

Trust- and Residual-Energy-Aware Edge Scheduling for Intermittent Urban IoT Sensing

Utham Kumar Anugula Sethupathy^{*}, Vijayanand Ananthanarayanan

Independent Researcher, Atlanta, GA, USA

Email: *mailuthamkumar@gmail.com, vijayanand31@gmail.com

How to cite this paper: Sethupathy, U.K.A. and Ananthanarayanan, V. (2026) Trust- and Residual-Energy-Aware Edge Scheduling for Intermittent Urban IoT Sensing. *Advances in Internet of Things*, **16**, 106-123. <https://doi.org/10.4236/ait.2026.164007>

Received: September 10, 2026

Accepted: October 20, 2026

Published: October 23, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Trust-aware Internet of Things scheduling requires reputation evidence that is observable and energy state that enters the scheduling decision explicitly. This study presents a joint trust-energy scheduler (JTES). Beta reputation updates only after known-answer challenges or redundant execution produces an imperfect validation label. The scheduler uses the exact 5th percentile of the Beta posterior at a documented controller refresh interval and includes both sensor residual battery and gateway residual energy. Twenty-four paired seeds, five compromise probabilities, seven schedulers, and three validator qualities produced 840 main and 72 sensitivity trials. Eight thousand task arrivals define one simulated day. At 15% compromise with moderate validation, full JTES achieved 94.80% clean timely completion (SD 3.21) and accepted 2.56% bad results (SD 3.17). Energy-only scheduling achieved 86.81% and accepted 11.32%. Posterior-mean scheduling achieved 97.01% clean timely completion, accepted 0.95% bad results, and consumed 138.98 mJ per completed task, yielding better values than the conservative-bound policy on these principal simulated metrics. Validator degradation changed accepted corruption from 2.26% to 3.64%. The simulation exposes integrity-energy and conservatism-efficiency trade-offs and does not claim production lifetime or security performance.

Keywords

Internet of Things, Edge Scheduling, Beta Reputation, Validation, Residual Energy

1. Introduction

Constrained nodes have limited memory, processing, communication, and energy resources [1]. IoT security baselines therefore emphasize observable device state

and defined security capabilities [2], while consumer-IoT guidance requires lifecycle controls rather than implicit trust [3].

Trust estimates can become overconfident if schedulers learn from outcomes that were never validated. Energy-aware scheduling is likewise incomplete when distance or queue length substitutes for residual battery and gateway energy. JTES therefore restricts trust updates to observable validation and represents both energy domains explicitly.

The research question is whether observable validation evidence and explicit dual-resource energy state can improve clean timely delivery under intermittent and compromised gateways, and what integrity cost follows from weaker validation or reduced replication. **Figure 1** summarizes the architecture.

Observable-evidence trust and energy scheduling

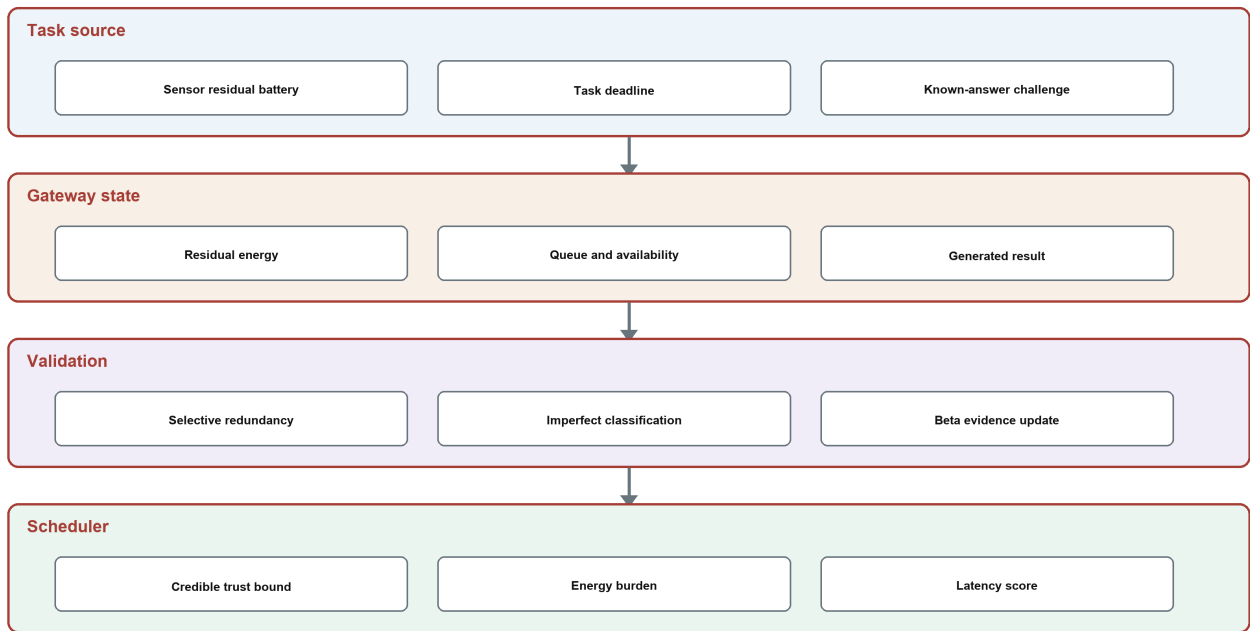


Figure 1. JTES architecture with observable validation, Beta evidence updates, and separate sensor and gateway energy states.

2. Related Work

Edge computing places computation near data sources to reduce response time and backhaul dependence [4]. Edge and fog architectures expose resource-allocation and reliability trade-offs [5] [6]. JTES studies a narrow scheduling layer within that broader architecture.

Beta reputation represents favorable and unfavorable evidence with compact sufficient statistics [7]. EigenTrust aggregates repeated local experience [8], and sensor-network reputation systems use observations to support integrity [9]. JTES follows their evidence principle: an unvalidated task creates no reputation update.

Radio-energy models for sensor networks relate electronic and amplifier cost to transmitted bits and distance [10]. Wireless-sensor surveys document how topol-

ogy, duty cycling, and routing affect lifetime [11]. The present experiment maintains sensor joules and gateway joules separately instead of treating distance as residual energy.

CoAP [12], object security [13], and constrained-device cryptography [14] illustrate mechanisms that could carry authenticated tasks and validation exchanges. Security and trust surveys [15] and IoT protocol surveys [16] emphasize that reputation does not replace identity or cryptographic protection. Simulation limitations remain substantial [17].

Recent edge studies jointly consider trust and information freshness [18], trusted data scheduling [19], and energy-aware offloading with learned trust [20]. Task dependencies and unreliable volunteer servers motivate trust-aware placement [21], while cooperative IoT scheduling [22], device-specific communication-computation allocation [23], and delay-energy optimization [24] show that service and energy objectives are tightly coupled. Real-time trust-aware fog scheduling [25] further supports evaluating trust as an operational scheduling signal rather than an isolated reputation score.

3. Synthetic IoT Environment

Figure 2 shows 400 sensors and six gateways in a 2 km by 2 km field. Gateway compromise is hidden from the scheduler. It affects only the generated result; the controller observes a noisy validation label when a challenge or redundant execution occurs.

Synthetic urban sensing field and gateway compromise ground truth

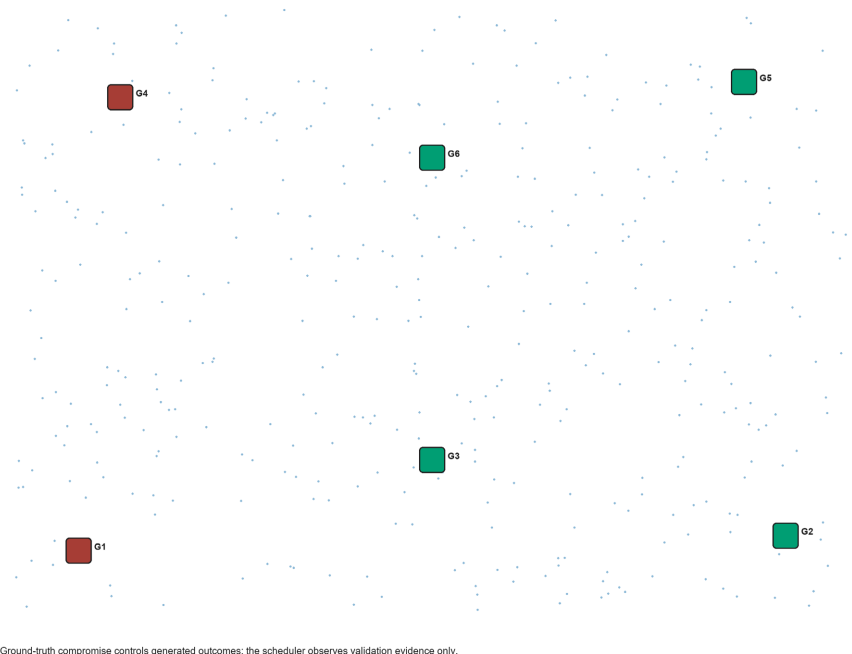


Figure 2. Representative sensing field. Compromise coloring is evaluator-only ground truth.

Table 1 defines the workload and physical units. Exactly 8000 arrivals represent one simulated day; the lifetime projection divides energy capacity by that day's per-device consumption.

Table 1. Simulation environment.

Parameter	Value
Sensors and gateways	400 and 6
Area	2 km × 2 km
Arrivals	8000 per simulated day
Critical tasks	22%
Known-answer challenges	3%
Sensor battery	18,000 J
Gateway energy budget	2,000,000 J
Compromise probability	0%, 5%, 10%, 15%, 20%

Task, Availability, and Execution Model

The simulation represents one synthetic day as 8000 discrete task arrivals. One task arrives per simulation step; no Poisson or wall-clock interarrival process is assumed. For each arrival, the originating sensor is selected uniformly from the 400 sensors. A task is marked critical independently with probability 0.22 and as a known-answer challenge independently with probability 0.03. Computational workload is sampled uniformly from 25 to 65 million cycles.

Task deadlines use a two-level distribution: 25% of tasks receive a 150 ms deadline and the remaining 75% receive a 220 ms deadline. Gateway availability is sampled independently for every task-gateway pair. The six gateway unavailability probabilities are 0.08, 0.12, 0.09, 0.14, 0.10, and 0.13, respectively; thus the corresponding availability probabilities are 0.92, 0.88, 0.91, 0.86, 0.90, and 0.87.

Before each task is scheduled, every gateway queue is reduced by 0.55 task units, bounded below by zero. After an execution is assigned to gateway g , the queue is incremented by one. The execution-success probability depends on the resulting queue:

$$P_{\text{fail},g} = 0.012 + 0.008 \min(8, Q_g),$$

so that

$$P_{\text{succ},g} = 1 - P_{\text{fail},g}.$$

The execution latency of a successful task on gateway g is

$$L_{i,g} = 22 + 0.032d_{i,g} + 6Q_g + \eta_{i,g},$$

where $d_{\{i,g\}}$ is sensor-gateway distance in meters and

$$\eta_{i,g} \sim \text{Lognormal}(2.0, 0.27).$$

A task is classified as timely only when its final accepted result is clean and its modeled task latency does not exceed that task's assigned deadline. Gateway com-

promise status is sampled once at the start of a simulated day and remains fixed for that trial. Each gateway is independently marked compromised with the scenario probability p_c . A benign gateway never generates a corrupted result in the model. A compromised gateway corrupts each attempted execution independently with probability 0.68:

$$P(\text{bad}|\text{compromised}) = 0.68, P(\text{bad}|\text{benign}) = 0.$$

The same corruption rule applies to primary executions, replicated executions, and known-answer challenge tasks. Corruption status affects the generated result but is never directly visible to the scheduler.

4. Observable Beta Reputation

4.1. Evidence Sources

A task updates reputation only when observable validation evidence is available. This occurs when a successful known-answer challenge returns a result or when redundant execution produces at least two successful gateway outcomes. Merely dispatching a replica does not guarantee a validation label, because one or both executions may fail. Validation is imperfect: in the moderate setting, a bad primary is labeled bad with 90% probability and a clean primary is falsely labeled bad with 3% probability. Unvalidated successes and failures do not change alpha or beta.

Equation (1) defines the posterior and the conservative score. The implementation numerically inverts the regularized incomplete Beta function; it does not use a Wilson approximation.

$$\theta_g | \alpha_g, \beta_g \sim \text{Beta}(\alpha_g, \beta_g), q_g = F_{\text{Beta}(\alpha_g, \beta_g)}^{-1}(0.05) \quad (1)$$

The prior is Beta (2, 1). Alpha increments after an observed clean label and beta increments after an observed bad label. Because the validation labels are imperfect, this posterior should not be interpreted as a calibrated posterior probability of true gateway integrity. Rather, it estimates the probability that a validated gateway execution will receive a clean validation label under the configured validator error model. The true-positive and false-positive rates are not explicitly inverted or incorporated into the Bayesian likelihood, so the posterior reflects validated-label cleanliness rather than latent physical integrity.

For controller efficiency, posterior counts update on every validation but q_g is refreshed after 50 new observations for that gateway. **Figure 3** shows posterior means and lower bounds in one fixed-seed run.

4.2. Validation Sensitivity

Table 2 lists the three validator scenarios. They are controlled error settings, not measured classifier performance.

4.3. Selective Replication and Result Resolution

Full JTES, Trust-only, and Posterior-mean scheduling use the same selective-

replication trigger. After the primary gateway is selected, a second execution is dispatched only when the task is critical and the primary gateway's current 5th-percentile trust bound is below 0.90. The second gateway is the next-lowest-cost currently active gateway under the same scheduling policy. Always replicate dispatches a second execution whenever a second eligible gateway is available.

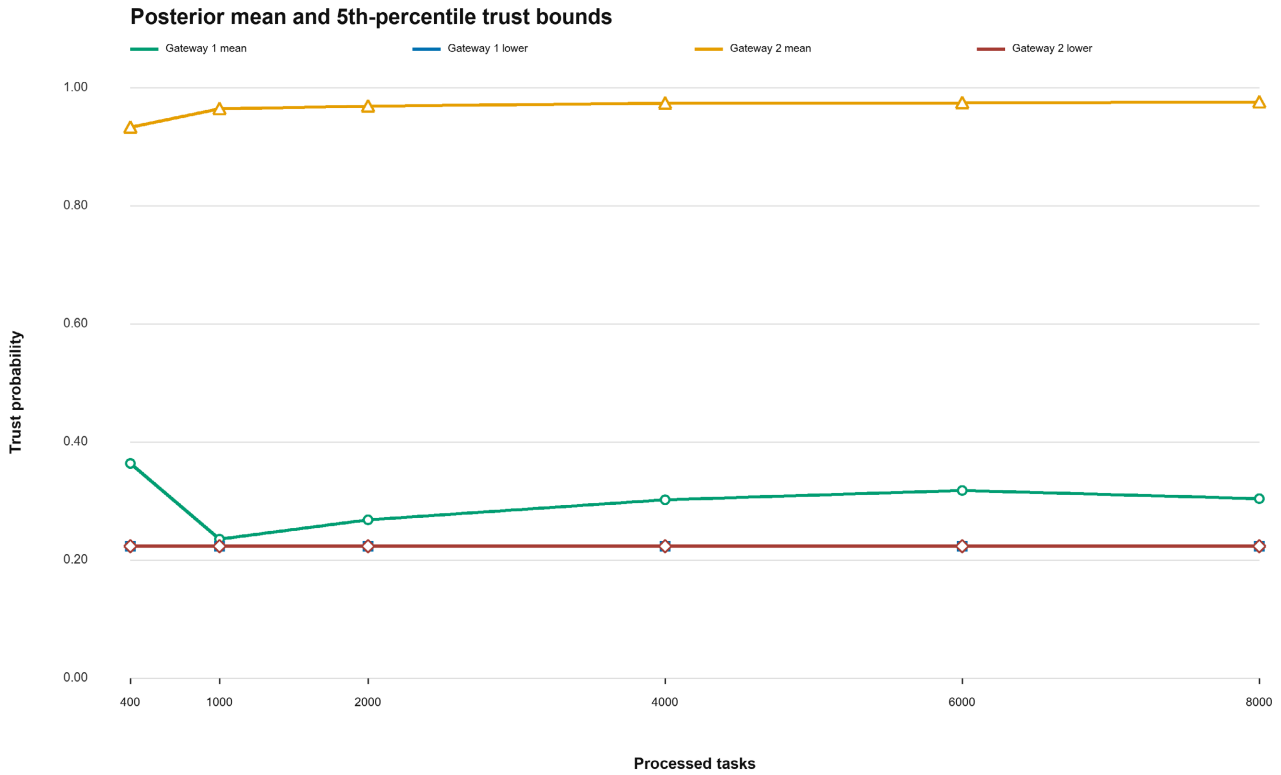


Figure 3. Posterior means and 5th-percentile Beta credible bounds during one representative run.

Table 2. Validator quality scenarios.

Quality	True-positive probability	False-positive probability
Strong	0.98	0.01
Moderate	0.90	0.03
Weak	0.75	0.08

Replica executions are modeled as parallel copies on different gateways rather than as a serial retry. If both executions succeed, the task latency is taken as the larger of the successful execution latencies because resolution waits for the returned evidence. If only one copy succeeds, that successful outcome remains available but redundant execution alone does not produce a two-result validation event.

Validation classifies the primary result. If the primary receives a bad label and a second successful result exists, the second result is accepted. Otherwise, the pri-

mary result is retained. Known-answer challenges update reputation but do not independently force task rejection or replacement. If no execution succeeds, the task is incomplete and produces no validation evidence.

5. Dual Resource Energy Model

Equation (2) gives sensor transmit energy in joules, using packet length b , electronics cost E_{elec} , amplifier coefficient ε_{amp} , and sensor-gateway distance.

$$E_{i,g}^{\text{tx}} = bE_{\text{elec}} + b\varepsilon_{\text{amp}}d_{i,g}^2 \quad (2)$$

The simulation uses $b = 4096$ bits, $E_{\text{elec}} = 50$ nJ/bit, and $\varepsilon_{\text{amp}} = 100$ pJ/bit/m².

Equation (3) defines gateway energy for idle execution time, computational work C in millions of cycles, and queue-dependent overhead.

$$E_{g,t}^{\text{gw}} = P_{\text{idle}}t_{\text{exec}} + \varepsilon_{\text{cpu}}C_t + \varepsilon_qQ_g(t) \quad (3)$$

where:

- $P_{\text{idle}} = 40$ W ;
- $t_{\text{exec}} = 0.020$ s ;
- $\varepsilon_{\text{cpu}} = 0.46$ J per million cycles;
- $\varepsilon_q = 0.16$ J per queued task.

Both costs are deducted after every attempted execution. The scheduler observes residual sensor battery B_i and residual gateway energy G_g . Equation (4) combines predicted latency, normalized energy burdens, and the conservative trust penalty.

$$\begin{aligned} \hat{L}_{i,g} &= \frac{22 + 0.032d_{i,g} + 6Q_g}{120} \\ \tilde{E}_{i,g}^s &= \frac{E_{i,g}^{\text{tx}}/B_i}{\text{median}_h(E_{i,h}^{\text{tx}}/B_i)} \\ \tilde{E}_g^g &= \frac{E_g^{\text{gw}}/G_g}{\text{median}_h(E_h^{\text{gw}}/G_h)} \\ E_{i,g} &= 0.55\tilde{E}_{i,g}^s + 0.45\tilde{E}_g^g \\ J(i,g) &= 0.34\hat{L}_{i,g} + 0.26E_{i,g} + 0.40(1 - q_g) \end{aligned} \quad (4)$$

Full JTES uses weights 0.34 and 0.26 split across the two energy ratios, and 0.40 for trust after within-task normalization.

The Energy-only baseline minimizes actual residual-energy burdens plus a small queue term. It no longer substitutes distance for residual energy. **Table 3** defines every ablation.

6. Experimental Design and Statistics

Table 4 describes the executed design. Task identities, gateway availability, result corruption draws, work sizes, and validator draws are generated once for a seed-compromise scenario and reused across policies.

Table 3. Schedulers and ablations.

Scheduler	Trust statistic	Energy state	Replication
Nearest active	None	None	None
Energy only	None	Sensor and gateway	None
Trust only	5th percentile	None	Selective
Full JTES	5th percentile	Sensor and gateway	Selective
No selective validation	5th percentile	Sensor and gateway	Challenges only
Posterior mean	Mean	Sensor and gateway	Selective
Always replicate	5th percentile	Sensor and gateway	Every task

Table 4. Executed design.

Factor	Values
Seeds	24, numbered 3001 - 3024
Tasks	8000 per simulated day
Compromise probabilities	0%, 5%, 10%, 15%, 20%
Schedulers	7
Main records	840
Validator sensitivity	72 records
Primary outcomes	Clean timely completion; accepted bad results; energy
Task source	Uniform random sensor among 400
Critical-task probability	22%
Known-answer challenge probability	3%
Workload	U(25, 65) million cycles
Deadline	150 ms with 25% probability; otherwise 220 ms
Gateway availability	Independent per arrival; active probabilities 0.92, 0.88, 0.91, 0.86, 0.90, 0.87
Compromised-result probability	68% per execution on a compromised gateway
Selective replication	Critical task and primary lower trust bound $q_g < 0.90$
Beta refresh	After 50 new validations per gateway

Outcome Metric Definitions

Clean timely completion is defined over all 8000 task arrivals:

$$R_{\text{clean,timely}} = 100 \frac{N_{\text{clean,timely}}}{8000}.$$

A task contributes to $N_{\text{clean,timely}}$ only when at least one execution succeeds, the final accepted result is clean, and the modeled task latency does not exceed its assigned deadline. Tasks with no active gateway, no successful execution, an accepted

corrupted result, or a missed deadline remain in the denominator and therefore count as non-clean-timely.

Accepted bad results are defined only among completed tasks:

$$R_{\text{bad}} = 100 \frac{N_{\text{accepted bad}}}{N_{\text{completed}}},$$

where a completed task is one for which at least one gateway execution succeeds. Tasks with no successful execution are excluded from the accepted-bad denominator because no result is accepted. Validation failures or missing replica results do not create a separate denominator; they affect reputation evidence and final-result resolution as described in Section 4.3.

Paired 95% confidence intervals use seed-level differences. Paired comparisons use two-sided Student- t confidence intervals over the 24 seed-level paired differences, with $df = 23$ and $t_{0.975,23} = 2.069$. Policy-level intervals likewise use the Student- t distribution across the 24 seed outcomes. Lifetime is computed by Equation (5) from one-day drain, then summarized using the tenth percentile across devices within each trial.

$$T_i^{\text{sensor}} = \frac{B_{i(0)}}{B_{i(0)} - B_{i(1 \text{ day})}}; T_g^{\text{gw}} = \frac{G_{g(0)}}{G_{g(0)} - G_{g(1 \text{ day})}}. \quad (5)$$

The projection assumes that the simulated daily workload repeats unchanged. It is dimensionally in days but is not a warranty or battery-aging model.

7. Results

7.1. Principal Integrity and Service Results

Figure 4 compares all schedulers at 15% compromise. Full JTES achieved 94.80% clean timely completion (SD 3.21) and accepted 2.56% bad results (SD 3.17). Energy-only achieved 86.81% and accepted 11.32%.

The paired full-minus-energy difference in clean timely completion was 7.99 points (95% paired t -CI 4.86 to 11.13); the accepted-bad difference was -8.76 points (95% paired t -CI -11.98 to -5.54).

Figure 5 extends accepted corruption across the compromise sweep. The challenge-only ablation remains close to untrusted scheduling because sparse challenges cannot steer assignments quickly enough.

7.2. Conservative Bound and Posterior Mean

The Posterior Mean policy outperformed the 5th-percentile Full JTES policy on all three principal comparison metrics. Clean timely completion was 97.01% versus 94.80%, a full-minus-mean paired difference of -2.21 points (95% paired t -CI -3.01 to -1.41). Accepted bad results were 0.95% versus 2.56%, a difference of 1.61 points (95% paired t -CI 0.82 to 2.40). Sensor energy was 138.98 versus 199.34 mJ per completed task, a difference of 60.37 mJ (95% paired t -CI 48.76 to 71.97).

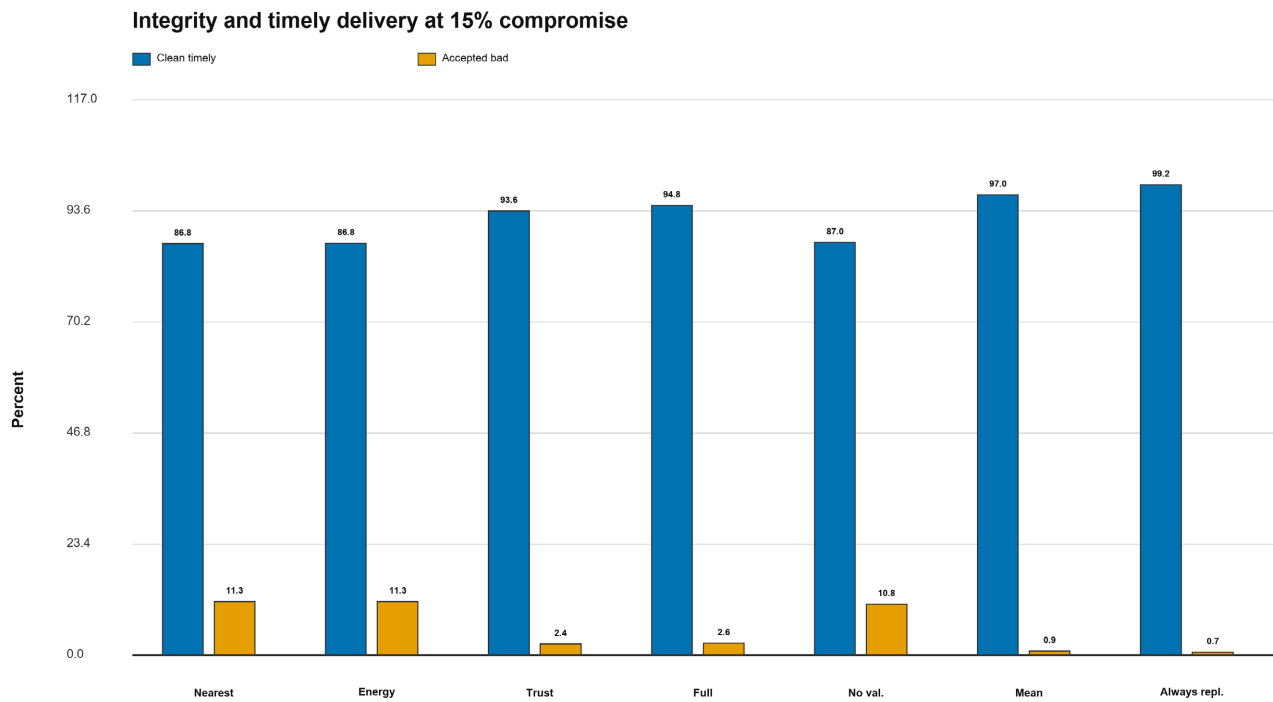


Figure 4. Clean timely completion and accepted bad results at 15% compromise with moderate validation.

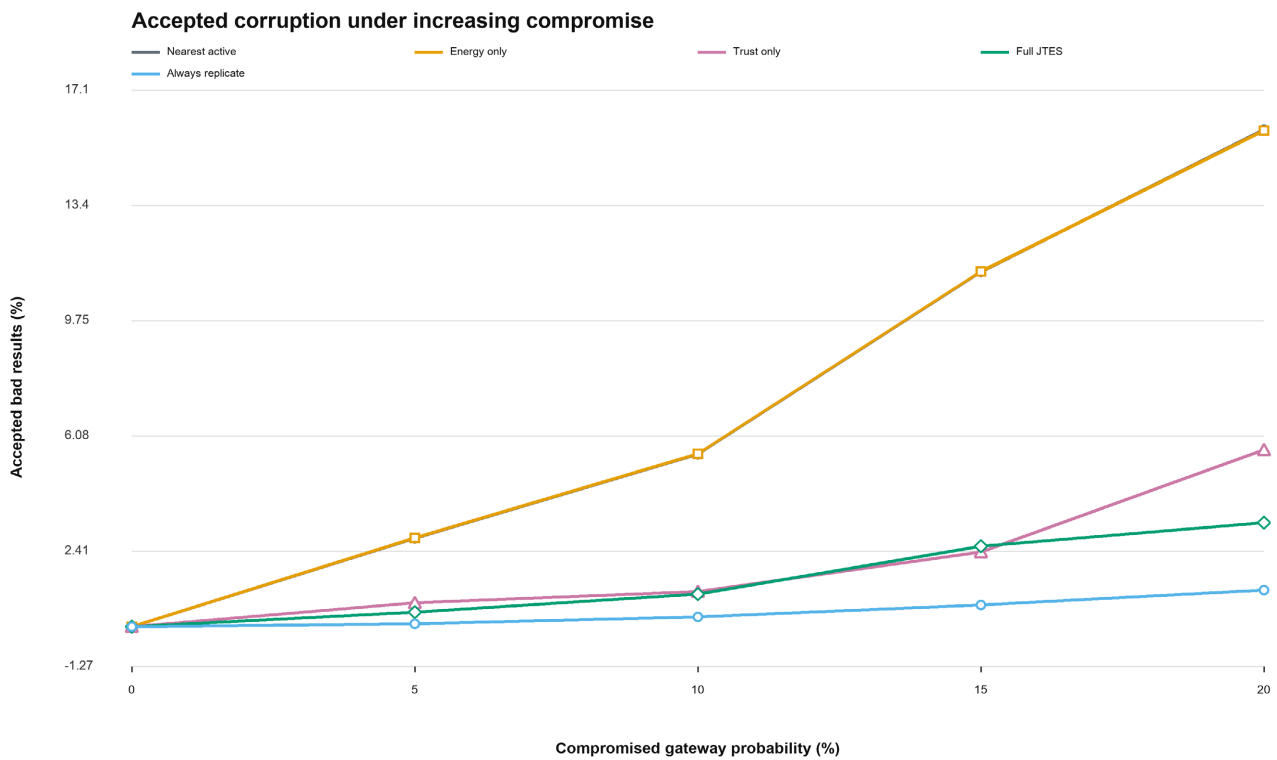


Figure 5. Accepted bad results across all compromise probabilities.

The conservative bound lowers scores for gateways with limited evidence and therefore encodes greater caution, but that caution did not provide a principal-

scenario protection advantage after 8000 task arrivals. In this feedback-coupled scheduler, it increased exploration and validation cost and sometimes delayed concentration on reliable gateways. The 5th-percentile bound is therefore not universally superior; under sufficient accumulated evidence, it can sacrifice service, energy efficiency, and even observed integrity relative to posterior-mean ranking.

7.3. Energy and Lifetime

Figure 6 reports both energy domains. Full JTES used 199.34 mJ of sensor energy and 23.38 J of gateway energy per completed task. The paired sensor-energy difference from Energy-only was 111.03 mJ (95% paired t -CI 103.65 to 118.42).

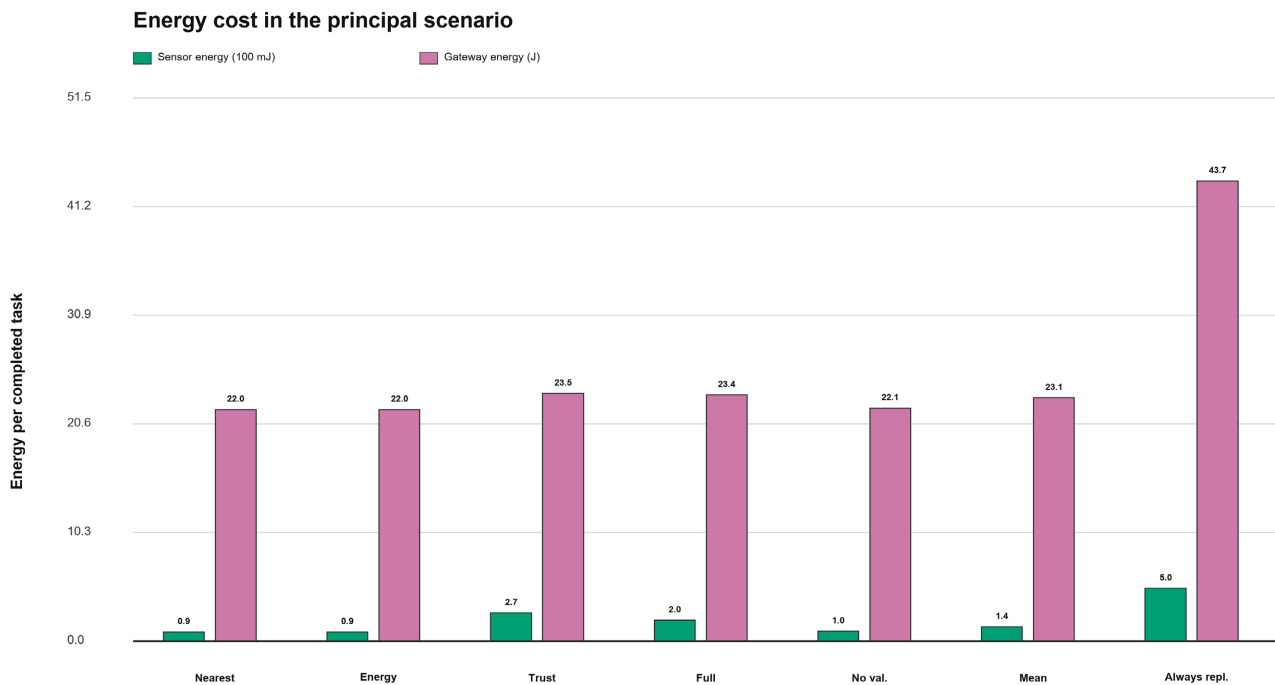


Figure 6. Sensor and gateway energy per completed task in the principal scenario. Sensor bars are scaled in units of 100 mJ.

Figure 7 applies the explicit one-day basis. Full JTES projected a sensor tenth-percentile lifetime of 2535.6 days and a gateway tenth-percentile lifetime of 29.5 days. These values assume identical daily repetition.

7.4. Validator Sensitivity and Ablation

Figure 8 shows validator sensitivity. Accepted bad results changed from 2.26% under strong validation to 3.64% under weak validation. The corresponding clean timely results were 95.00% and 93.96%.

Figure 9 distinguishes attempted replication from realized validation evidence. The replication rate records tasks for which a second execution was dispatched, whereas the validation rate records tasks that produced observable validation evidence. Under Always Replicate, a second execution was dispatched for essentially every eligible task (99.99% on average), but only 93.32% of arrivals produced a

validation event because some gateway executions failed and therefore did not yield two successful outcomes. Known-answer challenges can also create validation evidence when they produce a successful result. Thus, attempted replication and realized validation are intentionally different quantities.

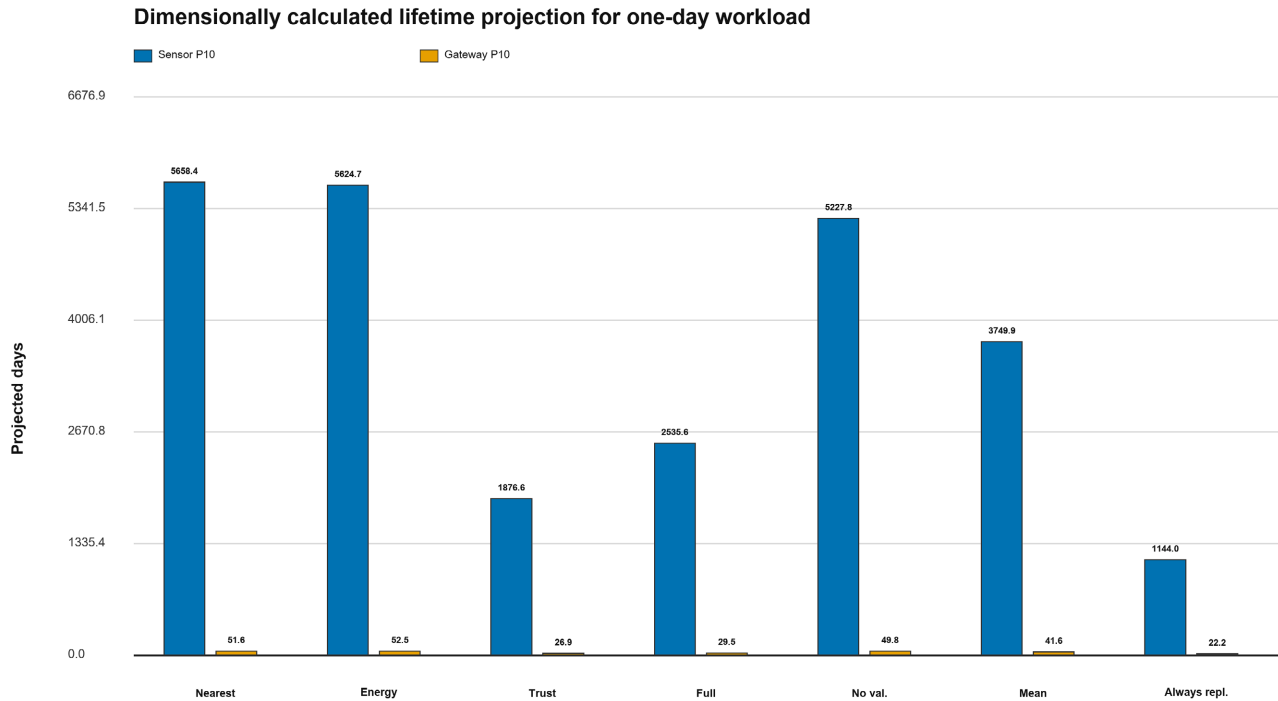


Figure 7. Dimensionally calculated sensor and gateway lifetime projections from one simulated day of drain.

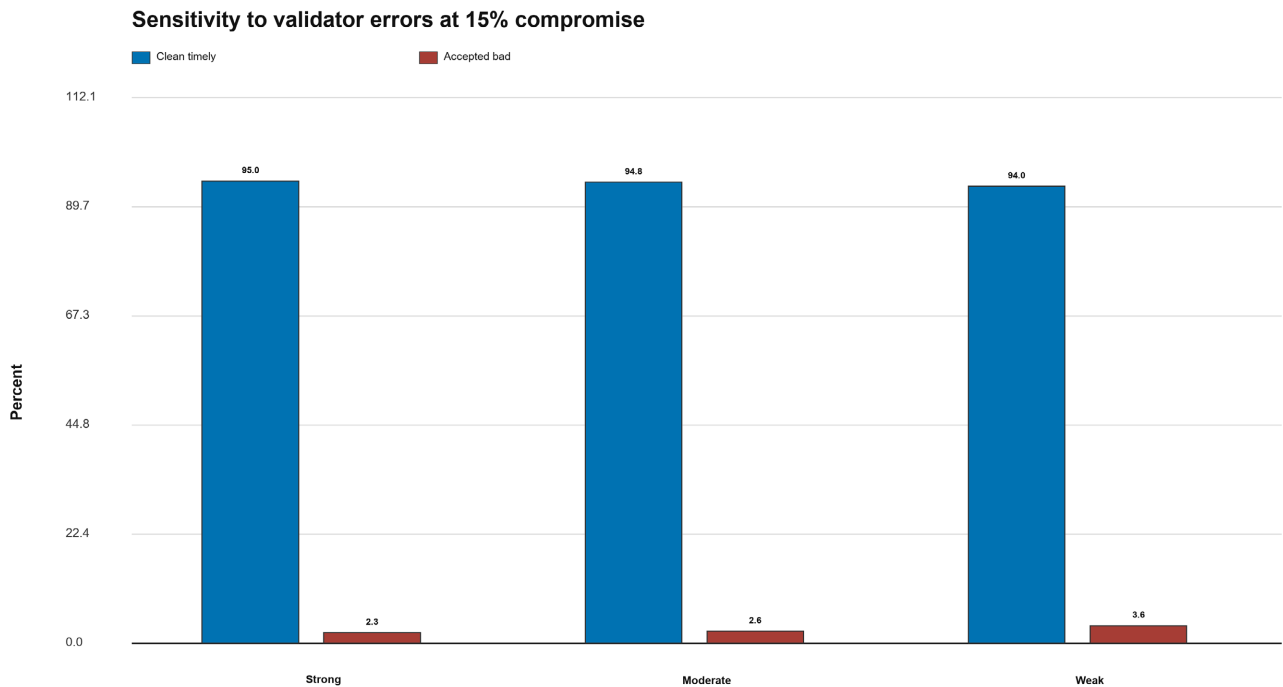


Figure 8. Full-JTES sensitivity to validator true-positive and false-positive error settings.

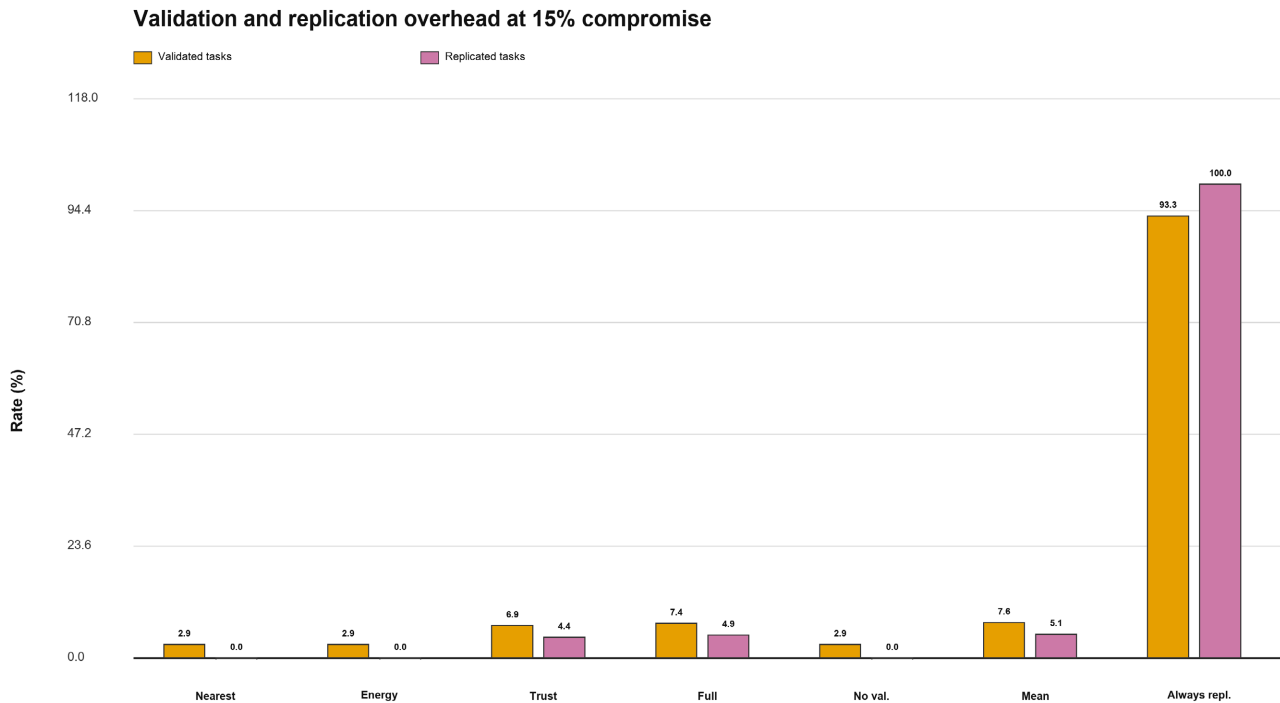


Figure 9. Validation and replication rates for all schedulers at 15% compromise.

Table 5 consolidates the principal results and exposes the trade-off among integrity, service, energy, and projected lifetime.

Table 5. Principal scenario results.

Policy	Clean timely mean $\pm$ SD (%)	Accepted bad (%)	Sensor mJ/task	Sensor P10 days
Nearest active	86.79 $\pm$ 10.23	11.31 $\pm$ 10.50	87.64 $\pm$ 2.73	5658.4
Energy only	86.81 $\pm$ 10.20	11.32 $\pm$ 10.46	88.31 $\pm$ 2.73	5624.7
Trust only	93.62 $\pm$ 4.12	2.38 $\pm$ 3.16	269.08 $\pm$ 62.87	1876.6
Full JTES	94.80 $\pm$ 3.21	2.56 $\pm$ 3.17	199.34 $\pm$ 17.88	2535.6
No selective validation	87.03 $\pm$ 9.76	10.80 $\pm$ 10.04	96.58 $\pm$ 8.91	5227.8
Posterior mean	97.01 $\pm$ 1.38	0.95 $\pm$ 1.38	138.98 $\pm$ 35.57	3749.9
Always replicate	99.16 $\pm$ 1.00	0.69 $\pm$ 0.98	504.60 $\pm$ 63.60	1144.0

8. Discussion

The experiment supports an integrity-energy trade-off. Full JTES reduces accepted corruption relative to energy-only scheduling, but it uses more sensor energy because trust-aware placement often chooses a farther gateway and selective validation adds transmissions.

Posterior-mean scheduling outperforms the conservative-bound variant on the reported principal service, integrity, and sensor-energy metrics. The lower bound discounts gateways with sparse evidence and can induce additional exploration and replication. Conservative ranking expresses epistemic caution, but the result

shows that caution can sacrifice efficiency after sufficient reputation evidence accumulates and does not guarantee lower accepted corruption in a feedback-coupled scheduler.

Always replication performs best on clean timely completion and accepted corruption but approximately doubles transmissions. Selective validation offers an intermediate operating point. Validator sensitivity confirms that trust quality is limited by the evidence mechanism, not only the Bayesian update.

Robustness was examined along two independent axes already included in the experimental design. First, gateway compromise probability was varied from 0% to 20%, allowing the integrity trend to be evaluated as adversarial exposure increased. Second, validator quality was varied from strong to weak error settings. The resulting degradation in accepted corruption and clean-timely service shows that the qualitative trade-off is not confined to one validator setting. However, these sweeps remain bounded by the fixed workload, static daily compromise status, and simplified execution model, so they should be interpreted as simulation robustness rather than deployment robustness.

From a deployment perspective, the results suggest that a conservative credible bound should not be selected automatically merely because it is statistically cautious. Posterior-mean ranking performed better after substantial evidence had accumulated, whereas conservative ranking imposed additional exploration and validation cost. A practical controller could therefore use more conservative trust treatment during cold start or high-consequence operation and transition toward posterior-mean ranking after sufficient validated evidence is available. The replication threshold, 50-observation trust-bound refresh interval, and validation intensity should be treated as operational control parameters whose values depend on available energy, gateway reliability, and application criticality.

9. Threats to Validity and Limitations

The validator is represented by fixed true-positive and false-positive probabilities, gateway compromise is static within a day, and corrupted results are binary. Colluding gateways, targeted challenges, concept drift, and adversarial validators are not modeled.

The radio and computational energy equations are simplified. Battery self-discharge, voltage effects, recharge, gateway cooling, and idle energy outside assigned tasks are excluded. Lifetime assumes daily repetition and should not be interpreted as a deployment forecast.

The exact Beta lower bound is refreshed after 50 observed validations per gateway to limit controller cost. A shorter interval would react faster and require more computation. This interval should be evaluated on hardware.

10. Conclusion

JTES updates trust only from imperfect observable validation and includes sensor and gateway residual energy in the scheduling score. At 15% compromise, the 5th-

percentile Full JTES policy achieved 94.80% clean timely completion and accepted 2.56% bad results while consuming 199.34 mJ of sensor energy per completed task. Posterior-mean ranking performed better on these principal metrics, showing that conservative credible bounds can sacrifice efficiency after substantial evidence accumulates and are not universally superior. Validator degradation and the ablations show that evidence quality, replication, and trust-statistic choice jointly determine the operating point. These conclusions are limited to the simulated workload and validation model.

Data and Code Availability

The study uses no proprietary or personal data. The complete simulation source, fixed random-seed ranges, seed-level result files, aggregated tables, and publication figures are included in the accompanying reproducibility package. The code regenerates all numerical results and figures reported in this manuscript.

Funding

This research received no external funding.

Author Contributions

Utham Kumar Anugula Sethupathy: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing of the original draft, review and editing, and visualization. Vijayanand Ananthanarayanan: Validation, formal analysis, and writing—review and editing.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Bormann, C., Ersue, M. and Keranen, A. (2014) RFC 7228: Terminology for Constrained-Node Networks. <https://doi.org/10.17487/RFC7228>
- [2] Fagan, M., Megas, K.N., Scarfone, K. and Smith, M. (2020) NISTIR 8259A: IoT Device Cybersecurity Capability Core Baseline. <https://doi.org/10.6028/NIST.IR.8259A>
- [3] ETSI (2020) Cyber Security for Consumer Internet of Things: Baseline Requirements. ETSI EN 303 645 V2.1.1. https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf
- [4] Satyanarayanan, M. (2017) The Emergence of Edge Computing. *Computer*, **50**, 30-39. <https://doi.org/10.1109/mc.2017.9>
- [5] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L. (2016) Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, **3**, 637-646. <https://doi.org/10.1109/jiot.2016.2579198>
- [6] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S. (2012) Fog Computing and Its Role in the Internet of Things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, Helsinki, 17 August 2012, 13-16.

- <https://doi.org/10.1145/2342509.2342513>
- [7] Ismail, R. and Josang, A. (2002) The Beta Reputation System. *BLED 2002 Proceedings*, Bled, 17-19 June 2002, Article 41. <https://aisel.aisnet.org/bled2002/41/>
 - [8] Kamvar, S.D., Schlosser, M.T. and Garcia-Molina, H. (2003) The EigenTrust Algorithm for Reputation Management in P2P Networks. *Proceedings of the Twelfth International Conference on World Wide Web*, Budapest, 20-24 May 2003, 640-651. <https://doi.org/10.1145/775152.775242>
 - [9] Ganeriwal, S., Balzano, L.K. and Srivastava, M.B. (2008) Reputation-Based Framework for High Integrity Sensor Networks. *ACM Transactions on Sensor Networks*, **4**, 1-37. <https://doi.org/10.1145/1362542.1362546>
 - [10] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, 7 January 2000, 10. <https://doi.org/10.1109/hicss.2000.926982>
 - [11] Yick, J., Mukherjee, B. and Ghosal, D. (2008) Wireless Sensor Network Survey. *Computer Networks*, **52**, 2292-2330. <https://doi.org/10.1016/j.comnet.2008.04.002>
 - [12] Shelby, Z., Hartke, K. and Bormann, C. (2014) RFC 7252: The Constrained Application Protocol. <https://doi.org/10.17487/RFC7252>
 - [13] Selander, G., Mattsson, J.P., Palombini, F. and Seitz, L. (2019) RFC 8613: Object Security for Constrained RESTful Environments. <https://doi.org/10.17487/RFC8613>
 - [14] Liu, A. and Ning, P. (2008) TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. 2008 *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, St. Louis, 22-24 April 2008, 245-249. <https://doi.org/10.1109/ipsn.2008.47>
 - [15] Sicari, S., Rizzardi, A., Grieco, L.A. and Coen-Porisini, A. (2015) Security, Privacy and Trust in Internet of Things: The Road Ahead. *Computer Networks*, **76**, 146-164. <https://doi.org/10.1016/j.comnet.2014.11.008>
 - [16] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, **17**, 2347-2376. <https://doi.org/10.1109/comst.2015.2444095>
 - [17] Paxson, V. and Floyd, S. (1997) Why We Don't Know How to Simulate the Internet. *Proceedings of the 29th Conference on Winter Simulation*, Atlanta, 7-10 December 1997, 1037-1044. <https://doi.org/10.1145/268437.268737>
 - [18] Wang, X., Zhang, J., Chen, C., He, J., Ma, Y. and Guan, X. (2024) Trust-AoI-Aware Codesign of Scheduling and Control for Edge-Enabled IIoT Systems. *IEEE Transactions on Industrial Informatics*, **20**, 2833-2842. <https://doi.org/10.1109/tii.2023.3299040>
 - [19] Liu, T., Liu, J., Zhang, T., Wang, J., Yuan, Z., Xu, M., *et al.* (2024) Blockchain and Trusted Hardware-Enabled Data Scheduling for Edge Learning in Wireless IIoT. *IEEE Internet of Things Journal*, **11**, 34229-34242. <https://doi.org/10.1109/jiot.2024.3443642>
 - [20] Baker, T., Al Aghbari, Z., Khedr, A.M., Ahmed, N. and Girija, S. (2024) EDITORS: Energy-Aware Dynamic Task Offloading Using Deep Reinforcement Transfer Learning in SDN-Enabled Edge Nodes. *Internet of Things*, **25**, Article 101118. <https://doi.org/10.1016/j.iot.2024.101118>
 - [21] Alsenani, Y. and Alnori, A.S. (2023) Trust-Aware Scheduling for Edge Computing with Task Dependencies and Unreliable Servers. *IEEE Access*, **11**, 113514-113525.

- <https://doi.org/10.1109/access.2023.3324178>
- [22] Al-Masri, E., Sourì, A., Mohamed, H., Yang, W., Olmsted, J. and Kotevska, O. (2023) Energy-Efficient Cooperative Resource Allocation and Task Scheduling for Internet of Things Environments. *Internet of Things*, **23**, Article 100832.
<https://doi.org/10.1016/j.iot.2023.100832>
- [23] Wang, Q., Wang, Q., Zhao, H., Zhang, H., Zhu, H. and Wang, X. (2024) Device-Specific QoE Enhancement through Joint Communication and Computation Resource Scheduling in Edge-Assisted IoT Systems. *IEEE Internet of Things Journal*, **11**, 13257-13270.
<https://doi.org/10.1109/jiot.2023.3332401>
- [24] Xu, C., Lv, M., Zhang, K., Cao, K., Wang, G., Wei, M., *et al.* (2024) Energy Consumption and Time-Delay Optimization of Dependency-Aware Tasks Offloading for Industry 5.0 Applications. *IEEE Transactions on Consumer Electronics*, **70**, 1590-1600.
<https://doi.org/10.1109/tce.2023.3338620>
- [25] Kaur, A. and Auluck, N. (2023) Real-Time Trust Aware Scheduling in Fog-Cloud Systems. *Concurrency and Computation: Practice and Experience*, **35**, e7680.
<https://doi.org/10.1002/cpe.7680>

Appendix: Reproduction Procedure

Table A1 reports policy-level two-sided 95% Student-*t* confidence intervals across the 24 seeds for accepted corruption and both energy domains.

Table A1. Two-sided 95% confidence intervals across 24 seeds.

Policy	Accepted bad (%)	Sensor energy (mJ/task)	Gateway energy (J/task)
Nearest active	6.87 - 15.74	86.49 - 88.79	21.96 - 22.01
Energy only	6.90 - 15.74	87.15 - 89.46	21.95 - 22.00
Trust only	1.04 - 3.71	242.54 - 295.63	23.31 - 23.71
Full JTES	1.22 - 3.90	191.79 - 206.89	23.20 - 23.55
No selective validation	6.56 - 15.04	92.82 - 100.34	22.08 - 22.16
Posterior mean	0.37 - 1.53	123.96 - 154.00	22.99 - 23.20
Always replicate	0.28 - 1.11	477.74 - 531.46	43.57 - 43.79

Run the supplied Python file using the following command: `python run_experiments.py --study ait`. The script writes all 840 main records, 72 validator-sensitivity records, summaries, figures, confidence intervals, and the experiment manifest; **Table A2** lists the expected checks.

Table A2. Reproduction checks.

Check	Expected value
Seed range	3001 - 3024
Main records	840
Sensitivity records	72
Trust updates without validation	Zero by construction
Workload time basis	8000 arrivals per day
Beta statistic	Numerical 5th percentile