical movement traces, enhancing service stability in urban deployments. However, uncertainty in mobility patterns—especially in non-deterministic or multi-modal transportation settings—still poses challenges for the accuracy and robustness of these migration strategies. Service migration is also increasingly extended to heterogeneous and aerial edge computing environments. For instance, Liu et al. [47] investigated dynamic task offloading in multi-UAV edge networks, where services are adaptively assigned to UAVs in proximity to moving users, improving both energy efficiency and responsiveness. Meanwhile, Zhang et al. [48] examined QoS-driven mobility-aware and dependence-aware service chain adaptation, where interdependent modules are dynamically reallocated in accordance with both user movement and latency constraints between modules.

To safeguard user privacy during mobility-aware orchestration, *FL* was introduced. Wang et al. [49] proposed an asynchronous *FL*-based framework that enables vehicles to collaboratively learn location popularity models and optimize caching decisions without exchanging raw mobility data. Building on this, Wu et al. [50] integrated *FL* with *DRL* to develop a proactive mobility-aware caching and migration system that effectively reduces service interruption under rapid mobility conditions.

In summary, mobility-aware and location-aware service migration is essential for delivering reliable and responsive services in MEC, especially in highly dynamic or privacy-sensitive environments.

3.3. Energy Efficiency and Device Lifespan Management

In edge computing systems, where distributed edge nodes often operate under stringent energy and hardware limitations, maintaining long-term service reliability requires not only efficient task allocation but also proactive workload redistribution. Service migration, when driven by energy-awareness and hardware health considerations, becomes an essential mechanism to offload stress from battery-depleted, thermally saturated, or aging devices. This not only helps reduce the risk of abrupt service termination but also enhances the overall sustainability of edge infrastructures.

Recent studies have increasingly highlighted the importance of energy-efficient service migration. Li et al. [13] developed an online decision-making framework that triggers service migration when node-level energy consumption exceeds adaptive thresholds, enabling reduced energy footprints across the system without degrading service quality. Similarly, Zhou et al. [51] investigated energy-aware migration strategies for dense multi-user networks, demonstrating that offloading computation from heavily loaded base stations to neighboring energy-sufficient nodes can significantly extend device operational time and lower network-wide energy cost.

Service migration also plays a vital role in managing thermal stress and preventing hardware wear. As prolonged high-load execution increases thermal cycling, researchers have proposed thermal-aware migration to shift workloads away from overheating units. For example, Toumi et al. [52] proposed a scheduling strategy that integrates thermal

monitoring with energy metrics to decide migration timing in industrial *IoT* scenarios. Additionally, Ning et al. [53] designed a lightweight imitation learning-based mechanism to migrate services between mobile edge nodes in real time, ensuring minimal energy consumption and extended hardware uptime, particularly under fluctuating workloads and constrained power budgets.

In mobility-heavy edge environments such as vehicular or drone-based systems, migration decisions must also take into account dynamic energy availability across moving nodes. Niu et al. [54] proposed a meta-RL approach that adapts service scheduling according to residual energy, node aging characteristics, and user demand density, achieving higher overall efficiency compared to static heuristics. Furthermore, Ma et al. [55] integrated energy consumption prediction into a trajectory-aware migration policy, reducing unnecessary retransmission and improving the energy–delay tradeoff in user-centric edge applications.

Finer-grained strategies have emerged at the microservice level. Tocze and Nadjm-Tehrani's work [56] on distributed microservice placement formulates an *integer linear programming (ILP)* model to jointly optimize energy consumption and latency for edge service request placement. Their framework takes into account the resource heterogeneity across edge nodes, varying microservice communication patterns, and the stochastic nature of service arrivals. By using workload-aware prediction and request batching, their solution enables proactive scheduling that reduces redundant migrations and limits energy-intensive task duplications.

Overall, energy efficiency and device lifespan management form one of the most pragmatic motivations for service migration in edge computing, especially as edge nodes scale in number, diversity, and mission-critical responsibilities. Migration offers a viable path to shift computation away from vulnerable nodes toward stable, energy-rich environments, ultimately contributing to more resilient, cost-effective, and sustainable edge ecosystems.

3.4. Latency Optimization and QoS Enhancement

In latency-critical applications such as autonomous driving, virtual reality, and industrial control, the delay in service response can significantly degrade system performance and user experience. MEC aims to meet these demands by offering computation closer to users; however, fluctuating workloads, user mobility, and dynamic network states often disrupt this goal. Service migration emerges as a vital mechanism for dynamically adjusting service placement, ensuring low latency and consistent QoS.

Recent studies have increasingly emphasized latency-aware and collaborative service migration frameworks. Zeng et al. [57] proposed a two-stage collaborative microservice migration strategy for DAG-based applications, utilizing DRL and network flow optimization to reduce end-to-end latency in MEC. Their approach effectively distributes microservices across edge clusters, addressing both workload imbalance and latency-sensitive dependencies. Similarly, Zhang et al. [58] introduced the quality-of-service-aware edge–cloud service migration framework, which performs dynamic service migration guided by QoS metrics and mobility patterns, adapting to time-varying latency constraints across edge–cloud infrastructures. To anticipate latency degradation before it occurs, predictive migration models have been widely explored. Ma et al. [59] developed a forecasting-based placement scheme using Deep Learning (DL) to anticipate network congestion and proactively reallocate services to minimize potential response delays. Likewise, Peng et al. [60] designed a transfer RL-based approach to handle computing and communication costs in vehicular edge networks, ensuring low-latency service continuity even under high mobility and workload surges.

In highly dynamic environments, probabilistic and context-aware models improve adaptability. Xu et al. [61] introduced a *probabilistic delay- and mobility-aware* migration

framework (*PDMA*) that uses trajectory uncertainty modeling and real-time delay profiling to trigger migration events. Additionally, in large-scale edge deployments, Chi et al. [62] proposed a multi-criteria decision-making model that jointly considers network load, latency violations, and node reliability to select optimal migration targets for microservices in ultra-large *MEC* infrastructures. Moreover, low-latency service migration also benefits from contextual scheduling. Saha et al. [63] explored a scheduling mechanism based on contextual information such as link quality, service criticality, and temporal load fluctuation to drive proactive service relocation. Their model demonstrated improved *QoS* stability and reduced average migration delay.

Together, these advancements reflect a shift toward more intelligent, proactive, and fine-grained latency-sensitive service migration mechanisms. By integrating predictive analytics, learning-based decision models, and real-time context awareness, *MEC* systems can sustain high *QoS* standards in increasingly heterogeneous and mobile network environments.

4. Key Techniques for Service Migration

In edge-native computing environments, dynamic resource availability, task heterogeneity, and user mobility necessitate intelligent, adaptive service migration techniques. To address these challenges, this section systematically presents six representative approaches, emphasizing their real-time decision-making capabilities, deployment rationality, and suitability under constrained edge scenarios.

4.1. Approximate Algorithms

In the context of edge computing, service migration should respond to highly dynamic environments characterized by constrained resources, heterogeneous device capabilities, and fluctuating user demands. Approximate optimization algorithms have emerged as effective tools for tackling the inherent complexity of multi-objective migration problems in such environments. These methods offer a favorable trade-off between computational efficiency and solution quality, making them suitable for latency-sensitive applications at the edge.

In this work, we adopt *NSGA-II* to solve a tri-objective migration optimization problem that balances delay, energy consumption, and system load imbalance. *NSGA-II* enables the construction of Pareto-optimal migration strategies by maintaining solution diversity and converging toward global optima, which is particularly advantageous in multi-service edge scenarios with dynamic interaction topologies. Compared with deterministic or exhaustive search methods, *NSGA-II* significantly reduces computational overhead, thereby supporting real-time reconfiguration under volatile system loads.

Inspired by the operational scale and concurrency of *IoT* applications, swarm intelligence algorithms such as particle swarm optimization (*PSO*) and artificial bee colony (*ABC*) are further employed to explore decentralized solution spaces. For example, based on the methodology proposed in [64], *PSO* enables rapid exploration of high-dimensional migration states by simulating the social behavior of agents. *ABC*, on the other hand, enhances global search ability through division of exploration and exploitation phases. These algorithms are well-suited for non-convex and nonlinear optimization problems that are frequently encountered in real-world edge networks.

In addition, recent work has demonstrated the effectiveness of parallel multi-objective approximate algorithms in *IoT* service composition, where the energy-aware integration of temporal service flows can be optimized under shared constraints. Building on this insight, we extend the use of approximate models to the migration domain, adapting their structural advantages to reduce the frequency and cost of unnecessary migrations while maximizing system-wide performance consistency.

The principal strength of these approximate models lies in their scalability and adaptability. Heuristic-enhanced approximations can maintain optimal task scheduling under dynamically detected event boundaries. Our system design therefore incorporates these approximate methods not merely as auxiliary optimizers but as core enablers of real-time decision-making in migration control capable of supporting distributed edge networks with minimal tuning effort.

4.2. Heuristic Algorithms

In latency-sensitive edge computing scenarios, where real-time responsiveness and low-overhead decision-making are essential, heuristic algorithms provide practical and lightweight solutions for dynamic service migration. These methods rely on problem-specific knowledge and rule-based logic to generate near-optimal results within constrained time budgets, making them ideal for environments with limited computational capacity and high service volatility.

This work integrates a hybrid heuristic framework that combines offline initialization with online learning. Specifically, we adopt a two-stage method in which the initial service deployment is constructed using a *minimum spanning tree (MST)* over service dependency graphs, ensuring a low-latency communication baseline among interconnected service modules. On top of this static topology, we deploy a Q-value learning based online decision module to adaptively trigger migration in response to environmental changes [65]. This reinforcement-enhanced heuristic balances exploitation of known optimal placements with exploration of new migration paths, facilitating continuous optimization during system operation.

To further improve responsiveness under bursty workloads, we implement a time-window-based greedy scheduling mechanism. This approach collects and analyzes short-term system metrics such as CPU usage, memory availability, and link bandwidth within sliding time windows. Based on this historical state profile, the algorithm selects target nodes for service migration by evaluating marginal resource gains and local utility scores. Compared to static rule sets, this dynamic heuristic enables early identification of potential hot spots and mitigates performance bottlenecks before they escalate.

The effectiveness of such hybrid heuristics has been demonstrated in recent edge-centric scheduling frameworks where real-time task adaptation to detected events significantly reduces average processing delay [66]. Moreover, the system outlined shows that heuristic selection strategies guided by service execution history and task completion feedback can outperform static placement under variable request patterns and constrained device availability.

In our implementation, heuristic models act as lightweight controllers embedded within each edge node. Their low algorithmic complexity ensures negligible latency overhead, while their rule-adaptable architecture enables fast convergence to high-performing migration plans. This makes them particularly suitable for fine-grained service orchestration in scenarios involving mobile users, constrained bandwidth, or time-sensitive applications such as video analytics and industrial monitoring.

4.3. Game Theory

In decentralized edge computing environments, service migration decisions often involve multiple autonomous nodes or services, each seeking to optimize its own performance under shared resource constraints. Game-theoretic approaches offer a powerful modeling framework to capture these interactive and potentially conflicting objectives. By formulating migration as a strategic decision process among rational agents, such mod-

els support the design of stable and self-organizing service placement strategies without relying on centralized control.

In this work, we construct a non-cooperative game model in which each service instance acts as a self-interested player aiming to minimize its individual migration cost [67]. The utility function integrates delay, energy consumption, and migration overhead:

$$C_i(n_j) = \phi_1 \cdot \text{delay}_{ij} + \phi_2 \cdot \text{energy}_{ij} + \phi_3 \cdot \text{migration_cost}_{ij} \quad (10)$$

Each service selects a target node n_j such that no unilateral move can further reduce its cost—resulting in a Nash equilibrium. Compared with purely heuristic policies, the game-theoretic formulation guarantees convergence to stable configurations even under partial observability and asynchronous updates. Then, decentralized task allocation models based on potential games and evolutionary game theory can improve system efficiency while preserving individual rationality.

To enhance global resource utilization, we further integrate cooperative game principles by introducing incentive-compatible mechanisms such as reputation-based credits and resource sharing rewards. These mechanisms encourage edge nodes to participate in collaborative scheduling and migration, particularly in federated or heterogeneous edge clusters [68]. Drawing from the framework, we design a token-based exchange model where nodes gain credits by accepting external migration requests and spend them when offloading their own services. This balances load across the network while maintaining fairness and autonomy.

The use of game-theoretic models offers several advantages for real-time edge migration. First, it enables distributed execution and local decision-making, significantly reducing migration latency and coordination overhead. Second, it accounts for dynamic system states and strategic behaviors, allowing robust adaptation to varying traffic, node availability, and service priorities. Third, game-theoretic frameworks naturally extend to multi-agent and multi-tenant scenarios, supporting scalability in complex edge infrastructures.

By embedding game-theoretic agents within the service migration controller, our architecture ensures that migration decisions are both resource-aware and strategically sound. The combination of non-cooperative equilibrium seeking and cooperative incentive alignment enables intelligent workload reallocation that respects both system-wide objectives and local node preferences, making it well-suited for heterogeneous, large-scale edge deployments.

4.4. Reinforcement Learning

To address the sequential and uncertain nature of service migration in dynamic edge computing environments, *RL* offers an effective framework for modeling migration decisions as *MDPs* [69]. Unlike heuristic or rule-based methods, *RL* enables the system to autonomously learn optimal migration strategies through trial-and-error interactions with the environment, making it particularly suitable for environments characterized by user mobility, fluctuating workloads, and partial observability.

In our architecture, we formulate the service migration problem as a continuous-time Markov decision process (*CTMDP*), where the system state s encapsulates node resource usage, bandwidth availability, and user request queues. The reward function penalizes high delay and energy consumption:

$$R(s, a) = -(\gamma_1 \cdot \text{delay}(s, a) + \gamma_2 \cdot \text{energy}(s, a)) \quad (11)$$

To solve the *CTMDP*, we employ temporal-difference (*TD*) learning to update the state-value function iteratively:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)] \quad (12)$$

This approach enables online policy refinement in response to newly observed system feedback, enhancing the model's real-time adaptability.

The effectiveness of *RL* in edge service migration has been validated, where dynamic scheduling policies trained using tabular Q-value learning or *TD* learning frameworks significantly outperform static policies under non-stationary conditions. In particular, these works highlight *RL*'s ability to handle diverse *QoS* metrics, including task urgency, queue latency, and node-specific failure rates [69].

To further improve convergence speed and policy robustness, our system integrates historical replay buffers and adaptive learning rates. The replay buffer stores past state transitions and reward signals, enabling more stable and sample-efficient learning, especially in bursty service scenarios where instantaneous data may not be representative. Additionally, we initialize policies using heuristic priors to reduce exploration time in early deployment phases.

Compared to approximate or game-theoretic approaches, *RL* offers several key advantages in edge computing contexts:

- **Online Adaptation:** *RL* dynamically adjusts migration decisions in response to real-time feedback, accommodating non-deterministic workloads and mobility patterns.
- **Policy Generalization:** Once trained, the learned policy can be reused or fine-tuned across similar edge environments, reducing retraining costs.
- **QoS Awareness:** *RL* inherently supports multi-objective optimization, enabling fine-grained control over delays, energy, and resource utilization.

These properties make *RL* an attractive component for intelligent migration control. In our design, *RL* agents are deployed at edge orchestrators, interacting with local node monitors and network profilers to continuously refine migration behavior. As edge environments grow in complexity and scale, *RL* provides the flexibility and autonomy needed to maintain performance and responsiveness under evolving operating conditions.

4.5. Deep Learning

In edge computing environments characterized by dynamic user demands and fluctuating resource availability, *DL* serves as a critical predictive engine for intelligent service migration. Rather than relying on static heuristics or reactive rules, *DL* empowers the system with foresight—enabling proactive migration decisions that anticipate changes in workload intensity, service reliability, and node availability.

This work incorporates three classes of *DL* models tailored to distinct contextual sensing tasks. Convolutional neural networks (*CNN*) are employed to encode service state matrices, capturing spatial correlations in resource occupancy and communication bandwidth. These structured encodings are then fed into long short-term memory (*LSTM*) networks, which excel in learning temporal patterns in time-series traces such as CPU loads, memory usage, and request rates [70]. Furthermore, to model the complex topological dependencies among distributed edge nodes, graph convolutional networks (*GCNs*) are utilized, which construct migration-affinity graphs based on latency, bandwidth, and workload similarity, facilitating global coordination during large-scale service relocation.

The system integrates *DL*-based prediction modules in a hybrid learning-control loop. When offline, these models are trained using historical logs and event traces extracted from the cloud–edge system, with feature extraction driven by sliding-window analysis and variance-aware sampling. When online, the models operate periodically, issuing preemptive migration signals when predicted resource bottlenecks or service disruptions exceed

a learned threshold. This predictive mechanism helps maintain service continuity under volatile traffic, especially in scenarios involving mobile users or bursty sensor streams.

While deep models introduce computational overhead, they are deployed as auxiliary modules—decoupled from the core real-time control loop and updated asynchronously. In this architecture, migration decisions are enacted by lightweight agents that reference *DL* outputs without incurring their full runtime cost. Such decoupling ensures that predictive accuracy is retained while the system adheres to stringent delay constraints imposed by edge applications like smart surveillance and industrial control.

The adoption of *DL* in our migration framework marks a shift from reactive adaptation to anticipatory intelligence. By embedding learning-driven predictions into edge orchestration, the system achieves enhanced resilience and adaptability across diverse operating conditions. Empirical results from recent *IoT* scheduling systems further confirm that *DL*-enabled prediction reduces unnecessary migrations by up to 38% while maintaining *SLA* compliance, offering a compelling trade-off between foresight and efficiency.

4.6. Deep Reinforcement Learning

In highly dynamic and resource-constrained edge environments, where service placement decisions should respond to uncertain workloads, user mobility, and volatile network conditions, *DRL* offers a powerful solution framework. By combining deep neural networks with sequential decision-making logic, *DRL* enables edge systems to learn adaptive service migration policies directly from experience, without relying on fixed rules or explicit performance models.

In this work, we employ *DRL* models to capture the long-term effects of migration actions in Markovian edge environments. Specifically, *DQN* are adopted to approximate state-action value functions based on system states such as the CPU load, node availability, and bandwidth utilization. The agent iteratively refines its migration policy by minimizing *TD* errors between predicted and actual rewards [71]. To mitigate instability during training, we use experience replay buffers and periodically updated target networks.

To address the limitations of discrete action spaces in *DQN*, we further implement an actor–critic structure. In this design, the actor network generates continuous migration actions, while the critic evaluates action-value estimates and provides gradient-based feedback. This architecture, aligned with recent advances such as *deep deterministic policy gradient (DDPG)* and *twin delayed deep deterministic policy gradient (TD3)*, is particularly suitable for multi-node resource scheduling where fine-grained migration control is required under latency and energy constraints.

In large-scale edge deployments, service migration often involves multiple decentralized entities. We thus extend our model to *multi-agent DRL (MADRL)*, where each edge node functions as an autonomous agent. The *MADRL* system supports decentralized learning with centralized training, enabling agents to learn coordinated policies while preserving local autonomy [72]. This is particularly valuable in vehicular and federated edge scenarios, where communication overhead must be minimized and partial observability is common.

The integration of *DRL* into our service migration architecture enables predictive, adaptive, and energy-aware orchestration. Agents continuously refine their behavior based on environmental feedback, learning to avoid overloading hotspots, reduce inter-node latency, and maximize overall system utility. The experimental results from multi-service edge scheduling scenarios demonstrate that *DRL*-based migration can reduce average response latency by over 30% while simultaneously lowering energy usage and *SLA* violation rates compared to heuristic and static approaches.

By endowing edge infrastructure with *DRL*-enhanced decision logic, the system transitions from reactive response to autonomous optimization—supporting long-term service sustainability in highly heterogeneous and dynamic operational environments.

5. Service Migration Application Scenarios

Service migration plays a pivotal role in optimizing the performance and efficiency of modern computing paradigms, particularly in edge computing and *IoT* environments. Below, we summarize its applications in four key domains: smart cities, smart homes, smart manufacturing, and smart healthcare (as illustrated in Figure 5).

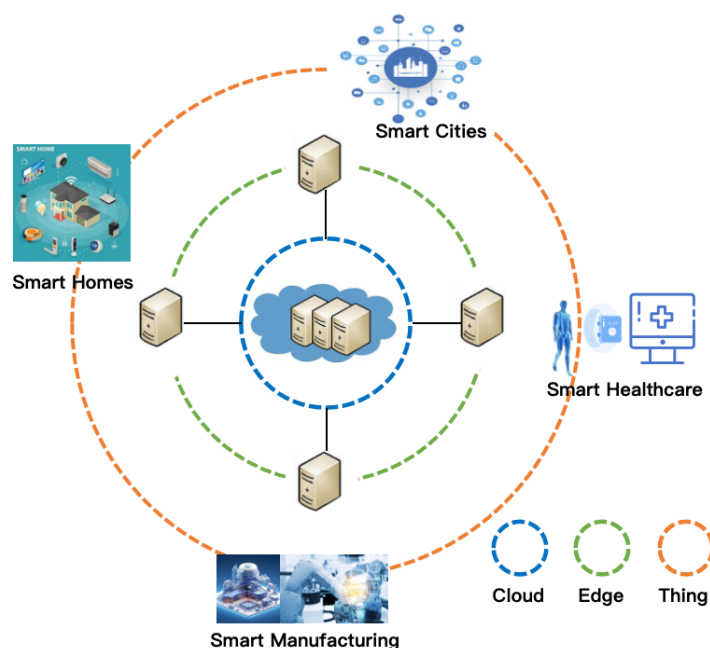


Figure 5. Service migration application scenarios.

5.1. Smart Cities

With the rapid advancement of technologies such as *IoT*, cloud computing, edge computing, and wireless communications, smart cities have evolved from theoretical frameworks into practical, interconnected ecosystems. These technologies empower urban infrastructure to deliver intelligent, efficient, and real-time services that significantly enhance residents' quality of life. However, as data volumes surge due to pervasive *IoT* deployment, traditional cloud-centric architectures encounter limitations, including high latency, bandwidth constraints, and heightened privacy risks. To mitigate these issues, edge computing combined with service migration has emerged as a promising solution that pushes processing closer to data sources, reducing response times and offloading core cloud resources [73].

Service migration enables dynamic relocation of computing tasks across distributed edge nodes in response to changing contexts, loads, and mobility. In traffic management systems, for instance, migrating traffic optimization algorithms between edge nodes near congested intersections allows real-time responsiveness and adaptive load balancing. Similarly, in smart surveillance applications, video analytics tasks can be relocated to local edge servers for rapid anomaly detection, improving both timeliness and data sovereignty. To optimize this migration process, Xu et al. [74] proposed a trust-aware *IoT* service provisioning strategy, integrating evolutionary algorithms with multi-criteria decision-making to balance load, energy consumption, and privacy. In healthcare and public services, real-time service migration ensures uninterrupted operation even under user mobility. Ganesan [75]

proposed a VM migration strategy for mobile cloud computing in healthcare applications, leveraging ant colony optimization to minimize latency and optimize resource utilization under dynamic user movement. Their model highlights the importance of predictive migration strategies in achieving quality-of-service guarantees in time-sensitive environments.

Further, Kientopf et al. [73] developed a service management platform under a fog computing architecture to support autonomous service migration. Their platform leverages network metrics such as the expected transmission count (ETX) and round trip time (RTT) to position services near active users in dense IoT environments, significantly reducing latency compared to centralized deployments. These insights reveal the potential of service-aware edge architectures in scaling smart city systems efficiently. Moreover, the smart city operating system (SCOS) [76] proposed by Vögler et al. provides a cloud-native framework for seamless migration and deployment of smart city services. With its microservice-based architecture and support for service mobility, SCOS enables applications to dynamically shift across edge, cloud, and hybrid infrastructures, ensuring resilience and agility in urban service delivery.

In summary, real-time service migration in edge networks enhances the responsiveness, scalability, and sustainability of smart cities. By dynamically reallocating computing services based on contextual factors—such as mobility, resource load, and network conditions—urban systems can maintain low-latency operation, support dynamic user demands, and uphold data governance standards. This adaptive migration paradigm forms a cornerstone of future-proof smart city infrastructure.

5.2. Smart Homes

The proliferation of smart home devices and the increasing demands for real-time, low-latency responses challenge traditional cloud-centric architectures due to constraints such as network congestion, limited bandwidth, and privacy concerns. Edge computing, particularly through service migration strategies, offers a viable solution by enabling services to dynamically relocate among local devices, edge gateways, and nearby servers based on contextual factors like user behavior, network conditions, and device status [77].

Recent approaches incorporate DRL for request-aware migration scheduling, demonstrating reduced latency and improved load balancing across heterogeneous edge nodes. Edge-centric operating systems for home environments introduce layered abstractions and lightweight orchestration, facilitating seamless service mobility while maintaining fault tolerance and efficient device control [78]. Additionally, intelligent sensing infrastructures such as the ZigBee-based Intelligent Self-Adjusting Sensor (ZiSAS) system introduce context- and event-aware service adjustments. ZiSAS enables dynamic reconfiguration of middleware, topology, and sensing rates, enhancing responsiveness and optimizing energy consumption in smart homes [79].

Privacy, a critical concern in smart environments, is strengthened by localized processing and policy-enforced migration frameworks. For example, HomePad ensures compliance-aware deployment of services, while BigReduce performs on-site data aggregation, reducing transmission volume without compromising analytical accuracy [80,81]. Meanwhile, blockchain-based models offer secure migration paths for mobile agents, protecting edge services against malicious code and ensuring data integrity during runtime relocation [82]. Moreover, user-centric factors—such as perceived reliability, usability, and system control—are essential for widespread smart home adoption. Studies show that user acceptance of real-time services depends heavily on perceived compatibility, connectedness, and ease of use [83]. Thus, real-time migration strategies should not only optimize technical performance but also align with user expectations and behavior patterns.

In summary, service migration in edge-enabled smart home environments is foundational to achieving intelligent, context-aware, and secure real-time service delivery. It reduces latency, enhances privacy, improves energy efficiency, and supports dynamic user demands—laying the groundwork for resilient and responsive next-generation home ecosystems.

5.3. Smart Manufacturing

Smart manufacturing requires ultra-low latency, reliability, and autonomy to support applications like predictive maintenance, process control, and job-shop scheduling. Traditional cloud-based models often fail to meet these demands due to network delays and centralized bottlenecks. Service migration within edge computing environments addresses this gap by enabling local decision-making and seamless task relocation [84].

In industrial settings, edge-based service migration supports adaptive task assignment across machines, production lines, and local servers. Lin et al. proposed a scheduling framework based on multiclass *DQN*, where services are migrated dynamically between edge nodes and the cloud to optimize makespan and maintain real-time responsiveness [85]. Li et al. [84] introduced a four-layer hybrid architecture that includes edge, cloud, device, and software-defined networking (*SDN*) control planes. Within this setup, a two-phase strategy—combining greedy and threshold-based algorithms—migrates tasks based on latency, queue depth, and processing load. This improves service efficiency and reduces power consumption.

To handle heterogeneous system environments, Sun et al. proposed an AI-enhanced offloading framework that performs intelligent migration considering both latency and service accuracy [86]. Services are selectively migrated to the cloud or retained at edge servers depending on workload profiles and device capabilities. Chen et al. outlined a layered model for smart manufacturing that enables cross-domain service migration (device, network, data, and application). Their architecture supports active maintenance, where services migrate to appropriate execution points based on real-time diagnostics, thereby conserving bandwidth and reducing downtime [87].

In summary, service migration empowers smart manufacturing with agility, reliability, and intelligence. By enabling dynamic relocation of tasks and services, edge computing ensures efficient resource coordination and promotes the integration of AI into industrial workflows.

5.4. Smart Healthcare

The rise of *IoT* has transformed healthcare through smart systems that enable personalized, real-time patient monitoring using multimodal data (e.g., vitals, biomedical waveforms, ambient context) from wearable, ambient, and mobile sensors [88,89]. However, traditional cloud models face challenges in meeting the latency, bandwidth, and privacy demands of these data-intensive workloads. Edge and fog computing mitigate these issues by moving computation closer to data sources, reducing delay, easing network congestion, and improving data governance. In this paradigm, real-time service migration, which is the relocation of application components or processing states across distributed edge resources, is critical to sustaining low latency, *QoS*, and data privacy as patients and workloads move [88,90,91].

Service migration spans both compute-path and data-path agility. *VM* and container placement across edge–fog–cloud tiers should balance the synchronization demands of *body area network (BAN)* data with dynamic pricing and resource constraints. Alam et al. propose an *edge-of-things (EoT)* brokerage model using portfolio optimization and a distributed Alternating Direction Method of Multipliers (*ADMM*)-based provisioning method for latency-sensitive flows [90]. At the extreme edge, constrained medical *IoT*

devices often offload tasks adaptively. Samie et al. address fragmentation in offloading granularity, which reduces gateway efficiency. Their method enhances resource utilization and battery life, informing fine-grained migration policies [91].

Edge-side analytics also drive migration outward from the cloud. Manogaran et al. present a smart wearable patch platform with edge computing and a Bayesian deep learning network to analyze physiological streams locally, supporting early warnings with reduced latency and energy usage [89]. For broader coordination, MEC with 5G enables multi-tier resource allocation in Internet of Medical Things (IoMT) deployments. Ning et al. model a MEC-enabled health monitoring system using a game-theoretic scheduler to jointly allocate computation and communication resources while minimizing a system's cost and energy use [92].

Dynamic workload and patient mobility also necessitate migration. Babbar et al. use SDN to shift services from overloaded to underutilized edge domains [93], while Oueida et al. orchestrate cloud–edge resources to reassign critical clinical staff in real time [94]. Islam et al. propose VM migration via ant colony optimization to follow mobile users across cloudlets, optimizing response time and load balance [95]. Moreover, security remains paramount. Nikoloudakis et al. introduce a *vulnerability assessment as a service* (VAaaS) system that uses fog-enabled SDN to scan, certify, and assign healthcare devices to secure trust zones before migration [96]. Similarly, Chaudhry et al. proposed the Autonomic Zero-knowledge Security Provisioning Mode (AZSPM), a zero-knowledge security model that remotely verifies mobile medical services in fog environments, supporting trustworthy dynamic composition and migration without centralized authorities [97].

In conclusion, real-time service migration is vital for edge–fog-enabled smart healthcare systems. By dynamically relocating services in response to device status, network conditions, user mobility, and clinical priorities—while upholding stringent security and privacy requirements—migration mechanisms play a critical role in meeting the demands of continuous, intelligent patient care [88,90,95].

6. Challenges and Future Directions

6.1. Inaccurate or Delayed Migration Decisions Limit Service Awareness

Recent years have seen rapid advances in service migration strategies in MEC, particularly in latency-critical, mobility-intensive, and resource-constrained environments. Many studies have proposed intelligent migration policies based on RL, trajectory forecasting, or utility-aware orchestration. For instance, Zeng et al. [57] developed a collaborative microservice migration model that leverages RL to minimize application latency across distributed edge clusters. Ma et al. [59] combined DL with proactive placement to anticipate service hotspots and reduce response delay before network saturation occurs. Peng et al. [60] designed a transfer RL approach to optimize migration decisions in vehicular edge computing, achieving low-latency responses under dynamic mobility and traffic loads.

Despite these achievements, migration decisions remain constrained by two key limitations: (i) insufficient timeliness due to reactive or batch-based migration mechanisms and (ii) limited observability of real-time network and user states, particularly in distributed or large-scale systems. As Xu et al. [61] pointed out in their PDMA framework, the probabilistic models that account for trajectory uncertainty may trigger delayed migrations when real-world mobility deviates from predictions. Furthermore, context-aware models such as the one proposed by Saha et al. [63] rely on accurate link quality estimation and temporal profiling, which are difficult to achieve in highly heterogeneous environments with fluctuating workloads. The consequences of delayed or inaccurate migration decisions include service degradation, increased handoff latency, redundant migrations, and suboptimal resource utilization. More critically, such inefficiencies compound in ultra-dense or

dynamic edge networks, where service demands are transient and compute availability fluctuates frequently.

To address these challenges, future work should focus on integrating fine-grained sensing and telemetry data—such as per-node queue delays, user trajectory entropy, and real-time hardware load—into migration triggers. Cross-layer designs that combine application-level service awareness with network-level congestion monitoring may help in building more responsive and predictive migration engines. Moreover, edge-native learning models, such as lightweight graph neural networks (GNNs) or federated meta-learning, could support adaptive migration across heterogeneous edge nodes while respecting local constraints.

6.2. Scheduling and Coordination Challenges in Large-Scale Heterogeneous Edge Networks

As edge computing ecosystems scale rapidly with the proliferation of *IoT*, vehicular nodes, *UAVs*, and micro-data centers, service migration in large-scale heterogeneous edge networks becomes increasingly complex. Efficient scheduling and coordination are essential to ensure that migration decisions can adapt to varying resource capacities, user demands, and service interdependencies across distributed edge domains.

Recent studies have explored scalable migration frameworks in multi-tier and multi-domain edge environments. For example, Zeng et al. [57] proposed a collaborative microservice migration strategy that integrates *DRL* with network flow optimization to reduce service response time in multi-cluster edge infrastructures. Similarly, Xu et al. [61] developed the *PDMA* framework, which introduces probabilistic decision-making for latency- and mobility-aware migration in vehicular networks, considering local context uncertainties and edge node load dynamics.

However, these approaches often face limitations when deployed in ultra-large-scale or resource-heterogeneous edge networks. The lack of global state visibility across domains, inconsistent node capabilities, and asynchronous resource changes can lead to suboptimal migration paths, redundant migrations, or service interruptions. Coordination complexity further increases when services are tightly coupled or require consistent state synchronization across edge clusters.

For instance, Chi et al. [62] highlighted that in ultra-large edge systems, optimal migration planning requires multi-criteria decisions that account for network congestion, node reliability, and delay variance. Yet, determining these decisions in real-time under incomplete knowledge remains computationally intensive and often infeasible. Similarly, Kim et al. [98] emphasized the difficulty in synchronizing stateful service migrations in mobile scenarios, especially when user mobility causes frequent shifts between edge domains.

Moreover, existing schedulers are often domain-specific and fail to generalize across different orchestration layers (e.g., from device edge to access edge or regional edge). This leads to fragmented coordination and inefficient resource utilization. While efforts such as hierarchical orchestration have been proposed, they introduce new bottlenecks in terms of scalability and decision propagation latency [63].

Future work should focus on building federated and context-aware migration controllers that support cross-domain coordination through lightweight communication and distributed decision-making. Leveraging meta-reinforcement learning, as explored by Niu et al. [54], offers a promising direction for adaptive policy generalization across heterogeneous conditions. Additionally, standardizing service descriptors and inter-node negotiation protocols may reduce coordination overhead and improve scheduling fairness across edge ecosystems.

6.3. Lack of Comprehensive Security and Privacy Protection During Service Migration

While service migration improves flexibility and responsiveness in MEC, it also introduces critical security and privacy challenges—particularly during the transfer and reallocation of service states. These risks become even more severe when migration occurs across heterogeneous nodes or administrative domains, where differing security policies and limited mutual trust significantly raise the likelihood of data leakage or unauthorized access.

To address these issues, various strategies have been proposed in recent years. For example, Yuan et al. [99] introduced a migration framework based on *trusted execution environments (TEEs)*, which allows remote integrity verification and encrypted data transfer using a virtual root of trust. Similarly, Pahl et al. [100] designed a blockchain-based migration architecture to maintain tamper-proof logs and decentralized access control, enhancing trust between unverified nodes. For privacy concerns, Abdrabou et al. [101] applied lightweight anonymization techniques to reduce personal data exposure during orchestration. Khalil et al. [102] used *FL* to preserve user behavior privacy while predicting service quality, demonstrating the importance of privacy at the inference stage.

However, these solutions are often limited to single-domain environments or assume a unified trust model. When services are migrated across domains, especially in real-time scenarios, additional mechanisms are required to manage inter-domain trust and policy heterogeneity. In response, Zheng et al. [103] proposed a trust-aware migration strategy where edge nodes are evaluated based on a dynamic trust score. Services are directed only to nodes that meet minimum trust thresholds, even if other options offer better latency or energy efficiency. Minmin et al. [104] developed a multi-ledger system using blockchain and federated identity management. Their approach enables anonymized service migration records and enforces domain-specific access rules through smart contracts. Moreover, identity-based authentication protocols [105] have been designed to enable mutual authentication across domains in real time without exposing sensitive credentials.

Despite these advancements, several critical challenges persist. Most current orchestrators still prioritize performance metrics like latency and throughput, while overlooking dynamic trust evaluation. Lightweight secure computing technologies such as Intel Software Guard Extensions (*ISGs*) or Advanced RISC Machines Ltd (*ARM*) TrustZone are not yet widely deployed at the network edge, limiting their applicability in real-time use cases. In addition, ensuring that application states are encrypted during transfer—without degrading responsiveness—remains an unresolved issue, especially for repeated or continuous migrations.

To move forward, future research should focus on four key areas: (i) dynamic, trust-aware migration planning that accounts for domain-level reputation; (ii) low-overhead protocols for cross-domain attestation and logging; (iii) co-optimization of privacy protection and service performance; and (iv) embedding domain-sensitive privacy policies directly into service scheduling and placement algorithms.

6.4. Lack of Adaptive and Context-Aware Autonomous Migration Mechanisms

Despite recent advancements in service migration strategies, many approaches still rely on static or rule-based triggers (e.g., load thresholds, mobility events), which fail to adapt dynamically to the complex, volatile, and context-rich environments of edge computing. This inflexibility often leads to suboptimal migration decisions, delayed reactions to environmental changes, and unnecessary resource consumption.

To improve adaptability, recent research has explored machine learning and *RL* techniques for autonomous migration. For instance, Kim et al. [98] introduces an adaptive migration engine that incorporates user mobility prediction and real-time environmental feedback to determine optimal migration timing and targets. Likewise, Xu et al. [61] pro-

posed *PDMA*, a probabilistic delay- and mobility-aware model that leverages real-time trajectories and delay metrics to make migration decisions under uncertainty. These efforts demonstrate the potential of intelligent migration mechanisms, yet most are still limited to specific use cases or environmental assumptions.

The lack of comprehensive context awareness remains a critical limitation. While some frameworks consider mobility and latency, they often overlook additional contextual signals such as link quality, service criticality, energy constraints, or user intent. Saha et al. [63] highlight the benefits of integrating contextual information into migration scheduling, including temporal workload trends and node reliability, but their framework still requires manual configuration for policy tuning. Moreover, heterogeneous edge nodes often differ significantly in processing power, network interfaces, energy reserves, and security capabilities—factors that are rarely jointly optimized in current systems.

Furthermore, achieving true autonomy requires service migration engines to operate with minimal external supervision while still ensuring system-wide coordination and compliance with global policies. Chi et al. [62] address part of this challenge by proposing a multi-criteria migration model for ultra-large-scale edge networks, which balances migration costs with latency and availability. However, these models often depend on centralized orchestration or complete network observability, which are infeasible in highly distributed and privacy-sensitive environments.

To address these challenges, future research must focus on designing fully autonomous, adaptive, and context-aware service migration frameworks. These frameworks should incorporate edge-native learning models capable of self-adjustment, handle partial observability via local inference and collaboration, and dynamically reconfigure migration policies based on evolving environmental and user-centric factors. The integration of federated learning, explainable AI, and semantic context modeling could offer new avenues to enhance adaptability without compromising transparency or efficiency.

6.5. Challenges and Opportunities of AI-Driven Service Migration

The integration of AI into service migration has shown promising potential to enhance the adaptability, responsiveness, and autonomy of edge computing systems. Particularly in highly dynamic environments such as *vehicular edge computing (VEC)*, AI can support real-time migration decisions that minimize latency and optimize resource allocation [106]. However, the deployment of AI-driven migration also introduces several technical challenges that merit further exploration.

Recent studies apply *DRL* and multi-agent systems to handle migration and task offloading under mobility and network uncertainty. For instance, Gao et al. [107] proposed a *DRL*-based service migration approach for *VEC* that jointly considers vehicle location, velocity, and delay constraints. Their *variational recurrent and critic-coached* strategy that reuses the *actor-critic (VRCCS-AC)* model demonstrates high adaptability in latency-critical traffic scenarios. Similarly, Malektaji et al. [108] introduced a *DRL*-enabled content migration strategy in *VEC*, achieving low migration cost and stable *QoS* by dynamically adjusting caching decisions and migration timing.

Despite these advancements, several challenges persist. One key issue is the scalability of AI models across large-scale and heterogeneous edge networks. Brandherm et al. [109] introduced *BigMEC*, a scalable migration system that combines *RL* and task grouping to support distributed migration at the mobile edge. While it achieves promising performance in emulated testbeds, its generalization to real-time environments with volatile node availability remains an open question. Another fundamental challenge lies in deploying AI models within constrained edge infrastructures. Al-Doghman et al. [110] emphasized the need for lightweight microservice architectures that support secure and flexible AI deployment

under hardware constraints. Their architecture facilitates fine-grained migration while embedding policy-driven control at the orchestration layer, a critical requirement for real-time AI inferencing on heterogeneous devices. Additionally, ensuring fairness, transparency, and coordination in multi-tenant edge systems is gaining importance. Zhang et al. [111] proposed the *edge as a service (EaaS)* framework, which emphasizes service-orientation and distributed intelligence. By supporting elastic service deployment and runtime migration under *SLA* constraints, it demonstrates how policy-awareness and AI integration can improve decision quality and fairness in shared environments.

Looking ahead, several opportunities can be identified. First, federated *RL* can mitigate data transfer costs and support privacy-preserving migration optimization across domains. Second, leveraging topology-aware models such as graph neural networks may enhance real-time awareness of migration contexts. Lastly, incorporating explainable AI techniques could improve trust and operational transparency in mission-critical edge applications.

6.6. Migration in the Age of 6G: Ultra-Dense and High-Mobility Networks

As 6G networks evolve to meet the demands of ultra-dense connectivity, ubiquitous intelligence, and extreme mobility, service migration strategies must be reimagined to operate in environments characterized by high user density, dynamic topologies, and stringent latency requirements [112]. In contrast to 5G systems, where edge service migration primarily addressed mobility within localized domains, 6G introduces new architectural paradigms—such as *integrated sensing and communications (ISACs)*, holographic beamforming, and AI-native orchestration—which drastically alter the landscape of edge computing and service management [113].

One of the key challenges in 6G-enabled migration is maintaining *ultra-reliable low-latency communication (URLLC)* under high-speed mobility. As noted by Dritsas et al. [114], conventional migration schemes based on periodic monitoring or static thresholds are insufficient for vehicular or drone-based edge nodes. To address this, the authors propose an *RL* framework that proactively adapts migration policies based on dynamic context inference. This model captures not only mobility patterns but also anticipates handover failures and delay jitter in heterogeneous networks. In addition, the scale and density of 6G deployments call for highly decentralized and scalable migration mechanisms. Ming et al. [115] highlighted the role of federated edge AI in supporting massive user participation in metaverse applications. They emphasize that migration in 6G should accommodate cooperative learning, privacy-preserving inference, and device heterogeneity, all while ensuring uninterrupted service delivery. As 6G edge nodes serve immersive and interactive applications, delay and synchronization become critical migration constraints. Moreover, network slicing and service function chaining are integral to 6G service delivery. Letaief et al. [116] proposed a federated *DRL* approach to proactively migrate service chains based on predicted slice mobility patterns. This ensures that end-to-end latency and *QoS* constraints are met across distributed network slices without centralized orchestration overhead. From a system-level perspective, collaborative intelligence between cloud, edge, and device tiers is becoming a central theme. Duan et al. [112] provide a comprehensive survey of *FL* embedded in edge infrastructures to facilitate ubiquitous intelligence. The synergy between *FL* and edge migration in 6G allows services to not only follow users across space but also evolve models continuously without relying on centralized data aggregation.

In summary, migration in 6G networks should go beyond simple workload displacement and adopt context-aware, learning-enabled, and federated paradigms to cope with ultra-dense, mobile, and latency-sensitive environments. Future work should focus on cross-layer migration protocols, explainable AI for migration policy interpretation, and energy-aware orchestration frameworks suitable for the 6G era.

6.7. Toward Sustainable and Energy-Aware Service Migration

With growing focus on energy efficiency and carbon awareness, sustainable service migration is becoming a key research direction in edge computing. In energy-constrained edge environments—such as *IoT* gateways, *RSUs*, and micro-data centers—migration strategies should not only ensure performance goals like latency and availability but also minimize the environmental footprint [13].

Recent works have proposed energy-aware frameworks for collaborative end-edge-cloud systems. Li et al. [13] developed an online migration mechanism that jointly considers container layering, data transmission overhead, and runtime energy consumption. Their results show that energy-efficient migration can reduce access latency and improve service success rates under bursty loads. Similarly, Sun et al. [117] presented a carbon-aware migration model that dynamically schedules services across heterogeneous nodes based on carbon intensity and energy availability, contributing to greener systems. *FL* has also emerged as a promising paradigm for sustainable migration. Chen et al. [118] introduced an *FL*-based *DRL* scheme that optimizes migration decisions with traffic prediction, reducing redundant migrations and unnecessary energy usage. Their approach enables localized learning without centralized training overhead, aligning well with sustainability goals. From a system architecture perspective, renewable-aware migration strategies are gaining traction. Alhartomi et al. [119] integrated solar energy prediction into migration triggers, allowing energy-harvesting edge nodes to defer or attract migrations based on predicted power surplus. This aligns with the broader vision of self-sustainable edge infrastructures. Furthermore, Heng et al. [120] reviewed cloud-edge service placement strategies with energy optimization in *IoT* communications, highlighting the need for multi-objective trade-offs between latency, energy, and *SLA* compliance.

One challenge lies in the heterogeneity of energy metrics across devices and geographic regions. As shown by Belkacem et al. [121], integrating local energy prices, renewable availability, and device aging into migration planning requires a cross-layer orchestration model. Their work advocates for predictive migration augmented with external energy signals and decentralized scheduling.

In conclusion, sustainable service migration calls for holistic optimization of energy, performance, and carbon goals. Future research should explore joint design of energy telemetry, carbon-aware AI triggers, and multi-objective *FL* frameworks, enabling edge systems to meet the twin imperatives of responsiveness and environmental responsibility.

6.8. Comparison and Discussion of Real-Time Service Migration Approaches

To provide a clearer picture of the capabilities and limitations of state-of-the-art migration strategies, we conduct a structured comparison of representative papers, as summarized in Table 2. Each method is described in terms of its key technique and qualitatively analyzed through its advantages and disadvantages. These reflect critical aspects such as application scenarios, latency performance, system complexity, and scalability.

Techniques utilizing *RL* (e.g., [35,57]) or predictive optimization frameworks (e.g., [55,59]) exhibit strong capabilities in latency control and adaptability. However, they often introduce high model complexity and demand large-scale training data or precise trajectory prediction, which may hinder real-time applicability in dynamic environments.

Works such as [60,122] target vehicular or heterogeneous edge scenarios by incorporating domain transfer learning or energy-aware policies. These strategies demonstrate promising adaptability under fluctuating conditions but often rely on pretrained models and stable connectivity, limiting their robustness in highly volatile edge contexts.

Table 2. Comparison of research papers on service migration.

Paper	Key Technique	Advantages	Disadvantages
[55]	Predictive trajectory-aware migration; <i>PSO</i> algorithm	Simultaneous reduction in delay, energy consumption, and load fluctuation; fault-tolerant replication for improved service reliability	Assumes accurate trajectory prediction, which may not hold in highly dynamic environments; <i>PSO</i> -based model increases computational complexity
[35]	<i>DDPG</i> for joint optimization	Learns optimal migration and mobility policies jointly; considers dynamic network states; improves latency and service continuity	Requires large training data and time; model interpretability is low; not optimized for energy consumption or load balance
[122]	Energy-aware service migration based on <i>DQN</i>	Enables dynamic migration across base stations; balances energy efficiency with delay	Assumes stable wireless link during migration; performance sensitive to <i>DQN</i> training quality
[60]	Transfer <i>RL</i> with communication and computation-aware reward function	Reduces migration cost via knowledge transfer from similar tasks; adapts to dynamic environments; jointly considers bandwidth and processing load	Requires pretrained source domain model; performance degrades if task similarity across domains is low
[49]	Asynchronous <i>FL</i> with mobility-aware caching	Reduces model aggregation delay; improves content hit ratio and personalization	Assumes accurate mobility prediction; training stability can be affected by stragglers and delayed updates
[59]	Predictive service placement using Lyapunov optimization	Achieves low latency through multi-timescale planning; considers both delay and reliability	Requires accurate mobility prediction; model complexity increases with prediction window size
[37]	Time-segmented multi-level reconfiguration using cloud–edge collaboration and edge clustering	Adapts to fluctuating loads via hierarchical edge–cloud scheduling; minimizes load shedding under bursty traffic	Relies on predefined time segmentation granularity; scalability limited by edge clustering overhead
[123]	Hierarchical cloud–edge–end collaborative inference with model partitioning, pruning, and early exiting	Reduces end-to-end latency via adaptive offloading and early exit; maintains robustness under dynamic network/device conditions	Requires accurate confidence estimation for early exit; system orchestration complexity increases with scale
[57]	Collaborative and latency-aware microservice migration based on delay-aware cost graph optimization and <i>DQN</i> learning	Reduces end-to-end latency via joint service chain optimization; improves responsiveness under dynamic workloads; enhances adaptability through <i>RL</i> -based policy refinement	High graph construction and optimization overhead; sensitive to profiling accuracy; scalability challenged in large service meshes
[124]	Context-aware neural configuration adaptation using latency-accuracy profiling and lightweight scheduling	Enables adaptive configuration migration with high success rate; balances inference latency and accuracy under mobility	Requires offline profiling for configuration library; targeted for video analytics and less generalizable to other domains
[125]	Two-stage mobility-aware edge server placement based on <i>NSGA-II</i> offline optimization and game-theoretic online control	Balances long-term server deployment efficiency with short-term load changes; achieves low-latency query resolution under mobility; supports incremental remapping	Only optimizes placement and not runtime migration; relies on historical mobility data and tuned parameters

A few approaches (e.g., [37,123]) adopt multi-layer cloud–edge–end orchestration to improve resource utilization and reduce overload. While these solutions provide system-level coordination and granularity control, their scalability is constrained by synchronization overhead and model orchestration complexity, especially under large-scale deployments.

Some works specialize in AR/VR or video analytics applications (e.g., [124]) by designing task-specific migration strategies such as neural configuration adaptation. These achieve high success rates and low latency within targeted use cases but offer limited generalizability.

It is noteworthy that methods such as [125] focus primarily on optimal initial placement rather than runtime migration. Although they enhance responsiveness in user-mobility scenarios, they may fall short in coping with real-time migration demands where service state transfer and session continuity are critical.

In conclusion, there is no one-size-fits-all solution for real-time service migration. Approaches must carefully balance trade-offs among latency, complexity, robustness, and application fit. Recent works indicate a trend toward hybrid designs that combine lightweight decision-making modules with system-level orchestration, aiming to enable adaptive and scalable service delivery across heterogeneous edge environments.

7. Conclusions

In this survey, we presented a comprehensive and structured review of real-time service migration in edge networks. We began by clarifying the concept of service migration and outlining its importance as a foundational mechanism in modern edge environments. Through a layered cloud–edge–device architecture and a unified modeling framework, we described how migration decisions are guided by system-level objectives such as latency reduction, energy efficiency, resource balancing, and service continuity. Specifically, the unified modeling framework refers to the integration of five models—network, latency, energy, utility, and resource utilization—which together form a consistent basis for analyzing and guiding migration strategies. Moreover, we identified four primary motivations that drive service migration, ensuring low-latency responses in real-time scenarios such as autonomous driving, smart cities, and industrial control. These motivations reflect the critical needs of both users and providers in increasingly dynamic and heterogeneous edge scenarios. To support these objectives, we reviewed six key enabling techniques across multiple categories. We further analyzed advanced approaches like *DRL* and multi-agent coordination, which offer promising solutions for migration in complex, large-scale edge environments. Four representative application scenarios demonstrate the practical significance of real-time service migration. Finally, we highlighted several open research challenges that limit current service migration strategies. Addressing these challenges requires the development of scalable, intelligent, and adaptive migration frameworks that can respond proactively to real-time dynamics in edge environment. We hope that this survey offers a solid foundation for advancing robust, real-time service migration in edge computing.

Author Contributions: Conceptualization, Y.Z. and K.Z.; methodology, Y.Z.; software, K.Z.; validation, Y.Y., Y.Z. and K.Z.; formal analysis, K.Z.; investigation, Y.Z.; resources, Y.Y.; data curation, K.Z.; writing—original draft preparation, Y.Z. and K.Z.; writing—review and editing, Y.Y. and Z.Z.; visualization, Y.Z. and K.Z.; supervision, Y.Y. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grants No. 62372420, 62402460), in part by the National Science and Technology Major Project (Grant No. 2024ZD1001900), in part by the China Geological Survey (CGS) work project: “Geoscience literature knowledge services and decision supporting” (Grant No. DD20230139), and in part by the Fundamental Research Funds for the Central Universities (Grants No. 2652023001, 2652023063).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Prangon, N.F.; Wu, J. AI and computing horizons: Cloud and edge in the modern era. *J. Sens. Actuator Netw.* **2024**, *13*, 44. [\[CrossRef\]](#)
2. Sodiya, E.O.; Umoga, U.J.; Obaigbena, A.; Jacks, B.S.; Ugwuanyi, E.D.; Daraojimba, A.I.; Lottu, O.A. Current state and prospects of edge computing within the Internet of Things (IoT) ecosystem. *Int. J. Sci. Res. Arch.* **2024**, *11*, 1863–1873. [\[CrossRef\]](#)
3. Akhlaqi, M.Y.; Hanapi, Z.B.M. Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions. *J. Netw. Comput. Appl.* **2023**, *212*, 103568. [\[CrossRef\]](#)
4. Mao, B.; Liu, J.; Wu, Y.; Kato, N. Security and privacy on 6G network edge: A survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1095–1127. [\[CrossRef\]](#)
5. Naseh, D.; Shinde, S.S.; Tarchi, D. Network sliced Distributed Learning-as-a-Service for Internet of Vehicles applications in 6G non-terrestrial network scenarios. *J. Sens. Actuator Netw.* **2024**, *13*, 14. [\[CrossRef\]](#)
6. Golpayegani, F.; Chen, N.; Afraz, N.; Gyamfi, E.; Malekjafarian, A.; Schäfer, D.; Krupitzer, C. Adaptation in edge computing: A review on design principles and research challenges. *ACM Trans. Auton. Adapt. Syst.* **2024**, *19*, 1–43. [\[CrossRef\]](#)
7. Ma, L.; Cheng, N.; Zhou, C.; Wang, X.; Lu, N.; Zhang, N.; Aldubaikhy, K.; Alqasir, A. Dynamic neural network-based resource management for mobile edge computing in 6G networks. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *10*, 953–967. [\[CrossRef\]](#)
8. Ramamoorthi, V. Real-Time Adaptive Orchestration of AI Microservices in Dynamic Edge Computing. *J. Adv. Comput. Syst.* **2023**, *3*, 1–9. [\[CrossRef\]](#)
9. Vaño, R.; Lacalle, I.; Sowiński, P.; S-Julián, R.; Palau, C.E. Cloud-native workload orchestration at the edge: A deployment review and future directions. *Sensors* **2023**, *23*, 2215. [\[CrossRef\]](#)
10. Iftikhar, S.; Ahmad, M.M.M.; Tuli, S.; Chowdhury, D.; Xu, M.; Gill, S.S.; Uhlig, S. HunterPlus: AI based energy-efficient task scheduling for cloud-fog computing environments. *Internet Things* **2023**, *21*, 100667. [\[CrossRef\]](#)
11. Alotaibi, B. A survey on industrial Internet of Things security: Requirements, attacks, AI-based solutions, and edge computing opportunities. *Sensors* **2023**, *23*, 7470. [\[CrossRef\]](#)
12. Zhang, X.; Debroy, S. Resource management in mobile edge computing: A comprehensive survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. [\[CrossRef\]](#)
13. Li, J.; Zhao, D.; Shi, Z.; Meng, L.; Gaaloul, W.; Zhou, Z. Energy-Efficient Online Service Migration in Edge Networks. *IEEE Internet Things J.* **2024**, *11*, 29689–29708. [\[CrossRef\]](#)
14. Bahrami, B.; Khayyambashi, M.R.; Mirjalili, S. Edge server placement problem in multi-access edge computing environment: Models, techniques, and applications. *Clust. Comput.* **2023**, *26*, 3237–3262. [\[CrossRef\]](#)
15. Ray, K.; Banerjee, A.; Narendra, N.C. Proactive microservice placement and migration for mobile edge computing. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 12–14 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 28–41.
16. Chiang, Y.; Zhang, Y.; Luo, H.; Chen, T.Y.; Chen, G.H.; Chen, H.T.; Wang, Y.J.; Wei, H.Y.; Chou, C.T. Management and orchestration of edge computing for IoT: A comprehensive survey. *IEEE Internet Things J.* **2023**, *10*, 14307–14331. [\[CrossRef\]](#)
17. Ullah, R.; Wu, D.; Harvey, P.; Kilpatrick, P.; Spence, I.; Varghese, B. FedFly: Toward migration in edge-based distributed federated learning. *IEEE Commun. Mag.* **2022**, *60*, 42–48. [\[CrossRef\]](#)
18. Shen, Y.; Shen, S.; Li, Q.; Zhou, H.; Wu, Z.; Qu, Y. Evolutionary privacy-preserving learning strategies for edge-based IoT data sharing schemes. *Digit. Commun. Netw.* **2023**, *9*, 906–919. [\[CrossRef\]](#)
19. Wu, L.; Sun, P.; Wang, Z.; Li, Y.; Yang, Y. Computation offloading in multi-cell networks with collaborative edge-cloud computing: A game theoretic approach. *IEEE Trans. Mob. Comput.* **2023**, *23*, 2093–2106. [\[CrossRef\]](#)
20. Peng, Y.; Liu, L.; Zhou, Y.; Shi, J.; Li, J. Deep reinforcement learning-based dynamic service migration in vehicular networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
21. Boulougaris, G.; Kolomvatsos, K. An inference mechanism for proactive service migration at the edge. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 4505–4516. [\[CrossRef\]](#)
22. Labrijj, I.; Meneghello, F.; Cecchinato, D.; Sesia, S.; Perraud, E.; Strinati, E.C.; Rossi, M. Mobility aware and dynamic migration of MEC services for the Internet of Vehicles. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 570–584. [\[CrossRef\]](#)

23. Chen, Z.; Huang, S.; Min, G.; Ning, Z.; Li, J.; Zhang, Y. Mobility-aware Seamless Service Migration and Resource Allocation in Multi-edge IoV Systems. *IEEE Trans. Mob. Comput.* **2025**, *24*, 6315–6332. [\[CrossRef\]](#)
24. Li, H.; Chen, Y.; Yang, Y.; Huang, J. Revenue-optimal contract design for content providers in IoT-edge caching. *IEEE Internet Things J.* **2024**, *11*, 23497–23508. [\[CrossRef\]](#)
25. Wang, H.; Lv, T.; Lin, Z.; Zeng, J. Energy-delay minimization of task migration based on game theory in MEC-assisted vehicular networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 8175–8188. [\[CrossRef\]](#)
26. Liu, J.; Wang, S.; Xu, H.; Xu, Y.; Liao, Y.; Huang, J.; Huang, H. Federated learning with experience-driven model migration in heterogeneous edge networks. *IEEE/ACM Trans. Netw.* **2024**, *32*, 3468–3484. [\[CrossRef\]](#)
27. Rui, L.; Zhang, M.; Gao, Z.; Qiu, X.; Wang, Z.; Xiong, A. Service migration in multi-access edge computing: A joint state adaptation and reinforcement learning mechanism. *J. Netw. Comput. Appl.* **2021**, *183*, 103058. [\[CrossRef\]](#)
28. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 223–236. [\[CrossRef\]](#)
29. Li, X.; Zhou, Z.; He, Q.; Shi, Z.; Gaaloul, W.; Yangui, S. Re-Scheduling IoT Services in Edge Networks. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 3233–3246. [\[CrossRef\]](#)
30. Li, X.; Zhou, Z.; Zhu, C.; Shu, L.; Zhou, J. Online Reconfiguration of Latency-Aware IoT Services in Edge Networks. *IEEE Internet Things J.* **2022**, *9*, 17035–17046. [\[CrossRef\]](#)
31. Choi, K.; Soma, R.; Pedram, M. Dynamic voltage and frequency scaling based on workload decomposition. In Proceedings of the 2004 International Symposium on Low Power Electronics and Design, New York, NY, USA, 9–11 August 2004; ISLPED '04, pp. 174–179. [\[CrossRef\]](#)
32. Wang, S.; Xu, J.; Zhang, N.; Liu, Y. A Survey on Service Migration in Mobile Edge Computing. *IEEE Access* **2018**, *6*, 23511–23528. [\[CrossRef\]](#)
33. Wang, S.; Urgaonkar, R.; Zafer, M.; He, T.; Chan, K.; Leung, K.K. Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1272–1288. [\[CrossRef\]](#)
34. Liu, F.; Yu, H.; Huang, J.; Taleb, T. Joint service migration and resource allocation in edge IoT system based on deep reinforcement learning. *IEEE Internet Things J.* **2023**, *11*, 11341–11352. [\[CrossRef\]](#)
35. Yuan, Q.; Li, J.; Zhou, H.; Lin, T.; Luo, G.; Shen, X. A Joint Service Migration and Mobility Optimization Approach for Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9041–9052. [\[CrossRef\]](#)
36. Chen, M.; Li, W.; Fortino, G.; Hao, Y.; Hu, L.; Humar, I. A Dynamic Service Migration Mechanism in Edge Cognitive Computing. *ACM Trans. Internet Technol.* **2019**, *19*, 1–15. [\[CrossRef\]](#)
37. Gao, H.; Ma, W.; He, S.; Wang, L.; Liu, J. Time-segmented multi-level reconfiguration in distribution network: A novel cloud-edge collaboration framework. *IEEE Trans. Smart Grid* **2022**, *13*, 3319–3322. [\[CrossRef\]](#)
38. Liang, Z.; Liu, Y.; Lok, T.M.; Huang, K. Multi-cell mobile edge computing: Joint service migration and resource allocation. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5898–5912. [\[CrossRef\]](#)
39. Yang, L.; Jia, J.; Lin, H.; Cao, J. Reliable dynamic service chain scheduling in 5G networks. *IEEE Trans. Mob. Comput.* **2022**, *22*, 4898–4911. [\[CrossRef\]](#)
40. Kaur, K.; Guillemin, F.; Sailhan, F. Container placement and migration strategies for cloud, fog, and edge data centers: A survey. *Int. J. Netw. Manag.* **2022**, *32*, e2212. [\[CrossRef\]](#)
41. Li, C.; Zhang, Q.; Luo, Y. A jointly non-cooperative game-based offloading and dynamic service migration approach in mobile edge computing. *Knowl. Inf. Syst.* **2023**, *65*, 2187–2223. [\[CrossRef\]](#)
42. Caro, M.; Tabani, H.; Abella, J.; Moll, F.; Morancho, E.; Canal, R.; Altet, J.; Calomarde, A.; Cazorla, F.J.; Rubio, A.; et al. An automotive case study on the limits of approximation for object detection. *J. Syst. Archit.* **2023**, *138*, 102872. [\[CrossRef\]](#)
43. Bozkaya-Aras, E. Optimizing Service Migration in IoT Edge Networks: Digital Twin-Based Computation and Energy-Efficient Approach. In Proceedings of the 2025 IEEE Wireless Communications and Networking Conference (WCNC), Milan, Italy, 24–27 March 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.
44. Jia, Y.; Zhang, C.; Huang, Y.; Zhang, W. Lyapunov optimization based mobile edge computing for Internet of Vehicles systems. *IEEE Trans. Commun.* **2022**, *70*, 7418–7433. [\[CrossRef\]](#)
45. Maleki, E.F.; Mashayekhy, L.; Nabavinejad, S.M. Mobility-aware computation offloading in edge computing using machine learning. *IEEE Trans. Mob. Comput.* **2021**, *22*, 328–340. [\[CrossRef\]](#)
46. Filiposka, S.; Mishev, A.; Gilly, K. Mobile-aware dynamic resource management for edge computing. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3626. [\[CrossRef\]](#)
47. Liu, J.; Wu, Z.; Liu, J.; Tu, X. Distributed location-aware task offloading in multi-UAVs enabled edge computing. *IEEE Access* **2022**, *10*, 72416–72428. [\[CrossRef\]](#)
48. Zhang, P.; Zhang, Y.; Dong, H.; Jin, H. Mobility and dependence-aware QoS monitoring in mobile edge computing. *IEEE Trans. Cloud Comput.* **2021**, *9*, 1143–1157. [\[CrossRef\]](#)

49. Wang, W.; Zhao, Y.; Wu, Q.; Fan, Q.; Zhang, C.; Li, Z. Asynchronous federated learning based mobility-aware caching in vehicular edge computing. In Proceedings of the 2022 14th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 14–17 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
50. Wu, Q.; Zhao, Y.; Fan, Q.; Fan, P.; Wang, J.; Zhang, C. Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning. *IEEE J. Sel. Top. Signal Process.* **2022**, *17*, 66–81. [\[CrossRef\]](#)
51. Zhou, X.; Ge, S.; Qiu, T.; Li, K.; Atiquzzaman, M. Energy-efficient service migration for multi-user heterogeneous dense cellular networks. *IEEE Trans. Mob. Comput.* **2021**, *22*, 890–905. [\[CrossRef\]](#)
52. Toumi, N.; Bagaa, M.; Ksentini, A. Machine learning for service migration: A survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1991–2020. [\[CrossRef\]](#)
53. Ning, Z.; Chen, H.; Ngai, E.C.; Wang, X.; Guo, L.; Liu, J. Lightweight imitation learning for real-time cooperative service migration. *IEEE Trans. Mob. Comput.* **2023**, *23*, 1503–1520. [\[CrossRef\]](#)
54. Niu, L.; Chen, X.; Zhang, N.; Zhu, Y.; Yin, R.; Wu, C.; Cao, Y. Multiagent meta-reinforcement learning for optimized task scheduling in heterogeneous edge computing systems. *IEEE Internet Things J.* **2023**, *10*, 10519–10531. [\[CrossRef\]](#)
55. Ma, Y.; Dai, M.; Xia, Y.; Liu, Z.; Shao, S.; Zhao, H.; Li, G.; Tang, Y.; Niu, X. A Novel Approach to Predictive-Trajectory-Aware Service Migration in Edge Computing. In Proceedings of the 2023 11th International Conference on Information Systems and Computing Technology (ISCTech), Qingdao, China, 30 July–1 August 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 40–45.
56. Toczé, K.; Nadjm-Tehrani, S. Energy-aware Distributed Microservice Request Placement at the Edge. *arXiv* **2024**, arXiv:2408.13748. [\[CrossRef\]](#)
57. Zeng, L.; Zhang, C.; Wang, Z.; Du, H.; Jia, X. Towards Collaborative and Latency-Aware Microservice Migration in Mobile Edge Computing. *IEEE Internet Things J.* **2025**, *12*, 25286–25299. [\[CrossRef\]](#)
58. Zhang, W.; Hu, Y.; Zhang, Y.; Raychaudhuri, D. Segue: Quality of service aware edge cloud service migration. In Proceedings of the 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, 12–15 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 344–351.
59. Ma, H.; Zhou, Z.; Chen, X. Leveraging the power of prediction: Predictive service placement for latency-sensitive mobile edge computing. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6454–6468. [\[CrossRef\]](#)
60. Peng, Y.; Tang, X.; Zhou, Y.; Li, J.; Qi, Y.; Liu, L.; Lin, H. Computing and communication cost-aware service migration enabled by transfer reinforcement learning for dynamic vehicular edge computing networks. *IEEE Trans. Mob. Comput.* **2022**, *23*, 257–269. [\[CrossRef\]](#)
61. Xu, M.; Zhou, Q.; Wu, H.; Lin, W.; Ye, K.; Xu, C. PDMA: Probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing. *Softw. Pract. Exp.* **2022**, *52*, 394–414. [\[CrossRef\]](#)
62. Chi, H.R.; Silva, R.; Santos, D.; Quevedo, J.; Corujo, D.; Abboud, O.; Radwan, A.; Hecker, A.; Aguiar, R.L. Multi-criteria dynamic service migration for ultra-large-scale edge computing networks. *IEEE Trans. Ind. Inform.* **2023**, *19*, 11115–11127. [\[CrossRef\]](#)
63. Saha, S.; Perumal, I.; Abbas, M.; Manimozhi, I.; Bhat, C.R. Contextual Information Based Scheduling for Service Migration in Mobile Edge Computing. *Int. J. Comput. Commun. Control.* **2024**, *19*. [\[CrossRef\]](#)
64. Zhao, Q.; Li, C. Two-Stage Multi-Swarm Particle Swarm Optimizer for Unconstrained and Constrained Global Optimization. *IEEE Access* **2020**, *8*, 124905–124927. [\[CrossRef\]](#)
65. Chen, L.; Zhou, P.; Gao, L.; Xu, J. Adaptive Fog Configuration for the Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4656–4664. [\[CrossRef\]](#)
66. Zhang, Y.; Zhou, Z.; Shi, Z.; Meng, L.; Zhang, Z. Online Scheduling Optimization for DAG-Based Requests Through Reinforcement Learning in Collaboration Edge Networks. *IEEE Access* **2020**, *8*, 72985–72996. [\[CrossRef\]](#)
67. Sun, M.; Zhou, Z.; Xue, X.; Gaaloul, W. Migration-Based Service Allocation Optimization in Dynamic IoT Networks. In *Proceedings of the Service-Oriented Computing*; Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H.Y., Eds.; Springer: Cham, Switzerland, 2021; pp. 385–399.
68. Monderer, D.; Shapley, L.S. Potential Games. *Games Econ. Behav.* **1996**, *14*, 124–143. [\[CrossRef\]](#)
69. Ding, Y.; Liu, C.; Zhou, X.; Liu, Z.; Tang, Z. A Code-Oriented Partitioning Computation Offloading Strategy for Multiple Users and Multiple Mobile Edge Computing Servers. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4800–4810. [\[CrossRef\]](#)
70. Cui, L.; Xu, C.; Yang, S.; Huang, J.Z.; Li, J.; Wang, X.; Ming, Z.; Lu, N. Joint Optimization of Energy Consumption and Latency in Mobile Edge Computing for Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 4791–4803. [\[CrossRef\]](#)
71. Zhou, A.; Wang, S.; Ma, X.; Yau, S.S. Towards Service Composition Aware Virtual Machine Migration Approach in the Cloud. *IEEE Trans. Serv. Comput.* **2020**, *13*, 735–744. [\[CrossRef\]](#)
72. Wang, S.; Zhou, A.; Bao, R.; Chou, W.; Yau, S.S. Towards Green Service Composition Approach in the Cloud. *IEEE Trans. Serv. Comput.* **2021**, *14*, 1238–1250. [\[CrossRef\]](#)
73. Kientopf, K.; Raza, S.; Lansing, S.; Güneş, M. Service management platform to support service migrations for IoT smart city applications. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5. [\[CrossRef\]](#)

74. Xu, X.; Liu, X.; Xu, Z.; Dai, F.; Zhang, X.; Qi, L. Trust-Oriented IoT Service Placement for Smart Cities in Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 4084–4091. [[CrossRef](#)]
75. Ganesan, P. Cloud Migration Techniques for Enhancing Critical Public Services: Mobile Cloud-Based Big Healthcare Data Processing in Smart Cities. *J. Sci. Eng. Res.* **2021**, *8*, 236–244.
76. Vögler, M.; Schleicher, J.M.; Inzinger, C.; Dustdar, S.; Ranjan, R. Migrating Smart City Applications to the Cloud. *IEEE Cloud Comput.* **2016**, *3*, 72–79. [[CrossRef](#)]
77. Zhai, Y.; Bao, T.; Zhu, L.; Shen, M.; Du, X.; Guizani, M. Toward Reinforcement-Learning-Based Service Deployment of 5G Mobile Edge Computing with Request-Aware Scheduling. *IEEE Wirel. Commun.* **2020**, *27*, 84–91. [[CrossRef](#)]
78. Cao, J.; Xu, L.; Abdallah, R.; Shi, W. EdgeOS_H: A home operating system for internet of everything. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1756–1764.
79. Byun, J.; Jeon, B.; Noh, J.; Kim, Y.; Park, S. An intelligent self-adjusting sensor for smart home services based on ZigBee communications. *IEEE Trans. Consum. Electron.* **2012**, *58*, 794–802. [[CrossRef](#)]
80. Zavalyshyn, I.; Duarte, N.O.; Santos, N. HomePad: A privacy-aware smart hub for home environments. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, 12–14 October 2017; IEEE: Piscataway, NJ, USA, 2018; pp. 58–73.
81. Wang, T.; Bhuiyan, M.Z.A.; Wang, G.; Rahman, M.A.; Wu, J.; Cao, J. Big data reduction for a smart city's critical infrastructural health monitoring. *IEEE Commun. Mag.* **2018**, *56*, 128–133. [[CrossRef](#)]
82. Sabir, B.E.; Youssfi, M.; Bouattane, O.; Allali, H. Towards a New Model to Secure IoT-based Smart Home Mobile Agents using Blockchain Technology. *Eng. Technol. Appl. Sci. Res.* **2020**, *10*, 5441–5447. [[CrossRef](#)]
83. Park, E.; Kim, S.; Kim, Y.; Kwon, S.J. Smart home services as the next mainstream of the ICT industry: Determinants of the adoption of smart home services. *Univers. Access Inf. Soc.* **2018**, *17*, 175–190. [[CrossRef](#)]
84. Li, X.; Wan, J.; Dai, H.N.; Imran, M.; Xia, M.; Celesti, A. A Hybrid Computing Solution and Resource Scheduling Strategy for Edge Computing in Smart Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4225–4234. [[CrossRef](#)]
85. Lin, C.C.; Deng, D.J.; Chih, Y.L.; Chiu, H.T. Smart Manufacturing Scheduling With Edge Computing Using Multiclass Deep Q Network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4276–4284. [[CrossRef](#)]
86. Sun, W.; Liu, J.; Yue, Y. AI-Enhanced Offloading in Edge Computing: When Machine Learning Meets Industrial IoT. *IEEE Netw.* **2019**, *33*, 68–74. [[CrossRef](#)]
87. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge Computing in IoT-Based Manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [[CrossRef](#)]
88. Hartmann, M.; Hashmi, U.S.; Imran, A. Edge computing in smart health care systems: Review, challenges, and research directions. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3710. [[CrossRef](#)]
89. Manogaran, G.; Shakeel, P.M.; Fouad, H.; Nam, Y.; Baskar, S.; Chilamkurti, N.; Sundarasekar, R. Wearable IoT Smart-Log Patch: An Edge Computing-Based Bayesian Deep Learning Network System for Multi Access Physical Monitoring System. *Sensors* **2019**, *19*, 3030. [[CrossRef](#)]
90. Alam, M.G.R.; Munir, M.S.; Uddin, M.Z.; Alam, M.S.; Dang, T.N.; Hong, C.S. Edge-of-things computing framework for cost-effective provisioning of healthcare data. *J. Parallel Distrib. Comput.* **2019**, *123*, 54–60. [[CrossRef](#)]
91. Samie, F.; Tsoutsouras, V.; Bauer, L.; Xydis, S.; Soudris, D.; Henkel, J. Computation offloading and resource allocation for low-power IoT edge devices. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 7–12. [[CrossRef](#)]
92. Ning, Z.; Dong, P.; Wang, X.; Hu, X.; Guo, L.; Hu, B.; Guo, Y.; Qiu, T.; Kwok, R.Y.K. Mobile Edge Computing Enabled 5G Health Monitoring for Internet of Medical Things: A Decentralized Game Theoretic Approach. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 463–478. [[CrossRef](#)]
93. Babbar, H.; Rani, S.; AlQahtani, S.A. Intelligent Edge Load Migration in SDN-IIoT for Smart Healthcare. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8058–8064. [[CrossRef](#)]
94. Oueida, S.; Kotb, Y.; Aloqaily, M.; Jararweh, Y.; Baker, T. An Edge Computing Based Smart Healthcare Framework for Resource Management. *Sensors* **2018**, *18*, 4307. [[CrossRef](#)] [[PubMed](#)]
95. Islam, M.M.; Razzaque, M.A.; Hassan, M.M.; Ismail, W.N.; Song, B. Mobile Cloud-Based Big Healthcare Data Processing in Smart Cities. *IEEE Access* **2017**, *5*, 11887–11899. [[CrossRef](#)]
96. Nikoloudakis, Y.; Pallis, E.; Mastorakis, G.; Mavromoustakis, C.; Skianis, C.; Markakis, E. Vulnerability assessment as a service for fog-centric ICT ecosystems: A healthcare use case. *Peer-to-Peer Netw. Appl.* **2019**, *12*, 1216–1224. [[CrossRef](#)]
97. Chaudhry, J.; Saleem, K.; Islam, R.; Selamat, A.; Ahmad, M.; Valli, C. AZSPM: Autonomic Zero-Knowledge Security Provisioning Model for Medical Control Systems in Fog Computing Environments. In Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), Singapore, 9–12 October 2017; pp. 121–127. [[CrossRef](#)]

98. Kim, T.; Sathyanarayana, S.D.; Chen, S.; Im, Y.; Zhang, X.; Ha, S.; Joe-Wong, C. Modems: Optimizing edge computing migrations for user mobility. *IEEE J. Sel. Areas Commun.* **2022**, *41*, 675–689. [\[CrossRef\]](#)
99. Yuan, J.; Shen, Y.; Xu, R.; Wei, X.; Liu, D. Elevating Security in Migration: An Enhanced Trusted Execution Environment-Based Generic Virtual Remote Attestation Scheme. *Information* **2024**, *15*, 470. [\[CrossRef\]](#)
100. Pahl, C.; El Ioini, N.; Le, V.T. Blockchain based service continuity in mobile edge computing. In Proceedings of the 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 22–25 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 136–141.
101. Jin, H.; Zhang, P.; Dong, H.; Wei, X.; Zhu, Y.; Gu, T. Mobility-aware and privacy-protecting QoS optimization in mobile edge networks. *IEEE Trans. Mob. Comput.* **2022**, *23*, 1169–1185. [\[CrossRef\]](#)
102. Jin, H.; Zhang, P.; Dong, H.; Zhu, Y.; Bouguettaya, A. Privacy-aware forecasting of quality of service in mobile edge computing. *IEEE Trans. Serv. Comput.* **2021**, *16*, 478–492. [\[CrossRef\]](#)
103. Zheng, G.; Navaie, K.; Ni, Q.; Pervaiz, H.; Zarakovitis, C. Energy-efficient secure dynamic service migration for edge-based 3-D networks. *Telecommun. Syst.* **2024**, *85*, 477–490. [\[CrossRef\]](#)
104. Minmin, H.; Lingyun, Y.; Xue, P.; Chuan, Z. Trusted edge and cross-domain privacy enhancement model under multi-blockchain. *Comput. Netw.* **2023**, *234*, 109881. [\[CrossRef\]](#)
105. Sun, H.; Tan, Y.; Li, C.; Lei, L.; Zhang, Q.; Hu, J. An edge-cloud collaborative cross-domain identity-based authentication protocol with privacy protection. *Chin. J. Electron.* **2022**, *31*, 721–731. [\[CrossRef\]](#)
106. Kalalas, C.; Mulinka, P.; Belmonte, G.C.; Fornell, M.; Dalgitsis, M.; Vera, F.P.; Sánchez, J.S.; Villares, C.V.; Sedar, R.; Datsika, E.; et al. AI-Driven Vehicle Condition Monitoring with Cell-Aware Edge Service Migration. *arXiv* **2025**, arXiv:2506.02785.
107. Gao, Z.; Yang, L.; Dai, Y. Vrcs-ac: Reinforcement learning for service migration in vehicular edge computing systems. *IEEE Trans. Serv. Comput.* **2024**, *17*, 4436–4450. [\[CrossRef\]](#)
108. Malektaji, S.; Ebrahimzadeh, A.; Elbiaze, H.; Glitho, R.H.; Kianpisheh, S. Deep reinforcement learning-based content migration for edge content delivery networks with vehicular nodes. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 3415–3431. [\[CrossRef\]](#)
109. Brandherm, F.; Gedeon, J.; Abboud, O.; Mühlhäuser, M. BigMEC: Scalable service migration for mobile edge computing. In Proceedings of the 2022 IEEE/ACM 7th Symposium on Edge Computing (SEC), Seattle, WA, USA, 5–8 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 136–148.
110. Al-Doghman, F.; Moustafa, N.; Khalil, I.; Sohrabi, N.; Tari, Z.; Zomaya, A.Y. AI-enabled secure microservices in edge computing: Opportunities and challenges. *IEEE Trans. Serv. Comput.* **2022**, *16*, 1485–1504. [\[CrossRef\]](#)
111. Zhang, M.; Cao, J.; Sahni, Y.; Chen, Q.; Jiang, S.; Wu, T. Eaas: A service-oriented edge computing framework towards distributed intelligence. In Proceedings of the 2022 IEEE International Conference on Service-Oriented System Engineering (SOSE), Newark, CA, USA, 15–18 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 165–175.
112. Duan, Q.; Huang, J.; Hu, S.; Deng, R.; Lu, Z.; Yu, S. Combining federated learning and edge computing toward ubiquitous intelligence in 6G network: Challenges, recent advances, and future directions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2892–2950. [\[CrossRef\]](#)
113. Chang, L.; Zhang, Z.; Li, P.; Xi, S.; Guo, W.; Shen, Y.; Xiong, Z.; Kang, J.; Niyato, D.; Qiao, X.; et al. 6G-enabled edge AI for metaverse: Challenges, methods, and future research directions. *J. Commun. Inf. Netw.* **2022**, *7*, 107–121. [\[CrossRef\]](#)
114. Dritsas, E.; Ramantas, K.; Verikoukis, C. A Mobility-Aware Reinforcement Learning Proactive Solution for State Data Migration in Edge Computing. In Proceedings of the 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 21–23 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
115. Ming, Z.; Yu, H.; Taleb, T. Federated deep reinforcement learning for prediction-based network slice mobility in 6G mobile networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 11937–11953. [\[CrossRef\]](#)
116. Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 5–36. [\[CrossRef\]](#)
117. Sun, P.; Lan, J.; Hu, Y.; Guo, Z.; Wu, C.; Wu, J. Realizing the Carbon-Aware Service Provision in ICT System. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 4090–4103. [\[CrossRef\]](#)
118. Chen, X.; Han, G.; Bi, Y.; Yuan, Z.; Marina, M.K.; Liu, Y.; Zhao, H. Traffic prediction-assisted federated deep reinforcement learning for service migration in digital twins-enabled MEC networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3212–3229. [\[CrossRef\]](#)
119. Alhartomi, M.; Salh, A.; Audah, L.; Alzahrani, S.; Alzahmi, A. Enhancing sustainable edge computing offloading via renewable prediction for energy harvesting. *IEEE Access* **2024**, *12*, 74011–74023. [\[CrossRef\]](#)
120. Heng, L.; Yin, G.; Zhao, X. Energy aware cloud-edge service placement approaches in the Internet of Things communications. *Int. J. Commun. Syst.* **2022**, *35*, e4899. [\[CrossRef\]](#)
121. Belkacem, K. Integrating Edge and Cloud Computing for Efficient Big Data Processing in IoT Environments: Enhancing Smart City Applications with Fog Computing. *Stud. Knowl. Discov. Intell. Syst. Distrib. Anal.* **2024**, *14*, 1–14.
122. Chen, Y.; Wang, D.; Wu, N.; Xiang, Z. Mobility-aware edge server placement for mobile edge computing. *Comput. Commun.* **2023**, *208*, 136–146. [\[CrossRef\]](#)

123. Zheng, X.; Zhang, W.; Hu, C.; Zhu, L.; Zhang, C. Cloud-Edge-End Collaborative Inference in Mobile Networks: Challenges and Solutions. *IEEE Network* **2025**, *39*, 90–96. [[CrossRef](#)]
124. Chen, N.; Quan, S.; Zhang, S.; Qian, Z.; Jin, Y.; Wu, J.; Li, W.; Lu, S. Cuttlefish: Neural configuration adaptation for video analysis in live augmented reality. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 830–841. [[CrossRef](#)]
125. Tayeb, H.; Bramas, B.; Faverge, M.; Guermouche, A. Dynamic Tasks Scheduling with Multiple Priorities on Heterogeneous Computing Systems. In Proceedings of the 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), San Francisco, CA, USA, 27–31 May 2024; pp. 31–40. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.