Volume 11, Issue 10, October – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

ContextWeave: Adaptive Context-Aware Lifecycle Orchestration for Privacy-Preserving Ephemeral Edge Computing

Sai Divyansh[1]

[1]Independent Researcher

Publication Date: 2026/10/27

Abstract: Edge orchestration research optimizes where tasks and services run, but usually assumes that the edge infrastructure itself is always present. Under mobile, bursty and event-driven demand, this assumption wastes energy on idle capacity and leaves sensitive service state resident long after the demand that justified it has ended. We treat the existence of the edge execution substrate as an orchestration variable and present CONTEXT-WEAVE, a context-aware lifecycle controller for ephemeral edge sessions. CONTEXTWEAVE observes only aggregate, privacy-minimized signals (arrival rates, trends, sensitive-task share and aggregate mobility inflow), forecasts short-horizon demand with an interpretable rule layer and an optional federated residual learner, converts the forecast and its uncertainty into the probability that a session has positive net value, and creates, extends or dissolves sessions with asymmetric evidence, hysteresis, minimum retention and cooldown. Dissolution includes state deletion and credential revocation, and privacy exposure is measured as resident, idle-resident and cloud-transfer components. In a simulation study spanning pedestrian, vehicular, IoT and bursty workloads with 20 held-out seeds, CONTEXTWEAVE reduces energy by 25–71% and resident exposure by 36–82% relative to continuous provisioning, at a cost of 0.2–2.1 percentage points of additional deadline misses. It achieves a lower objective than a reactive keep-alive autoscaler in all four scenarios and comes within 2.3% of an omniscient look-ahead. Ablations identify demandtrend and mobility-inflow signals and the churn safeguards as the main sources of benefit, and show where ephemeral operation does not pay.

Keywords: *Edge Computing, Ephemeral Cloud, Context Awareness, Service Lifecycle Orchestration, Privacy Exposure, Energy Efficiency, Mobility, Federated Learning.*

How to Cite: Sai Divyansh (2026) ContextWeave: Adaptive Context-Aware Lifecycle Orchestration for Privacy-Preserving Ephemeral Edge Computing. *International Journal of Innovative Science and Research Technology*, 11(10), 1932-1945. https://doi.org/10.38124/ijisrt/26oct866

I. INTRODUCTION

Edge computing moves computation toward the producers of data so that interactive and data-intensive applications can meet response-time targets that a distant cloud cannot. Two properties of real edge workloads, however, make fixed provisioning a poor fit. First, users move: a mobile user crosses edge domains, and keeping a service close to that user requires migration whose cost must be traded against user-perceived latency [1]. Second, demand fluctuates: IoT and mobile tasks arrive with statistics that are hard to know in advance, which has motivated a large body of online offloading and resourceallocation methods [2], [3]. A provider that keeps every edge service running everywhere pays for that uncertainty in idle energy; one that provisions only on average demand pays for it in latency.

This paper argues that a third quantity is equally affected by the provisioning decision and is usually left out of it: privacy exposure over time. Features of offloaded tasks can reveal whether a user is present [4], offloading in connected-vehicle networks exposes users to tracking and identity tampering [5], and historical service requests can disclose sensitive user information [6]. Encryption protects content in transit [7] but does not bound how long sensitive service state remains resident on infrastructure after the demand that justified it has ended. An edge service that is kept warm "just in case" is, from a privacy standpoint, an exposure surface that persists without delivering value. The decision of whether the execution substrate should exist at all is therefore a joint latency, energy and privacy decision.

We call an edge deployment whose existence is itself an orchestration variable an ephemeral edge cloud: a short-lived session that is created when local context indicates sufficient value, extended while that value persists, and dis-solved, including deletion of temporary state and revocation of credentials, when it no longer does. The notion builds on short-lived personal cloud environments that are created for a task and dissolved automatically, with time-based key decay

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

so that no long-term data remains [8]. Existing work provides strong machinery for what happens inside such a session: task placement [9], [10], online offloading [2], [3], service-function-chain orchestration [11], [12], and privacyaware offloading and placement [4], [6]. What it does not provide is a tested controller for the session lifecycle: one that turns noisy, privacy-minimized context into create, extend and dissolve actions, while guarding against premature teardown, wasteful pre-provisioning and oscillation.

➢ *Contributions*

This paper makes four contributions.

- *A Lifecycle Formulation.*

We formalize ephemeral edge sessions as a four-state machine (dormant, booting, active, retiring) and define a rolling objective that prices latency, deadline misses, energy, a three-part privacyexposure measure, and lifecycle churn (Section III). The exposure measure separates resident sensitive state, sensitive data shipped to the remote cloud, and resident state that serves no task.

- *The CONTEXTWEAVE Controller.*

We design a contextaware lifecycle controller that (i) observes only aggregate, privacy-minimized signals, (ii) forecasts short-horizon demand with an interpretable rule layer and an optional federated residual learner, (iii) converts the forecast and its uncertainty into the probability that a session has positive net value, and (iv) acts with asymmetric evidence, hysteresis, minimum retention, cooldown and a privacy-driven fast-retire path (Section IV). Lifecycle control is kept separate from inner task placement so that either can be replaced.

- *An Open, Reproducible Evaluation.*

We evaluate CONTEXTWEAVE in a discrete-time simulator across pedestrian-mobility, vehicular, IoT and bursty workloads against static placement, continuous provisioning, Lyapunov online offloading, a reactive keep-alive autoscaler and an omniscient look-ahead reference, using 20 heldout seeds per configuration (Sections V–VI).

- *Evidence on what matters.*

Through ablations, sensitivity sweeps and impaired-context experiments we identify which context signals and safeguards carry the benefit, and where ephemeral operation does *not* pay (Sections VI–VII).

The remainder of the paper is organized as follows. Section II reviews related work. Section III presents the system model and problem. Section IV describes CONTEXTWEAVE. Section V details the evaluation methodology, and Section VI reports results. Section VII discusses limitations, and Sec-tion VIII concludes.

II. RELATED WORK

➢ *Mobility-Aware Placement and Pre-Provisioning*

Mobility is a direct source of placement instability. Ouyang *et al.* [1] use Lyapunov optimization to turn latency minimization under a long-term migration-cost budget into perslot decisions without prior knowledge of user mobility, and address the NP-hard per-slot problem with Markov approximation and, for large user populations, distributed best-response updates. The formulation links migration frequency, service performance and operational cost, but assumes that the candidate edge infrastructure already exists; it decides where a service lives rather than whether a resource domain should be instantiated. Bolla *et al.* [13] show that orchestration for service continuity leaves pre-provisioned network functions idle for redundancy, resilience, scaling or migration, and argue that context can separate resources in active use from those retained for anticipated events. Together these works expose the tension we target: continuity calls for prediction, whereas energy efficiency calls for permission to withdraw capacity when predicted demand loses confidence.

➢ *Workload, Workflow and Service-Function-Chain Orchestration*

Sonmez *et al.* select execution locations with a fuzzy-logic orchestrator that encodes administrator-like rules over task and resource state [9], and later with a two-stage machine-learning orchestrator for vehicular edge computing that accounts for task completion and service time [10]. Zhou *et al.* [11] jointly optimize resource provisioning and traffic routing for crossedge service-function chains via online decomposition and randomized rounding. Sun *et al.* [12] show that mapping workflow-structured requests, rather than linear chains, onto edge nodes affects response time and blocking. These methods assign tasks or functions to an already-admitted infrastructure; their primary action is placement rather than lifecycle control of a temporary session.

➢ *Online Offloading and Resource Allocation*

Online offloading formulates latency, energy and allocation as decisions under stochastic arrivals. Chen *et al.* [2] use stochastic optimization to decompose dynamic offloading into subproblems that are solved concurrently, online and in a distributed manner, without prior statistics of task generation; Liu *et al.* [3] combine Lyapunov optimization and duality theory for near-optimal edge–cloud task offloading and resource allocation under heterogeneous tasks and random arrivals. Ren *et al.* [14] show that partial compression offloading reduces end-to-end latency compared with purely local or purely edge processing. Zhang *et al.* [7] redistribute users across small base stations to avoid overload and add an AES-based security layer, jointly reducing system time and energy relative to local execution. These methods are strong candidates for the inner operation of an ephemeral session, and we use a drift-pluspenalty policy as a baseline. None of them defines a session boundary, an exposure budget or a stopping rule for infrastructure after demand subsides.

➢ *Privacy-Preserving Offloading and Distributed Learning*

PEACE [4] addresses inference of user presence from offloaded feature tasks while maintaining cost-efficient scheduling. Xu *et al.* [5] treat tracking, identity tampering and hijacking in connected vehicles through multi-objective route

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

selection with NSGA-II. Qian *et al.* [6] cast service placement as a 0–1 problem solved with centralized greedy selection and federated learning so that raw request records are not centralized, and Lu *et al.* [15] show asynchronous federated learning in which participants join and leave, which matches the transient membership of ephemeral sessions. Li *et al.* [16] aggregate encrypted device data at an edge server with source authentication and integrity. These works protect content and training data; they do not treat the *duration* for which sensitive state stays resident as a cost.

➢ *Context Awareness Across the Continuum*

Chen *et al.* [17] select offloading targets from context that changes with mobility (location, network conditions and available cloud resources) together with per-service computation cost, and da Silva and Sofia [18] review the context-awareness indicators used by automated edge-selection algorithms in the IoT cloud/edge continuum. Both establish the relevance of context but output a selected resource. Neither specifies a persistent estimator that fuses noisy observations into a lifecycle state, nor quantifies the relative value of indicator groups, which our ablation addresses.

Table 1 Positioning of Contextweave Against Representative Work

Work	Mob.	Energy	Privacy	Temporal exposure	Lifecycle control	Churn control
[1]	✓					partial
[2], [3], [9], [12] [10]	✓					
[11]					partial	
[13]		✓			partial	
[7]		✓	partial			
[4]			✓			
[5]	✓	✓	✓			
[6], [15]			✓			
[8]			✓	✓	✓	
[17]	✓					
CONTEXTWEAVE	✓	✓	✓	✓	✓	✓

➢ *Positioning*

Table 1 summarizes the gap; a check mark indicates that the property is an explicit design goal or evaluated quantity of the work, and "partial" that it is addressed indirectly.

Edge orchestration methods optimize placement, offloading or consolidation after infrastructure is assumed to exist. The ephemeral family cloud [8] does make environments short-lived, but for task-scoped household services rather than for context-driven, mobility-aware edge sessions, and its environ-ments are dissolved when their task ends rather than according to forecast demand. CONTEXTWEAVE makes the existence of the edge substrate the controlled variable, and treats timeintegrated exposure, churn and context uncertainty as first-class quantities alongside latency and energy.

III. SYSTEM MODEL AND PROBLEM FORMULATION

➢ *Edge Sites and Ephemeral Sessions*

We consider a set M = { 1, . . . , *M*} of edge sites (access points, small cells or roadside units), each able to host one ephemeral session on a server of capacity *C* (Gcycles/s). Time is slotted with slot length $\tau = 1$ s. Each site has neighbors N *m* reachable over a backhaul, and all sites reach a remote cloud over a wide-area network (WAN). The session at site *m* is in one of four states $x_m(t) \in$ {D ORMANT, BOOTING, ACTIVE, RETIRING (Fig.} 1). Creat- ing a session moves it to BOOTING for T_b slots, during which the substrate is instantiated at power P_b; it then becomes.

ACTIVE and accepts tasks. Dissolving moves it to RETIRING for T_c slots at power P_c, during which the session drains, deletes temporary storage, revokes credentials and releases compute and network allocations; only then does it return to DORMANT. An ACTIVE session draws base power P_0 regardless

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

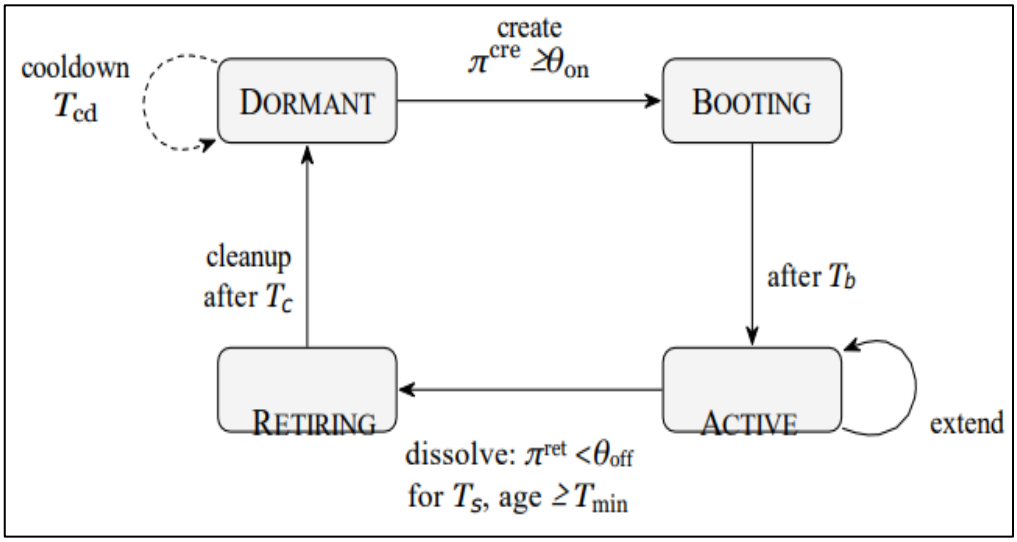


Fig 1 Ephemeral Session Lifecycle. Only ACTIVE Sessions Accept Tasks; Resident Sensitive State is Deleted Only When RETIRING Completes.

➢ *Task Model*

Tasks arrive at the site that currently serves their originating device. A task of class c has computational demand w_c (Gcycles), input size s_c (MB) and deadline d_c; a fraction of tasks is privacy-sensitive. We use two classes: *interactive* (w_0 = 0.4, s_0 = 0.3, d_0 = 100 ms), representative of augmented-reality or perception offloading, and *analytic* (w_1 = 2.0, s_1 = 1.5, d_1= 1 s). Let $a_{m,c}(t)$ and a^s (t) denote all and sensitive m,c arrivals.

➢ *Latency Model*

A task served at its local ACTIVE session incurs

$$\ell^{edge} = \frac{8s_c}{R_u} + \delta_e + W_m + \frac{w_c}{C}\xi, \quad (1)$$

Where Ru is the uplink rate, δe the access propagation delay, Wm the queueing delay and ξ ~ Γ(4, 1) models service time variability. W 4 m is exponentially distributed with mean Bm/C +w¯ mρm/(2(1 ρm)), where Bm is the backlog carried from the previous slot and ρm the slot utilization (capped at 0.95). A task served at a neighbor adds a backhaul hop δh + 8sc/Rb. A task served in the cloud incurs of load. A session that stops accepting tasks but retains data is not considered dissolved: resident state is cleared only on completion of RETIRING.

$$\ell^{cloud} = \frac{8s_c}{R_u} + \mathrm{RTT} + \frac{8s_c}{R_b} + \frac{w_c}{C_{cl}}\xi + J, \quad (2)$$

With log-normal WAN jitter J. A task misses its deadline when its latency exceeds d_c.

➢ *Energy Model*

Site energy over a slot is $P_0\tau$ when ACTIVE, $P_b\tau$ when BOOTING and $P_c\tau$ when RETIRING, plus dynamic energy $\kappa_e w$ per Gcycle executed; cloud energy is $\kappa_{cl} w + \epsilon_{wan} s$ per task, including facility overhead and WAN transfer. Dormant sites consume no session energy. Total energy E is the sum over sites, slots and tasks.

➢ *Privacy-Exposure Model*

Encryption addresses content in transit [7], [16]; the threats of inference from task features and of tracking [4], [5] are distinct. We therefore measure exposure as three separate components rather than a single score:

L L

- *Resident Exposure* $_{res}$ = $_{t\ m}$ $h_m(t)$, the site-time E during which sensitive state is resident. $h_m(t)$ = 1 from the first sensitive task a site serves until the session's cleanup completes.

For persistent infrastructure, which has no dissolution event, we apply a retention policy that purges sensitive state T_r = 300 s after the last sensitive task; this is generous to the persistent baselines.

- *Idle Resident Exposure* $_{idle}$E , the subset of $_{res}$ accrued in slots E in which the site serves no task: exposure that buys no utility.

- *Cloud Exposure* $E_{cl} = s_c$ over sensitive tasks shipped to the remote cloud (MB), capturing transfer volume and the extra processing location.

➢ *Lifecycle Problem*

The controller chooses lifecycle actions $u_m(t) \in$ {create , extend, dissolve ;} the inner placement then assigns tasks among ACTIVE sessions and the cloud. The rolling objective weights latency L, deadline misses D, energy E, privacy exposure P and lifecycle churn S:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

$$J = w_L \sum_{\text{tasks}} \ell + w_M D + w_E E + w_P^{\text{mb}} E_{\text{cl}} + w_P^{\text{res}} E_{\text{res}} + w_S N_{\text{cre}}, \quad (3)$$

Where N_{cre} counts session creations. The weights are scenario parameters, not hidden constants. Feasibility requires that tasks be admitted to a session only up to a utilization cap $\bar{\rho}$, that service-chain dependencies be preserved, and that only ACTIVE sessions serve tasks. Even with perfect knowledge of arrivals, the joint choice of lifecycle and placement is a combinatorial facility-location problem with switching costs; in practice arrivals and mobility are not known in advance, and the controller observes them only through aggregated, delayed and noisy context. We therefore design an online controller and report all objective components separately, since latency, energy and privacy can move in different directions [4], [5].

IV. CONTEXTWEAVE DESIGN

➢ *Architecture*

CONTEXTWEAVE separates observation, state estimation and action selection (Fig. 2). A *context broker* at each site produces privacy-minimized summaries. A *state estimator* converts them into a short-horizon demand forecast with an uncertainty estimate. A *lifecycle controller* maps the forecast to create, extend or dissolve actions. A *resource manager* instantiates and tears down the session, and a *privacy monitor* records resident state, processing locations and cloud transfers. While a session is ACTIVE, an *inner placement policy* assigns tasks. The lifecycle controller owns session state only; any placement or offloading method from Section II can be used inside a session without changing it.

➢ *Privacy-Minimized Observation*

The broker never exports per-device identifiers. Every control epoch ($\Delta = 5$ s) it reports, for its site: arrival rates over 5, 30 and 120 s windows (r_5, r_{30}, r_{120}); the trend $r_5 - r_{30}$; the sensitive fraction σ over 120 s; the count of active devices currently attached; and an *inflow* count, the number of active devices in neighboring cells whose extrapolated position falls in this cell within 30 s. Inflow and presence are aggregate counts, converted to task rates by the mean per-device rate. Where devices cannot report heading, these two features are absent and the estimator falls back on rate features, as in our IoT and bursty scenarios. Model training, when used, exchanges only model weights [6], [15]; the telemetry itself could additionally be protected with authenticated aggregation [16].

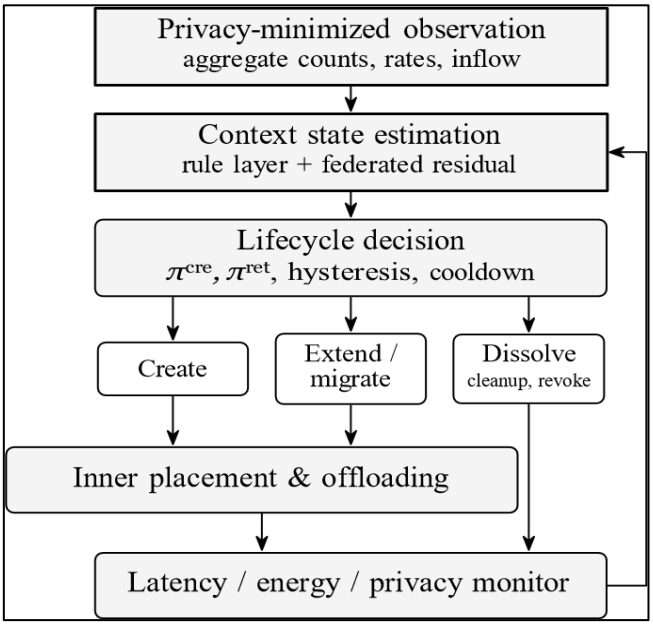


Fig 2 CONTEXTWEAVE Architecture. Lifecycle Control (Top) is Separated from Inner Placement (Bottom); Monitored Outcomes Feed Back into State Estimation.

➢ *Context State Estimation*

Let $\mathbf{f}_m(t) \in \mathrm{R}^8$ be the feature vector [1, r_5, r_{30}, r_{120}, $r_5 - r_{30}$, ι, p, σ], where ι is inflow and p presence (both as task rates). The estimator forecasts $\hat{D}_m(t)$, the mean arrival rate over the next $H = 60$ s.

Rule layer. An interpretable forecast takes the recent peak rate and adds half of the predicted inflow:

$$\hat{D}_m^{\text{rule}} = \max\{r_{30}, r_5\} + \tfrac{1}{2}\iota, \quad (4)$$

The rule is deliberately conservative toward emerging demand (via r_5 and ι) and smooth toward declining demand (via r_{30}), which matches the asymmetric costs of missed demand and of creation.

- *Federated residual learner (optional).* A linear correction $\hat{D}_m = \max\{0, \hat{D}_m^{\text{rule}} + \boldsymbol{\beta}_m^{\top}\mathbf{f}_m\}$ is trained online by normalized least-mean-squares on labels that become available H seconds later, and the per-site weights $\boldsymbol{\beta}_m$ are averaged across sites every 60 s (FedAvg). No arrival records leave a site. We denote this variant CW-FedLearn.
- *Uncertainty.* Each site tracks the exponentially weighted mean squared forecast error v_m; forecast uncertainty is $s_m = \sqrt{v_m + s_0}$ with a floor $s_0 = 0.2$ tasks/s.

➢ *Value of a Session*

CONTEXTWEAVE decides by comparing the value a local session would add over the horizon with what it would cost. Let b_c be the per-task objective gain of serving a class-c task locally instead of at the fallback, evaluated from (1)–(3) at nominal load:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

$$b_c = w_L(\ell_c^{\text{fb}} - \ell_c^{\text{edge}}) + w_M(q_c^{\text{fb}} - q_c^{\text{edge}}) + w_E \frac{e_c^{\text{fb}} - e_c^{\text{edge}}}{1000} + w_P \sigma_{s_c} \mathbf{1}_{\text{cl}}, \quad (5)$$

Where q^{-} denotes miss probability and the last term applies only when the fallback is the cloud. The fallback is a mix: a fraction α_m of forecast load could be absorbed by spare capacity at ACTIVE neighbors (each neighbor's spare capacity below $0.2C$ is shared among its own non-active neighbors) and the rest goes to the cloud, so $b = \alpha_m b^{\text{nb}} + (1-\alpha_m) b^{\text{cl}}$ with $b^{(\cdot)}$ averaged over the class mix. A session costs $c = w_E P_0/1000 + w^{\text{res}}{}_P \mathbf{1}$[sensitive state resident or expected] per second, plus a one-time creation cost $C_0 = w_E(P_b T_b + P_c T_c)/1000 + w_S$. The net value over the horizon is $H(Db\ c) - C - \mathbf{01}$ [dormant], so retaining an active session is worthwhile when demand ex- ceeds $D\text{ret} = c/b$, and creating a dormant one when it exceeds $D^{\text{cre}} = (cH + C_0)/(bH)$. Modeling demand as Gaussian with mean $\hat{D}_m$ and deviation s_m yields the probabilities that each action has positive value:

$$\pi_m^{\text{ret}} = 1 - \Phi\left(\frac{D_m^{\text{ret}} - \hat{D}_m}{s_m}\right), \qquad \pi_m^{\text{cre}} = 1 - \Phi\left(\frac{D_m^{\text{cre}} - \hat{D}_m}{s_m}\right). \quad (6)$$

➢ *Complexity*

Per epoch, CONTEXTWEAVE performs $O(M(F + n^{-}))$ work

Algorithm 1 CONTEXTWEAVE lifecycle control at epoch t

```
for each site m do
    f_m ← aggregate context from broker (stale if missing)
    D̂_m ← forecast (4) (+ federated residual); s_m ← sqrt(v_m + s_0)
    compute α_m, b, c, C_0; D^ret, D^cre; π^ret, π^cre by (6)
    if x_m = DORMANT and t − t^dis ≥ T_cd and π^cre ≥ θ_on then
        CREATE(m); low_m ← 0
    else if x_m = ACTIVE then
        if π^ret_m ≥ θ_off then low_m ← 0                ▷ extend
        else
            low_m ← low_m + Δ; T ← T_s
            if h_m and D̂_m σ_m < 0.2 then T ← max(Δ, T_s/2)
            end if
            if low_m ≥ T and age_m ≥ T_min then
                DISSOLVE(m)                ▷ drain, delete, revoke, release
            end if
        end if
    end if
end for
update forecast error v_m and (CW-FedLearn) β_m; FedAvg every 60 s
```

➢ *Lifecycle Policy*

Decisions use asymmetric evidence (Algorithm 1). A dor-mant site is created when $\pi^{\text{cre}}\ \theta \geq$on and its cooldown T_{cd} since the last dissolution has elapsed. An active site is extended while $\pi^{\text{ret}}\ \theta_{\text{off}}$, with $\geq \theta_{\text{off}} < \theta_{\text{on}}$ providing hysteresis. It is dissolved only after π^{ret} has stayed below θ_{off} for T_s seconds and the session is at least T_{min} old. A privacy safeguard halves T_s when the session holds sensitive state but the expected sensitive arrival rate $\hat{D}\ \sigma$ is below 0.2 tasks/s, so sessions that hold sensitive data without sensitive demand retire sooner.

Defaults are $\theta_{\text{on}} = 0.7$, $\theta_{\text{off}} = 0.3$, $T_s = 20$ s, $T_{\text{min}} = 30$ s and $T_{\text{cd}} = 30$ s, chosen on a development seed that is excluded from all reported results.

➢ *Inner Placement*

Within active sessions every lifecycle policy in our study uses the same inner placement, so that differences between them are attributable to lifecycle control alone. Each slot, a site admits its own tasks up to utilization $\rho^{-} = 0.85$, ordering sensitive before non-sensitive and interactive before analytic tasks; overflow goes to the ACTIVE neighbor with the most spare capacity, and remaining tasks go to the cloud. Tasks arriving at a site that is BOOTING or RETIRING are handled in the same way, which is how cold starts appear in latency.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

Table 2 Simulation Parameters

Parameter	Value
Edge capacity C, admission cap $\bar{\rho}$	25 Gcycles/s, 0.85
Uplink R_u / backhaul R_b	100 Mb/s / 1 Gb/s
Access / neighbor-hop delay δ_e, δ_h	2 ms / 5 ms
WAN RTT / jitter median, σ	40 ms / 10 ms, 0.9
Cloud speed C_{cl}	40 Gcycles/s per task
Session base power P_o	70 W
Boot P_b, T_b / cleanup P_c, T_c	200 W, 8 s / 120 W, 3 s
Dynamic energy κ_e / κ_{cl}	4 J/Gcycle / 3 J/Gcycle
WAN transfer energy ϵ_{wan}	0.2 J/MB
Retention TTL (persistent) T_r	300 s
Weights w_L, w_M, w_E	2/s, 0.5/miss, 4/kJ
Weights w^{mb}, w^{res}, w_S	0.1/MB, 0.002/s, 0.5
Epoch Δ^P ^{P}H	5 s, 60 s

For F features and mean neighbor degree $\bar{n}$, and FedAvg costs, horizon

$O(MF)$ every minute. Each site's decision depends only on its own features and its neighbors' states, so the controller can run fully distributed at the sites.

V. EVALUATION METHODOLOGY

➢ *Simulator*

We implemented a discrete-time simulator (1 s slots, 1 h horizon) of the model in Section III in Python. For each scenario and seed, the workload and mobility trace is generated once and replayed for every policy (common random numbers), so policy comparisons are paired. Table II lists the parameters. Base power, boot and cleanup costs reflect a small edge server instantiating a containerized or unikernel service; the WAN round-trip and jitter reflect a regional cloud. All tuning was done on a single development seed (seed 1); reported results use 20 held-out seeds (100–119) for the main comparison and 10 (100–109) for ablation and sensitivity. We report means with 95% confidence intervals across seeds.

➢ *Scenarios*

Four scenarios stress different lifecycle decisions.

- *Pedestrian* (9 sites, 3 3 grid of 200 m cells): 120 users ×follow random waypoints at 1 m/s to 2 m/s with exponential pauses; each alternates app activity (mean on 120 s, off 300 s) and issues 1 task/s when active. Three crowd events per hour draw 40% of users to a venue cell for 10 min, producing mobility-driven demand. Mix: 70% interactive; 40% sensitive. *Vehicular* (8 roadside units, 500 m segments): vehicles enter in traffic waves (alternating 240 s to 600 s peak and off-peak phases) at 20 m/s to 33 m/s and issue 2 tasks/s each, so demand propagates downstream. Mix: 85% interactive; 30% sensitive. *IoT* (9 sites): static deployments with heterogeneous base rates (0.3 to 7 tasks/s), slow periodic variation and a multiplicative random walk; no mobility signal. Mix: 20% interac-tive; 50% sensitive.

- *Bursty* (9 sites): Markov-modulated on/off demand with idle periods of mean 400 s and bursts of mean 60 s at 8 to 18 tasks/s with a 10 s ramp; no mobility signal. Mix: 60% interactive; 30% sensitive.

➢ *Baselines*

- *Static*:

Sessions are permanently active at sites whose longrun mean demand exceeds the retain break-even *D*ret, computed with perfect knowledge of the whole trace (a best-case planning baseline). *Continuous*: all sites permanently active. *Lyapunov*: all sites permanently active; each slot a drift-pluspenalty rule ($V = 10$) selects the share of local arrivals to offload to the cloud and a DVFS level $f \in \{0.6, 0.8, 1.0\}$, in the spirit of Lyapunov-based online offloading [3]. *ReactiveTTL*: a single-threshold autoscaler that creates a session when a smoothed rate (10 s time constant) exceeds *D*cre and dis-solves it after 60 s below that threshold, typical of keep-alive autoscaling. *Oracle*: an omniscient look-ahead reference that knows the true arrivals over the next H (shifted by T_b for dormant sites) and creates or dissolves at the break-evens; it is not an optimal bound, because it ignores churn and neighbor effects, but it isolates the value of perfect demand knowledge. CONTEXTWEAVE uses the rule layer by default; CW-FedLearn adds the federated residual learner.

➢ *Metrics*

Latency: Mean and 95th-percentile task latency, and deadline-miss rate. *Energy*: total edge, lifecycle and cloud energy per hour. *Privacy exposure*: $_{res}$E and $_{idle}$ Ein site-hours per hour, and $_{cl}$ in MB/h. E *Stability*: creations per hour, oscillation rate (share of dissolutions followed by re-creation at the same site within 60 s), and transition fallback (share of tasks arriving at a BOOTING or RETIRING site). *Detection*:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

at each epoch a site *needs* a session when its true mean demand over the next H exceeds Dret; we report precision and recall of the BOOTING ACTIVE stateU against this label, the falseactivation rate (creations with no need), and, for CONTEXTWEAVE, the expected calibration error (ECE) and Brier score of π^{ret}. We also report the objective J of (3) per hour.

VI. RESULTS

➢ *Overall Comparison*

Table III reports all metrics, and Fig. 3 shows the energy– fallback 2.3% versus 10.3%), and worse in the vehicular miss trade-off. Unless stated otherwise, differences below are and IoT scenarios, because a greedy break-even rule that paired over 20 seeds and significant at $p < 0.01$ (paired t-test).

- *Against Persistent Infrastructure.* Relative to Continuous provisioning, CONTEXTWEAVE reduces energy by 24.5% (pedestrian), 26.9% (vehicular), 45.4% (IoT) and 70.8% (bursty), and resident exposure $_{res}$E by 35.7%, 37.4%, 58.8% and 81.9%. Idle resident exposure, sensitive state held while no task is being served, falls by 88–98% in every scenario (Fig. 4). These reductions are not free: deadline misses rise by 1.1, 0.2, 1.8 and 2.1 percentage points, respectively, mostly because tasks arriving during a cold start or at a dormant site are served by the cloud, and because sensitive data then travels to the cloud ($_{cl}$ of 450E MB/h to 4550 MB/h instead of near zero). Under the objective weights of Table II, the net effect on J ranges from no significant change in the pedestrian scenario (0.07%, p =− 0.61) to 3.4% (vehicular),− 1.6% (IoT, p =− 0.03) and 20.0% (bursty). The more demand is concentrated in time − and space, the more ephemeral operation pays; when demand is spread over most sites most of the time, it mainly converts energy and exposure into a small latency cost.

- *Against reactive autoscaling.* The single-threshold keepalive policy is the natural lifecycle competitor. CONTEXTWEAVE attains a lower J in all four scenarios (− 0.24%, p = 0.025; −1 .4%; − 2.0%; − 1.7%). How it does so differs by scenario. In the pedestrian and bursty scenarios it uses 10.0% and 17.8% less energy and 13.2% and 29.5% less resident exposure for a 0.3-point increase in misses; in the vehicular scenario it is better on energy (4.4%), exposure (6.8%) and misses (− 0.24 points) simultaneously. In the IoT scenario it moves in the opposite direction: it keeps sessions longer at moderately loaded sites (11.3% more energy), which cuts cloud exposure by 36% and misses by 0.75 points. In every scenario it changes state less erratically (Table IV): oscillation falls from 44.4% to 22.9% (pedestrian), 24.5% to 14.6% (vehicular) and 21.9% to 0% (IoT), and creations fall by 15% and 38% in the pedestrian and IoT scenarios.

- *Against online offloading.* The Lyapunov baseline keeps every site powered and therefore matches Continuous provisioning on energy and exposure. Its per-slot offloading, however, is a stronger *inner* policy than our shared placement: by sending analytic tasks to the faster cloud when local queues build, it achieves the lowest misses of all policies in the pedestrian, vehicular and IoT scenarios and the lowest J in the pedestrian scenario (J 4.3% below CONTEXTWEAVE). In the IoT scenario the two are statistically tied (p = 0.28), and CONTEXTWEAVE is better in the vehicular (− 1.3%, p = 0.02) and bursty (− 19.8%) scenarios. Because CONTEXTWEAVE separates lifecycle control from placement, the two mechanisms are complementary rather than competing; we return to this in Section VII.

- *Against perfect foresight.* The Oracle look-ahead is within± 2.3% of CONTEXTWEAVE in J in every scenario. It is better in the bursty scenario (+1.9% for CONTEXTWEAVE), where it pre-boots ahead of each burst (Table IV, transition ignores churn and neighbor effects is not optimal even with perfect knowledge. The small gap indicates that, at these weights, most of the attainable lifecycle benefit is captured; the remaining loss concentrates in bursty demand without precursors.

Table 3 Main Results (Mean Over 20 Held-Out Seeds Per Scenario). Lower Is Better For Every Column. Bold: Best Non-Oracle Policy.

Policy	**p95 lat.**	**Miss**	**Energy**	***E* res**	***E* idle**	***E* cl**	**Creations**	**J**
	(ms)	(%)	(kJ/h)	(site-h/h)	(site-h/h)	(MB/h)	(/h)	(/h)
Pedestrian Static	303	10.5	2816	8.70	1.08	5	–	63.94
Continuous	302	10.5	2892	9.00	1.22	**0**	–	64.04
Lyapunov	**271**	**9.8**	2871	9.00	1.37	3367	–	**61.35**
Reactive-TTL	308	11.3	2426	6.67	0.34	588	71	64.14
CONTEXTWEAVE	311	11.6	2184	5.79	0.14	595	59	63.99
CW-FedLearn	313	11.9	**2108**	**5.52**	**0.11**	777	56	64.24
Oracle	310	11.8	2191	5.85	0.13	1053	55	64.18
Vehicular Static	246	7.6	2677	7.65	2.71	**14**	–	65.83
Continuous	246	7.6	2677	7.65	2.71	14	–	65.83
Lyapunov	**243**	**7.0**	2675	7.65	2.71	46	–	64.41
Reactive-TTL	247	8.0	2053	5.15	0.73	541	55	64.64
CONTEXTWEAVE	246	7.8	1960	4.79	**0.31**	453	53	**63.72**
CW-FedLearn	247	7.9	**1910**	**4.69**	0.34	684	40	63.79
Oracle	247	8.1	1861	4.32	0.12	841	39	64.04

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

IoT Static	488	7.0	1662	3.95	0.11	4463	–	60.91
Continuous	446	5.0	2949	8.99	2.54	**12**	–	61.21
Lyapunov	**423**	**4.9**	2938	8.99	2.74	930	–	**59.95**
Reactive-TTL	496	7.6	**1460**	**3.05**	**0.02**	7071	20	61.68
CONTEXTWEAVE	482	6.8	1621	3.71	0.05	4547	12	60.43
CW-FedLearn	488	7.1	1556	3.45	0.04	5456	13	60.89
Oracle	500	7.7	1424	2.95	0.02	7345	13	61.88
Bursty Static	**327**	13.9	789	2.28	1.24	3478	–	23.99
Continuous	332	11.2	2491	8.52	6.88	**0**	–	28.77
Lyapunov	331	**11.2**	2490	8.52	6.96	1	–	28.73
Reactive-TTL	336	13.1	884	2.19	0.99	1110	59	23.53
CONTEXTWEAVE	335	13.4	**728**	**1.55**	**0.48**	1576	64	**23.13**
CW-FedLearn	336	12.8	990	2.64	1.34	863	54	23.76
Oracle	335	12.9	709	1.40	0.28	1121	48	22.70

➢ *Lifecycle Behavior and Detection Quality*

Fig. 5 shows lifecycle traces at one site per scenario. In the pedestrian trace, all three lifecycle policies keep the venue site active through the crowd event (minutes 22–40). Before the event, demand at this site sits only modestly above the cloud-fallback break-even, and each policy dissolves the site at least once; CONTEXTWEAVE does so earliest because it counts spare capacity at active neighbors as a fallback, which lowers the value of a local session. In the bursty trace, all lifecycle policies follow the bursts. The Oracle boots before each burst begins, whereas CONTEXTWEAVE and Reactive-TTL boot after the ramp has started; this cold-start penalty is the main source of CONTEXTWEAVE's residual gap in that scenario.

Table IV quantifies detection against the ground-truth need label. In the three scenarios whose demand has temporal structure (pedestrian, vehicular, IoT), CONTEXTWEAVE reaches 85– 87% precision and 78–91% recall, with false activations of 6–9% in the pedestrian and IoT scenarios. The vehicular scenario shows 28% false activations for both CONTEXTWEAVE and Reactive-TTL: wave fronts briefly push a segment above the creation threshold before the wave thins out. CWFedLearn reduces this to 12% and cuts vehicular oscillation from 14.6% to 4.3%, the clearest benefit of the learned residual. Bursty demand without precursors is hard for every causal policy (precision 40–59%, recall 55–62%), while the Oracle reaches 98%/92%. Confidence is reasonably calibrated: the ECE of π^{ret} is 0.07 (IoT) to 0.15 (pedestrian, bursty) for CONTEXTWEAVE, and 0.02–0.18 for CW-FedLearn.

➢ *Which Context Signals Matter*

Table V removes one signal group, or the safeguards, at a time. Four findings stand out.

- *Trend features are the most broadly useful signal.* Removing r_5 and $r_5 - r_{30}$ raises J in every scenario (by 0.1% to 1.9%) because the controller reacts later to emerging demand: energy falls by 4–11%, but misses rise and cloud exposure grows by 62–118%.

- *Mobility helps exactly where mobility exists.* Removing inflow and presence raises J by 0.47% (pedestrian) and 1.03% (vehicular), with misses up by 0.23 and 0.36 points; in the IoT and bursty scenarios, where no mobility signal is available, it has no effect, as expected.

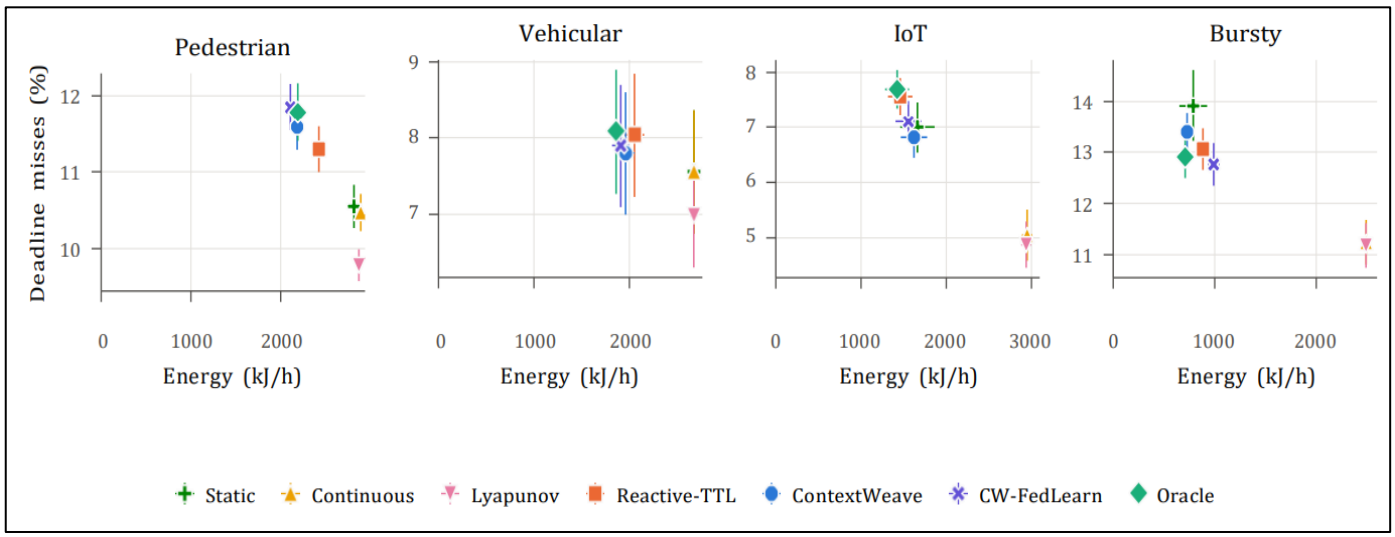


Fig 3 Energy versus deadline-miss rate (mean ± 95% CI over 20 seeds). Persistent policies (Continuous, Lyapunov) sit on the right; lifecycle policies trade a small increase in misses for large energy savings. Static coincides with Continuous in the vehicular scenario, where every site exceeds the break-even on average.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

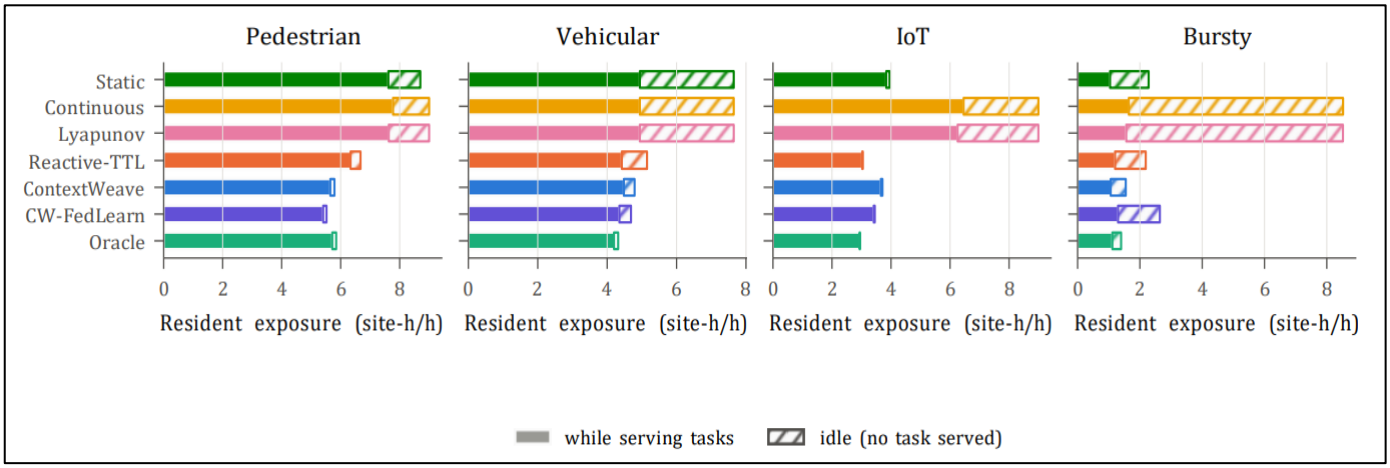


Fig 4 Resident privacy exposure E_{res} , split into exposure accrued while the site serves tasks (solid) and while it serves none (hatched). Ephemeral operation removes most idle exposure.

Table 4 Lifecycle Stability and Detection Quality (20 Seeds)

Policy	Osc. (%)	Trans. fb (%)	Prec. (%)	Rec. (%)	False act. (%)	ECE
Pedestrian Reactive-TTL	44.4	1.1	85.0	87.0	14.5	–
Pedestrian CONTEXTWEAVE	22.9	1.0	87.4	77.7	9.2	0.15
Pedestrian CW-FedLearn	17.4	1.0	88.2	74.8	7.2	0.18
Pedestrian *Oracle*	17.6	0.5	100.0	89.6	0.0	–
Vehicular Reactive-TTL	24.5	1.0	79.0	88.1	28.7	–
Vehicular CONTEXTWEAVE	14.6	0.6	85.5	88.9	28.3	0.11
Vehicular CW-FedLearn	4.3	0.5	86.9	88.0	11.8	0.11
Vehicular *Oracle*	2.2	0.2	99.5	96.9	0.0	–
IoT Reactive-TTL	21.9	0.7	96.5	84.3	3.4	–
IoT CONTEXTWEAVE	0.0	0.5	85.4	90.8	5.6	0.07
IoT CW-FedLearn	0.4	0.5	89.6	88.5	5.0	0.02
IoT *Oracle*	0.0	0.5	100.0	84.3	0.0	–
Bursty Reactive-TTL	15.5	9.8	45.3	58.4	32.3	–
Bursty CONTEXTWEAVE	13.3	10.3	59.0	55.1	32.4	0.12
Bursty CW-FedLearn	12.1	8.7	40.2	61.8	32.2	0.15
Bursty *Oracle*	9.6	2.3	97.7	92.0	0.0	–

- *The privacy signal is decisive when sensitive data is common.* In the IoT scenario (50% sensitive), dropping the privacy terms makes the controller value only latency and energy. It then dissolves far more aggressively, saving 38.5% energy but quadrupling cloud exposure (+422%) and raising misses by 1.9 points; *J* rises by 3.9%. The effect is smaller but still significant in the pedestrian and bursty scenarios, and negligible in the vehicular one.

- *Safeguards prevent churn.* Without hysteresis, minimum retention and cooldown, creations rise by 138% (pedestrian) and 173% (IoT), and *J* rises by 2.4% and 2.9%. In the bursty scenario, where sessions are short-lived anyway, the safeguards neither help nor hurt.

- The *resource* signal (neighbor absorption) is the one component with mixed value. It improves *J* in the vehicular scenario (+0.39% when removed) but slightly worsens it in the pedestrian scenario (0.−42 % when removed, at the edge of significance): without it, energy rises by 18% while misses fall by 0.75 points, so neighbor consolidation trades latency for energy there. This signal therefore encodes a policy preference more than a uniformly better estimate.

➢ *Sensitivity to Lifecycle Parameters*

Fig. 6 sweeps the three churn controls. The retire-evidence interval T_s is the strongest lever: raising it from 5 s to 80 s halves creations in the pedestrian scenario (80 to 39 per hour) and lowers misses from 11.9% to 11.2%, at 8.6% more energy. Hysteresis width has a similar but weaker effect on creations (84 to 52 per hour as $\theta_{on} - \theta_{off}$ grows from 0 to 0.6) and almost none in the bursty scenario, where bursts are far above or below the break-even. A cooldown of at least 60 s eliminates re-creation within a minute by construction, but at 120 s it begins to cost recall (77.8% to 73.4% in the pedestrian scenario). Within the tested ranges, no parameter changed *J* by more than 1.5%, so the controller is not fragile to these settings; they mainly select a point on the churn–energy curve.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

Table 5 Ablation: Change Relative To Full Contextweave When One Signal Group Or The Safeguards Are Removed (10 Seeds). ΔJ And ΔE In %, ΔMiss In Percentage Points. Bold ΔJ: 95% Ci Excludes Zero. Positive ΔJ Means The Removed Component Was Helping.

	−Trend			−Mobility			−Resource			−Privacy			−Safeguards		
Scenario	ΔJ	ΔE	ΔMiss	ΔJ	ΔE	ΔMiss	ΔJ	ΔE	ΔMiss	ΔJ	ΔE	ΔMiss	ΔJ	ΔE	ΔMiss
Pedestrian	**+1.12**	−7	+0.68	**+0.47**	−2	+0.23	**−0.42**	+18	−0.75	**+0.90**	−4	+0.48	**+2.42**	−4	+1.05
Vehicular	**+0.10**	−4	+0.15	**+1.03**	−3	+0.36	**+0.39**	+6	−0.08	+0.04	0	+0.01	+0.11	−2	+0.07
IoT	**+1.93**	−11	+0.77	0.00	0	0.00	−0.07	+1	−0.04	**+3.93**	−39	+1.91	**+2.90**	−14	+1.04
Bursty	**+0.93**	−7	+0.56	0.00	0	0.00	0.00	0	0.00	**+0.52**	+5	−0.02	0.00	−4	+0.22

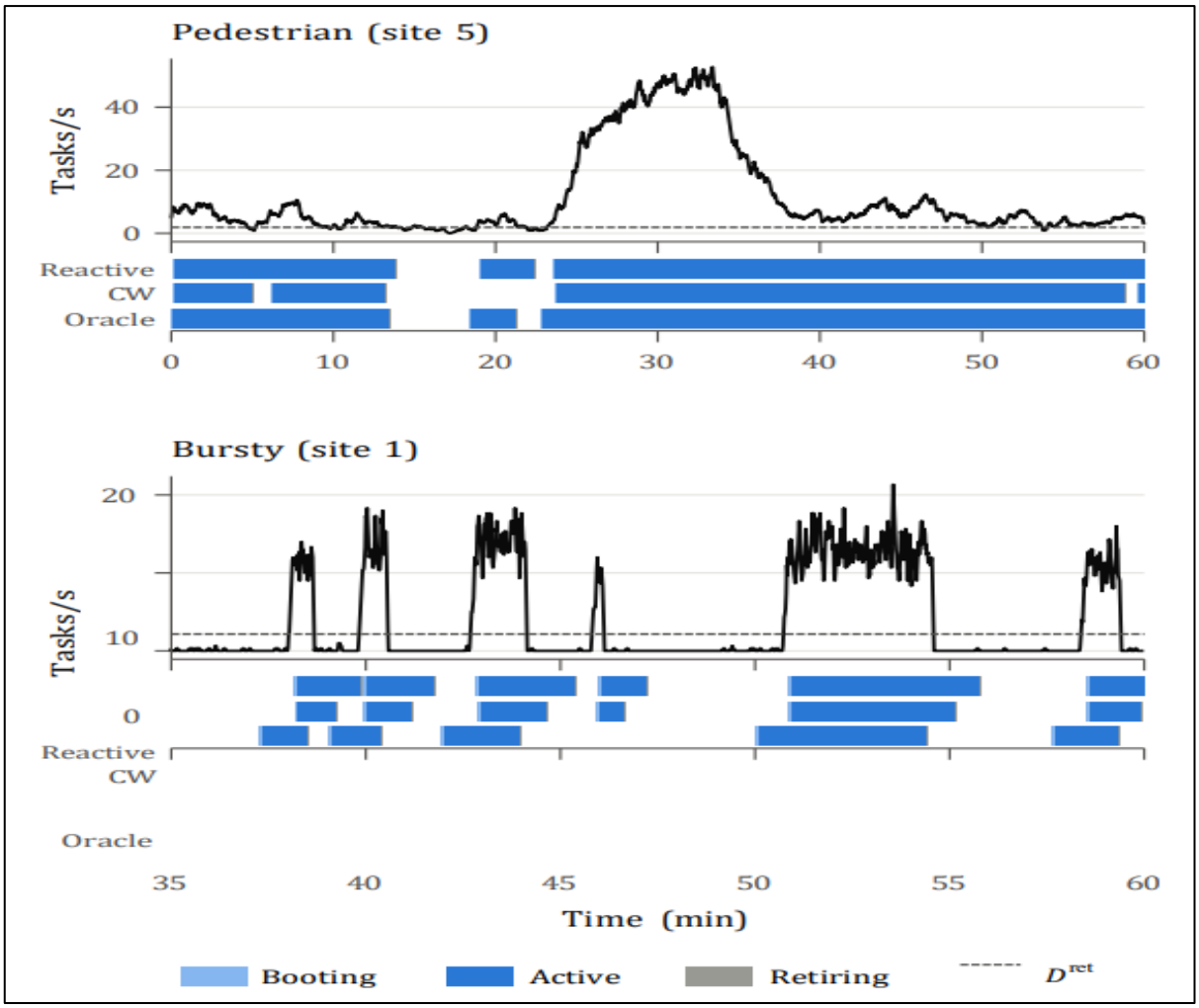


Fig 5 Demand (smoothed) and lifecycle state at one site for Reactive-TTL, CONTEXTWEAVE (CW) and the Oracle. Top: pedestrian scenario, full hour; bottom: bursty scenario, minutes 35–60. The dashed line is the cloud-fallback retain break-even.

➢ *Trade-off Frontiers*

Varying the privacy weight (×{0, 1, 3, 10, 30 on } w^{mb} and $_P$ w^{res}) and the energy weight ($_P$ 0.5, 1×{, 2) traces the frontier between } resident exposure and deadline misses (Fig. 7). In the bursty scenario the CONTEXTWEAVE frontier lies below ReactiveTTL's wherever both are defined up to about 2.4 site-h/h: at equal exposure, CONTEXTWEAVE misses fewer dead-lines (for example 12.9% versus 13.2% at 2.2 site-h/h). Only ReactiveTTL's most exposure-heavy settings reach lower miss rates. In the pedestrian scenario the frontiers of CONTEXT-WEAVE, Reactive-TTL and the Oracle largely overlap, with CONTEXTWEAVE lower at intermediate exposure. Over the 15 weight settings, CONTEXTWEAVE achieves a lower *J* than Reactive-TTL in 10 settings in each scenario. Persistent policies sit at the high-exposure end of every frontier: they buy their lower miss rate with 1.5–5.5 more resident exposure than × CONTEXTWEAVE at default weights.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

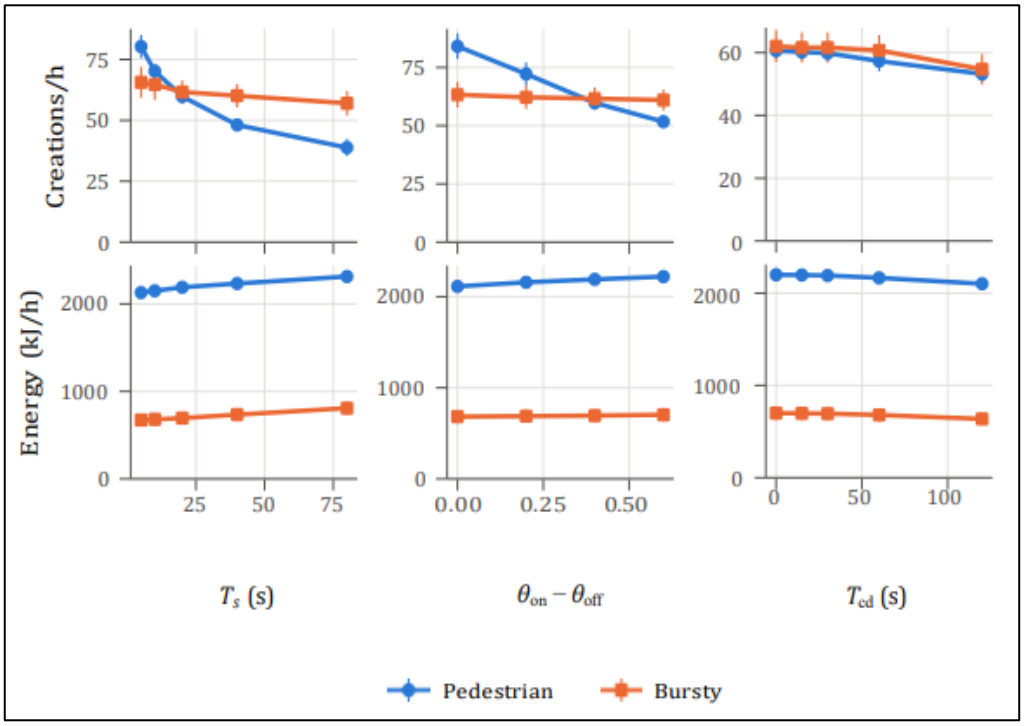


Fig 6 Sensitivity of Creations per Hour (top) and Energy (Bottom) to the Retire-Evidence Interval T_s, Hysteresis Width, and Cooldown T_{cd} (10 Seeds, 95% CI).

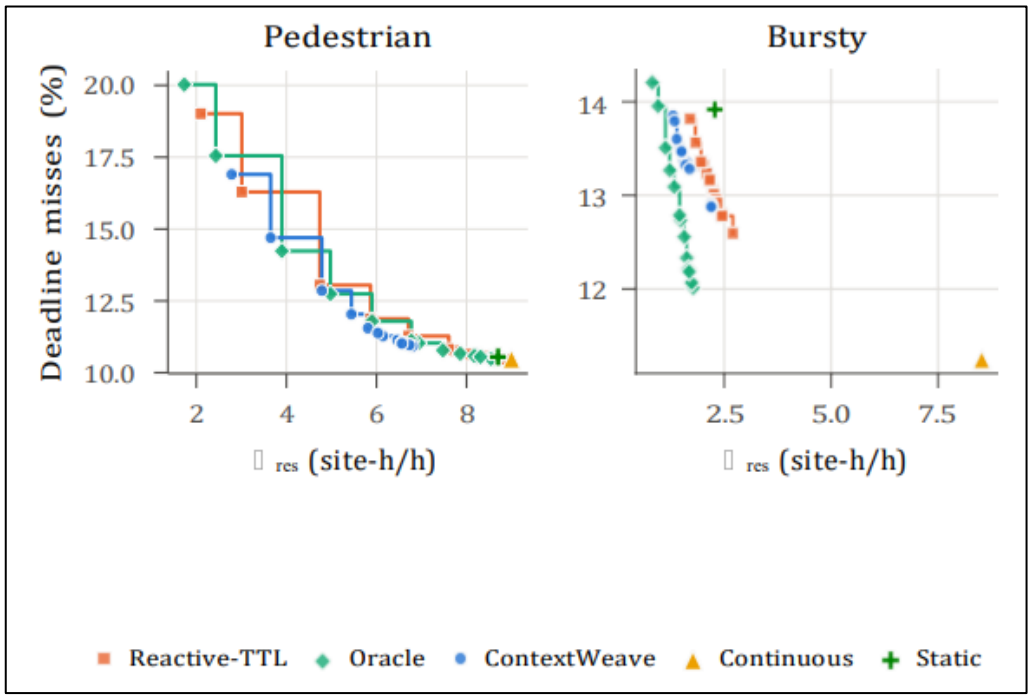


Fig.7 Resident Exposure Versus Deadline Misses as the Privacy and Energy Weights Vary (15 Settings, 10 Seeds Each). Lines are the Non-Dominated Frontier of Each Lifecycle Policy.

➢ *Robustness to Impaired Context*

Online controllers cannot assume perfect observations, so we delayed, dropped and perturbed the context stream (Fig. 8). Delaying context by 30 s raises *J* by 0.5% in the pedestrian scenario and 2.1% in the bursty scenario, where recall falls First, ephemeral operation shifts exposure: dissolving a session reduces resident exposure but can increase cloud exposure, which is why the privacy terms in the objective and the privacy safeguard matter (Table V, IoT). Second, lifecycle events are themselves metadata. A site that is created only when a crowd gathers reveals that it gathered. Protecting this side channel, for example by randomized activation or cover sessions, is outside our scope and is an important direction for future work.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

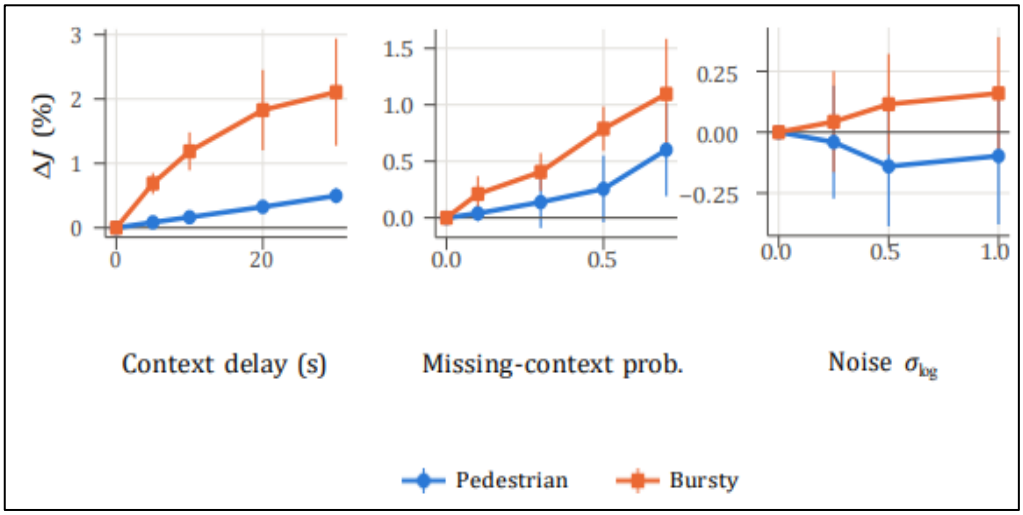


Fig 8 Change in Objective J Under Delayed, Missing and Noisy Context, Relative to Unimpaired Context (10 seeds, 95% CI).

From 54% to 38% because short bursts are already half over when they are seen. Losing 70% of context reports costs 0.6% and 1.1%. Multiplicative noise up to $\sigma_{log} = 1$ changes J by less than 0.2%, because the peak-seeking rule layer over-reacts to noise: it keeps sessions longer (20% more energy in the pedestrian scenario at $\sigma_{log} = 1$) and thereby reduces misses. Timeliness of context therefore matters more than its precision, which argues for local, low-latency brokers at each site.

VII. DISCUSSIONS

➢ *When Ephemeral Operation Pays*

The results suggest a simple rule of thumb. The benefit of lifecycle control grows with the share of site-time during which demand is below the retain break-even, which is high under bursty, event-driven or wave-like demand and low when most sites carry steady load. In the pedestrian scenario, where demand is spread across a grid of busy cells, CONTEXT-WEAVE saves a quarter of the energy and a third of the resident exposure, yet the objective is unchanged because the latency cost offsets the savings at our weights. Operators should therefore treat the weights as an explicit policy statement. The frontiers in Fig. 7 show which latency is purchased with which exposure, and the weights pick one point on that curve.

➢ *Lifecycle and Placement Are Complementary*

The Lyapunov baseline outperforms CONTEXTWEAVE on latency because its inner offloading policy is better than the simple placement we deliberately held fixed across lifecycle policies, not because persistent infrastructure is better. CONTEXTWEAVE's lifecycle layer and a Lyapunov-based online offloading policy [3] act on different variables (whether a session exists versus how tasks are split among existing sessions and the cloud), so they can be composed directly. Likewise, the cold-start penalty that dominates the bursty scenario could be reduced by lighter-weight substrates or by online service-function-chain provisioning that adjusts capacity and routing as demand evolves [11].

➢ *Privacy Interpretation*

Our exposure measures are deliberately operational: how long sensitive state is resident, how much of that time is idle, and how much sensitive data leaves for the cloud. They do not quantify inference risk in an information-theoretic sense, and they treat all sensitive tasks alike. Two caveats follow.

➢ *Threats to Validity*

Simulation fidelity. Queueing uses a fluid approximation with stochastic waiting and service times rather than packetor container-level emulation; boot and cleanup are modeled as fixed durations and powers. Absolute numbers should be read in this light; the paired, seed-matched comparisons are more robust than absolute levels. *Workloads.* The traces are synthetic, with structure (crowd events, traffic waves, on/off bursts) chosen to stress lifecycle decisions; trace-driven validation on real mobility and request logs remains necessary. *Parameters.* Controller parameters were set on one development seed; Section VI shows that J varies by at most 1.5% across the tested ranges, but other environments may favor different settings. *Baselines.* Our Oracle is a greedy look-ahead, not an optimal offline solution, so it is a reference point rather than a bound. *Scope.* We model one session per site and one server per session. Multi-tenant sessions, partial chains and heterogeneous hardware are natural extensions. To support replication, the simulator, configuration and analysis scripts that produced every number in this paper are released with it.

VIII. CONCLUSION

This paper treated the existence of the edge execution substrate as an orchestration variable and asked whether it can be controlled from privacy-minimized context. CONTEXTWEAVE combines an interpretable demand forecast, an uncertaintyaware estimate of a session's net value, and asymmetric, hysteretic lifecycle rules that are kept separate from inner placement. Across four workload families and 20 held-out seeds, it reduced energy by 25–71% and resident privacy exposure by 36–82% relative to continuous provisioning, eliminated most idle exposure, and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26oct866

cost 0.2–2.1 percentage points of additional deadline misses. Against a reactive keepalive autoscaler it achieved a lower objective in every scenario while roughly halving oscillation where demand is noisy, and it came within 2.3% of an omniscient look-ahead. Ablations showed that trend features and, where available, mobility inflow carry most of the detection value, that the privacy signal prevents the controller from trading local exposure for cloud exposure, and that the churn safeguards matter wherever demand hovers near the break-even. Future work will compose CONTEXTWEAVE with stronger inner offloading policies, evaluate it on real traces and an emulated container platform, and address lifecycle metadata as a privacy side channel.

REFERENCES

[1.] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobilityaware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, 2018.

[2.] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4242–4251, 2019.

[3.] T. Liu, L. Fang, Y. Zhu, W. Tong, and Y. Yang, "A near-optimal approach for online task offloading and resource allocation in edge-cloud orchestrated computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2687–2700, 2022.

[4.] X. He, R. Jin, and H. Dai, "PEACE: Privacy-preserving and cost-efficient task offloading for mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1814–1824, 2020.

[5.] X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, and S. Wan, "An edge computing-enabled computation offloading method with privacy preservation for Internet of connected vehicles," *Future Gener. Comput. Syst.*, vol. 96, pp. 89–100, 2019.

[6.] Y. Qian, L. Hu, J. Chen, X. Guan, M. M. Hassan, and A. Alelaiwi, "Privacy-aware service placement for mobile edge computing via federated learning," *Inf. Sci.*, vol. 505, pp. 562–570, 2019.

[7.] W.-Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, and A. A. A. El-Latif, "Secure and optimized load balancing for multitier IoT and edge-cloud computing systems," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8119–8132, 2021.

[8.] A. Rautaray, "The "ephemeral" family cloud: A secure, short-lived, privacy-preserving personal cloud architecture for modern households," *J. Inf. Syst. Eng. Manage.*, vol. 7, no. 4, 2022.

[9.] C. Sonmez, A. O¨ zgo¨vde, and C. Ersoy, "Fuzzy workload orchestration for edge computing," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 2, pp. 769–782, 2019.

[10.] C. Sonmez, C. Tunca, A. O¨ zgo¨vde, and C. Ersoy, "Machine learningbased workload orchestrator for vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2239–2251, 2021.

[11.] Z. Zhou, Q. Wu, and X. Chen, "Online orchestration of cross-edge service function chaining for cost-efficient edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1866–1880, 2019.

[12.] G. Sun, Y. Li, Y. Li, D. Liao, and V. Chang, "Low-latency orchestration for workflow-oriented service function chain in edge computing," *Future Gener. Comput. Syst.*, vol. 85, pp. 116–128, 2018.

[13.] R. Bolla, A. Carrega, M. Repetto, and G. Robino, "Improving efficiency of edge computing infrastructures through orchestration models," *Computers*, vol. 7, no. 2, 2018, art. no. 36.

[14.] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, 2018.

[15.] X. Lu, Y. Liao, P. Lio`, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48 970–48 981, 2020.

[16.] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4755–4763, 2019.

[17.] X. Chen, S. Chen, X. Zeng, X. Zheng, Y. Zhang, and C. Rong,

[18.] "Framework for context-aware computation offloading in mobile cloud computing," *J. Cloud Comput.*, vol. 6, no. 1, 2017, art. no. 1.

[19.] D. M. A. da Silva and R. C. Sofia, "A discussion on context-awareness to better support the IoT cloud/edge continuum," *IEEE Access*, vol. 8, pp. 193 686–193 694, 2020.