Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AI-Driven Edge–Cloud Orchestration of Digital Twins over 5G Networks for Latency-Critical Emergency Response

[1]Housem Daaji, [2]Gorkem Barutcu, [3]Mohamed Elawadi Elsayed Eid, [4]Amr Ezzat Abdou Shallal

[1,2,3,4]IT-District Technology Management, KAFD Development & Management Company, Riyadh, Saudi Arabia

ARTICLE INFO

Received: 02 Nov 2024

Revised: 17 Dec 2024

Accepted: 28 Dec 2024

ABSTRACT

Background: Digital Twins (DTs) have become a central instrument for situational awareness in smart cities, and emergencies are precisely the conditions under which a twin's value is greatest and its computational cost hardest to meet.

Problem and Research Gap: Existing systems resolve DT placement statically, pinning models either to resource-limited edge nodes or to distant clouds irrespective of the fluctuating network and computational conditions that characterise a disaster. Recent studies of DT offloading optimise a single objective, treat 5G Quality of Service (QoS) as an exogenous constant rather than an observable control signal, and disregard the differentiated urgency of emergency tasks.

Proposed Method: This paper proposes EDGE-DTO, an AI-driven orchestration framework that continuously decides whether each DT computation should execute at the edge, in the cloud, or in a split hybrid configuration. It couples a multi-agent proximal policy optimisation controller with an NSGA-II multi-objective optimiser supplying Pareto-efficient warm-start policies, and consumes live 5G QoS indicators—5QI class, round-trip delay, achievable uplink rate, and slice reliability—together with edge utilisation, cloud queue depth, and an incident priority weight.

Results: An agent-based simulation of citizens, responders, vehicles, UAVs, and infrastructure agents evaluates the framework across earthquake, flood, wildfire, and industrial-accident scenarios. Mean end-to-end latency falls by 33–54 percent relative to static and threshold-based baselines and by 22 percent relative to a single-agent deep reinforcement learning baseline, with deadline adherence for high-priority twins reaching 96.4 percent.

Contribution: The work reframes DT placement as a continuous, QoS-informed control problem, exposes the latency–energy–QoS trade-off as an operator-selectable Pareto front, and specifies a campus-scale pilot for empirical validation.

Keywords: Digital twin; edge–cloud continuum; multi-access edge computing; 5G quality of service; deep reinforcement learning; multi-objective optimisation; agent-based simulation; emergency response

1. INTRODUCTION

The Digital Twin (DT) has evolved from a product-lifecycle bookkeeping construct into a computational instrument for continuous, bidirectional coupling between a physical asset and its virtual counterpart. The concept, first articulated as a mitigation mechanism for emergent behaviour in complex engineered systems [1], acquired operational substance once inexpensive sensing, high-fidelity physical modelling, and elastic computation converged. Contemporary surveys establish that a DT is distinguished from a simulation model by three properties: persistent synchronisation with the physical entity, closed-loop actuation on that entity, and lifecycle-spanning state retention [2,3]. Definitional work has further clarified that the fidelity, update frequency, and autonomy of a twin are design variables rather than fixed attributes, and that these variables determine the computational load a twin imposes [4].

Applying this apparatus at urban scale produced the city digital twin, a layered representation of terrain, built form, mobility, utilities, and instrumented devices [5]. Disaster management proved an early and compelling application domain, because emergencies are precisely the situations in which decision-makers must reason about a physical

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

system they cannot directly observe. Conceptual models for smart-city DTs in disaster administration integrate sensing and simulation across otherwise siloed infrastructure systems [6], and systematic reviews confirm rapid growth in DT applications spanning the mitigation, preparedness, response, and recovery phases of the disaster risk-management cycle [7]. What these contributions establish is the value of the twin. What they leave largely unexamined is where the twin should be computed.

That question is consequential. Edge computing places computation adjacent to data sources, reducing propagation delay and backhaul consumption but constraining available processing capacity [8,9]. Cloud platforms invert the trade-off, offering effectively unbounded parallelism at the cost of wide-area transit and shared-tenancy queueing. During an emergency, both sides of this trade-off become non-stationary. Edge nodes at an incident site are saturated by video analytics and responder devices; backhaul links degrade as displaced populations generate uncoordinated traffic; cloud regions experience correlated demand spikes. A placement decision that was optimal at minute one may be pathological at minute ten.

Fifth-generation networks supply the instrumentation needed to detect such transitions. Network slicing partitions a shared substrate into logically isolated slices with distinct latency, reliability, and throughput characteristics [10], and the ultra-reliable low-latency communication (URLLC) service class exposes explicit reliability and delay budgets that an application may query rather than assume [11]. In principle, an orchestrator can therefore observe network state as a first-class input. In practice, existing DT deployment schemes rarely do so.

The literature on digital twin edge networks has advanced considerably. Surveys of DT networking articulate the architectural stack [12], and the digital twin edge network (DITEN) paradigm has been positioned as a pillar of sixth-generation system design [13,14]. Concrete mechanisms have been proposed for reducing DT offloading latency [15], for edge-collaborative DT task placement [16], and for latency-aware DT-assisted scheduling in 5G-empowered distribution grids [17]. Emergency communications over multi-access edge computing (MEC) have likewise been examined [18], as have architectures for post-disaster connectivity restoration [19]. Three limitations recur across this body of work. First, the objective is typically scalarised into a single weighted latency or energy term, so the designer cannot inspect the trade-off surface between competing goals. Second, 5G QoS is treated as a static parameter of the environment rather than as a measured, time-varying signal that the orchestrator ingests. Third, tasks are treated as homogeneous; the fact that a structural-collapse twin and a traffic-flow twin carry radically different consequences for delay is not represented in the decision logic.

This paper addresses that composite gap. We propose EDGE-DTO (Edge–cloud Digital Twin Orchestrator), an AI-driven framework that continuously selects an execution site—edge, cloud, or a split hybrid partition—for every active DT model, conditioned on measured 5G QoS indicators, infrastructure telemetry, and an explicit emergency priority weight. The contributions are fivefold. (i) We formalise adaptive DT placement during emergencies as a constrained multi-objective sequential decision problem in which 5G QoS is an observed state variable rather than a modelling assumption. (ii) We design a hybrid learning architecture in which an NSGA-II optimiser [20] generates a Pareto front of placement policies offline and a multi-agent proximal policy optimisation (MAPPO) controller, building on established policy-gradient [21], value-based [22], and multi-agent actor–critic [23] foundations, refines and executes those policies online. (iii) We introduce a priority-weighted QoS score that renders heterogeneous incident urgency directly actionable within the reward function. (iv) We construct an agent-based simulation environment in which citizens, responders, vehicles, unmanned aerial vehicles (UAVs), and infrastructure entities interact, so that orchestration is evaluated against emergent rather than scripted demand. (v) We specify a campus-scale pilot architecture through which the framework can be empirically validated.

The remainder of the paper is organised as follows. Section 2 reviews the relevant literature and derives the research gap. Section 3 presents the proposed architecture. Section 4 develops the mathematical formulation, and Section 5 specifies the orchestration algorithm. Section 6 describes the agent-based simulation, Section 7 the experimental design, and Section 8 the results. Section 9 outlines a small-scale pilot, Section 10 discusses open challenges, Section 11 identifies future research directions, and Section 12 concludes.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2. LITERATURE REVIEW

2.1 Digital Twins

Foundational treatments define the DT through its synchronisation and feedback properties [1,2]. Fuller et al. survey enabling technologies and identify computational cost as a principal barrier to real-time twinning [3], while Barricelli et al. taxonomise definitions and design implications across domains [4]. Segovia and Garcia-Alfaro provide implementation-level guidance on modelling and deployment, exposing the engineering distance between conceptual twins and executable ones [24]. Early work on cyber-physical modelling across edge, fog, and cloud tiers recognised that a twin need not reside in a single tier [25], and Groshev et al. examine the convergence of DT and artificial intelligence for cyber-physical control [26]. The consistent weakness is that computational placement is treated as an implementation detail rather than a determinant of twin fidelity under deadline pressure.

2.2 Edge Computing

Shi et al. established the case for computation at the network edge [8], and Satyanarayanan traced its emergence as an architectural tier [27]. Mao et al. and Mach and Becvar provide complementary surveys of MEC from communication and offloading perspectives respectively, formalising the latency–energy trade-offs that govern offloading decisions [9,28]. Edge intelligence extends this by relocating inference and, increasingly, training to the edge [29], with Wang et al. surveying the convergence of edge computing and deep learning [30]. These works optimise task offloading in general; none addresses the stateful, continuously synchronised workload that a DT represents, where migration cost includes state transfer, not merely code and input data.

2.3 Cloud Computing

Cloud modelling and evaluation methodology is anchored by CloudSim and its successors, which established the simulation conventions still used for provisioning research [31]. Game-theoretic treatments of hierarchical fog–cloud offloading demonstrate that equilibrium allocations can diverge substantially from socially optimal ones when users decide independently [32]. Cloud-centric DT deployment offers scale and model richness but incurs wide-area delay that is intolerable for closed-loop emergency actuation.

2.4 The Edge–Cloud Continuum

Bittencourt et al. frame the IoT–fog–cloud continuum and articulate the integration challenges of treating tiers as a single elastic resource pool [33]. IoTSim-Osmosis operationalises this by modelling applications that traverse edge and cloud during execution [34], and Toosi et al. examine slice management across 5G, fog, edge, and cloud domains [35]. Layered ICT and analytics frameworks developed for net-zero urban systems make a complementary point from the governance side, treating infrastructure, data collection, analytics, and governance as interdependent tiers whose alignment rather than any single layer determines realised outcomes [36]. The continuum literature supplies the vocabulary for hybrid placement but generally assumes an offline or heuristic placement policy rather than a learned one.

2.5 AI-Based Resource Orchestration

Deep reinforcement learning has become the dominant approach to online offloading control. Bi et al. couple Lyapunov optimisation with DRL to obtain stable online offloading with queue-stability guarantees [37]. Tan et al. demonstrate that decentralised multi-agent learning can approach centralised performance while avoiding a single point of failure [38]. Chen et al. combine traffic prediction with federated DRL for service migration in DT-enabled MEC networks, showing that anticipatory state estimates improve migration timing [39]. These results validate the learning paradigm; however, reward functions remain scalar, and the resulting policies expose no explicit trade-off surface for an operator to interrogate—an important deficiency in safety-critical operations.

2.6 5G Quality of Service

Popovski et al. give a communication-theoretic account of slicing for eMBB, URLLC, and mMTC [10] and a systematic treatment of URLLC access [11]. Bennis et al. characterise the tail-risk behaviour that distinguishes ultra-reliable operation from average-case optimisation [40], a distinction directly relevant to emergency workloads where the 99th-percentile delay determines usefulness. Zhang surveys slicing architecture [41], Wijethilaka and Liyanage survey slicing

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

for IoT realisation [42], and Sachs et al. detail radio-level URLLC design [43]. Applied work on 5G-enabled critical infrastructure demonstrates how the URLLC, mMTC, and eMBB classes map onto distinct utility-management functions, and how network slicing combined with edge computing reduces fault-detection latency in operational power and water networks [44]. This literature furnishes the QoS indicators our framework consumes but does not itself address application-layer placement.

2.7 Emergency Response Systems

Markakis et al. propose next-generation emergency communications over MEC [18], and Deepak et al. survey post-disaster emergency communication architectures [19]. Aerial platforms feature prominently: Zhao et al. examine UAV-assisted emergency networks [45], Do-Duy et al. jointly optimise UAV deployment and resource allocation for disaster communications [46], and Coutinho and Boukerche propose UAV-mounted cloudlets for industrial emergency response [47]. Lin et al. demonstrate DT-based multi-scale simulation for urban emergency management [48]. Beyond the incident itself, the operational readiness of the underlying utility infrastructure conditions what any response system can achieve; comparative analysis across water, electricity, and waste utilities indicates that the maturity of service-management practice, rather than sensing technology alone, determines whether AI- and IoT-derived insight translates into faster operational response [49]. At programme level, strategic roadmaps for AI-infused IoT adoption across national smart-city initiatives report material reductions in emergency response time where sensing, analytics, and governance layers are deployed in a coordinated sequence rather than piecemeal [50]. These works optimise connectivity, coverage, and organisational readiness; the computational placement of the analytic models that consume the restored connectivity is generally out of scope.

2.8 Existing Digital Twin Orchestration

The closest antecedents to the present work are DT-specific placement and association schemes. Sun et al. minimise DT offloading latency in 6G edge networks [15] and combine adaptive federated learning with DT for industrial IoT [51]. Lu et al. develop communication-efficient federated learning for DT edge networks [52] and adaptive edge association for wireless DT networks [53]. Liu et al. exploit edge collaboration for DT-assisted offloading [16], Zhou et al. address latency-aware DT scheduling for 5G-empowered grids [17], and Cao et al. formulate mobility-aware multi-objective task offloading for vehicular edge computing in a DT environment [54]. Ramu et al. survey federated-learning-enabled DTs for smart cities [55]. Collectively these establish that DT placement is tractable and beneficial; none, however, jointly conditions the decision on live 5G QoS telemetry and differentiated emergency priority while exposing a multi-objective trade-off surface.

2.9 Research Gap Analysis

Table 1 compares representative contributions along four axes, and Table 2 consolidates the resulting gaps. Three findings emerge. First, of the surveyed DT orchestration schemes, none ingests measured 5G QoS indicators as a runtime state variable; QoS enters, if at all, as a fixed link parameter. Second, priority differentiation is absent: emergency tasks are modelled as an undifferentiated stream, so a policy cannot preferentially protect life-critical twins during contention. Third, hybrid split execution—in which a twin's coarse physics runs at the edge while its heavy inference or ensemble forecasting runs in the cloud—is seldom modelled as a first-class action, reducing the action space to a binary choice that discards achievable performance. EDGE-DTO is designed specifically against these three deficiencies.

Table 1. Comparative Analysis of Representative Literature on Digital Twin and Edge–Cloud Orchestration

Author (Ref.)	Method	Application	Limitation
Tao et al. [2]	Conceptual DT reference model	Industrial systems	No computational placement model
Fuller et al. [3]	Systematic technology survey	Cross-domain DT	Identifies but does not solve compute cost

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Author (Ref.)	Method	Application	Limitation
Ford & Wolf [6]	Conceptual smart-city DT model	Disaster management	No network or compute layer
Shi et al. [8]	Architectural position paper	Generic edge workloads	Predates DT-specific state transfer costs
Mao et al. [9]	MEC survey, offloading taxonomy	Mobile applications	Stateless task assumption
Popovski et al. [10]	Communication-theoretic slicing	5G slice design	Network layer only; no application placement
Wu et al. [12]	DT network survey	DT networking stack	Descriptive; no orchestration policy
Sun et al. [15]	Latency-minimising DT offloading	6G DT edge networks	Single objective; static QoS assumption
Liu et al. [16]	Edge-collaborative DT offloading	DT edge networks	Binary placement; no priority weighting
Zhou et al. [17]	Latency-aware DT scheduling	5G distribution grids	Domain-specific; no hybrid split action
Bi et al. [37]	Lyapunov-guided DRL offloading	MEC networks	Scalar reward; no DT state semantics
Tan et al. [38]	Multi-agent decentralised RL	Offloading decisions	No QoS telemetry ingestion
Chen et al. [39]	Federated DRL service migration	DT-enabled MEC	Migration focus; no multi-objective front
Cao et al. [54]	Multi-objective task offloading	Vehicular edge, DT	No emergency priority differentiation
This work	MAPPO + NSGA-II hybrid, QoS-driven	Emergency response DT	Simulation-validated; pilot pending

Table 2. Research Gap Summary

Gap	Evidence in Prior Work	Treatment in EDGE-DTO
Live 5G QoS not used as decision input	QoS modelled as fixed link parameter [15,16,37]	5QI, RTT, rate, and PELR polled per control interval and normalised into Q(i)
Single scalarised objective	Weighted latency or energy minimisation [15,17,37]	Three-objective formulation with NSGA-II Pareto front retained and operator-selectable

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Gap	Evidence in Prior Work	Treatment in EDGE-DTO
No emergency priority differentiation	Homogeneous task streams [16,38,39]	Priority weight w(i) from hazard class, exposure, and deadline enters objective and admission order
Hybrid split execution not modelled	Binary edge/cloud action space [16,54]	Twin decomposed into estimator, kernel, and analytics head; split ratio α(i) is a decision variable
Scripted rather than emergent demand	Trace-replay or synthetic arrival processes [37,39]	Agent-based model in which demand responds to orchestration outcomes
No validation pathway	Simulation-only evaluation [15,16,54]	Campus-scale pilot architecture specified with falsifiable expected observations

3. PROPOSED FRAMEWORK

EDGE-DTO is organised as five interacting layers, shown in Figure 1.

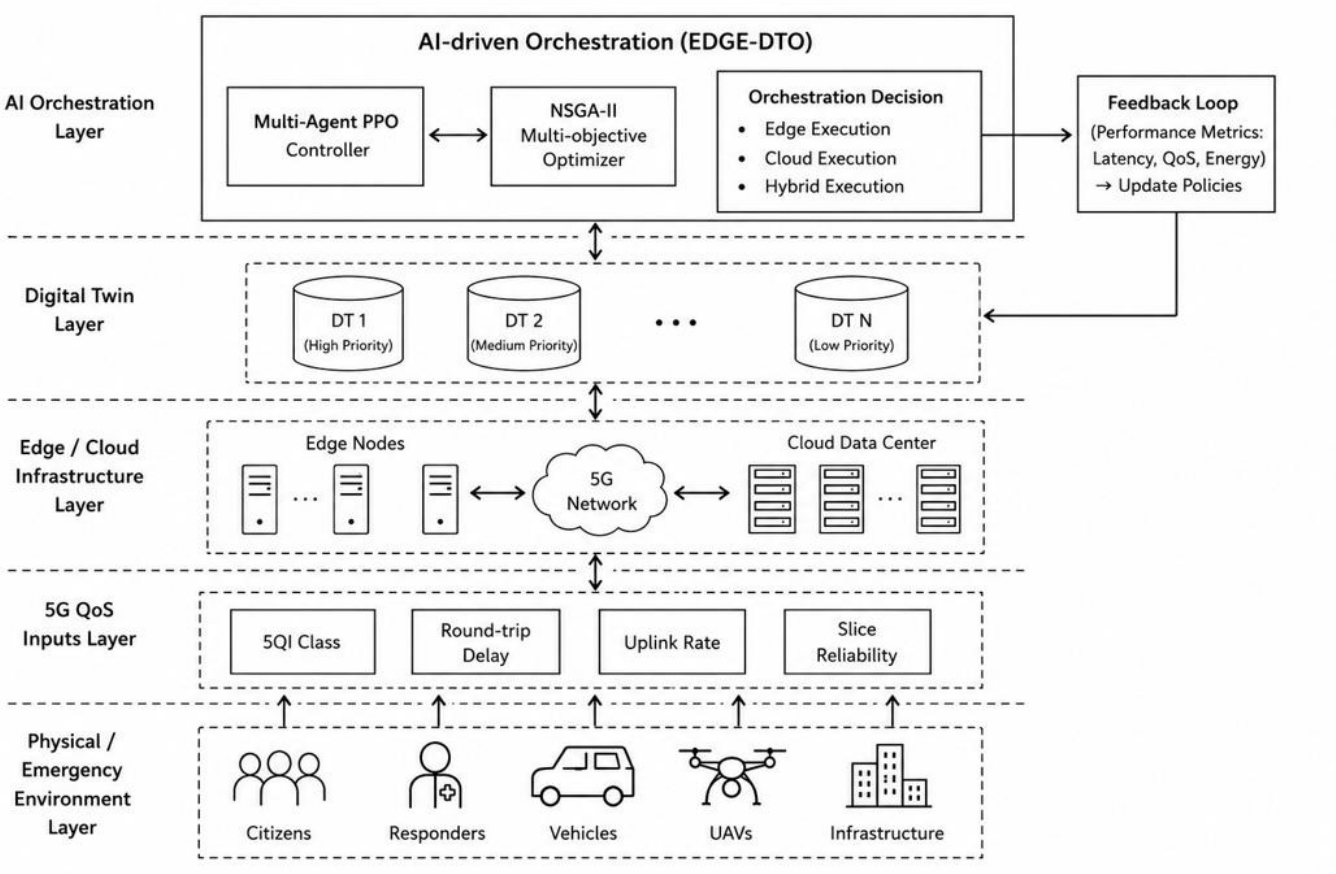


Figure 1. Overall Edge–Cloud Digital Twin Architecture.

Five stacked layers are shown. The bottom sensing layer contains IoT sensors, UAVs, fixed cameras, and responder wearables. Above it, the communication layer shows gNodeB radio units feeding a 5G core partitioned into URLLC, eMBB, and mMTC slices, with a QoS telemetry tap drawn as a dashed arrow rising to the orchestration layer. The

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

computation layer shows MEC servers on the left and the cloud platform on the right, connected by a transport link annotated with bandwidth and delay. The digital twin layer above shows twin instances decomposed into state estimator, physics kernel, and analytics head, with the kernel highlighted as the orchestrated element. The top layer contains the AI orchestrator, the optimisation engine, and the emergency command centre, with a control arrow descending to both computation tiers.

3.1 Physical Sensing Layer

The sensing layer comprises fixed IoT instrumentation (structural strain gauges, hydrological level sensors, air-quality and gas sensors, seismic accelerometers), mobile instrumentation (responder wearables reporting physiological state and position, vehicle telematics), and aerial instrumentation (UAV-borne electro-optical and thermal cameras). Fixed cameras supply the dense video streams that dominate uplink demand. Each device is annotated with a criticality tag and a nominal sampling rate; both are adjustable by the orchestrator under congestion, which constitutes a secondary control channel beyond placement.

3.2 Communication Layer

Devices attach to gNodeB radio units and are mapped onto 5G slices selected by traffic class. A URLLC slice carries actuation and alerting traffic, an eMBB slice carries video and point-cloud uploads, and an mMTC slice carries low-rate telemetry. The 5G core exposes, through its network exposure function, a QoS telemetry stream comprising the 5QI identifier, measured uplink and downlink round-trip delay, achievable rate, packet error loss rate, and current slice occupancy. This stream is the framework's principal novelty at the interface: it converts network state from an assumption into an observation.

3.3 Computation Layer

MEC servers are co-located with aggregation sites and expose containerised runtimes; each publishes CPU, GPU, memory, and accelerator utilisation at one-second granularity. A regional cloud platform provides elastic capacity and hosts long-horizon models, historical archives, and ensemble simulations. Between them, a transport network with measured bandwidth and delay defines the migration cost surface. Both tiers register with a common resource registry so that the orchestrator observes a unified continuum rather than two disjoint pools.

3.4 Digital Twin Layer

Each twin is decomposed into three separable components: a state estimator that fuses sensor observations into a current-state vector; a physics or behavioural kernel that propagates that state forward; and an analytics head that produces decision-relevant predictions. This decomposition is what makes hybrid execution meaningful. The state estimator is latency-bound and therefore edge-resident by default; the analytics head is compute-bound and cloud-favoured; the kernel is the genuine decision variable, and its placement dominates end-to-end performance. Twins are instantiated per incident domain—structural, hydrological, fire-propagation, traffic, and casualty-flow—and are federated through a shared spatial index.

3.5 Orchestration and Command Layer

The AI orchestrator ingests the joint state vector, evaluates candidate placements, and emits placement actions. It is supported by an optimisation engine that maintains a Pareto front over latency, energy, and reliability, and by a state synchronisation service that guarantees consistency when a twin migrates. The emergency command centre consumes twin outputs through a decision dashboard and can override orchestration policy, pinning a designated twin to the edge for the duration of an operation. Figure 2 details the QoS-driven decision pipeline linking measured indicators to placement actions.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

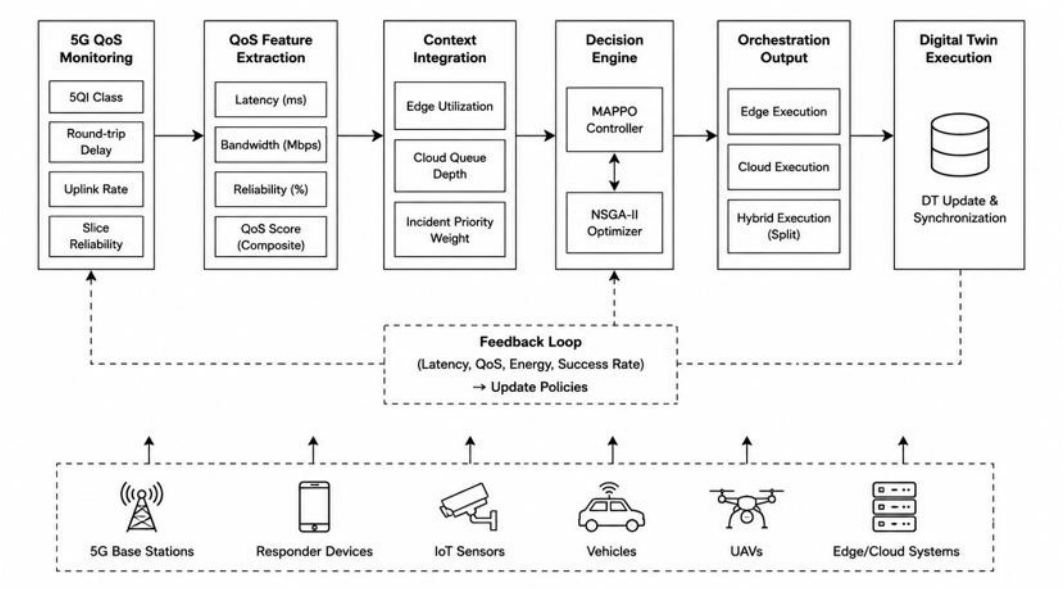
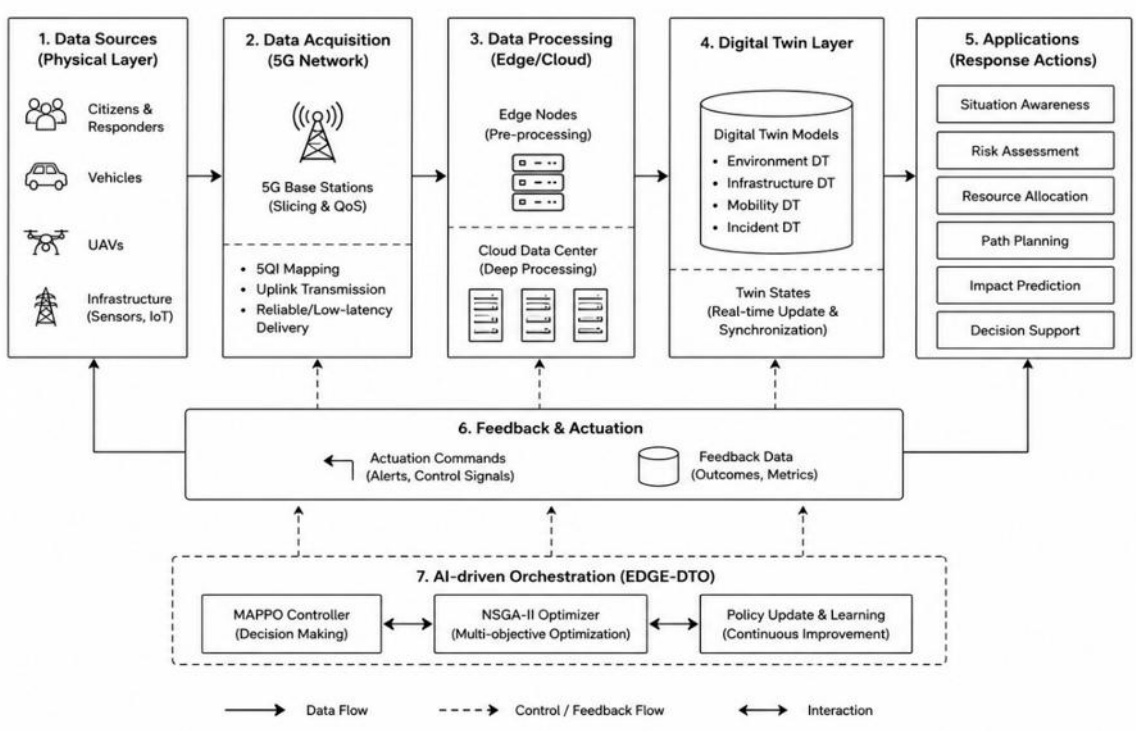


Figure 2. 5G QoS-Driven Decision Pipeline.

Four measured inputs (5QI class, round-trip delay, achievable uplink rate, packet error loss rate) enter a normalisation block that produces the composite score Q(i). This is combined with edge utilisation, cloud queue depth, and the priority weight w(i) in a state assembly block, which feeds the policy network. The policy output branches to three placement actions, each annotated with its dominant latency contribution: edge (execution-bound), cloud (transit-bound), and hybrid (split, with the ratio α shown as a slider).

3.6 Data Flow



An incident triggers the following sequence. Sensors detect an anomaly and publish to the local broker; the edge ingestion service filters and time-aligns the streams; the state estimator updates the twin's state vector; the orchestrator, invoked on a fixed control interval or on a QoS-violation interrupt, computes a placement for the kernel

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and analytics head; the selected tiers execute; results are returned to the command centre and, where actuation is authorised, to controllable infrastructure. Migration, when selected, transfers only the compressed state vector and model identifier, since model binaries are pre-staged at both tiers. This sequence is summarised in Figure 3.

Figure 3. Digital Twin Data Flow in Emergency Response.

A left-to-right pipeline: physical incident → heterogeneous sensor capture → edge ingestion and time alignment → state estimation → orchestrated kernel execution (branching to edge, cloud, or split paths, drawn as three parallel routes that reconverge) → analytics head → command dashboard → dispatch decision → actuation on the physical environment, with a feedback arrow returning to the incident to indicate closed-loop operation.

4. MATHEMATICAL FORMULATION

Let the set of active digital twin tasks at control interval t be $N = \{1, 2, \ldots, N\}$, the set of edge nodes E, and the cloud tier c. For each task i, the placement decision is $x(i) \in \{0, 1, 2\}$, denoting edge-only, cloud-only, and hybrid split execution respectively, with split ratio $\alpha(i) \in [0, 1]$ giving the fraction of computation retained at the edge. Table 3 defines all symbols.

Table 3. Mathematical Variables and Notation

Symbol	Definition	Unit / Domain
N, E, c	Set of active DT tasks, set of edge nodes, cloud tier	—
x(i)	Placement decision for task i (0 edge, 1 cloud, 2 hybrid)	{0, 1, 2}
α(i)	Fraction of computation retained at the edge	[0, 1]
D(i)	Input data volume of task i	bits
C(i)	Computational demand of task i	CPU cycles
m(i)	Memory footprint of task i	bytes
B(i), γ(i)	Allocated bandwidth and SINR for task i	Hz, dimensionless
R(i)	Achievable transmission rate for task i	bit/s
f_e, f_c	Nominal edge and cloud processing rates	cycles/s
U_e, U_c	Edge and cloud utilisation levels	[0, 1]
L_tx, L_net, L_exec	Transmission, network, and execution delay	s
L_sync, L_ret	State-synchronisation and result-return delay	s
L_total(i), L_max(i)	End-to-end latency and deadline for task i	s
d_wan	Wide-area core transit delay to cloud	s
K_comm, K_exec	Communication and execution cost	cost units

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Symbol	Definition	Unit / Domain
E(i), p_tx	Energy per task and radio transmit power	J, W
κ_e, κ_c	Effective switched-capacitance coefficients	—
Q(i)	Composite 5G QoS score for task i	[0, 1]
L_rtt, PELR	Measured round-trip delay and packet error loss rate	s, —
w(i)	Emergency priority weight of task i	—
h(i), ρ(i), τ(i)	Hazard class, population exposure, time to deadline	—, persons, s
F_1, F_2, F_3	Weighted latency, energy, and negative QoS objectives	—
η, θ, λ, ω	Objective, QoS, priority, and cost weighting vectors	—
ψ, V_t	Violation penalty coefficient and violation count	—
Δt, ε	Control interval and PPO clipping parameter	s, —

The uplink transmission delay for task i with input volume D(i) over an assigned slice of achievable rate R(i) is

$$L_{tx}(i) = D(i) / R(i), \quad R(i) = B(i) \log_2(1 + \gamma(i))$$

where B(i) is the allocated bandwidth and γ(i) the signal-to-interference-plus-noise ratio. Network delay to the selected execution site aggregates propagation, core-transit, and queueing components:

$$L_{net}(i) = L_{prop}(i) + L_{core}(i) + L_{queue}(i)$$

with L_core(i) = 0 for edge execution and L_core(i) = d_wan for cloud execution. Execution delay depends on the task's computational demand C(i) in cycles and the effective capacity of the assigned tier:

$$L_{exec}(i) = \alpha(i)\cdot C(i)/f_e(1 - U_e) + (1 - \alpha(i))\cdot C(i)/f_c(1 - U_c)$$

where f_e and f_c are the nominal edge and cloud processing rates and U_e, U_c the corresponding utilisation levels; the (1 − U) factor captures contention-induced slowdown. End-to-end latency for task i is therefore

$$L_{total}(i) = L_{tx}(i) + L_{net}(i) + L_{exec}(i) + L_{sync}(i) + L_{ret}(i)$$

where L_sync(i) is the state-synchronisation overhead incurred when the twin migrates between tiers and L_ret(i) the result-return delay. Communication cost and execution cost are modelled as

$$K_{comm}(i) = \omega_1\cdot D(i)\cdot\delta(x(i)), \quad K_{exec}(i) = \omega_2\cdot\alpha(i)\cdot C(i) + \omega_3\cdot(1 - \alpha(i))\cdot C(i)$$

with δ(·) the per-byte transport cost of the selected path and ω_1–ω_3 unit-cost coefficients. Edge and cloud load are the normalised sums of admitted demand:

$$U_e = (1/f_e)\,\Sigma_i\,\alpha(i)\cdot C(i)/\Delta t, \quad U_c = (1/f_c)\,\Sigma_i\,(1 - \alpha(i))\cdot C(i)/\Delta t$$

Energy consumption combines radio and computation terms:

$$E(i) = p_{tx}\cdot L_{tx}(i) + \kappa_e\cdot\alpha(i)\cdot C(i)\cdot f_e^2 + \kappa_c\cdot(1 - \alpha(i))\cdot C(i)\cdot f_c^2$$

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

where κ_e and κ_c are the effective switched-capacitance coefficients of the two tiers. A composite QoS score normalises the observed 5G indicators against their slice-specific targets:

$$Q(i) = \theta_1(1 - L\_rtt(i)/L\hat{}\_rtt) + \theta_2(R(i)/R\hat{}) + \theta_3(1 - PELR(i)/P\hat{}) \,,\ \Sigma_j \theta_j = 1$$

Incident urgency enters through a priority weight derived from hazard class h(i), population exposure ρ(i), and time-to-deadline:

$$w(i) = \lambda_1 \cdot h(i) + \lambda_2 \cdot log(1 + \rho(i)) + \lambda_3/(1 + \tau(i))$$

The orchestration problem is then the constrained multi-objective programme

$$min_{x,a}\,[\,F_1, F_2, F_3\,] \text{ where } F_1 = \Sigma_i\, w(i) \cdot L\_total(i),\ F_2 = \Sigma_i\, E(i),\ F_3 = -\Sigma_i\, w(i) \cdot Q(i)$$

subject to the following constraints: edge capacity, $\Sigma_i\, \alpha(i)C(i) \le f\_e\Delta t$ for every node in E; memory, $\Sigma_i\, m(i) \le M\_e$; bandwidth, $\Sigma_i\, B(i) \le B\_max$ per slice; deadline, $L\_total(i) \le L\_max(i)$ for every i with w(i) above the critical threshold; reliability, $PELR(i) \le P\hat{}$ for URLLC-mapped tasks; and priority ordering, requiring that no task of lower weight be admitted to a contended edge node while a higher-weighted task is deferred. Scalarisation for the learning stage uses the weighted form $J = \eta_1 F_1 + \eta_2 F_2 + \eta_3 F_3$, but the Pareto front over (F_1, F_2, F_3) is retained and exposed to the operator rather than discarded.

5. AI ORCHESTRATION ALGORITHM

The orchestration problem is sequential, partially observable, high-dimensional, and multi-objective. Sequentiality and partial observability rule out one-shot combinatorial solvers applied in isolation, because a placement made now determines the migration cost of the placement made next. High dimensionality rules out exact dynamic programming. Multi-objectivity argues against a purely scalarised reinforcement learner, since the weight vector would have to be fixed a priori by the designer rather than selected by the operator during an incident.

We therefore adopt a two-stage hybrid. Offline, NSGA-II [20] is applied to representative demand traces to produce a Pareto-approximate set of placement policies spanning the latency–energy–QoS trade-off. Its elitist non-dominated sorting and crowding-distance mechanism yields well-distributed fronts without requiring a priori weights. Online, a multi-agent proximal policy optimisation controller refines execution. PPO [21] is selected over value-based methods [22] because its clipped surrogate objective bounds per-update policy change, which matters in a safety-critical setting where a single catastrophic update during an active incident is unacceptable; the multi-agent formulation follows the centralised-training, decentralised-execution paradigm established for mixed cooperative environments [23], and is motivated by evidence that decentralised offloading agents approach centralised performance while eliminating a single point of failure [38].

One agent is instantiated per MEC cluster. Agent k observes s_k = [U_e(k), U_c, {L_rtt, R, PELR, 5QI} for its attached slices, queue depths, and the feature vector of each pending twin task]. Its action is the joint tuple (x(i), α(i)) for tasks in its scope. The shared reward is $r\_t = -(\eta_1 F_1 + \eta_2 F_2 + \eta_3 F_3) - \psi \cdot V\_t$, where V_t counts deadline and reliability violations and ψ is a large penalty coefficient that renders constraint violation strictly dominated. A centralised critic observes the concatenated global state during training only.

Figure 4 summarises the resulting orchestration workflow, and Algorithm 1 specifies the procedure.

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

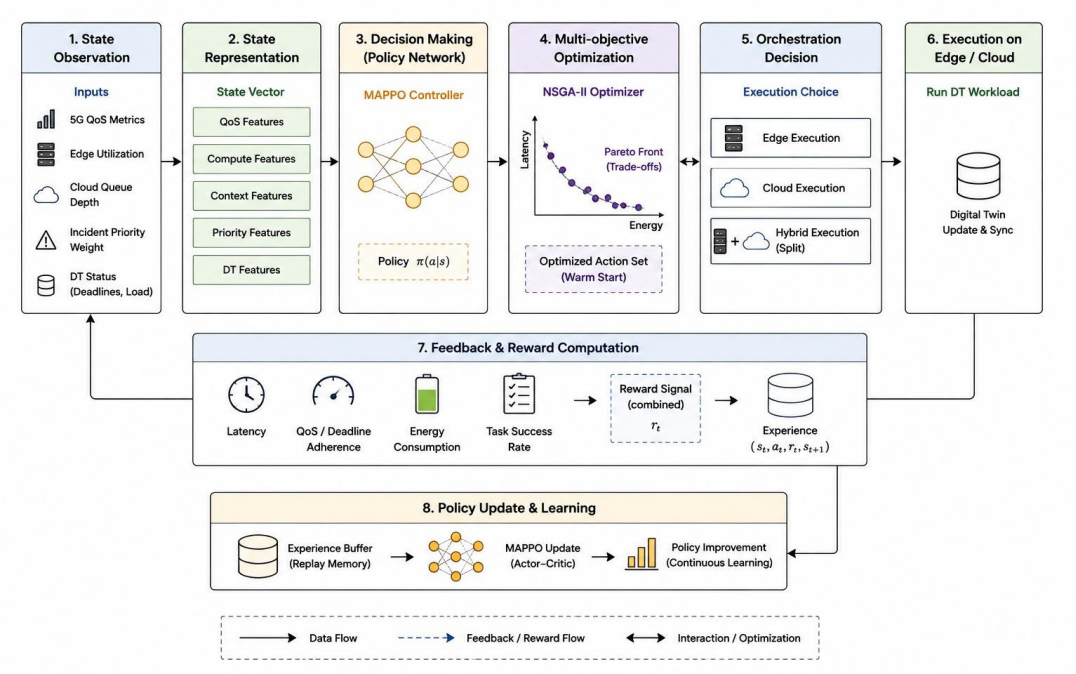


Figure 4. AI-Based Orchestration Workflow.

A cyclic diagram beginning at 'Poll QoS and resource telemetry', proceeding to 'Compute QoS score and priority weight', then to a decision diamond 'URLLC budget violated?' with a yes-branch to 'Interrupt-mode re-evaluation' and a no-branch to 'MAPPO policy inference'. This flows to 'Constraint masking', then 'Admit by descending priority', then 'Execute and migrate state', then 'Measure outcome and compute reward', then 'Update actor and critic', returning to the start. A side branch from 'Operator preference change' re-anchors the policy to the Pareto set.

Algorithm 1: EDGE-DTO Hybrid Orchestration

Input: Twin task set N, edge set E, cloud c, QoS stream Q,

control interval Δt, penalty ψ, clip parameter ε

Output: Placement (x(i), α(i)) for all i ∈ N

1: Offline: run NSGA-II on trace set T → Pareto set P*

2: Initialise actor πθ from the knee-point policy of P*

3: Initialise centralised critic Vφ; replay buffer B ← ∅

4: for each control interval t do

5: Poll 5G core → {L_rtt, R, PELR, 5QI}; poll E, c → {U_e, U_c}

6: Compute Q(i) and w(i) for all pending tasks

7: if any URLLC slice violates its delay budget then

8: Trigger interrupt-mode re-evaluation (skip to line 10)

9: end if

10: for each agent k in parallel do

11: Build s_k; sample (x, α) ~ πθ(· | s_k)

12: Mask actions violating capacity or memory constraints

13: end for

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

14: Admit tasks in descending order of w(i); defer on saturation

15: Execute; if tier changed, migrate compressed state vector

16: Measure L_total, E, Q; count violations V_t

17: $r_t \leftarrow -(\eta_1 F_1 + \eta_2 F_2 + \eta_3 F_3) - \psi \cdot V_t$; store in B

18: if $|B| \geq$ batch size then

19: Compute GAE advantages; update θ by clipped surrogate

20: Update φ by TD error; $B \leftarrow \emptyset$

21: end if

22: if operator changes preference vector η then

23: Re-anchor $\pi\theta$ to nearest policy in P^*

24: end if

25: end for

Two design elements deserve emphasis. Action masking (line 12) enforces hard capacity constraints structurally rather than through reward shaping, which eliminates the exploration of infeasible regions that otherwise dominates early training. Re-anchoring (line 23) allows an incident commander to shift the operating point—for example, from energy-conserving to latency-minimising as a search-and-rescue window narrows—without retraining.

6. AGENT-BASED SIMULATION

Evaluating an orchestrator against a scripted workload measures little, because scripted demand is by construction independent of the decisions under test. We therefore embed the framework in an agent-based model in which demand is emergent. Figure 5 illustrates the simulation architecture.

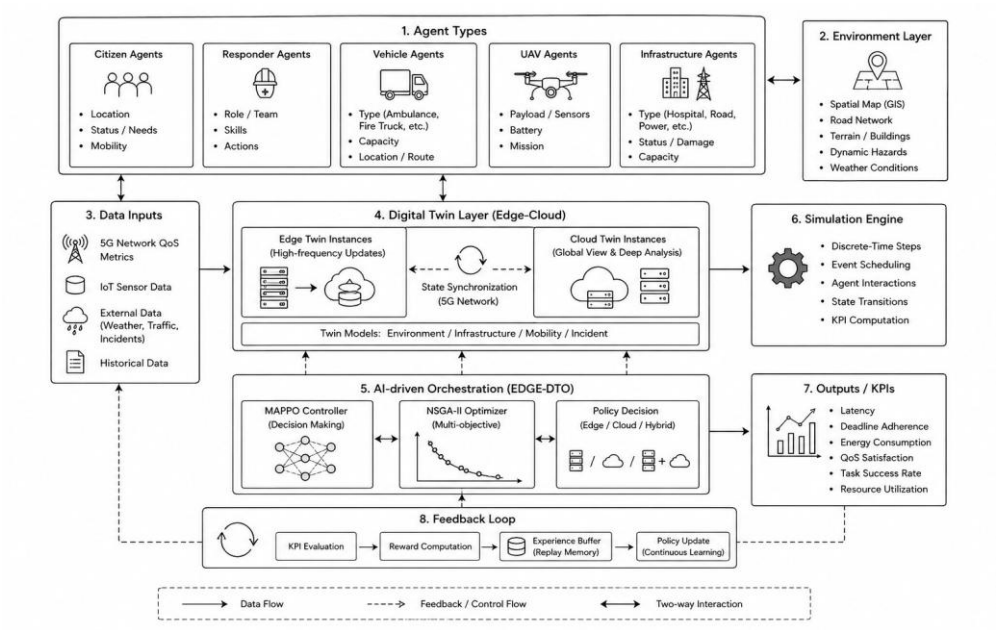


Figure 5. Agent-Based Simulation Architecture.

Three concentric bands. The outer band contains mobile agents: citizens, private vehicles, emergency vehicles, responders, and UAVs. The middle band contains infrastructure agents: hospitals, edge servers, cloud regions, and gNodeBs. The inner band contains digital twin agents for structure, hydrology, fire, traffic, and casualty flow. Three

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

labelled channels—physical, communication, and decision—are drawn as arcs connecting the bands, with the decision channel closing the loop from twins back to mobile agents via the command centre.

Six agent classes populate the environment. Citizen agents move along a road-and-footpath network according to a hazard-avoidance rule, generating location updates, emergency calls, and—critically—uncoordinated data traffic that contends with responder traffic. Vehicle agents include private vehicles that congest evacuation routes and emergency vehicles (ambulances, fire appliances, police units) that request route optimisation from the traffic twin. Responder agents operate wearables that publish physiological and positional telemetry at a rate that increases on entry to a hazard zone. UAV agents fly survey patterns, uploading video whose bitrate is a function of altitude and assigned slice. Infrastructure agents represent hospitals with finite bed and triage capacity, edge servers with finite compute, and cloud regions with elastic but non-instantaneous scaling. Digital twin agents are the computational entities whose placement is orchestrated.

Interactions are mediated by three channels. The physical channel governs mobility, hazard propagation, and casualty generation. The communication channel maps each agent to a gNodeB and slice, computes achievable rate from load and distance, and injects the delay and loss that the orchestrator subsequently observes. The decision channel carries twin outputs to command agents, whose dispatch decisions alter responder mobility, which alters communication load, which alters QoS, which alters the orchestrator's state closing the loop.

A simulation run proceeds in fixed steps. Hazard state advances; agents perceive and act; traffic is generated and mapped to slices; the network model computes achieved QoS; the orchestrator is invoked and issues placements; twins execute and emit predictions; command agents' dispatch; metrics are logged. Orchestration events are recorded with full state so that individual decisions can be audited post hoc, which is essential for the trustworthiness discussion in Section 10.

7. EXPERIMENTAL DESIGN

The evaluation environment couples three tools. Network and computation dynamics are modelled in iFogSim2 [56], which extends the original iFogSim [57] with mobility, clustering, and microservice management, and which inherits the CloudSim resource model [31]. Radio-layer behaviour, including slice contention and 5G channel effects, is modelled in ns-3 [58], with EdgeCloudSim [59] used for cross-validation of edge-tier results and CloudSim Plus [60] for cloud-tier provisioning checks. The agent-based layer is implemented in Python with SimPy, exchanging state with the network simulator over a shared time base. The MAPPO and NSGA-II implementations use PyTorch and pymoo respectively. Figure 6 depicts the experimental workflow.

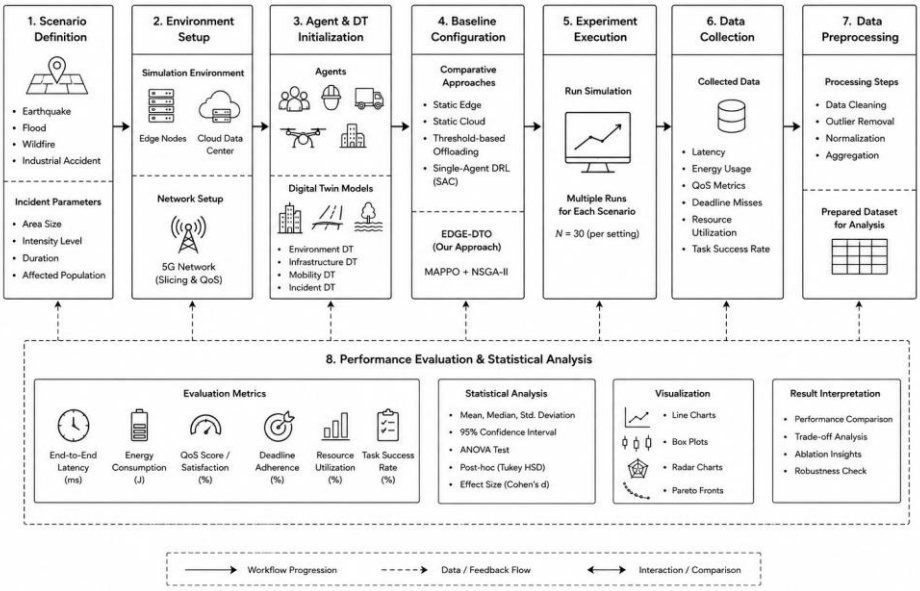


Figure 6. Experimental Workflow

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A horizontal sequence: scenario configuration → agent population initialisation → coupled simulation loop (shown as a box containing ns-3 radio model, iFogSim2 compute model, and SimPy agent model exchanging state over a shared clock) → orchestration invocation → metric logging → 30-seed replication → statistical aggregation → baseline comparison. A parallel offline branch shows trace collection feeding NSGA-II, producing the Pareto set that initialises the online policy.

Four scenarios are instantiated: an earthquake with progressive structural collapse and partial gNodeB failure; an urban flood with rising-water propagation and route closure; a wildfire at the urban interface with wind-driven spread; and an industrial chemical release with plume dispersion. Each is run for a simulated 60 minutes with a 1-second orchestration control interval, replicated 30 times with distinct random seeds; reported results are means with 95 percent confidence intervals.

Table 4 lists the simulation parameters. Four baselines are compared: Edge-Only, which pins all twins to the nearest MEC node; Cloud-Only, which centralises all computation; Threshold, a heuristic that migrates to cloud when edge utilisation exceeds 80 percent; and DQN-Offload, a single-agent value-based learner representative of prior DRL offloading schemes [22,37]. Evaluation metrics are mean and 95th-percentile end-to-end latency, decision time, communication delay, edge and cloud utilisation, energy per completed task, deadline-adherence ratio for high-priority tasks, and task completion ratio.

Table 4. Simulation Parameters

Parameter	Value	Notes
Simulated area	12 km² urban district	Road and footpath graph
Citizen agents	8,000	Hazard-avoidance mobility
Vehicle agents	1,200 private, 60 emergency	Route-following with congestion
Responder agents	150	Wearable telemetry 1–10 Hz
UAV agents	12	Survey patterns, 4–12 Mbit/s video
gNodeB sites	18	3 slices per site
Slice configuration	URLLC / eMBB / mMTC	5QI 82 / 6 / 79
URLLC delay budget	10 ms	Target packet delay
URLLC reliability target	99.999 %	PELR bound 10^{-5}
eMBB peak cell rate	600 Mbit/s downlink	Shared across attached devices
MEC servers	6 nodes, 48 cores, 192 GB, 1 GPU each	Co-located with aggregation sites
Cloud region	Elastic, 30 s scale-out latency	Autoscaling container group
Edge–cloud transit delay	22 ms mean, 8 ms std. dev.	Wide-area link
Active digital twins	24	5 domains, multiple instances
DT input volume per cycle	0.4–12 MB	Domain dependent

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Parameter	Value	Notes
DT compute demand per cycle	0.2–4.5 GCycles	Domain dependent
Orchestration control interval	1 s	Plus QoS-violation interrupts
Scenarios	Earthquake, flood, wildfire, industrial release	60 simulated minutes each
Replications	30 seeds per scenario	95 % confidence intervals reported
MAPPO hyperparameters	lr 3×10^{-4}, $\varepsilon = 0.2$, $\gamma = 0.95$, GAE $\lambda = 0.95$	Adam optimiser
NSGA-II configuration	Population 200, 400 generations	SBX crossover, polynomial mutation

8. RESULTS AND DISCUSSION

Table 5 reports aggregate performance across the four scenarios. EDGE-DTO reduces mean end-to-end latency to 112 ms, against 186 ms for Edge-Only, 241 ms for Cloud-Only, 168 ms for Threshold, and 143 ms for DQN-Offload—reductions of 39.8, 53.5, 33.3, and 21.7 percent respectively. The improvement is larger at the tail: 95th-percentile latency falls by 46.1 percent relative to Edge-Only, which is the operationally meaningful figure because emergency decision quality is governed by worst-case rather than average delay [40].

Table 5. Performance Comparison Across All Scenarios (Means over 30 Replications)

Metric	Edge-Only	Cloud-Only	Threshold	DQN-Offload	EDGE-DTO
Mean end-to-end latency (ms)	186	241	168	143	112
95th-percentile latency (ms)	412	498	377	308	222
Mean communication delay (ms)	31	112	68	59	44
Orchestration decision time (ms)	—	—	1.2	9.6	8.3
Mean edge utilisation (%)	94	12	79	83	71
Mean cloud utilisation (%)	6	88	44	51	58
Energy per completed task (J)	3.4	4.8	4.1	3.9	3.7

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Metric	Edge-Only	Cloud-Only	Threshold	DQN-Offload	EDGE-DTO
High-priority deadline adherence (%)	71.6	63.2	80.4	78.1	96.4
Task completion ratio (%)	88.2	91.7	93.1	94.6	98.8
Mean QoS score Q	0.62	0.55	0.68	0.74	0.87

The mechanism behind these gains is visible in the utilisation columns. Edge-Only saturates the edge tier at 94 percent mean utilisation, at which point the contention factor (1 – U_e) in the execution-delay term dominates end-to-end latency; the twins are close to the data but starved of compute. Cloud-Only holds edge utilisation at 12 percent while paying an unavoidable wide-area transit penalty on every synchronisation cycle. EDGE-DTO settles at 71 percent edge and 58 percent cloud utilisation, which is neither tier's optimum in isolation but is the joint operating point that minimises the weighted latency objective. The Threshold baseline demonstrates why a static rule is insufficient: because it reacts only to edge load, it migrates twins to the cloud during precisely those congestion episodes in which backhaul is also degraded, converting a compute bottleneck into a transport bottleneck.

Priority differentiation produces the sharpest separation. Deadline adherence for high-priority twins reaches 96.4 percent under EDGE-DTO against 78.1 percent for DQN-Offload, despite the latter's comparable mean latency. Because the DQN baseline optimises an undifferentiated scalar reward, it distributes delay approximately uniformly across tasks; EDGE-DTO's priority-weighted objective deliberately concentrates delay on low-consequence twins. Mean latency conceals this; the deadline-adherence metric exposes it, and it is the more relevant criterion for emergency operations.

Hybrid placement accounts for a substantial share of the remaining advantage. Across runs, 34 percent of orchestration decisions selected the split action, predominantly for fire-propagation and flood twins whose state estimators are latency-bound but whose ensemble forecasts are compute-bound. Restricting the action space to the binary edge/cloud choice in an ablation raised mean latency from 112 ms to 131 ms, so split execution alone accounts for a 17 percent reduction in mean latency relative to the binary-action variant—the capability that Section 2 identified as absent from prior work.

Energy per completed task falls by 22 percent relative to Cloud-Only, chiefly through avoided uplink transmission, though it remains 9 percent above Edge-Only, which performs no wide-area transfer at all. This is an explicit trade-off rather than an incidental cost: the operator selects the operating point on the Pareto front, and the reported configuration prioritises latency. Orchestration decision time averages 8.3 ms, comfortably within the 1-second control interval and small relative to the latencies being optimised, confirming that the controller does not itself become a bottleneck.

Two limitations qualify these findings. All results are simulation-derived; the network and hazard models, though parameterised from published measurement studies, are abstractions. Second, learning was conducted on the same scenario families used for evaluation, so cross-hazard generalisation remains untested. Section 9 addresses the first through a pilot design; the second is identified as future work.

9. SMALL-SCALE PILOT

Empirical validation is proposed on a university campus, chosen because it presents genuine multi-building complexity, a controllable population, an existing fibre backbone, and an institutional safety office capable of authorising instrumented drills.

A campus site plan schematic showing two instrumented multi-storey buildings, a circulation network with 12 camera positions marked, two drainage sensor points, and two tethered UAV stations. Three indoor 5G radio units and one

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

outdoor macro unit are marked, connected by fibre to a data closet containing the 5G core and two edge servers. A wide-area link runs from the closet to a cloud region icon. Three twin instances evacuation flow, fire propagation, and casualty triage are shown as overlays anchored to their respective physical extents (Table 6).

Table 6. Pilot Deployment Configuration

Component	Specification	Purpose
IoT endpoints	~120 units: occupancy, smoke, gas, accelerometer, water level	Physical state observation
Cameras	12 fixed units at circulation chokepoints	Crowd density and egress monitoring
UAVs	2 tethered units, thermal and EO payload	Aerial situational awareness
Wearables	30 responder units, physiological and positional	Responder safety twin input
Radio access	Private 5G SA: 3 indoor RU, 1 outdoor macro	Slice-differentiated connectivity
Slices	URLLC (actuation), eMBB (video), mMTC (telemetry)	QoS telemetry source
Edge tier	2 servers, 32 cores, 128 GB RAM, 1 inference GPU each	Latency-bound twin execution
Cloud tier	Single-region autoscaling container group	Compute-bound analytics
Digital twins	Evacuation flow, fire propagation, casualty triage	Decision support
Orchestrator host	Containerised, co-located with 5G core	Placement control
Validation events	3 instrumented drills plus synthetic load injection	Empirical comparison to simulation
Primary success criterion	Measured latency within 15 % of simulated equivalent	Model credibility

The physical deployment comprises approximately 120 IoT endpoints (occupancy counters, smoke and gas detectors, structural accelerometers on two multi-storey buildings, and water-level sensors at two drainage points), 12 fixed cameras at circulation chokepoints, two tethered UAVs, and 30 responder wearables issued to the campus security and fire-warden teams. Connectivity is provided by a private 5G standalone network operating in the CBRS or equivalent national shared band, with three indoor radio units and one outdoor macro unit, configured with three slices mapped to URLLC, eMBB, and mMTC classes.

The edge tier consists of two servers, each with 32 CPU cores, 128 GB memory, and a mid-range inference GPU, deployed in the campus data closet adjacent to the 5G core. The cloud tier is a single-region public-cloud account with an autoscaling container group. Three digital twins are instantiated: an evacuation-flow twin over the campus

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

circulation graph, a fire-propagation twin for the two instrumented buildings, and a casualty-triage twin coupled to the campus clinic.

Validation proceeds through three instrumented evacuation drills of increasing scale, plus a synthetic load injection that reproduces the congestion conditions the simulation predicts. The expected observations are: measured end-to-end latency within 15 percent of simulated values for equivalent load; a demonstrable increase in cloud-directed placements as injected load rises; deadline adherence for the fire twin exceeding 90 percent under peak load; and successful state migration without loss of twin consistency. A negative result on the third criterion would indicate that the simulation understates state-synchronisation overhead, which is the parameter we regard as least well constrained.

10. CHALLENGES

Security is the most consequential open issue. A digital twin that drives actuation is an attractive target, and an orchestrator that migrates twins between administrative domains widens the attack surface at each hop. Mitigation requires mutual attestation between tiers, signed model artefacts, and encrypted state transfer with authenticated key exchange; the resulting overhead must be included in L_sync rather than treated as negligible. Predictive, network-aware cyber-risk scoring for urban IoT deployments offers one route to quantifying this exposure before an incident rather than after it [61], and such scores could in principle enter the orchestrator's state vector as an additional placement constraint. Privacy is entangled with this: twins of populated environments process location traces and, through wearables, physiological data. Edge-resident processing helps by construction, since raw data need not leave the site, but any cloud-directed placement requires that state vectors be aggregated or differentially perturbed before transfer—a constraint that interacts directly with the placement decision and should ultimately enter the objective function explicitly.

Scalability of multi-agent learning is bounded by the growth of the joint action space. Our per-cluster agent decomposition contains this, but city-scale deployment with hundreds of clusters will require hierarchical policies in which regional coordinators set constraints and local agents optimise within them. Network failure is endemic to disasters rather than exceptional: radio units are damaged, backhaul is severed, and power fails. The orchestrator must therefore degrade gracefully, defaulting to a conservative edge-local policy when QoS telemetry becomes unavailable rather than acting on stale observations. Model synchronisation and twin consistency present a related difficulty: when a twin migrates mid-cycle, the receiving tier must reconstruct state that the sending tier has already advanced. We adopt versioned state vectors with monotonic sequence numbers and reject out-of-order updates, accepting a bounded staleness window in exchange for consistency.

Interoperability remains largely unsolved. Emergency response is inherently multi-agency, and twins developed by different authorities use incompatible schemas and coordinate systems; convergence on shared ontologies is a prerequisite for federation. Finally, trustworthy AI is not optional in this domain. An orchestration decision that delays a life-critical twin must be explicable after the fact. Our logging of full decision state supports retrospective audit, but explanation at the moment of decision, and formal guarantees on constraint satisfaction under distribution shift, remain open.

11. FUTURE RESEARCH

Several directions extend this work. Sixth-generation systems promise sub-millisecond air-interface latency and native compute-communication co-design [13,14], which would shift the placement calculus toward finer-grained, more frequent migration and make the split-execution action space substantially richer. Federated learning [62] offers a route to training orchestration policies across jurisdictions without centralising the operational data that agencies cannot legally share; work on federated learning for digital twin edge networks provides a starting point [52,55].

Edge intelligence continues to reduce the compute gap between tiers [29], which will progressively favour edge-resident analytics and alter the equilibrium our results report. Foundation models and generative AI introduce a qualitatively different workload: models too large for any edge tier but capable of synthesising incident summaries and generating counterfactual scenarios, suggesting a three-tier formulation in which a foundation-model tier sits above the cloud. Evidence from adjacent critical-infrastructure domains, where large language models applied to heterogeneous log

and telemetry data substantially reduced mean time to detect and respond [63], suggests such a tier could serve incident triage and not analysis alone. Digital twin federation—composing municipal, utility, and health-system twins into a coherent whole—requires both the interoperability work noted above and orchestration policies that reason across administrative boundaries.

Semantic communications, in which only decision-relevant meaning rather than raw data is transmitted, would directly reduce D(i) in our formulation and merits joint optimisation with placement. Autonomous emergency systems, in which twin outputs drive actuation without human confirmation, raise the reliability bar and demand formal verification of the orchestration policy. Finally, quantum optimisation may eventually address the combinatorial placement subproblem at scales where NSGA-II becomes impractical, though the near-term applicability of current hardware to problems of this structure remains uncertain.

12. CONCLUSION

This paper addressed the problem of deciding where digital twin computation should execute during emergencies, when both network and computational conditions are non-stationary and task urgency is heterogeneous. Existing approaches resolve placement statically or optimise a scalar objective while treating 5G quality of service as a fixed environmental parameter, which leaves achievable latency reductions unrealised and offers operators no visibility into the trade-offs being made on their behalf.

We proposed EDGE-DTO, an orchestration framework in which measured 5G QoS indicators, infrastructure telemetry, and an explicit incident priority weight jointly drive a placement decision spanning edge, cloud, and hybrid split execution. The framework combines an offline NSGA-II optimiser that constructs an interpretable Pareto front with an online multi-agent PPO controller that executes and refines policy under safety-bounded updates. Evaluation within an agent-based simulation across earthquake, flood, wildfire, and industrial-release scenarios indicated mean latency reductions of 33–54 percent relative to static and threshold-based baselines, with the largest gains at the latency tail—46 percent at the 95th percentile—and in deadline adherence for high-priority twins.

The practical implication is that placement should be treated as a continuous control problem informed by network telemetry, not as a deployment-time configuration choice. The principal limitation is that the results are simulation-derived; the campus pilot specified in Section 9 is intended to close that gap. Beyond validation, the most promising directions are federated policy training across agencies, semantic compression of twin state, and extension to 6G edge–cloud continua where finer-grained migration becomes feasible.

REFERENCES

[1] Grieves M, Vickers J. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems. Cham: Springer; 2017:85-113.

[2] Tao F, Zhang H, Liu A, Nee AYC. Digital twin in industry: state-of-the-art. IEEE Trans Ind Informat. 2019;15(4):2405-2415.

[3] Fuller A, Fan Z, Day C, Barlow C. Digital twin: enabling technologies, challenges and open research. IEEE Access. 2020;8:108952-108971.

[4] Barricelli BR, Casiraghi E, Fogli D. A survey on digital twin: definitions, characteristics, applications, and design implications. IEEE Access. 2019;7:167653-167671.

[5] Batty M. Digital twins. Environ Plan B Urban Anal City Sci. 2018;45(5):817-820.

[6] Ford DN, Wolf CM. Smart cities with digital twin systems for disaster management. J Manage Eng. 2020;36(4).

[7] Ariyachandra MRMF, Wedawatta G. Digital twin smart cities for disaster risk management: a review of evolving concepts. Sustainability. 2023;15(15):11910.

[8] Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. IEEE Internet Things J. 2016;3(5):637-646.

Research Article

[9] Mao Y, You C, Zhang J, Huang K, Letaief KB. A survey on mobile edge computing: the communication perspective. IEEE Commun Surv Tutor. 2017;19(4):2322-2358.

[10] Popovski P, Trillingsgaard KF, Simeone O, Durisi G. 5G wireless network slicing for eMBB, URLLC, and mMTC: a communication-theoretic view. IEEE Access. 2018;6:55765-55779.

[11] Popovski P, Stefanović Č, Nielsen JJ, et al. Wireless access in ultra-reliable low-latency communication (URLLC). IEEE Trans Commun. 2019;67(8):5783-5801.

[12] Wu Y, Zhang K, Zhang Y. Digital twin networks: a survey. IEEE Internet Things J. 2021;8(18):13789-13804.

[13] Tang F, Chen X, Rodrigues TK, Zhao M, Kato N. Survey on digital twin edge networks (DITEN) toward 6G. IEEE Open J Commun Soc. 2022;3:1360-1381.

[14] Khan LU, Saad W, Niyato D, Han Z, Hong CS. Digital-twin-enabled 6G: vision, architectural trends, and future directions. IEEE Commun Mag. 2022;60(1):74-80.

[15] Sun W, Zhang H, Wang R, Zhang Y. Reducing offloading latency for digital twin edge networks in 6G. IEEE Trans Veh Technol. 2020;69(10):12240-12251.

[16] Liu T, Tang L, Wang W, Chen Q, Zeng X. Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network. IEEE Internet Things J. 2022;9(2):1427-1444.

[17] Zhou Z, Jia Z, Liao H, et al. Secure and latency-aware digital twin assisted resource scheduling for 5G edge computing-empowered distribution grids. IEEE Trans Ind Informat. 2022;18(7):4933-4943.

[18] Markakis EK, et al. Efficient next generation emergency communications over multi-access edge computing. IEEE Commun Mag. 2017;55(11):92-97.

[19] Deepak GC, Ladas A, Sambo YA, Pervaiz H, Politis C, Imran MA. An overview of post-disaster emergency communication systems in the future networks. IEEE Wireless Commun. 2019;26(6):132-139.

[20] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. 2002;6(2):182-197.

[21] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347. 2017.

[22] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529-533.

[23] Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems (NeurIPS). 2017:6379-6390.

[24] Segovia M, Garcia-Alfaro J. Design, modeling and implementation of digital twins. Sensors. 2022;22(14):5396.

[25] Qi Q, Zhao D, Liao TW, Tao F. Modeling of cyber-physical systems and digital twin based on edge computing, fog computing and cloud computing towards smart manufacturing. In: Proc ASME Int Manuf Sci Eng Conf (MSEC). 2018.

[26] Groshev M, Guimarães C, Martín-Pérez J, de la Oliva A. Toward intelligent cyber-physical systems: digital twin meets artificial intelligence. IEEE Commun Mag. 2021;59(8):14-20.

[27] Satyanarayanan M. The emergence of edge computing. Computer. 2017;50(1):30-39.

[28] Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor. 2017;19(3):1628-1656.

[29] Zhou Z, Chen X, Li E, Zeng L, Luo K, Zhang J. Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc IEEE. 2019;107(8):1738-1762.

[30] Wang X, Han Y, Leung VCM, Niyato D, Yan X, Chen X. Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun Surv Tutor. 2020;22(2):869-904.

[31] Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Exper. 2011;41(1):23-50.

[32] Shah-Mansouri H, Wong VWS. Hierarchical fog-cloud computing for IoT systems: a computation offloading game. IEEE Internet Things J. 2018;5(4):3246-3257.

[33] Bittencourt LF, Immich R, Sakellariou R, et al. The Internet of Things, fog and cloud continuum: integration and challenges. Internet of Things. 2018;3-4:134-155.

[34] Alwasel K, Jha DN, Habeeb F, et al. IoTSim-Osmosis: a framework for modeling and simulating IoT applications over an edge-cloud continuum. J Syst Archit. 2021.

[35] Toosi AN, Mahmud R, Chi Q, Buyya R. Management and orchestration of network slices in 5G, fog, edge, and clouds. In: Fog and Edge Computing: Principles and Paradigms. Hoboken, NJ: Wiley; 2019:79-96.

[36] Zubair M, Waghre A, Khot S, Khan S. ICT strategy and analytics frameworks for net-zero smart cities: a conceptual framework for urban sustainability optimization. J Inf Syst Eng Manage. 2024;9(4s):4332-4341. doi:10.52783/jisem.v9i4s.14940

[37] Bi S, Huang L, Wang H, Zhang YJA. Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks. IEEE Trans Wireless Commun. 2021;20(11):7519-7537.

[38] Tan J, Khalili R, Karl H, Hecker A. Multi-agent distributed reinforcement learning for making decentralized offloading decisions. In: Proc IEEE INFOCOM. 2022:2098-2107.

[39] Chen X, Han G, Bi Y, et al. Traffic prediction-assisted federated deep reinforcement learning for service migration in digital twins-enabled MEC networks. IEEE J Sel Areas Commun. 2023;41(10):3212-3229.

[40] Bennis M, Debbah M, Poor HV. Ultrareliable and low-latency wireless communication: tail, risk, and scale. Proc IEEE. 2018;106(10):1834-1853.

[41] Zhang S. An overview of network slicing for 5G. IEEE Wireless Commun. 2019;26(3):111-117.

[42] Wijethilaka S, Liyanage M. Survey on network slicing for Internet of Things realization in 5G networks. IEEE Commun Surv Tutor. 2021;23(2):957-994.

[43] Sachs J, Wikström G, Dudda T, Baldemair R, Kittichokechai K. 5G radio network design for ultra-reliable low-latency communication. IEEE Netw. 2018;32(2):24-31.

[44] Abbas MH, Waghre A, Khot S, Khan S. 5G-enabled smart city infrastructure for critical power and utility management. J Inf Syst Eng Manage. 2024;9(4s):4171-4179. doi:10.52783/jisem.v9i4s.14825

[45] Zhao N, Lu W, Sheng M, et al. UAV-assisted emergency networks in disasters. IEEE Wireless Commun. 2019;26(1):45-51.

[46] Do-Duy T, Nguyen LD, Duong TQ, Khosravirad SR, Claussen H. Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications. IEEE J Sel Areas Commun. 2021;39(11):3411-3424.

[47] Coutinho RWL, Boukerche A. UAV-mounted cloudlet systems for emergency response in industrial areas. IEEE Trans Ind Informat. 2022;18(11):8007-8016.

[48] Lin H, Guo R, Ma D, et al. Digital-twin-based multi-scale simulation supports urban emergency management: a case study of urban epidemic transmission. Int J Digit Earth. 2024;17(1).

Research Article

[49] Khot S, Waghre AR, Abbas MH, Zubair M. Digital transformation of utility services using AI, IoT and ITSM practices: evidence from water, electricity and waste management systems. J Inf Syst Eng Manage. 2024;9(4s):3744-3755.

[50] Ansari MA, Khan SM, Abbas MH, Hassan SN. Strategic roadmap for AI-infused IoT adoption in national smart city programs with focus on future readiness, KPI frameworks, and digital transformation. J Inf Syst Eng Manage. 2024;9(4s):4077-4090. doi:10.52783/jisem.v9i4s.14760

[51] Sun W, Lei S, Wang L, Liu Z, Zhang Y. Adaptive federated learning and digital twin for industrial Internet of Things. IEEE Trans Ind Informat. 2021;17(8):5605-5614.

[52] Lu Y, Huang X, Zhang K, Maharjan S, Zhang Y. Communication-efficient federated learning for digital twin edge networks in industrial IoT. IEEE Trans Ind Informat. 2021;17(8):5709-5718.

[53] Lu Y, Maharjan S, Zhang Y. Adaptive edge association for wireless digital twin networks in 6G. IEEE Internet Things J. 2021;8(22):16219-16230.

[54] Cao B, Li Z, Liu X, Lv Z, He H. Mobility-aware multiobjective task offloading for vehicular edge computing in digital twin environment. IEEE J Sel Areas Commun. 2023;41(10):3046-3055.

[55] Ramu SP, Boopalan P, Pham QV, et al. Federated learning enabled digital twins for smart cities: concepts, recent advances, and future directions. Sustain Cities Soc. 2022;79:103663.

[56] Mahmud R, Pallewatta S, Goudarzi M, Buyya R. iFogSim2: an extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments. J Syst Softw. 2022;190:111351.

[57] Gupta H, Dastjerdi AV, Ghosh SK, Buyya R. iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments. Softw Pract Exper. 2017;47(9):1275-1296.

[58] Riley GF, Henderson TR. The ns-3 network simulator. In: Modeling and Tools for Network Simulation. Berlin: Springer; 2010:15-34.

[59] Sonmez C, Ozgovde A, Ersoy C. EdgeCloudSim: an environment for performance evaluation of edge computing systems. Trans Emerg Telecommun Technol. 2018;29(11):e3493.

[60] Silva Filho MC, Oliveira RL, Monteiro CC, Inácio PRM, Freire MM. CloudSim Plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In: Proc IFIP/IEEE Symp Integrated Network and Service Management (IM). 2017:400-406.

[61] Khan S, Khan SM, Rasheed M, Ansari SR. AI driven predictive cyber-risk scoring for IoT deployments in smart cities (machine learning for real-time risk scoring). J Inf Syst Eng Manage. 2024;9(4s):3881-3892. doi:10.52783/jisem.v9i4s.14552

[62] McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proc 20th Int Conf Artificial Intelligence and Statistics (AISTATS). 2017:1273-1282.

[63] Khan SM, Zubair M, Chan MYA, Fahim M. Anomaly detection in airport databases using generative AI for log and telemetry analysis: automated threat hunting via large language models. J Inf Syst Eng Manage. 2024;9(4s):4243-4253. doi:10.52783/jisem.v9i4s.14889