

# A SURVEY ON RESOURCE MANAGEMENT SOLUTIONS, CHALLENGES, AND EMERGING OPPORTUNITIES IN FOG COMPUTING

Firas Ghoneim

ATA Software Solutions, Beni Suef, Egypt

## ABSTRACT

*Fog computing positions computation and storage resources in close proximity to data sources, promising millisecond-scale responsiveness and data handling that the distant cloud cannot deliver. Over the past decade, researchers have proposed a constellation of mechanisms such as virtualisation, latency-aware service placement, predictive load balancing, rapid state migration, replica-centric fault tolerance, energy-adaptive scaling, QoS orchestration, and locality-sensitive caching, where each addressing a slice of the management challenge. However, these contributions often operate in isolation, optimise a single metric, or assume benign network and energy conditions, leaving unanswered questions about cross-layer interactions and deployment realism. This survey consolidates the state-of-art studies into a coherent pipeline view of fog resource management, tracing how hardware evolves into an elastic, fault-resilient micro-cloud. The analysis distils convergent design patterns, exposes hidden assumptions, and highlights goals among layers. Building on these insights the paper identifies cross-cutting research opportunities such as device-centric observability, federated resource markets, on-device continual learning, sustainable operations, and jurisdiction-aware SLA management. These paradigms must be addressed before fog infrastructures can underpin safety-critical and continent-spanning services. By mapping current achievements to unresolved challenges, the survey offers researchers and practitioners a clear agenda for the next wave of fog innovation.*

## KEYWORDS

*Fog computing, distributed virtualisation, service placement, load balancing, load migration, fault tolerance, resource scaling, quality-of-service, edge caching*

## 1. INTRODUCTION

Fog computing has emerged as the architectural bridge between the hyper-scale data center and the hyper-dynamic physical world, responding to a fundamental shift in application expectations. Here, users of augmented-reality goggles, autonomous vehicles, smart-grid controllers, and remote medical devices no longer view a two-hundred-millisecond round-trip as acceptable latency, nor will regulators tolerate a health record or industrial telemetry stream that meanders through unknown sovereign territories before returning actionable insights. Instead, these applications demand sub-second processing, deterministic jitter, and data locality, requirements that traditional cloud regions, i.e., located dozens or hundreds of kilometres from endpoints struggle to satisfy even when fronted by content-delivery networks or routing. Fog computing answers this by embedding compute, storage, and networking capabilities directly into the last-mile infrastructure. Yet delivering on that promise demands a comprehensive resource-management stack able to tame unprecedented heterogeneity and volatility. Resource managers must make millisecond-grade decisions about where to instantiate micro-services, when to

migrate or replicate them, how to down-clock a CPU to save carbon without breaking real-time guarantees, and which caching strategy minimises both hop count.

Recently, an expanding literature has been developed, proposing container engines tailored to single-board computers, tabu-search and genetic algorithms that collect service-function chains into patchy topologies, priority queues that stretch or shrink node fleets according to queuing-theory insights, swarm-inspired replicas that flutter between carriages of an urban light-rail system, and diversity-coded protection schemes that restore 5G slices within microseconds of a fibre cut. While these contributions advance the state of the art, they rarely interact. For instance, a placement algorithm may assume unlimited energy, a migration study may ignore how cache coherency is preserved, a fault-tolerance paper may posit reliable back-haul, and an energy-aware scaler may treat latency targets as soft hints rather than contractual obligations. The absence of a unifying perspective complicates reproduction and hampers the emergence of robust design patterns. Equally important, few evaluations extend beyond synthetic topologies or single-metric objectives, leaving open whether techniques compose harmoniously when combined in real deployments.

Motivated by this gap, this survey adopts a holistic lens, framing fog-resource management as a pipeline whose successive layers, i.e., virtualisation, service placement, load balancing, load migration, fault tolerance, resource scaling, quality-of-service enforcement, and caching, transform edge hardware into a responsive, energy-frugal, and regulation-aware micro-cloud. Within each layer, the survey catalogues representative studies, extracts shared architectural primitives, and teases out collisions among their assumptions. The analysis demonstrates, for example, how aggressive pre-copy migration can nullify the gains of energy-aware down-clocking when working-set churn is overlooked, or how blockchain-anchored fault logging inflates the network overhead that load balancers strive to reduce. By juxtaposing techniques across disciplines, the survey illuminates where incremental progress has become incrementalism, optimising nanoseconds while ignoring missing cross-layer glue and where genuine voids remain, such as observability frameworks that must operate opportunistically across disconnected fog islands, or SLA constructs capable of expressing per-packet privacy invariants alongside millisecond tail-latency ceilings. Out of this synthesis emerge research opportunities that cut across the surveyed terrain. The need for device-centric telemetry that streams encrypted micro-events without starving CPU cycles, real-time resource markets that allow autonomous domains such as municipalities and logistics fleets to lend and borrow idle compute under cryptographic escrow, continual-learning pipelines that refine models locally while bounding catastrophic forgetting and privacy leakage, energy-adaptive schedulers that fuse photovoltaic forecasts, thermal envelopes, and carbon-intensity curves into placement calculus; and SLA management engines that transform multidimensional obligations into machine-verifiable policies enforceable by automated orchestration. Addressing these opportunities demands collaboration between researchers, and network theorists, promises to elevate fog computing from prototype deployments to a dependable substrate underpinning continent-spanning, safety-critical services. This survey therefore sets out not merely to describe the current landscape but to chart a route through the remaining wilderness, equipping the community with both a clearer understanding of how existing pieces interlock and a forward-looking agenda to ensure that fog potential becomes practical reality.

## **2. RESOURCE MANAGEMENT IN FOG COMPUTING**

This section traces how virtualisation, service placement, load control, migration, resilience, elastic scaling, QoS enforcement and caching collectively convert scattered fog nodes into a dependable, low-latency utility. Each subsequent subsection dissects one of those levers.

## 2.1. Virtualization Management

Virtualization management is pivotal for realizing fog computing's potential. The work in [1] establishes a comprehensive virtual fog framework, integrating object, service, and network function virtualization across IoT, fog, and cloud layers. It concretizes fog features by low latency, high multitenancy, scalability via a layered architecture validated in smart living scenarios, demonstrating latency/jitter reduction and operational cost savings. Critically, it achieves seamless cloud-fog-things integration while remaining generic for 5G use cases. Accelerator virtualization poses distinct challenges at the edge. The study in [2] identifies multiple interconnected hurdles integrating accelerators as primary processors in edge stacks, scheduling amid heterogeneous resources, virtualization overheads under QoS constraints, and redesigning accelerators for non-HPC edge workloads. These demand architectural rethinking to harness edge accelerators effectively. For information-centric networks, [3] proposes cognitive network function virtualization (NFV) enabled by fog. It virtualizes on-demand caching and control functions, deploying them on fog nodes to enable dynamic content-aware resource configuration. This approach optimizes routing, caching policies, and security in decentralized environments. Simplifying fog deployment, [4] introduces fog function virtualization (FFV), coupled with node constellations and orchestration policies. This work abstracts infrastructure complexities, allowing developers to deploy services solely via application code. By virtualizing "things" as services and enabling horizontal fog communication, FFV democratizes fog access while reducing costs. Focusing on service chains, [5] designs a delay-aware SFC controller for fog environments. Unlike MEC-focused works, it optimizes container placement across cloud-fog hierarchies using Kubernetes extensions, ensuring bidirectional communication guarantees. Security in untrusted fog infrastructures drives in [6] leverages Intel SGX-secured containers. It supports two attestation modes, a full chain-of-trust for trusted hosts and hardware-backed enclaves for untrusted hosts. This ensures trusted execution without mandating third-party infrastructure modifications. Network-aware container placement is addressed in [7] via a genetic algorithm incorporating heterogeneous communication fabrics. It minimizes application response time by co-locating high-interaction containers and selecting optimal networking modes. To maintain proximity for mobile IoT, [8] demonstrates multi-container service migration. Using an AR application, it statefully migrates an MQTT broker and statelessly migrates processing containers, showcasing QoS preservation during device mobility via live testbed validation. Orchestration needs are analysed in [9], which evaluates container performance on fog nodes and proposes a Docker swarm-based framework. It balances orchestration overhead with IoT application constraints, enabling efficient service deployment across distributed fog resources. A systematic mapping in [10] surveys container orchestration on SBC-based fog. It reveals limited support for dynamic mobility and real-time workloads, underscoring the immaturity of orchestration solutions for resource-constrained fog. The Con-Pi framework [11] enables distributed container management across SBCs. It supports renewable energy integration and microservice orchestration in mesh/hierarchical topologies, improving response time and energy efficiency. Finally, [12] advances container-based SFC mapping by exploiting VNF dependencies. Its dependency-aware scheme batches requests and optimizes placement, outperforming VM-based approaches in resource utilization, processing time, and request acceptance within resource-constrained substrates.

## 2.2. Service Management

Service management constitutes an emergent research domain focused on optimizing the allocation of application components, i.e., monolithic services, microservices, or VNFs across resource-constrained fog nodes. This discipline addresses the triple constraint of stringent QoS (latency, reliability), dynamic infrastructure (node heterogeneity, mobility), and operational efficiency (energy, cost) intrinsic to edge environments. Placing services at the edge is the

moment where fog computing either fulfils its promise of “cloud-like power without cloud-scale delay” or collapses under mis-located workloads, empty nodes, and congested links. Because fog devices are heterogeneous, energy-limited, and scattered over wide areas, the placement problem must respect latency budgets, mobility, fault likelihood, micro-service dependency graphs, all while keeping migration and signalling overhead below the very delays it tries to avoid. The work in [13] frames Industry 4.0 workloads as “context-aware” objects whose input-data rates, packet sizes, and deadline strictness vary by sensor line. It cross-checks those stream signatures against each fog node’s remaining head-room, selecting the host whose compute–network mix yields the shortest end-to-end service time. In [14], services roam a three-tier wireless-multimedia fog as users move and a dynamic strategy constantly re-examines whether the current host still minimises joint energy and delay. Raspberry Pi test-beds running *OpenWrt* confirm that the algorithm’s quick re-placement cuts video-frame lag and halves joule burn relative to static pins, turning fog into a practical “USB-like” service dock. Tabu search powers the SFC engine in [15]. It explores swaps of VNFs across nodes, banning recently tried configurations to escape local minima, and pairs that with load-aware shortest-path routing. Batch requests in [16] are sorted by highest-resource, lowest-resource, or random order before mapping. An enhanced breadth-first search chooses the shortest path while a weighted-least-connection rule balances load. Tests show that processing high-demand chains first maximises overall acceptance, whereas low-demand-first smooths utilisation. Delay, cost, and load-minimisation heuristics in [17] tailor node and link selection across small lower-tier and beefy upper-tier fogs. Results indicate that the delay-centred heuristic excels for real-time IoT, cost wins when bandwidth prices dominate, and load balancing prolongs cluster life, underscoring the need for objective-aware placement menus. The work in [18] recasts placement as a mixed-integer multi-objective optimisation trading response time against residual capacity. A moDSP algorithm searches that landscape and, inside iFogSim, lands services on “just-fit” nodes that keep median latency down while boosting utilisation. The work in [19] revisits the plug-and-play notion for wireless multimedia but emphasises three-tier bandwidth asymmetries, coarse placement decisions happen at the backhaul-aware top tier, fine re-placements near users. The strategy trims both energy and hop-count versus prior wireless-fog schemes by pre-emptively shunting bulky streams to nodes on wide links before congestion materialises. Reliability headlines in [20] include micro-service graphs that are modelled as  $k$ -out-of- $n$  serial–parallel systems, and a two-level hybrid of particle-swarm (throughput-aware) and NSGA-II (redundancy-aware) searches for replica sets that meet availability targets under both independent and correlated failures.

Delay-cost trade-offs drive DAFC in [21], which slots chains into a single fog layer for tight deadlines, cluster-head fogs for tolerant jobs, and cloud only when edge is exhausted. A heuristic shortest-path plus load filter meets 15–40% more traffic than pure fog, yet still slashes delay over cloud-only mapping, proving the value of hybrid tactically but fog-first philosophically. LSTM in [22] predicts which VNFs will surge next, pre-fetching popular ones onto high-capacity fog nodes and relegating rare functions to lighter boxes. By front-loading caches, the system doubles hit ratio and carves substantial energy savings. Parallel collaboration underpins *PColl* in [23] where each fog cluster solves its placement sub-problem then duplicates undeployed-application requests to neighbours, letting a heuristic “first finisher” claim the job. Authors in [24] formalise placement across all common service-topology variants and feed it to *ChocoSolver*. It returns solutions competitive with hand-tuned heuristics and accepts new constraints with minimal code. A single-tier heterogeneous layout in [25] consists of super-fog nodes shoulder heavy compute while ordinary nodes deal with moderate tasks. Hooke-Jeeves search then packs VNFs to exploit super nodes first, achieving lower delay and energy than multi-tier hybrids as traffic never leaves the immediate edge. A teaching-learning-based optimisation emulator in [26] alternates “*teacher*” (best solution) and “*learner*” phases to refine placements that minimise diagnostic latency while respecting privacy. Simulation indicates swift convergence and resilience to workload swings, critical for emergency telemetry.

A cloud–fog middleware in [27] watches active/idle cycles and formulates placement as a combinatorial plan that off-loads hot nodes while waking just enough idle peers. Optimal redistribution spreads heat and chops overall power draw without breaching QoS. Genetic-algorithm meets Monte-Carlo in [30], where chromosomes encode multi-service mappings, fitness blends QoS assurance with cost, and random sampling estimates stochastic link delays. Multilevel clustering is shaped in [31], i.e., geo-location forms macro clusters, then CPU/memory/bandwidth similarity sub-clusters nodes. Services tagged as CPU-, memory-, or bandwidth-bound drop into the matching bucket, cutting placement time and flow completion by over 20 % compared to geo-only clustering. A multi-tier fog for smart-grid traffic in [32] leverages time-of-day and QoS tags to push services outward or inward dynamically. Simulations reveal latency and energy drops versus cloud-centric deployment, underscoring the power of temporal awareness. Deep-learning sector prediction in [33] reduces initial access overhead, indirectly freeing compute cycles and link budgets for placement engines, illustrating cross-layer synergy. Matching theory surfaces in [34], where VNFs and fog nodes draft preference lists, and two adaptations (chain-aware, chain-agnostic) reach stable matchings that slash completion deadlines and outages versus greedy and potential-game baselines. Graph partitions is proposed in [35]. Applications are first mapped to well-connected device communities, then their interrelated services land on the highest-fitness nodes within that sub-graph. A hierarchical-autonomous fog in [36] organises nodes into logical zones and employs a fully distributed cost-minimising selector that works without global state, outperforming centralised and heuristic peers on both latency and overhead in large topologies. Finally, [37] recasts placement as an iterative combinatorial auction without a central auctioneer. Here, fog nodes bid for bundles of micro-services, applications allocate in greedy rounds, and the process converges in bounded steps. Numerical tests reveal lower energy and bandwidth cost than existing distributed heuristics while preserving privacy and bounded runtime.

### 2.3. Load Balancing

As fog nodes run on limited cores and power budgets, an overloaded subset can choke the whole locality. Load-balancing research therefore strives to push tasks toward the “just-enough-idle” neighbourhood node, doing so with microscopic delay budget, minimal probing and without wasting back-haul capacity that fog was meant to save. The studies below show how designers borrow SDN, graph theory, randomised algorithms and deep learning to share work fairly while still honouring real-time contracts. The work in [38] merges IoV, SDN and parked vehicles. Namely, an SDN controller surveys city-wide demand while local fog managers act on second-by-second queue lengths. When a roadside unit nears saturation, time-sensitive jobs are steered first to idle processors inside nearby parked cars, falling back to the cloud only if no edge capacity remains. In [39], physical gateways are atomised into uniform virtual machines, edges of the communication graph carry traffic weights, and a dynamic Fiduccia-Mattheyses cut re-partitions the graph whenever hotspots form. Because only a sliver of VMs moves on each adjustment, migration overhead stays low while utilisation remains near-optimal across hierarchy levels created by the earlier cloud atomisation step. Power-of-random-choices to the edge is adapted in [40]. Each node with queue length above a certain threshold samples random peers and diverts the job to the least-loaded among them, but only if that peer’s queue is shorter. The study in [41] offers a systematic taxonomy of fifty-plus fog load-balancing papers, categorising them into approximate, exact, fundamental and hybrid families; it catalogues the KPIs each camp optimises, the simulators they rely on, and pin-points blind spots, i.e., energy-delay trade-offs and explicit mobility support that later works begin to address. The authors in [42] cast workload allocation in an IoT–fog–cloud triad as a queuing-drift minimisation, train an LSTM to forecast burst arrivals, and use Lyapunov drift-plus-penalty to decide the fog/cloud split one step ahead. Latency itself becomes the balancing metric in [43]. Each node measures its own user-perceived service delay and migrates tasks to neighbours until latencies converge. A continuous-time

differential model proves convergence to a Wardrop equilibrium, and a twenty-Pi cluster shows the heuristic evens out delay within four percent across heterogeneous hardware while moving far less data than CPU-based balancers. The proposed SFC provisioner in [44] first embeds chains to minimise end-to-end delay and then triggers live VNF migration whenever a node's utilisation breaches a load threshold, seeking the lightly-loaded neighbour that keeps path delay low. Results show higher admission and lower latency than delay-only or load-only benchmarks, giving NFV operators a joint placement-and-balancing recipe. Finally, the hierarchy-descending strategy in [45] flips typical bottom-up mapping. Under light load, grouped VNFs are placed on resource-rich upper-tier nodes, producing short paths and leaving lower tiers idle; as traffic swells, the algorithm gradually cascades single VNFs downwards, thereby postponing congestion and shortening provisioning delay compared with schemes that always start mapping at the lowest tier.

## 2.4. Load Migration

When an edge node tips into saturation the entire latency advantage of fog evaporates, yet pushing overflow all the way to the cloud is rarely acceptable for real-time traffic. Migration research therefore asks how to sense overload early, pick a viable neighbour, and move state quickly enough that users do not notice. The studies below chronicle a shift from single-node, delay-only heuristics to probabilistic foresight, mobility awareness, and dual-phase diffusion strategies that weigh delay, cost, and energy in tandem.

The work in [46] casts dynamic CPU pressure itself as a first-class trigger. Each node runs a finite-state machine whose transition probabilities forecast future load, and an ILP then chooses the nearest edge with the lowest expected utilisation. A Karush-Kuhn-Tucker derivation yields closed-form thresholds that MobFogSim replay shows cutting average delay by nearly a third and halving migration count versus reactive "move-to-nearest" baselines. Paper [47] argues VM moves are too heavy and instead prototypes live container migration with LXD, CRIU checkpoints, and ZFS copy-on-write layers. Experiments on a three-node test-bed keep pause time below one second while CPU overhead stays under seven percent, proving that lightweight state capture can fit inside fog's narrow energy envelope. Reference [48] introduces a coarse-grained diffusion scheme for NFV chains: whole VNF bundles shift only when utilisation crosses a threshold, and destination choice toggles between least-load and least-delay policies according to application class. Simulations show the method slashes control traffic by half and holds latency within five percent of fine-grained strategies even under torrent-level traffic bursts. Luus-Jaakola search powers the meta-heuristic in [49]. Here an optimisation model minimises migration delay while maximising success probability, and the algorithm perturbs candidate nodes in shrinking hyper-rectangles until it finds a lightly-loaded host.

Vehicular fog animation appears in [50] with FEE, an online algorithm that blends hidden-Markov trajectory prediction, interference-aware server-load weighting, and per-slot latency minimisation. Rome-taxi traces confirm a 15 percent latency cut and twelve-point rise in deadline-hit ratio against four state-of-the-art vehicular baselines. LBATSM in [51] treats migration as a two-phase puzzle, a heuristic first selects tasks whose removal yields the steepest utilisation drop, then a modified binary PSO with a mirrored sigmoid transfer maps them to targets. Authors in [52] use coordinated generalised pattern search to migrate VNFs either exclusively or in a shared mode. The policy favours rapid admission recovery, while shared mode trims bandwidth, giving operators a tunable knob between speed and cost. Mobility dictates the mixed-integer plan in [53]. Here, sojourn time of each user follows an exponential stay model, a Gini-coefficient selector chooses the fog whose dwell time maximises expected revenue, and a

GA allocator divides CPU. Before a host actually saturates, the work in [54] copies cold VNF images to standby neighbours, so fail-over is merely a flow reroute.

Hybrid-grained rebalancing in [55] adapts chunk size to traffic density and fine moves under light load, coarse bundles when bursts loom, achieving the fewest iterations and lowest delay across all three density regimes, and demonstrating that one granularity never fits all. Users are grouped in [56] into overlapping geographic and virtual cliques, and resources migrate within the smallest community that still meets latency targets, thereby reducing cross-region traffic and maintaining optimal placement as mobiles drift. Docker's gossip layer in [57] elects the closest healthy surrogate after a blackout, automatically redeploying containers and restoring service in under three seconds, where no external orchestrator is required. Paper [58] leverages a heterogeneous super/ordinary-fog topology. The entire service chains first crowd onto resource-rich super nodes, then excess VNFs diffuse like heat into idle ordinary peers once a saturation alarm rings, lifting utilisation and saving power without cloud core. Finally, [59] proposes three dual-purpose strategies. Content-specific that moves excess only to nodes already hosting related VNFs, buffer-based reserves a blank standby node for instant relief, and a hybrid begins with the buffer then switches to content-specific as load grows.

## 2.5. Fault-Tolerance

The work in [60] tackles replica placement by mining each fog node's failure history, classifying devices into high- and low-risk groups with a K-nearest-neighbour model, and then copying only the hottest files from brittle nodes onto the most reliable peers. Security joins reliability in [61], where the authors blend attribute-based access control with a private blockchain ledger that doubles as a fault-tolerant log. Each fog gateway authenticates via a wallet, and a min-min load balancer replicates the micro-authenticator onto multiple nodes, so even a local crash leaves the ledger intact; encryption of policy attributes hides user identities from the public chain, giving the system confidentiality, resilience, and scalability in a single strike.

The study in [62] shifts focus to smart-city mosaics. It catalogues how node outages ripple through traffic lights, water grids, and pollution sensors, then sketches a middleware layer that embeds failure detection, task migration, and state checkpoints as reusable services. Energy and timeliness take centre stage in [63], where a modified particle-swarm optimiser schedules IoT tasks while maintaining a reactive fault-tolerance rule. If a node fails mid-execution, the task is immediately resubmitted to the best remaining candidate. Authors in [64] present a dynamic fault-tolerant learning Automata scheduler. Each automaton adapts its replication versus re-execution decision as runtime statistics drift, backed by a normal-distribution model of task length. The algorithm converges on near-optimal energy-reliability trade-offs. SFCs dominate carrier edge clouds, so [65] designs two polynomial-time heuristics (end-to-end and intermediate restoration) that reroute broken chains only after a failure strikes. The study in [66] develops pre-fault survivability mapping for the same SFC context. Greedy but delay-aware heuristics embed primary and shadow VNFs across multi-tier fog layers; batch-sorting of incoming requests further improves admission. Healthcare IoT inspires the REDPF framework in [67]. Here, a hybrid directed-diffusion plus limited-flooding protocol recollects lost sensor data, while a reduced-variable-neighbourhood-search queue orders packets by medical urgency instead of arrival time. Markov chains guide the fault model in [68]. Each fog node's state transitions define a continuous-time chain whose steady-state probabilities estimate future failures. An improved simulated-annealing optimiser then selects the cheapest subset of standby nodes whose combined reliability meets a global target. Finally, [69] brings 5G NFVs into the picture with diversity coding. Instead of pre-computing alternate paths or retransmitting lost packets, a single protection

stream encodes parity across all primary flows. When a link or node fails, decoding at the destination yields instant recovery without feedback.

## 2.6. Resource Scaling

Resource-scaling research seeks algorithms that predict traffic pressure before queues expand, stretch just the right slice of CPU or replica count, and then taper back without violating real-time constraints. The work in [70] grounds that vision in classical queuing theory. Tasks arrive into two non-pre-emptive M/M/1 priority lines, one for delay-sensitive traffic and one for background jobs. A closed-form expression links queueing delay to the number of active fog nodes, so a controller periodically solves for the minimal node count that keeps the high-priority mean waiting time below a user-tuned bound. If traffic ebbs, idle fog boards are placed into low-power sleep, yielding a 14.5 % energy saving; if bursts hit, additional boards boot and instantly join the priority server pool. Where [70] toggles node granularity, the authors of [71] zoom into silicon itself and treat CPU frequency as the actuation lever. They model each heterogeneous fog board as a DVFS-capable core whose dynamic power follows the familiar cubic dependency on clock speed, then pose a minimisation: choose the lowest frequency that still clears the task-delay requirement under current arrival rate, data size and word-length constraints. Using proximal operators and Lagrange multipliers, they derive a closed-form scaling factor that can be computed in micro-seconds, allowing per-request adaptation; numerical experiments on Consumer-IoT traces cut energy at least 36 % while keeping every job below its latency budget. FScaler in [72] replaces analytic knobs with learning: the authors encode horizontal container scaling as a Markov Decision Process whose states include current replica count, CPU-RAM saturation, and recent demand trend, while actions add or remove pods and place them on whichever fog node minimises a composite cost. A SARSA agent explores this space online, guided only by observed latency penalties and deployment costs, and converges to an optimal policy after a few hundred episodes on a real Comcast-video trace. Once trained, FScaler anticipates surges and spins replicas ahead of time, slicing p95 response latency by 28 % over Kubernetes' reactive HPA and reducing needless scale-outs by throttling back during demand troughs.

## 2.7. Quality-of-Service (QoS) Management

Fog systems sit directly on the latency-critical path between sensing and actuation, so any mismatch between workload surges and the micro-resources at the edge translates instantly into deadline misses. an outcome that collapses industrial control loops, degrades tele-medicine. QoS research therefore pursues two intertwined goals, predict when and where pressure will arise, and orchestrate compute, storage, and bandwidth so that the first hop always honours the application's timeliness contract. The work in [73] recognises that edge schedulers too often chase a single metric. Instead, the authors build a robust resource-management engine that minimises execution time, link utilisation, energy draw, and end-to-end latency simultaneously. They weave those objectives into a composite cost, derive gradient rules that drive container migrations and DVFS steps, and implement the loop in iFogSim across healthcare and smart-traffic traces the engine trims average delay by a quarter while holding energy inside a tight five-percent envelope. The study in [74] equips a swarm of UAVs with two distinct personalities some drones host fog micro-datacentres, others act as mobile relays and lets a path-loss-aware planner choreograph flight routes so that every task finds the closest compute slot. The survey in [75] dissects multiple resource-management studies, sorting them into placement, scheduling, allocation, and provisioning buckets and mapping each to the QoS knobs it controls. Meta-analysis highlights that only a handful embrace holistic orchestration across compute, storage, and radio, and even fewer propagate user-level QoS back into placement loops. Focusing on medical telemetry, the authors of [76] build OpenHealthQ, an SDN-enhanced fog fabric where

OpenFlow queues carry priority codes rooted in HIPAA data classes. A Ryu controller injects per-flow bandwidth guarantees on demand. Testbed experiments under COVID-scale traffic confirm zero packet loss for emergency ECG streams, a 44 % shorter response time for high-priority flows, and negligible jitter inflation for best-effort traffic, proving policy-driven slicing can coexist with commodity edge switches. Multimedia latency motivates the QoS-aware streaming stack in [77]. Bandwidth between fog and client is predicted via a lightweight ARIMA filter, while a cross-layer broker adaptively selects video representations and may even transcode at the edge. Large-scale emulation shows 47 % fewer re-buffer events than cloud-only DASH and a 30 % gain in fog-node utilisation because transcoding is triggered only where path predictions warrant the cost.

FOGPLAN in [78] tackles service placement as a rolling horizon problem. Every few seconds a fog node observes current RTTs and queue depths, feeds them into a greedy knapsack that weighs penalty against deployment cost, and decides whether to spawn, migrate, or retire each micro-service. Despite relying on nothing more than local statistics, the planner adds under one millisecond extra latency and slices leasing cost by about a third when replayed against real city-bus mobility traces. Latency violations inevitably arise, so the QoTAM strategy in [79] switches from allocation to rescue: it monitors each task's slack, identifies endangered jobs, and migrates them to the fog node that maximises residual capacity minus transmission delay. Compared with static allocation the migration logic lifts task-completion ratio ten percentage points and proves especially resilient when virtual-machine failures burst above five per hour. Finally, the UAFN framework proposed in [80] argues that dense UAV-fog collectives should be treated as a first-class substrate: a global planner slices the sky into service zones, assigns data acquisition to relay drones and heavy processing to fog-capable peers, and continuously replans under battery constraints. Trace-driven studies for disaster-response scenes demonstrate that end-to-end latency stays below 70 ms while coverage and throughput scale linearly with drone count, thus illustrating how aerial fogs convert line-of-sight mobility from a liability into a QoS asset.

## 2.8. Caching

Caching sits at the heart of fog computing because the entire paradigm is highly impacted by milliseconds. Once data or services are pre-positioned a single hop from the consumer, back-haul congestion evaporates, uplink energy plummets, and application designers can finally treat the edge as a predictable micro-cloud rather than a best-effort relay. Yet fog nodes are small, user tastes mutate hourly, and mobility rearranges the demand map faster than any cloud scheduler can react. Against that backdrop, the following studies attack the same core tension, i.e., what to cache, where, and for how long from different technical angles.

The work in [81] attacks the surge of 5 G IoT requests with a many-to-one matching game that views each fog cell as a scarce radio-compute bundle and each device as a latency-hungry suitor; by applying deferred-acceptance rounds the authors prove the system always converges to a stable partition and, in simulation, they show cache-hit probability climbs while back-haul shrinks. Firefly intelligence animates the study in [82]. Each cache is a luminous agent whose brightness encodes hit-ratio gain, and multi-objective flight paths steer replicas toward nodes where they jointly maximise hits, minimise intra-fog link stress, and cap query time. Millimetre-wave F-RAN topologies motivate the Nelder–Mead placement in [83]. Here the simplex search wanders across potential RRH–fog pairings until it finds the one that admits the largest library given power, cost and beam-forming constraints. The work in [84] revisits the same mmWave scene but contrasts two opposing philosophies: clustered caching hoards popular VNFs inside a hand-picked RRH clique, whereas distributed caching sprinkles them network-wide.

Fog servers are scarce, so [85] swaps the ubiquitous shortest-path heuristic for a Steiner-tree planner that chains requester, content pairs through a lightweight subset of inter-fog links; by proving the tree's weight lower-bounds total caching cost, the authors rationalise why their placements incur less bandwidth and shorter paths than hop-by-hop greed. Mobility drives the multi-layer vehicular blueprint in [86]. Data begin life in a top-tier roadside cache, hitch rides in passing cars as transient replicas, and finally trickle down into on-board memories near expected requesters; extensive SUMO traces show the approach cuts miss rate and backbone load because cars themselves become roving fog nodes. Privacy and scale intertwine in [87]. The authors push popularity prediction down to IoT devices via federated learning, integrate differential noise for provable privacy, then let a hierarchical planner spread content across neighbour-to-neighbour F-AP coalitions and an over-watch BBU tier. Content-centric networks enter the picture in [88], where sensor clusters expose CCN names and a distributed fog broker orchestrates which node copies which chunk; energy models and cache-replacement curves reveal that Leave-Copy-Everywhere wastes power, leave-copy-down hoards bandwidth, whereas the proposed DFCS keeps both metrics in check by negotiating replica placement in-band with CCN interests.

The authors in [89] pivot to user-side energy by treating each phone as a potential micro-cache; an analytical two-hop power model ranks whether cloud-edge or peer-to-peer transfer wins for a given file, and a multi-population GA then selects the replica map that slashes aggregate transmission joules. Popularity-stratified placement guides study [90], Files fall into hot, warm and cold tiers, and fog clusters elect highly connected leaders to store the hottest tier while mid-active and sleep-prone nodes hold gradually colder sets. The scheme trims radio energy by deactivating seldom-needed cells without starving remote users of trending content. The work in [91] fuses a three-layer cloud-fog-user graph with hierarchical federated learning: user devices cluster via hedonic coalition games, predict future hits locally, and push only model deltas up the hierarchy. Computational tasks themselves become cachable artefacts in [92]. Here fog nodes play a coalition game, decide whether to admit a neighbour's overflowing DAG, and share pay-offs according to shapley-fair splits. Simulation shows the grand coalition forms only when latency penalties justify the collaboration, ensuring resources are not squandered on perfunctory partnerships. Integrity rather than placement drives FogAudit in [93]. Bilinear-map tags and a lightweight consensus protocol allow untrusted fog caches to audit one another, expel malicious peers and restore lost chunks without cloud guardians.

In [94], probabilistic caching is coupled with adaptive modulation. A joint optimisation selects both which file each node stores and which QAM level it uses to serve it, convex relaxations yield closed-form policies, and Monte-Carlo trials in heterogeneous layouts show double-digit energy savings for identical throughput. The work in [95] clusters groups of fog nodes by similarity, and applies collaborative filtering that forecasts the next-hour hit list, and intra-plus cross-layer cooperation, which allow any miss in one tier to be satisfied by a neighbour before falling back to cloud. Simulations report simultaneous gains on hit ratio, user satisfaction, latency and watt draw. Finally, authors in [96] argue that also application semantics (not just popularity) should steer replica choices. Namely, by tagging data with its originating IoT vertical, the system pre-fetches smart-health streams ahead of entertainment traffic during critical windows, trading a modest bandwidth hike for 30 % faster response and higher hit rates.

### 3. OPEN RESEARCH AND OPPORTUNITIES

Fog computing has graduated from proof-of-concept testbeds to deployments that shoulder real industrial monitoring, multimedia, and vehicular workloads. Yet the survey in Section 2 makes clear that today's solutions remain siloed, each optimising a narrow slice of the stack—virtualisation, placement, scaling or recovery—while leaving systemic questions unanswered.

Turning these isolated advances into a dependable, sustainable, and economically sound fabric demands fresh investigations that cut across layers, administrative domains, and trust boundaries. The following research directions highlight where new theory, protocols and toolchains can unlock the full potential of cloud-like services at planetary edge scale.

### **3.1. Congestion Control**

Back-haul congestion is no longer the main bottleneck. Today it is the 100-meter hop between dozens of co-located fog nodes that share unlicensed spectrum, low-end Ethernet switches, and flash-based storage pools. Conventional TCP pacing or SDN queue weights cannot react quickly enough when AR/VR bursts, sensor floods, and background model updates collide in the same micro-datacentre. Future research must combine link-layer awareness with workload semantics: radios could expose real-time SINR and contention windows to the scheduler, which would defer non-interactive micro-services or redirect replicas to under-utilised neighbour nodes before queues back up. At the storage tier, write-back caches can adopt priority admission control, admitting telemetry frames only if a probabilistic admission function predicts they will meet their SLA budget, otherwise rerouting them opportunistically. Control-plane signalling must remain lightweight; gossip-based rate feedback encoded as one-bit congestion marks in service-mesh headers could inform all senders within tens of milliseconds. Ultimately, a cross-layer congestion loop—spanning Wi-Fi 6/7 airtime, edge switches, and container queues will be indispensable for sustaining hard 20-millisecond deadlines even when bursty edge workloads share the same fog substrate.

### **3.2. Resource Monitoring**

Fog schedulers can only act as fast as their sensors report. Scraping CPU and temperature every few seconds—standard in data-centres—misses the millisecond swings that matter when containers migrate or radios fade. Upcoming work must push observation into the kernel with eBPF probes that sample context-switch latency, cache storms and per-packet RSSI at micro-second granularity yet sip power on watt-scale ARM boards. Energy prediction is equally critical: tiny neural models embedded in the power-management unit can forecast battery and solar yield hours ahead, letting controllers admit jobs only if projected draw stays within a safe reserve. Because raw traces often contain sensitive data, secure multiparty aggregation and homomorphic sketches should summarise node health fleet-wide without revealing individual histories. Finally, pairing these rich streams with explainable AI can translate into a human-readable alert, closing the loop between low-level telemetry and policy-level action.

### **3.3. Data Synchronization**

Fog computing elevates the difficulty of keeping distributed state coherent: nodes talk over lossy Wi-Fi, hop between 5G slices, or drop to BLE when a smart pole's fibre back-haul fails. Classic pre-copy memory streaming cannot keep pace with gigabit video analytics whose object detectors update weight maps every few milliseconds. A fog-aware alternative is layer-aware semantic diffing. Instead of shipping raw pages, the container runtime would decompose each micro-service into logical objects—model checkpoints, inference queues, user sessions—and transmit only field-level deltas. Content-defined chunking plus vector-clock ordering would ensure that even when packets arrive out of order the reconciliation engine can stitch a causally correct state. Stateless VNFs can push their ephemeral counters into conflict-free replicated data types, allowing multiple replicas to merge without a coordinator; stateful services can split working sets into cold segments dripped over BLE beacons and hot deltas blasted via Wi-Fi 6E bursts, exploiting multipath diversity while respecting battery budgets on solar-powered gateways.

Orchestrators will need formal causal-consistency contracts so that, when a drone briefly loses contact during hand-off, synchronisation resumes without violating application invariants. Machine-learned network predictors could feed these contracts with link-quality forecasts, letting migration schedulers decide whether to defer, compress harder, or opportunistically switch to a passing vehicle's 5G sidelink. The grand challenge is to blend these techniques into a self-optimising pipeline that keeps state fresh enough for real-time control yet frugal enough for street-level power and spectrum constraints, turning volatile fog substrates into dependable substrates for live analytics, fail-over, and collaborative learning.

### **3.4. Data Integrity and Validation**

Fog nodes sit on lamp-posts, in cafés, inside buses and even atop drones—locations where physical access is easy and supply-chain provenance far murkier than in a locked cloud datacentre. Every hop that data take through this mesh therefore needs built-in, tamper-evident safeguards. Lightweight yet cryptographically strong validation pipelines should travel with the data, not live exclusively in the core. Streaming zero-knowledge proofs can accompany each transformation—de-identifying faces, filtering sensor outliers, aggregating health telemetry—allowing downstream fog or cloud consumers to verify that the operation was performed faithfully without re-executing the algorithm or seeing the raw inputs. These proofs can be rooted in hardware enclaves such as ARM TrustZone or RISC-V Keystone, but orchestration layers must also reason about correlated risk: if several poles share the same firmware lineage or supply-chain, replicating shards across them offers little extra assurance. Probabilistic attestation graphs that capture such hidden dependencies can guide replica placement, audit cadence and quorum size. Merging these integrity guarantees with content-addressable storage yields a self-verifying data fabric where every object carries a cryptographically bound provenance trail from sensor origin, through each fog hop, to final analytics ready for regulators and auditors to inspect on demand. This fog-grade integrity layer is the missing trust scaffold that will let safety-critical AI pipelines, privacy-sensitive health applications and city-wide digital twins rely on computations performed out in the wild, far from the comfort of sterile server rooms.

### **3.5. Service-Level-Agreement Management and Fog-Grade Assurance**

Guaranteeing performance at the cloud edge was hard enough when contracts revolved around “three nines” availability and monthly mean latency, but the very fabric of fog computing invalidates those coarse metrics. Micro-datacentres mounted on lamp-posts, buses or base-stations face wildly fluctuating load, spotty back-haul and jurisdiction-specific privacy mandates. Therefore, a contract that merely promises an average is meaningless when a single 40 ms stall can derail an AR overlay or interrupt closed-loop industrial control. Future SLAs must shift from retrospective bookkeeping to real-time, tail-centred precision, spelling out per-request ceilings on end-to-end latency, jitter envelopes over sub-second windows, bounded loss for bursty sensor streams and cryptographic guarantees that packets tagged with a region-of-origin label never traverse disallowed hops. Enforcing those multidimensional promises demands an event-driven control plane that fuses live observability feeds with predictive analytics. Lightweight models running on the same fog fabric can extrapolate queue lengths, radio congestion and thermal headroom minutes ahead, then issue proactive mitigations: spinning an extra micro-VM replica on a neighbouring pole-router, nudging DVFS governors to turbo frequencies for the next camera burst, or leasing micro-spectrum slices from a municipal wireless exchange for the duration of a rush-hour surge. Because resources near the users are finite, SLA enforcement will increasingly embed market semantics. Token-bucket or micro-payment layers can incentivise cooperative degradation—during a cardiac tele-surgery session, a digital billboard operator might voluntarily yield compute credits in exchange for notarised goodwill tokens redeemable in quieter hours. Orchestration stacks will need solvers that ingest these tokenised preferences plus formal SLA

clauses expressed in machine-readable policy languages and compile them into placement, scaling and routing actions that are provably safe under worst-case contention. Just as crucial are transparent auditing pipelines: signed telemetry digests and zero-knowledge proofs can allow providers to publish cryptographically verifiable compliance logs without revealing tenant secrets. Public benchmarks and certification suites, analogous to PCI-DSS for payments or ISO 26262 for automotive safety, must evolve to stress-test fog infrastructures under bursty, locality-sensitive loads, ensuring that claimed micro-SLA capabilities hold across firmware updates and hardware refreshes. When such continuous, verifiable and economically nuanced SLA management matures, fog computing will graduate from today's best-effort convenience layer to a fully-fledged critical-infrastructure substrate on which healthcare, transportation and finance can stake mission-critical operations with quantifiable, enforceable confidence.

### **3.6. Fog-Native Intelligence and on-Device Continual Learning**

The latest generations of single-board accelerators and low-power GPUs mean that fog nodes can now refine models in situ rather than shuttling data to the cloud, yet the field still lacks a cohesive blueprint for lifelong learning under the peculiar constraints of fog computing. Each node observes a skewed, often non-IID slice of the environment—traffic cameras capture local weather patterns, warehouse robots see only their aisle—and those micro-datasets evolve as seasons, lighting and user behaviour drift. Research must therefore design continual-learning pipelines that incrementally fine-tune models locally, exchange only highly compressed gradient or weight deltas and, crucially, guard against catastrophic forgetting when concept drift or sudden domain shifts occur. Memory-lean optimisers that fit within a few megabytes of SRAM, sparsity-aware gradient codecs that exploit temporal redundancy, and asynchronous aggregation protocols that tolerate battery-induced stragglers or intermittent back-haul links are pressing engineering gaps. Equally vital is a governance layer: federated updates must carry cryptographic provenance, differential-privacy noise budgets and automated “safe-region” checks so that rogue or poisoned edge contributions cannot degrade global accuracy or introduce bias. Trust-region scheduling could throttle update rates when confidence drops, while federated replay buffers may re-inject rare edge cases to stabilise training without full raw-data disclosure. Cross-node anomaly consensus—where neighbouring fog clusters vote on suspicious parameter shifts—promises an additional safeguard, but formal guarantees for real-time, adversarial, non-IID streams remain elusive. Solving this triad of algorithmic efficiency, privacy preservation and verifiable safety will unlock self-improving fog services that adapt to hyper-local nuances in agriculture, autonomous logistics and urban sensing without ever surrendering data sovereignty or violating stringent latency budgets.

### **3.7. Sustainable, Energy-Adaptive Fog Operations**

As thousands of fog micro-datacentres sprout on lamp-posts, rooftops, and roving delivery vans, sustainability shifts from a footnote to a first-class design goal. Tomorrow's schedulers must treat carbon, watt-hours, and thermal headroom as co-equal constraints with latency and reliability. This means ingesting fine-grained solar-irradiance forecasts, dynamic electricity-price feeds, and battery state-of-health metrics, then reshaping service placement and replication plans so that peak compute gravitates toward nodes basking in sunshine or drawing from off-peak grids. Co-designing those policies with variable-voltage hardware and adaptive workloads opens deeper savings: a gateway that predicts looming cloud cover can pre-emptively down-clock its AI accelerator, while upstream cameras trim frame rates or switch to event-driven codecs, all without violating QoS. Life-cycle carbon accounting complicates the optimisation, as moving a service to a greener node may trigger extra radio hops or accelerate battery wear that later demands truck-roll maintenance. Embedding embodied-emissions costs and disposal impacts into

real-time controllers will nudge ecosystems toward recyclable enclosures, second-life batteries, and carbon-aware application code that gracefully scales fidelity to match the energy envelope. Building such multi-horizon, multi-metric feedback loops, where workload orchestration, power electronics, and sensing hardware negotiate continuously remains an open research frontier and a prerequisite for climate-resilient fog infrastructures.

### **3.8. Fog-side Intelligence and on-device Continual Learning**

Fog hardware has reached the point where it can refine models as well as run inferences, yet the community still lacks a principled way to orchestrate lifelong learning when every micro-cloud samples a different corner of reality. Continual-learning pipelines for fog must fine-tune weights in situ, ship only tightly-compressed gradient sketches, and guard against catastrophic forgetting as demand patterns drift from morning traffic to midnight maintenance. Memory-lean optimisers that squeeze into a few megabytes of SRAM, sparsity-aware codecs that drop near-zero updates, and asynchronous aggregation loops resilient to low-battery stragglers are still open engineering puzzles. Governance layers are equally critical: before locally refined parameters flow sideways to neighbouring clusters—or upward into cloud “gold” models—they must pass verifiable safety gates that bound privacy leakage and certify behaviour on hold-out tests. Trust-region schedulers that throttle step-sizes, federated replay buffers that inject representative past samples without raw-data transfer, and cross-node anomaly voting that quarantines suspect updates all look promising, but rigorous robustness and privacy guarantees for non-IID, real-time streams at the fog edge remain to be demonstrated.

### **3.9. Inter-cluster Federation and Real-time Resource Markets for Fog Domains**

Today’s fog installations are still fenced inside a single campus, factory, or municipality, yet emerging continent-scale services—wildfire megafire detection, cross-border mixed-reality tourism, roaming fleet tele-operation—will need friction-free peering across dozens of independently run fog clusters. The grand challenge is to build resource markets that clear in milliseconds, respect heterogeneous data-sovereignty laws, and thwart free-riding behaviours. One promising avenue couples cryptographic escrows with zero-knowledge proofs so neighbouring districts can lend spare compute cycles without exposing their internal topologies or tenant identities. Stream-level micro-payments, settled via lightweight side-chains, could remunerate transient contributors such as idle delivery vans that double as rolling CDN caches. Yet economic clearing alone is not enough, policy compliance and liability demand programmable provenance. If an encrypted ECG stream bursts from a hospital fog cell into a retail gateway two blocks away, the SLA substrate must stitch an unforgeable audit chain that names every intermediary, records consent artefacts, and flags jurisdictional boundary crossings in real time. Only when these trust, accounting, and regulatory primitives co-evolve with ultra-low-latency auctioneers will inter-cluster federation graduate from lab demos to the always-on, planet-wide fog fabric that next-generation applications will expect.

### **3.10. Node-centric Observability and Opportunistic Telemetry for Fog Platforms**

Classic distributed tracing presumes static hosts wired to fat back-haul links—conditions rarely met by fog landscapes where compute nodes ride on buses, perch on streetlights, or drift with coastal buoys. New observability stacks must therefore spread tracing intelligence across the fog fabric itself, gossiping partial spans whenever nodes come within radio reach yet still re-stitching end-to-end timelines with micro-second fidelity. Techniques such as compressed vector clocks, probabilistic hashing of event IDs, and lightweight homomorphic encryption could let untrusted relays forward telemetry without exposing payloads or topologies. During connectivity outages,

fog-side anomaly detectors need to flag performance regressions locally and journal evidence until the next contact opportunity. Marrying these delay-tolerant traces with explainable-AI analytics would give operators, auditors, and even autonomous control loops clear insight into why a microservice hopped clusters, why jitter spiked at a roadside gateway, or which link failure cascaded into a QoS breach capabilities that are prerequisites for certifying safety-critical fog deployments in transport, healthcare, and smart-city infrastructure.

#### 4. CONCLUSIONS

The survey has mapped the fog-computing resource-management literature onto a coherent pipeline and, in doing so, revealed both impressive progress and critical lacunae. Researchers have produced ingenious point solutions: enclave-protected containers that boot on watt-scale devices, placement engines that exploit topology awareness and predictive caching, migration schemes that juggle mobility and energy, and diversity-coded replicas capable of sub-millisecond fail-over. Yet these achievements remain largely siloed, each optimising a single metric under narrow assumptions about fault rates, bandwidth, and governance. Real-world deployments will demand holistic integration where placement choices respect energy budgets forecast by solar models, where migration controllers coordinate with cache-coherency layers, and where SLA engines translate multi-jurisdiction privacy rules into enforceable orchestration constraints. The paper has distilled cross-cutting research directions that can unlock this integration, e.g., device-centric observability to deliver trustworthy, low-overhead telemetry, federated resource markets that clear in real time while preserving competitive secrecy; edge-native continual learning that evolves models without leaking data or forgetting past knowledge; sustainable, energy-adaptive operation that minimises lifetime carbon alongside latency; and jurisdiction-aware SLA management that elevates privacy and timeliness to first-class contractual terms. Progress on these fronts will require interdisciplinary collaboration and rigorous, open benchmarking, but the payoff is significant: a fog fabric capable of supporting autonomous transport, immersive media, and resilient smart-city infrastructure with the predictability and security today associated only with hyperscale clouds. By clarifying where the community stands and what obstacles still loom, the survey aims to accelerate that journey from promising prototypes to dependable global infrastructure.

#### REFERENCES

- [1] Li, J., Jin, J., Yuan, D., & Zhang, H. (2017) "Virtual fog: A virtualization enabled fog computing framework for Internet of Things," *IEEE Internet of Things Journal*, Vol. 5, No. 1, pp. 121-131.
- [2] Varghese, B., Reano, C., & Silla, F. (2018) "Accelerator virtualization in fog computing: Moving from the cloud to the edge," *IEEE Cloud Computing*, Vol. 5, No. 6, pp. 28-37.
- [3] Wu, J., Dong, M., Ota, K., Li, J., Yang, W., & Wang, M. (2019) "Fog-computing-enabled cognitive network function virtualization for an information-centric future Internet," *IEEE Communications Magazine*, Vol. 57, No. 7, pp. 48-54.
- [4] Roca, D., Quiroga, J. V., Valero, M., & Nemirovsky, M. (2017) "Fog function virtualization: A flexible solution for IoT applications," In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, IEEE, pp. 74-80.
- [5] Santos, J., Wauters, T., Volckaert, B., & De Turck, F. (2020) "Towards delay-aware container-based service function chaining in fog computing," In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, IEEE, pp. 1-9.
- [6] Bazm, M. M., Lacoste, M., Südholt, M., & Menaud, J. M. (2018) "Secure distributed computing on untrusted fog infrastructures using trusted linux containers," In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, IEEE, pp. 239-242.
- [7] Bourhim, E. H., Elbiaze, H., & Dieye, M. (2019) "Inter-container communication aware container placement in fog computing," In *2019 15th International Conference on Network and Service Management (CNSM)*, IEEE, pp. 1-6.

- [8] Puliafito, C., Virdis, A., & Mingozzi, E. (2020) "Migration of multi-container services in the fog to support things mobility," In 2020 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, pp. 259-261.
- [9] Hoque, S., De Brito, M. S., Willner, A., Keil, O., & Magedanz, T. (2017) "Towards container orchestration in fog computing infrastructures," In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Vol. 2, IEEE, pp. 294-299.
- [10] do Espírito Santo, W., Júnior, R. D. S. M., Ribeiro, A. D. R. L., Silva, D. S., & Santos, R. (2019) "Systematic mapping on orchestration of container-based applications in fog computing," In 2019 15th International Conference on Network and Service Management (CNSM), IEEE, pp. 1-7.
- [11] Mahmud, R., & Toosi, A. N. (2021) "Con-Pi: A distributed container-based edge and fog computing framework," IEEE Internet of Things Journal, Vol. 9, No. 6, pp. 4125-4138.
- [12] Siasi, N., Jasim, M.A., Crichigno, J., & Ghani, N. (2019) "Container-based service function chain mapping," In 2019 SoutheastCon, IEEE, pp. 1-6.
- [13] Mahmud, R., Toosi, A. N., Ramamohanarao, K., & Buyya, R. (2019) "Context-aware placement of industry 4.0 applications in fog computing environments," IEEE Transactions on Industrial Informatics, Vol. 16, No. 11, pp. 7004-7013.
- [14] Xu, J., Ota, K., & Dong, M. (2020) "A real plug-and-play fog: Implementation of service placement in wireless multimedia networks," China Communications, Vol. 16, No. 10, pp. 191-201.
- [15] Siasi, N., Jaesim, A., & Ghani, N. (2019) "Tabu search for efficient service function chain provisioning in fog networks," In 2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC), IEEE, pp. 145-150.
- [16] Siasi, N., & Jaesim, A. (2020) "Priority-aware SFC provisioning in fog computing," In 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), IEEE, pp. 1-6.
- [17] Siasi, N., Jaesim, A., & Ghani, N. (2020) "Service function chain provisioning schemes for multi-layer fog networks," IEEE Networking Letters, Vol. 2, No. 1, pp. 38-42.
- [18] Trabelsi, M., Bendaoud, N.M.M., & Ahmed, S.B. (2023) "Multi-objective Optimization for Dynamic Service Placement Strategy for Real-Time Applications in Fog Infrastructure," In 2023 IEEE Symposium on Computers and Communications (ISCC), IEEE, pp. 607-612.
- [19] Xu, J., Ota, K., & Dong, M. (2018) "Plug-and-play for fog: Dynamic service placement in wireless multimedia networks," In 2018 IEEE/CIC International Conference on Communications in China (ICCC), IEEE, pp. 490-494.
- [20] Pallewatta, S., Kostakos, V., & Buyya, R. (2024) "Reliability-aware proactive placement of microservices-based IoT applications in fog computing environments," IEEE Transactions on Mobile Computing, Vol. 23, No. 12, pp. 11326-11341.
- [21] Siasi, N., Jasim, M., Aldalbahi, A., & Ghani, N. (2020) "Delay-aware SFC provisioning in hybrid fog-cloud computing architectures," IEEE Access, Vol. 8, pp. 167383-167396.
- [22] Siasi, N., Jasim, M., Aldalbahi, A., & Ghani, N. (2020) "Deep learning for service function chain provisioning in fog computing," IEEE Access, Vol. 8, pp. 167665-167683.
- [23] Aljohani, A., & Sakellariou, R. (2024) "Improved Application Placement in Fog Environments Through Parallel Collaboration," IEEE Access.
- [24] Ait Salaht, F., Desprez, F., Lebre, A., Prud'Homme, C., & Abderrahim, M. (2019) "Service placement in fog computing using constraint programming," In 2019 IEEE International Conference on Services Computing (SCC), IEEE, pp. 19-27.
- [25] Jasim, M., Siasi, N., Malapaka, S., Oliveira, D., & Ugweje, O. (2020) "A single-tier fog architecture for delay-sensitive and computation-intensive SFC requests," In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, pp. 0654-0660.
- [26] Baskar, R., Mohanraj, E., Sneka, T., Yazhini, S., & Vasanth, S. (2024) "Teaching learning-based optimization for medical IoT applications service placement in fog computing," In 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), IEEE, pp. 1-6.
- [27] Apat, H.K., Sahoo, B., & Maiti, P. (2018) "Service placement in fog computing environment," In 2018 International Conference on Information Technology (ICIT), IEEE, pp. 272-277.
- [28] Jaesim, A., Siasi, N., Ababneh, M., Elkourdi, M., & Nour, A. (2020) "Application-specific service function chain provisioning in heterogeneous fog networks," In 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), IEEE, pp. 1-6.

- [29] Siasi, N., Jasim, M., & Ghani, N. (2020) "Service function chain mapping in fog networks," *IEEE Communications Letters*, Vol. 25, No. 1, pp. 99-102.
- [30] Brogi, A., Forti, S., Guerrero, C., & Lera, I. (2019) "Meet genetic algorithms in Monte Carlo: Optimised placement of multi-service applications in the fog," In 2019 IEEE International Conference on Edge Computing (EDGE), IEEE, pp. 13-17.
- [31] Borelli, H., Costa, F.M., & Carvalho, S.T. (2022) "Use of multilevel resource clustering for service placement in fog computing environments," In 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), IEEE, pp. 360-365.
- [32] Minh, Q.T., Nguyen, D.T., Van Le, A., Nguyen, H.D., & Truong, A. (2017) "Toward service placement on fog computing landscape," In 2017 4th NAFOSTED Conference on Information and Computer Science, IEEE, pp. 291-296.
- [33] Aldalbahi, A., Jasim, M.A., Shahabi, F., Mazin, A., Siasi, N., & Oliveira, D. (2021) "Deep learning for primary sector prediction in FR2 new radio systems," *IEEE Access*, Vol. 9, pp. 157522-157539.
- [34] Chiti, F., Fantacci, R., Paganelli, F., & Picano, B. (2019) "Virtual functions placement with time constraints in fog computing: A matching theory perspective," *IEEE Transactions on Network and Service Management*, Vol. 16, No. 3, pp. 980-989.
- [35] Lera, I., Guerrero, C., & Juiz, C. (2018) "Availability-aware service placement policy in fog computing based on graph partitions," *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 3641-3651.
- [36] Shaik, S., & Baskiyar, S. (2021) "A scalable approach to service placement in fog/cloud environments," In 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), IEEE, pp. 1-8.
- [37] Kayal, P., & Liebeherr, J. (2019) "Distributed service placement in fog computing: An iterative combinatorial auction approach," In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, pp. 2145-2156.
- [38] Kadhim, A.J., & Naser, J.I. (2021) "Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN," *China Communications*, Vol. 18, No. 2, pp. 271-289.
- [39] Ningning, S., Chao, G., Xingshuo, A., & Qiang, Z. (2016) "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Communications*, Vol. 13, No. 3, pp. 156-164.
- [40] Kashani, M.H., & Mahdipour, E. (2022) "Load balancing algorithms in fog computing," *IEEE Transactions on Services Computing*, Vol. 16, No. 2, pp. 1505-1521.
- [41] Venkatesh, M., Polisetty, S.N.K., Satpathy, R., & Neelima, P. (2022) "A Novel Deep Learning Mechanism for Workload Balancing in Fog Computing," In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, pp. 515-519.
- [42] Mattia, G.P., Pietrabissa, A., & Beraldi, R. (2023) "A load balancing algorithm for equalising latency across fog or edge computing nodes," *IEEE Transactions on Services Computing*, Vol. 16, No. 5, pp. 3129-3140.
- [43] Jasim, M.A., Siasi, N., Rahouti, M., & Ghani, N. (2022) "SFC provisioning with load balancing method in multi-tier fog networks," *IEEE Networking Letters*, Vol. 4, No. 2, pp. 82-86.
- [44] Jasim, M.A., Siasi, N., & Ghani, N. (2022) "Hierarchy descending SFC provisioning scheme with load balancing in fog computing," *IEEE Communications Letters*, Vol. 26, No. 9, pp. 2096-2100.
- [45] Zaki, A.M., & Sorour, S. (2022) "Proactive migration for dynamic computation load in edge computing," In ICC 2022-IEEE International Conference on Communications, IEEE, pp. 4275-4280.
- [46] Bhardwaj, A., Gupta, U., Budhiraja, I., & Chaudhary, R. (2023) "Container-based migration technique for fog computing architecture," In 2023 International Conference for Advancement in Technology (ICONAT), IEEE, pp. 1-6.
- [47] Jasim, M.A., Siasi, N., Almalag, M., Honary, V., & Aldalbahi, A. (2021) "Asynchronous coarse-grained load migration scheme for IoT applications in fog networks," In 2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), IEEE, pp. 82-87.
- [48] Jasim, M.A., Siasi, N., & Ghani, N. (2022) "Efficient load migration scheme for fog networks," In 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), IEEE, pp. 1-6.
- [49] Ge, S., Cheng, M., & Zhou, X. (2020) "Interference aware service migration in vehicular fog computing," *IEEE Access*, Vol. 8, pp. 84272-84281.

- [50] Singh, R.M., Sikka, G., & Awasthi, L.K. (2024) "LBATSM: load balancing aware task selection and migration approach in fog computing environment," *IEEE Systems Journal*, Vol. 18, No. 2, pp. 796-804.
- [51] Siasi, N., & Jasim, M.A. (2023) "Migration of VNF instances for service continuity," *Internet Technology Letters*, Vol. 6, No. 4, p. e422.
- [52] Wang, D., Liu, Z., Wang, X., & Lan, Y. (2019) "Mobility-aware task offloading and migration schemes in fog computing networks," *IEEE Access*, Vol. 7, pp. 43356-43368.
- [53] Jasim, M.A., Siasi, N., & Aldalbahi, A. (2023) "Pre-overload migration scheme for NFV-based fog computing," In *Proceedings of the Int'l ACM Symposium on Mobility Management and Wireless Access*, IEEE, pp. 99-103.
- [54] Jasim, M., & Siasi, N. (2024) "Hybrid-grained migration method for load redistribution in heavily-loaded fog nodes," *Cluster Computing*, Vol. 27, No. 7, pp. 9497-9507.
- [55] Filiposka, S., Mishev, A., & Gilly, K. (2018) "Community-based allocation and migration strategies for fog computing," In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, pp. 1-6.
- [56] Azhaguramyaa, V.R., Janet, J., Tharuneshwar, V., SriRam, S., & Kumar, T.S. (2021) "An Efficient Approach Towards Surrogate Node Selection For Container Migration In Fog Computing," In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1, IEEE, pp. 1209-1214.
- [57] Jasim, M., & Siasi, N. (2024) "Local load migration in high-capacity fog computing," *ACM Transactions on Internet Technology*, Vol. 24, No. 4, pp. 1-31.
- [58] Jasim, M.A., (2024) "Content-Specific and Buffer-Based Migration Schemes for Fog Computing," *IEEE Transactions on Services Computing*.
- [59] Kumari, P., Dubey, V., & Shrivastava, S. (2023) "Content replica placement method for fault tolerance in fog computing environment," In *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*, IEEE, pp. 497-502.
- [60] Mounnan, O., Mouatasim, A.E., Manad, O., Hidar, T., El Kalam, A.A., & Idboufker, N. (2020) "Privacy-Aware and authentication based on Blockchain with Fault Tolerance for IoT enabled Fog Computing," pp. 347-352 [online].
- [61] Mohamed, N., Al-Jaroodi, J., & Jawhar, I. (2019) "Towards fault tolerant fog computing for IoT-based smart city applications," In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, pp. 0752-0757.
- [62] Khan, S., Shah, I.A., Aurangzeb, K., Ahmad, S., Khan, J.A., & Anwar, M.S. (2024) "Energy-efficient task scheduling using fault tolerance technique for IoT applications in fog computing environment," *IEEE Internet of Things Journal*, Vol. 11, No. 24, pp. 39009-39019.
- [63] Ghanavati, S., Abawajy, J., & Izadi, D. (2020) "Automata-based dynamic fault tolerant task scheduling approach in fog computing," *IEEE Transactions on Emerging Topics in Computing*, Vol. 10, No. 1, pp. 488-499.
- [64] Siasi, N., Jasim, M.A., & Ghani, N. (2024) "Post-fault restoration of service function chains in fog networks," *Computer Networks*, Vol. 251, p. 110580.
- [65] Siasi, N., Jasim, M.A., Yayimli, A., & Ghani, N. (2021) "Service function chain survivability provisioning in fog networks," *IEEE Transactions on Network and Service Management*, Vol. 19, No. 2, pp. 1117-1128.
- [66] Wang, K., Shao, Y., Xie, L., Wu, J., & Guo, S. (2018) "Adaptive and fault-tolerant data processing in healthcare IoT based on fog computing," *IEEE Transactions on Network Science and Engineering*, Vol. 7, No. 1, pp. 263-273.
- [67] Zhang, P., Chen, Y., Zhou, M., Xu, G., Huang, W., Al-Turki, Y., & Abusorrah, A. (2021) "A fault-tolerant model for performance optimization of a fog computing system," *IEEE Internet of Things Journal*, Vol. 9, No. 3, pp. 1725-1736.
- [68] Siasi, N., Jaesim, A., Aldalbahi, A., & Ghani, N. (2019) "Link failure recovery in NFV for 5G and beyond," In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, pp. 144-148.
- [69] Bhushan, S., & Mat, M. (2021) "Priority-queue based dynamic scaling for efficient resource allocation in fog computing," In *2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, IEEE, pp. 1-6.

- [70] Saxena, M.K., Dev, K., & Kumar, S. (2024) "Dynamic CPU Frequency Scaling for Efficient Resource Allocation in Heterogeneous Fog Networks for CIoT Applications," *IEEE Transactions on Consumer Electronics*.
- [71] Sami, H., Mourad, A., Otrok, H., & Bentahar, J. (2020) "Fscaler: Automatic resource scaling of containers in fog clusters using reinforcement learning," In 2020 International Wireless Communications and Mobile Computing (IWCMC), IEEE, pp. 1824-1829.
- [72] Jana, G.C., & Banerjee, S. (2017) "Enhancement of QoS for fog computing model aspect of robust resource management," In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, pp. 1462-1466.
- [73] Gupta, A., & Gupta, S.K. (2022) "Intelligent collaboration of multi-agent flying UAV-fog networking for better QoS," In 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), IEEE, pp. 1-6.
- [74] Apat, H.K., Maiti, P., & Patel, P. (2020) "Review on QoS aware resource management in fog computing environment," In 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), IEEE, pp. 1-6.
- [75] Bardalai, P., Medhi, N., Bargayary, B., & Saikia, D.K. (2021) "Openhealthq: Openflow based QoS management of healthcare data in a software-defined fog environment," In ICC 2021-IEEE International Conference on Communications, IEEE, pp. 1-6.
- [76] Lai, C.F., Song, D.Y., Hwang, R.H., & Lai, Y.X. (2016) "A QoS-aware streaming service over fog computing infrastructures," In 2016 Digital Media Industry & Academic Forum (DMIAF), IEEE, pp. 94-98.
- [77] Yousefpour, A., Patil, A., Ishigaki, G., Kim, I., Wang, X., Cankaya, H.C., Zhang, Q., Xie, W., & Jue, J.P. (2019) "FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework," *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5080-5096.
- [78] Joshi, N., & Srivastava, S. (2023) "QoS-aware task migration strategy for fog IoT," In 2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC), IEEE, pp. 583-588.
- [79] Gupta, A., & Gupta, S.K. (2022) "UAV aided fog network (UAFN): A proposal framework for better QoS," In 2022 2nd International Conference on Computing and Information Technology (ICCIT), IEEE, pp. 265-270.
- [80] Assila, B., Kobbane, A., & El Koutbi, M. (2018) "A many-to-one matching game approach to achieve low-latency exploiting fogs and caching," In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, pp. 1-2.
- [81] Elnagar, M.R., Mohamed, A.A., Tawfik, B.S., & Refaat, H.E. (2024) "Enhancement of Fog Caching Using Nature Inspiration Optimization Technique Based on Cloud Computing," *IEEE Access*, Vol. 12, pp. 101484-101496.
- [82] Aldalbahi, A., Jasim, M.A., Bouzguenda, M., Enshasy, H., Sumsudeen, R., & Siasi, N. (2022) "Caching of popular content at F-RAN for mmWave new radio," *International Journal of Communication Systems*, Vol. 35, No. 18, p. e5353.
- [83] Aldalbahi, A., Jasim, M.A., Siasi, N., Bouzguenda, M., Enshasy, H., & Sumsudeen, R. (2022) "Clustered and distributed caching methods for F-RAN-based mmWave communications," *Applied Sciences*, Vol. 12, No. 14, p. 7111.
- [84] Su, J., Lin, F., Zhou, X., & Lu, X. (2015) "Steiner tree based optimal resource caching scheme in fog computing," *China Communications*, Vol. 12, No. 8, pp. 161-168.
- [85] Hassan, K.N., & Robson, E. (2022) "Mobility-based multi-layered caching and data distribution in vehicular fog computing," In 2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real-Time Applications (DS-RT), IEEE, pp. 164-167.
- [86] Yu, Z., Hu, J., Min, G., Wang, Z., Miao, W., & Li, S. (2021) "Privacy-preserving federated deep learning for cooperative hierarchical caching in fog computing," *IEEE Internet of Things Journal*, Vol. 9, No. 22, pp. 22246-22255.
- [87] Sellami, Y., Jaber, G., Lounis, A., Lakhlef, H., & Bouabdallah, A. (2022) "A Cooperative Caching Scheme in Fog/Sensor Nodes for CCN," In 2022 International Wireless Communications and Mobile Computing (IWCMC), IEEE, pp. 481-486.
- [88] Wang, S., Huang, X., Liu, Y., & Yu, R. (2016) "CachinMobile: An energy-efficient users caching scheme for fog computing," In 2016 IEEE/CIC International Conference on Communications in China (ICCC), IEEE, pp. 1-6.

- [89] Althamary, I., Huang, C.W., Lin, P., Yang, S.R., & Cheng, C.W. (2018) "Popularity-based cache placement for fog networks," In 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, pp. 800-804.
- [90] Huang, X., Yu, C., Wang, F., & Chen, Q. (2023) "Hierarchical Federated Learning for Collaborative Caching in Fog Computing," In 2023 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, pp. 898-903.
- [91] Ennya, Z., Hadi, M.Y., & Abouaomar, A. (2018) "Computing tasks distribution in fog computing: Coalition game model," In 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, pp. 1-4.
- [92] Zhang, Y., Wang, S., Qu, X., Li, C., & Xu, E. (2023) "Toward Secure and Efficient Collaborative Cached Data Auditing for Distributed Fog Computing," IEEE Internet of Things Journal, Vol. 10, No. 23, pp. 20941-20954.
- [93] Wang, K., Li, J., Yang, Y., Chen, W., & Hanzo, L. (2020) "Energy-efficient multi-tier caching and node association in heterogeneous fog networks," In 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), IEEE, pp. 1-5.
- [94] Ahmad, A., Aldalbahi, A., Ahmad, F., & Ali, S. (2024) "Service Caching in Multi-Tier Fog Radio Access Networks," IEEE Access.
- [95] Almobaideen, W.A., & Malkawi, O.M. (2020) "Application based caching in fog computing to improve quality of service," In 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, pp. 20-27.
- [96] Zhao, X., Zong, L., & Xu, Y. (2022) "Optimization of fog computing for smart cities with IoT," In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1-6.

## AUTHOR

**Firas Ghoneim** is a software engineer at ATA software solutions company in Beni Suef, Egypt. He finished his bachelor's degree in computer science from Ain Shams University and master's degree from Helwan University in software engineering. His research focuses on virtualization, datacenters, and software architecture.

