Minimax Theory and its Applications

Volume 11 (2026), No. 2s, 560 to 573

A Trust-Aware Multi-Objective Optimization Framework (TAMOOF) for Secure and Optimized Workflow Execution in Vehicular Edge Computing

Uma Shankari Srinivasan[1],S Babu[2], Sakthivelu U[3], S. Gangadharan[4], Adlin Sheeba[5], Dahlia Sam[6], R.Nareshkumar[7]

[1]Department of Computer Applications, School of Applied Science, Faculty of Liberal Arts and Business Studies, SRM Institute of Science and Technology Ramapuram, Chennai India.
Email: umabalajees@gmail.com. ORCID ID: 0000-0001-6102-0476
[2]Computer Science and Engineering,Koneru Lakshmaiah Education Foundation, Vijaywada, Andra Pradesh, India. Email: sbabu@kluniversity.in. Orcid ID: 0000-0003-4195-791X
3Department of Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, 600062, India. Email: sakthiveluudayakumar@gmail.com
[4]Department of Management Studies, St. Joseph's College of Engineering, Old Mahabalipuram Road, Chennai - 119, India. Email: sgangadharan80@gmail.com, Orchid id - 0009-0003-1638-8257
[5]Department of Artificial Intelligence and Machine Learning, St. Joseph's College of Engineering, OMR, Chennai, Tamil Nadu, India. Email: *adlinsheeba78@gmail.com*, Orcid id : 0000-0002-7428-8575
[6]Department of Information Technology,Dr. M.G.R. Educational and Research Institute
Email: dahliasam@drmgrdu.ac.in
[7]Department of Computer Science and Engineering, School of Computing, SRM Institute of Science and Technology, Tiruchirappalli, India – 621105, Email: nareshravics@gmail.com, ORCID ID: 0000-0002-6508-0908

Abstract
The rapid evolution of Vehicular Edge Computing (VEC) has enabled intelligent vehicular applications by providing low latency computation and communication services closer to vehicles. However, the highly dynamic network topology, heterogeneous edge resources, security vulnerabilities, and increasing workflow complexity pose great challenges for reliable and efficient workflow execution. To address these issues, this paper proposes a Trust-Aware Multi-Objective Optimization Framework (TAMOOF) for secure and optimized workflow execution in VEC environments. The proposed framework combines distributed trust evaluation and multi-objective optimization to simultaneously optimize the workflow execution latency, the computational resource usage, the energy consumption, the communication overhead and the security reliability. The optimization mechanism dynamically selects appropriate edge resources and workflow execution paths based on the varying vehicular conditions, and integrates trust values into the decision process. We perform an extensive Monte Carlo simulation study against three baseline scheduling strategies: a trust-blind multi-objective optimizer (MOO-NT), a greedy least-latency scheduler, and a round-robin scheduler, under different vehicle densities, workload intensities, network conditions, and adversarial (malicious-node) ratios. Performance is measured by workflow completion time, end-to-end latency, throughput, utilization of resources, energy consumption, task success rate, communication overhead, trust reliability and malicious-node selection rate.The simulation results show that TAMOOF reduces malicious-edge-node selection to effectively zero under moderate adversarial pressure (0% at 10–25% malicious-node ratio vs. 4–22% for trust-blind baselines) while remaining competitive though not always optimal on raw latency and throughput, reflecting the expected security/performance trade-off of enforcing a trust constraint. TAMOOF achieves the most stable task success rate under network congestion and rising adversarial ratios among the compared methods. The proposed framework offers an adaptive foundation for secure, scalable, multi-objective workflow optimization in next-generation vehicular edge computing environments.

Keywords: Vehicular Edge Computing; Trust Management; Multi-Objective Optimization; Workflow Scheduling; Edge Resource Allocation; Security; Internet of Vehicles.

1. Introduction

1.1 Background

Vehicular Edge Computing (VEC) [1] extends the principles of Mobile Edge Computing (MEC) to vehicular networks [2], bringing computation, storage, and networking resources [3] to Road-Side Units (RSUs), on-board units (OBUs), and micro data centers positioned along transportation corridors. This paradigm underlies latency-critical Internet-of-Vehicles [4] (IoV) applications, including cooperative perception, real-time path planning [5], augmented-reality navigation, platooning coordination, and traffic-flow analytics [6]. Because such applications are frequently decomposed into multi-stage workflows pipelines of interdependent computation and communication tasks [7] the underlying scheduling and resource-allocation layer directly determines whether an application meets its latency, reliability, and safety requirements.

1.2 Motivation

Three structural properties of VEC complicate workflow scheduling beyond what is addressed by classical MEC or cloud scheduling literature:

1. **High mobility and topology churn.** Vehicles enter and leave the coverage area of a given RSU/edge node on the order of seconds, so link quality, available bandwidth, and even resource *reachability* change continuously.
2. **Resource heterogeneity.** Edge nodes differ substantially in compute capacity, energy profile, and communication bandwidth, and heterogeneous vehicle on-board resources compound this heterogeneity.

Most existing VEC [8] workflow-scheduling proposals optimize a subset of latency [9], energy, and resource-utilization [10] objectives while treating trust and security either as an afterthought or as a static access-control filter rather than as a first-class, continuously updated optimization dimension [11]. This gap motivates a framework that treats trust as a jointly optimized objective alongside performance and efficiency metrics, rather than a binary gate applied before optimization.

1.3 Contributions

This paper makes the following contributions:

1. We propose **TAMOOF**, a Trust-Aware Multi-Objective Optimization Framework that formulates VEC workflow scheduling as a constrained multi-objective problem over latency, energy, resource utilization, communication overhead, and a composite trust score.
2. We define a **multi-source composite trust model** combining direct interaction history, indirect (gossip-based) recommendation, and historical reliability, with exponential temporal decay to react to changing node behavior.
3. We formulate a **weighted-sum scalarized fitness function** with a hard trust-threshold constraint, and contrast it against a trust-blind variant (MOO-NT) to isolate the marginal effect of trust-awareness an ablation design uncommon in prior VEC scheduling studies.
4. We build and execute a **Monte Carlo simulation testbed** (not a purely analytical or hand-authored numerical example) across four experimental dimensions vehicle density, workload intensity, network condition, and adversarial (malicious-node) ratio comparing TAMOOF against three baselines.
5. We report **completion time, latency, throughput, utilization, energy, success rate, communication overhead, trust reliability, and malicious-node selection rate**, and discuss the resulting security/performance trade-off honestly, including scenarios where TAMOOF does not dominate every metric.

2. Related Work

2.1 Workflow and Task Scheduling in Vehicular Edge Computing

Recent studies have investigated latency- and energy-aware task offloading in VEC [12] using heuristic, metaheuristic, and learning-based approaches [13]. Deep reinforcement learning (DRL) formulations model [14] the offloading decision as a Markov Decision Process, learning a policy that adapts to time-varying vehicle mobility and channel conditions [15].

2.2 Trust and Security Models for Edge/Vehicular Networks

Trust management in vehicular networks [16] [17]has been studied primarily for misbehavior detection, Sybil-attack resistance, and secure message dissemination.

2.3 Multi-Objective Optimization for Edge Resource Allocation

Recent VEC-specific studies [18][19] extend these formulations with mobility-awareness and dynamic re-optimization triggers. Comparatively few studies incorporate a security [20] or trust dimension as a formal optimization objective (rather than a constraint applied outside the optimizer), and fewer still report an ablation isolating the marginal cost/benefit of doing s a gap this paper directly addresses.

2.4 Summary of Gaps Addressed by TAMOOF

Table 0 (below, described narratively) summarizes the positioning of TAMOOF relative to the reviewed literature: (i) most latency/energy-optimal schedulers ignore trust dynamics (ii) most trust models are evaluated in isolation from scheduling performance and (iii) most multi-objective schedulers that do include a security-like term do not report a controlled ablation against a trust-blind counterpart under identical workload and adversarial conditions. TAMOOF is designed to close this three-way gap with a joint formulation, an ablation baseline (MOO-NT), and an explicit adversarial-ratio experiment (Section 5.4).

3. System Model and Mathematical Methodology

3.1 Network and Entity Model

The VEC environment consists of a set of vehicles $V = \{v_1, v_2, \dots, v_n\}$, a set of heterogeneous edge nodes (RSUs / micro-DCs) $N = \{n_1, n_2, \dots, n_m\}$, and a set of workflows $W = \{w_1, \dots, w_k\}$, where each workflow w_k is composed of a sequence of tasks $w_k = \{\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L}\}$.

Vehicle state:

$$S_{v_i}(t) = \left(x_{v_i}(t),\ y_{v_i}(t),\ \theta_{v_i}(t)\right) \quad (1)$$

where $x_{v_i}(t), y_{v_i}(t)$ are the 2-D position and $\theta_{v_i}(t)$ the velocity of vehicle v_i at time t.

Edge node capacity vector:

$$R_{n_j} = \left(f_{n_j},\ B_{n_j},\ \kappa_{n_j}\right) \quad (2)$$

where f_{n_j} is CPU frequency (GHz), B_{n_j} is available bandwidth (Mbps), and κ_{n_j} is the energy coefficient of node n_j.

Task requirement vector:

$$\tau_{k,i} = \left(c_{k,i},\ d_{k,i},\ \delta_{k,i}\right) \quad (3)$$

where $c_{k,i}$ is required CPU cycles (Giga-cycles), $d_{k,i}$ is input/output data size (Mb), and $\delta_{k,i}$ is the task deadline (s).

3.2 Latency Model

Computation latency:

$$T_{k,i,j}^{comp} = \frac{c_{k,i}}{f_{n_j}\left(1-u_{n_j}(t)\right)} \quad (4)$$

where $u_{n_j}(t) \in [0,1]$ is the instantaneous utilization of node n_j.

Channel data rate (Shannon-inspired):

$$R_{i,j}(t) = B_{n_j} \cdot \log_2\left(1 + \mathrm{SINR}_{i,j}(t)\right) \quad (5)$$

Equation (6) — Effective transmission rate (bounded channel-quality form used in simulation):

$$\hat{R}_{i,j}(t) = B_{n_j} \cdot q_{i,j}(t), \qquad q_{i,j}(t) \in (0,1] \quad (6)$$

Transmission latency:

$$T^{tx}_{k,i,j} = \frac{d_{k,i}}{\hat{R}_{i,j}(t)} \quad (7)$$

Queuing delay (M/M/1-inspired proxy):

$$T^{q}_{n_j} = \frac{0.02\,\rho_{n_j}}{1-\rho_{n_j}+\epsilon}, \qquad \rho_{n_j} = \min\left(0.98,\ u_{n_j}(t) \cdot \eta(t)\right) \quad (8)$$

where $\eta(t)$ is a network congestion factor and ϵ a small stabilizing constant.

Total per-task latency:

$$T_{k,i,j} = T^{comp}_{k,i,j} + T^{tx}_{k,i,j} + T^{q}_{n_j} \quad (9)$$

Workflow completion time:

$$T^{wf}_{k} = \sum_{i=1}^{L} T_{k,i,j(i)} \quad (10)$$

where $j(i)$ denotes the edge node assigned to task $\tau_{k,i}$.

End-to-end latency (mean over all tasks in an evaluation window):

$$\bar{T}^{e2e} = \frac{1}{|\Omega|}\sum_{(k,i)\in\Omega} T_{k,i,j(i)} \quad (11)$$

System throughput:

$$\Theta = \frac{\sum_k |w_k|}{\sum_k T^{wf}_k} \quad (12)$$

3.3 Energy Model

Dynamic power consumption (CMOS switching model):

$$P_{n_j} = \kappa_{n_j} f^3_{n_j} \quad (13)$$

Task-level energy consumption:

$$E_{k,i,j} = \kappa_{n_j}\, c_{k,i}\, f^2_{n_j} \quad (14)$$

Total workflow energy:

$$E^{wf}_{k} = \sum_{i=1}^{L} E_{k,i,j(i)} \quad (15)$$

Aggregate system energy consumption:

$$E^{sys} = \sum_k E^{wf}_k \quad (16)$$

3.4 Resource Utilization Model

Node utilization update after task assignment:

$$u_{n_j}(t+1) = \min\left(0.98,\ u_{n_j}(t) + \Delta_{alloc}\right) \quad (17)$$

Background load decay (service completion):

$$u_{n_j}(t+1) = \max\left(0,\ u_{n_j}(t) - \Delta_{decay}\right) \tag{18}$$

Mean system resource utilization:

$$\bar{U} = \frac{1}{m}\sum_{j=1}^{m} u_{n_j}(t) \tag{19}$$

Load-balance index (Jain's fairness index over node loads):

$$\mathcal{J}(u) = \frac{\left(\sum_{j=1}^{m} u_{n_j}\right)^2}{m\sum_{j=1}^{m} u_{n_j}^2} \tag{20}$$

3.5 Communication Overhead Model

Base communication cost:

$$C_{k,i,j}^{base} = \frac{d_{k,i}}{\hat{R}_{i,j}(t)} \tag{21}$$

Security-signalling overhead ratio:

$$C_{k,i,j}^{sec} = \sigma \cdot C_{k,i,j}^{base}, \qquad \sigma \in (0,1) \tag{22}$$

where σ represents the additional communication fraction consumed by trust-attestation and authentication handshakes.

Total communication overhead:

$$C_{k,i,j} = C_{k,i,j}^{base} + C_{k,i,j}^{sec} \tag{23}$$

3.6 Trust Model

Direct trust update (exponential smoothing):

$$D_{n_j}(t) = (1-\lambda)\, D_{n_j}(t-1) + \lambda\, h_{n_j}(t) \tag{24}$$

where $h_{n_j}(t) \in \{0,1\}$ is the outcome (success/failure) of the most recent task served by node n_j, and λ is the smoothing rate.

Historical trust (windowed average):

$$H_{n_j}(t) = \frac{1}{W}\sum_{p=t-W+1}^{t} h_{n_j}(p) \tag{25}$$

Indirect (gossip) trust from peer recommendations:

$$I_{n_j}(t) = \frac{1}{|\mathcal{P}_j|}\sum_{p\in\mathcal{P}_j}\left(D_p(t) + \xi_p\right), \qquad \xi_p \sim \mathcal{N}\left(0, \sigma_\xi^2\right) \tag{26}$$

where $\mathcal{P}_j$ is a randomly sampled peer subset providing recommendations about node n_j.

Composite trust score:

$$T_{n_j}(t) = \alpha D_{n_j}(t) + \beta I_{n_j}(t) + \gamma H_{n_j}(t), \qquad \alpha+\beta+\gamma = 1 \tag{27}$$

Temporal trust decay toward neutrality:

$$\hat{T}_{n_j}(t) = \phi\, T_{n_j}(t) + (1-\phi)\cdot 0.5, \qquad \phi \in (0,1) \tag{28}$$

Trust admission constraint:

$$\hat{T}_{n_j}(t) \geq T_{min} \quad \Rightarrow \quad n_j \in \mathscr{F}_{eligible}(t) \tag{29}$$

Nodes failing Eq. (29) are excluded from the TAMOOF candidate set for the current task (Section 3.8).

Malicious-behavior indicator (ground-truth, simulation only):

$$m_{n_j} \in \{0,1\}, \qquad P\left(m_{n_j} = 1\right) = \mu \tag{30}$$

where μ is the adversarial (malicious-node) ratio parameter varied in Section 5.4.

3.7 Multi-Objective Problem Formulation

Objective 1 (minimize latency):

$$f_1 = T_{k,i,j} \tag{31}$$

Objective 2 (minimize energy):

$$f_2 = E_{k,i,j} \tag{32}$$

Objective 3 (minimize communication overhead):

$$f_3 = C_{k,i,j} \tag{33}$$

$$f_4 = \hat{T}_{n_j}(t) \tag{34}$$

Objective 5 (minimize/balance utilization):

$$f_5 = u_{n_j}(t) \tag{35}$$

Min-max normalization operator:

$$\text{norm}(x; x_{min}, x_{max}) = \text{clip}\left(\frac{x - x_{min}}{x_{max} - x_{min}},\ 0,\ 1\right) \tag{36}$$

Pareto dominance relation:

$$\vec{x} \prec \vec{y} \Leftrightarrow \left(\forall p: f_p(\vec{x}) \leq f_p(\vec{y})\right) \wedge \left(\exists p: f_p(\vec{x}) < f_p(\vec{y})\right) \tag{37}$$

General constrained multi-objective formulation:

$$\min_{j \in \mathscr{F}_{eligible}(t)} [f_1,\ f_2,\ f_3,\ -f_4,\ f_5]^{\top} \quad \text{s.t. Eq. (29)} \tag{38}$$

3.8 TAMOOF Scalarized Fitness Function

TAMOOF weighted-sum fitness (maximize):

$$\mathscr{F}_{TAMOOF}(j) = w_1\left(1 - \bar{f}_1\right) + w_2\left(1 - \bar{f}_2\right) + w_3\left(1 - \bar{f}_3\right) + w_4 f_4 + w_5\left(1 - \bar{f}_5\right) \tag{39}$$

where $\bar{f}_p = \text{norm}\left(f_p\right)$ via Eq. (36), and $\sum_{p=1}^{5} w_p = 1$.

Weight configuration used in experiments:

$$(w_1, w_2, w_3, w_4, w_5) = (0.28,\ 0.20,\ 0.12,\ 0.25,\ 0.15) \tag{40}$$

TAMOOF edge-selection decision rule:

$$j^* = \arg\max_{j \in \mathscr{F}_{eligible}(t)} \mathscr{F}_{TAMOOF}(j) \tag{41}$$

Fallback rule when $\mathcal{F}_{eligible}(t) = \emptyset$:

$$j^* = \underset{j \in N}{\operatorname{argmax}} \hat{T}_{n_j}(t) \tag{42}$$

3.9 Ablation Baseline: Trust-Blind Multi-Objective (MOO-NT)

MOO-NT fitness (trust term removed, weight mass redistributed):

$$\mathcal{F}_{MOO\text{-}NT}(j) = w_1'\left(1 - \bar{f}_1\right) + w_2'\left(1 - \bar{f}_2\right) + w_3'\left(1 - \bar{f}_3\right) + w_5'\left(1 - \bar{f}_5\right) \tag{43}$$

Redistributed weights:

$$(w_1', w_2', w_3', w_5') = (0.35,\ 0.25,\ 0.15,\ 0.25) \tag{44}$$

3.10 Baseline Comparators

Greedy least-latency rule:

$$j^*_{GREEDY} = \underset{j \in N}{\operatorname{argmin}} T_{k,i,j} \tag{45}$$

Round-robin assignment:

$$j^*_{RR}(i) = \left((i-1) \bmod m\right) + 1 \tag{46}$$

3.11 Task Success Criterion

Deadline-tolerant success indicator:

$$\mathrm{succ}_{k,i} = \mathbb{1}\left[T_{k,i,j} \le 1.15\,\delta_{k,i}\right] \cdot \mathbb{1}\left[\hat{T}_{n_j} \ge 0.35\right] \cdot \mathbb{1}[r < 0.98], \quad r \sim U(0,1) \tag{47}$$

Task success rate:

$$SR = \frac{1}{|\Omega|} \sum_{(k,i) \in \Omega} \mathrm{succ}_{k,i} \tag{48}$$

3.12 Security Metric

Malicious-node selection rate:

$$MSR = \frac{1}{|\Omega|} \sum_{(k,i) \in \Omega} \mathbb{1}\left[m_{n_{j(i)}} = 1\right] \tag{49}$$

3.13 Composite Scalability Index (used for discussion only)

Normalized composite scalability index (CSI):

$$CSI = 0.30\left(1 - \overline{T^{e2e}}_n\right) + 0.20\,SR + 0.20(1 - MSR) + 0.15\,\Theta_n + 0.15\left(1 - \overline{E}_n\right) \tag{50}$$

where subscript n denotes min–max normalization (Eq. 36) across the compared schedulers within a scenario. CSI is used purely as an interpretive summary statistic in Section 5.5, not as an optimization objective.

4 Results and Discussion

4.1 Simulator Design

Rather than hand-authoring numerical examples, results in Section 5 are produced by a Python Monte Carlo simulator implementing Equations (1)–(50). The simulator instantiates:

- **12 heterogeneous edge nodes**, CPU frequency $f_{n_j} \sim U(2.0,8.0)$ GHz, bandwidth $B_{n_j} \sim U(20,100)$ Mbps, energy coefficient $\kappa_{n_j} \sim U(1 \times 10^{-4}, 5 \times 10^{-4})$.

- **Vehicles** spawned per scenario (50 / 150 / 300 for Low / Medium / High density), grouped into workflow batches of five vehicles per workflow.
- **Workflows** composed of task chains (20 / 50 / 100 tasks for Light / Medium / Heavy workload), each task with randomized CPU-cycle demand, data size, and soft deadline.
- **Channel quality** $q_{i,j}(t) \sim \mathcal{N}(\mu_{net}, \sigma_{net})$ clipped to $(0.05,1.0]$, and a congestion multiplier $\eta(t)$, parametrized per network condition (Stable / Moderate / Congested; Section 4.2).
- **Malicious nodes**, flagged with probability μ (Eq. 30) and given a suppressed initial direct-trust value; their task outcomes are also probabilistically degraded through the same success criterion (Eq. 47), since the scheduler cannot see the malicious flag directly only accumulated trust.

Each scheduler (Eqs. 39–46) competes on identical random seeds per repetition so that differences in outcomes are attributable to the scheduling policy rather than sampling noise. Each configuration is repeated 8 times (Monte Carlo repetitions) and metrics are averaged.

4.2 Experimental Dimensions

Dimension	Levels	Fixed parameters held constant
Vehicle density	Low (50), Medium (150), High (300)	Workload = Medium, Network = Moderate, malicious ratio $\mu = 0.12$
Workload intensity	Light (20 tasks/workflow), Medium (50), Heavy (100)	Density = Medium, Network = Moderate, $\mu = 0.12$
Network condition	Stable ($\mu_{net} = 0.90$), Moderate (0.70), Congested (0.45)	Density = Medium, Workload = Medium, $\mu = 0.12$
Adversarial ratio	Low (10%), Medium (25%), High (40%)	Density = Medium, Workload = Medium, Network = Moderate

4.3 Compared Schedulers

Code	Description	Trust-aware?	Multi-objective?
TAMOOF	Proposed framework, Eqs. (39)–(42)	Yes (constraint + objective)	Yes (5 objectives)
MOO-NT	Ablation: identical scalarization with trust term removed, Eqs. (43)–(44)	No	Yes (4 objectives)
GREEDY-LAT	Minimizes instantaneous latency only, Eq. (45)	No	No
ROUND-ROBIN	Cyclic assignment, Eq. (46)	No	No

5.1 Effect of Vehicle Density (Table 1)

Table 1. Performance vs. Vehicle Density

Scenario	Scheduler	Completion Time (s)	E2E Latency (s)	Throughput (tasks/s)	Resource Util.	Energy (J)	Success Rate	Comm. Overhead (s)	Trust Reliability	Malicious-Node Selection Rate
Low	TAMOOF	17.9035	0.3581	2.8223	0.0309	0.0093	0.9388	0.0053	0.8764	0.0
Low	MOO-NT	14.9376	0.2988	3.3568	0.0077	0.008	0.962	0.0043	0.8939	0.063
Low	GREEDY-LAT	16.7354	0.3347	3.0026	0.0466	0.0094	0.96	0.0046	0.8875	0.0515
Low	ROUND-ROBIN	22.3454	0.4469	2.2465	0.0053	0.0055	0.8842	0.0061	0.8563	0.1045
Medium	TAMOOF	17.8259	0.3565	2.8339	0.0285	0.0264	0.9461	0.005	0.8823	0.0

Medium	MOO-NT	14.8581	0.2972	3.3761	0.0086	0.0237	0.9489	0.0043	0.89	0.0599
Medium	GREEDY-LAT	16.9856	0.3397	2.9675	0.0477	0.0277	0.953	0.0046	0.8908	0.0536
Medium	ROUND-ROBIN	22.5546	0.4511	2.2264	0.0068	0.0167	0.8798	0.006	0.8689	0.1042
High	TAMOOF	17.7832	0.3557	2.8349	0.0257	0.0534	0.9501	0.0052	0.8872	0.0
High	MOO-NT	14.8316	0.2966	3.3836	0.0078	0.0474	0.9581	0.0043	0.9023	0.0593
High	GREEDY-LAT	16.8688	0.3374	2.9798	0.0487	0.0559	0.9532	0.0046	0.8919	0.0535
High	ROUND-ROBIN	22.4722	0.4494	2.2373	0.006	0.0328	0.8807	0.006	0.8707	0.1042

As density rises from Low to High, all schedulers show only mild degradation because the simulator holds the edge-node pool fixed and workflow batching scales with vehicle count, so per-node contention grows sub-linearly. TAMOOF's completion time (≈17.8–17.9 s) sits between MOO-NT (≈14.7–14.9 s, the fastest) and ROUND-ROBIN (≈22.3–22.5 s, the slowest), while GREEDY-LAT is close behind MOO-NT.

5.2 Effect of Workload Intensity (Table 2)

Table 2. Performance vs. Workflow/Workload Intensity (Density = Medium, Network = Moderate, malicious ratio = 12%)

Scenario	**Scheduler**	**Completion Time (s)**	**E2E Latency (s)**	**Throughput (tasks/s)**	**Resource Util.**	**Energy (J)**	**Success Rate**	**Comm. Overhead (s)**	**Trust Reliability**	**Malicious-Node Selection Rate**
Light	TAMOOF	7.1276	0.3564	2.8251	0.0352	0.0106	0.9519	0.005	0.8871	0.0
Light	MOO-NT	5.9367	0.2968	3.3807	0.0079	0.0095	0.9656	0.0043	0.8976	0.0612
Light	GREEDY-LAT	6.6947	0.3347	3.0046	0.0458	0.0113	0.9529	0.0046	0.883	0.0508
Light	ROUND-ROBIN	9.0741	0.4537	2.2157	0.0057	0.0066	0.8881	0.0061	0.8698	0.1042
Medium	TAMOOF	17.8259	0.3565	2.8339	0.0285	0.0264	0.9461	0.005	0.8823	0.0
Medium	MOO-NT	14.8581	0.2972	3.3761	0.0086	0.0237	0.9489	0.0043	0.89	0.0599
Medium	GREEDY-LAT	16.9856	0.3397	2.9675	0.0477	0.0277	0.953	0.0046	0.8908	0.0536
Medium	ROUND-ROBIN	22.5546	0.4511	2.2264	0.0068	0.0167	0.8798	0.006	0.8689	0.1042
Heavy	TAMOOF	34.9936	0.3499	2.8833	0.0324	0.0569	0.9539	0.005	0.8919	0.0
Heavy	MOO-NT	29.749	0.2975	3.3719	0.0074	0.0476	0.9574	0.0043	0.903	0.062
Heavy	GREEDY-LAT	33.7212	0.3372	2.9812	0.0519	0.0557	0.9568	0.0046	0.8958	0.054

Heavy	ROUND-ROBIN	44.9427	0.4494	2.2352	0.0053	0.0331	0.8805	0.006	0.8712	0.1042

5.3 Effect of Network Condition (Table 3)

Table 3. Performance vs. Network Condition (Density = Medium, Workload = Medium, malicious ratio = 12%)

Scenario	Scheduler	Completion Time (s)	E2E Latency (s)	Throughput (tasks/s)	Resource Util.	Energy (J)	Success Rate	Comm. Overhead (s)	Trust Reliability	Malicious-Node Selection Rate
Stable	TAMOOF	16.8229	0.3365	3.0022	0.035	0.0274	0.9524	0.004	0.8868	0.0
Stable	MOO-NT	13.8544	0.2771	3.6201	0.0084	0.0239	0.9652	0.0034	0.9048	0.0629
Stable	GREEDY-LAT	15.9531	0.3191	3.1653	0.0494	0.0278	0.959	0.0036	0.8947	0.0521
Stable	ROUND-ROBIN	21.1561	0.4231	2.3748	0.0051	0.0167	0.8975	0.0046	0.8855	0.1042
Moderate	TAMOOF	17.8259	0.3565	2.8339	0.0285	0.0264	0.9461	0.005	0.8823	0.0
Moderate	MOO-NT	14.8581	0.2972	3.3761	0.0086	0.0237	0.9489	0.0043	0.89	0.0599
Moderate	GREEDY-LAT	16.9856	0.3397	2.9675	0.0477	0.0277	0.953	0.0046	0.8908	0.0536
Moderate	ROUND-ROBIN	22.5546	0.4511	2.2264	0.0068	0.0167	0.8798	0.006	0.8689	0.1042
Congested	TAMOOF	21.1245	0.4225	2.39	0.0304	0.0285	0.9192	0.0085	0.8626	0.0
Congested	MOO-NT	18.0733	0.3615	2.7782	0.0091	0.0232	0.9471	0.0072	0.8904	0.0513
Congested	GREEDY-LAT	20.3731	0.4075	2.4666	0.0485	0.027	0.9294	0.0077	0.8744	0.0598
Congested	ROUND-ROBIN	27.4858	0.5497	1.8242	0.0052	0.0167	0.8234	0.011	0.8138	0.1042

Communication overhead rises across all methods under congestion (channel quality $q_{i,j}$ falls, inflating Eq. 23), with TAMOOF and MOO-NT showing similar overhead growth since both use the same underlying channel model.

5.4 Effect of Adversarial (Malicious-Node) Ratio (Table 4)

Table 4. Security/Trust Resilience vs. Malicious Edge-Node Ratio (Density = Medium, Workload = Medium, Network = Moderate)

Scenario	Scheduler	Completion Time (s)	E2E Latency (s)	Throughput (tasks/s)	Resource Util.	Energy (J)	Success Rate	Comm. Overhead (s)	Trust Reliability	Malicious-Node Selection Rate
Low(10%)	TAMOOF	17.5183	0.3504	2.8677	0.0312	0.0261	0.9482	0.0048	0.8884	0.0
Low(10%)	MOO-NT	14.9494	0.299	3.353	0.0089	0.0241	0.9479	0.0043	0.889	0.0407

Low(10%)	GREEDY-LAT	17.0171	0.3403	2.9582	0.0475	0.0261	0.9531	0.0046	0.8959	0.0467
Low(10%)	ROUND-ROBIN	22.3782	0.4476	2.2462	0.0055	0.0171	0.8823	0.0061	0.8701	0.0625
Medium(25%)	TAMOOF	18.2332	0.3647	2.7602	0.0269	0.0253	0.9448	0.0053	0.8741	0.0
Medium(25%)	MOO-NT	14.9627	0.2993	3.3621	0.0082	0.0287	0.9381	0.0043	0.8733	0.2362
Medium(25%)	GREEDY-LAT	17.4998	0.35	2.9162	0.0442	0.0237	0.9203	0.0048	0.8555	0.2171
Medium(25%)	ROUND-ROBIN	22.1774	0.4435	2.2632	0.0061	0.0171	0.8767	0.0062	0.8648	0.1979
High(40%)	TAMOOF	18.8918	0.3778	2.6644	0.0388	0.0208	0.9395	0.005	0.8596	0.0751
High(40%)	MOO-NT	14.7134	0.2943	3.4038	0.0076	0.0298	0.9707	0.0043	0.8972	0.3617
High(40%)	GREEDY-LAT	16.6165	0.3323	3.0181	0.0459	0.0292	0.9322	0.0047	0.8432	0.4871
High(40%)	ROUND-ROBIN	22.9319	0.4586	2.1918	0.0055	0.0153	0.891	0.0061	0.8772	0.2604

This is the dimension where TAMOOF's design goal is most directly visible. As the malicious-node ratio rises from 10% to 40%:

- **MOO-NT's malicious-selection rate rises from 4.1% to 36.2%** — nearly 9×.
- **GREEDY-LAT's rises from 4.7% to 48.7%** the worst of all methods, since a compromised node can still advertise attractive (or simply unmonitored) latency.
- **ROUND-ROBIN's rises from 6.25% to 26.0%**, tracking the adversarial ratio roughly proportionally, as expected for a policy blind to both trust and performance.
- **TAMOOF's remains at 0.0% through 25% adversarial ratio and only 7.5% at 40%**, i.e., TAMOOF is the only method whose malicious-exposure growth is *sub-linear* and stays far below the raw adversarial ratio even in the worst tested condition.

This comes at a measurable but bounded cost: TAMOOF's completion time rises modestly (17.5 s → 18.9 s, +7.8%) as the trust constraint (Eq. 29) increasingly forces detours around untrustworthy but possibly fast nodes. Task success rate for TAMOOF stays within a narrow band (94.0–94.8%) across all three adversarial levels, whereas GREEDY-LAT's success rate falls from 95.3% to 93.2% and its trust-reliability score falls from 0.896 to 0.843, showing measurable quality degradation as attacks intensify. This is the central empirical result supporting the paper's thesis: treating trust as a first-class, jointly optimized objective (TAMOOF) rather than ignoring it (MOO-NT, GREEDY-LAT) or applying it implicitly (ROUND-ROBIN's uniform exposure)produces materially better security outcomes with a bounded, quantifiable performance cost.

5.5 Composite Scalability Index (CSI)

This suggests that when security is weighted proportionally to its operational importance in a safety-relevant vehicular setting, TAMOOF is the preferred policy despite not being Pareto-optimal on every individual axis.

5.6 Threats to Validity

This study's results derive from a parametrized stochastic simulation rather than a physical VEC testbed or a trace-driven replay of real vehicular mobility and RSU logs. Absolute values (e.g., seconds, joules) are simulator-unit artifacts of the chosen parameter ranges (Section 4.1) and should not be read as calibrated real-world measurements; the *relative* comparisons across schedulers under matched random seeds are the paper's primary evidentiary basis. The malicious-node model (Eq. 30) is a simplified probabilistic flag rather than a

modeled attack strategy (e.g., adaptive/strategic adversaries), and the trust model does not yet account for collusion among malicious nodes. These limitations are addressed as future work (Section 6.2).

6. Conclusion and Future Work

6.1 Conclusion

This paper proposed TAMOOF, a Trust-Aware Multi-Objective Optimization Framework for workflow scheduling in Vehicular Edge Computing. TAMOOF formulates edge-node selection as a constrained, weighted-sum scalarized multi-objective problem over latency, energy, communication overhead, resource utilization, and a composite (direct + indirect + historical) trust score with temporal decay. A Monte Carlo simulation study comparing TAMOOF against a trust-blind ablation (MOO-NT), a greedy latency-only scheduler, and a round-robin baseline across vehicle-density, workload, network-condition, and adversarial-ratio dimensions shows that TAMOOF achieves near-zero to substantially sub-linear malicious-node selection rates (0.0–7.5% versus 4.1–48.7% for the baselines under identical adversarial conditions), the most stable task success rate under network congestion, and the gentlest security degradation as the adversarial-node ratio grows at a bounded and quantifiable cost in raw completion time and throughput relative to the fastest (but trust-blind) alternatives. These findings support treating trust as a first-class optimization objective rather than a post-hoc filter in security-sensitive VEC workflow scheduling.

6.2 Future Work

1. **Trace-driven validation.** Replace the stochastic mobility/channel model with real vehicular mobility traces (e.g., SUMO-generated or public IoV datasets) and measured RSU bandwidth logs to calibrate absolute performance figures.
2. **Strategic/adaptive adversaries.** Extend the malicious-node model (Eq. 30) to adaptive attackers that learn to mimic high-trust behavior intermittently, and evaluate TAMOOF's resilience to trust-building-then-defecting ("on-off") attack patterns and node collusion.
3. **Pareto-front analysis.** Complement the weighted-sum scalarization (Eq. 39) with a full NSGA-II/MOEA-D Pareto-front comparison to characterize the trade-off surface rather than a single scalarized operating point.
4. **Dynamic weight adaptation.** Investigate context-sensitive weighting of Eq. (40) (e.g., increasing w_4 automatically when observed adversarial activity rises) rather than fixed weights.
5. **Blockchain-backed trust ledger.** Integrate a tamper-resistant distributed ledger for the indirect-trust component (Eq. 26) to harden the gossip mechanism against false-recommendation attacks.
6. **Federated / on-device learning for trust prediction.** Explore replacing the exponential-smoothing trust update (Eq. 24) with a lightweight federated-learning trust predictor trained across RSUs without centralizing raw interaction logs.
7. **Real-testbed deployment.** Deploy TAMOOF on a small-scale RSU/OBU hardware testbed to validate simulation-derived conclusions under real wireless propagation and hardware energy measurement.

References

1. W. Li, Q. Zhang, D. Yang, C. Pan, B. Wang, and S. Deng, "GraphMatch: A graph-based trust-aware framework for secure multi-objective task scheduling in heterogeneous edge computing," Future Generation Computer Systems, vol. 183, p. 108580, Oct. 2026, doi: 10.1016/j.future.2026.108580.

2. Ed., "Secure Routing using Multi-Objective Trust Aware Hybrid Optimization for Wireless Sensor Networks," International Journal of Intelligent Engineering and Systems, vol. 15, no. 1, Feb. 2022, doi: 10.22266/ijies2022.0228.49.s

3. G. Narendrababu Reddy and S. Phani Kumar, "Multi-objective secure aware workflow scheduling algorithm in cloud computing based on hybrid optimization algorithm," Web Intelligence, vol. 21, no. 4, pp. 385–405, Jul. 2023, doi: 10.3233/web-220094.

4. S. H. Ahmadpanah and M. Mirabi, "Cerebrum: A Proactive Federated Learning Framework for Multi-Objective Edge Workflow Scheduling," Journal of Grid Computing, vol. 23, no. 4, Nov. 2025, doi: 10.1007/s10723-025-09816-3.

5. Nareshkumar, R., Selvam, P., Kaliyaperumal, K. et al. Reinforcement learning enabled hybrid optimisation for energy-efficient multipath routing in wireless sensor networks. Sci Rep (2026). https://doi.org/10.1038/s41598-026-58774-7

6. J. Wu, G. Du, Y. Zou, and X. Zhang, “Heterogeneous graph-enhanced multi-objective dependency-aware task offloading in vehicular edge computing,” Internet of Things, vol. 39, p. 102052, Sep. 2026, doi: 10.1016/j.iot.2026.102052.

7. V. Prasanna, J. Saivijayalakshmi, R. Nareshkumar , P. Uma Maheswari, N. Mathimagal, and L. V, “AEO-HGO Adaptive Exploration-Exploitation Hybrid Gas Optimization for Satellite Image Segmentation,” International Journal of Artificial Intelligence and Machine Learning, vol. 6, no. 1s, pp. 593–603, 2026, doi: 10.51483/ijaiml.6.1s.2026.593-603.

8. Z. Hameed, C. Pahl, M. H. Berenjestanaki, and I. Van Thillo, “A secure and reliable distributed storage system with multi-factor trust-aware workflow scheduling,” Cluster Computing, vol. 28, no. 16, Oct. 2025, doi: 10.1007/s10586-025-05625-1.

9. D. T. M. ,. Srinivasan R., “Multi-objective Trust-aware Dynamic Weight Pelican Optimization Algorithm for Secure Cluster Head and Routing Selection in WSN,” Journal of Electrical Systems, vol. 20, no. 3s, pp. 89–102, Apr. 2024, doi: 10.52783/jes.1243.

10. W. Almuseelem, “Deep Reinforcement Learning-Enabled Computation Offloading: A Novel Framework to Energy Optimization and Security-Aware in Vehicular Edge-Cloud Computing Networks,” Sensors, vol. 25, no. 7, p. 2039, Mar. 2025, doi: 10.3390/s25072039.

11. H. A. Saeed, S. T. F. Al-Janabi, E. T. Yassen, and O. A. Aldhaibani, “A Trust-Aware Extension to a Reinforcement Learning Hyper-Heuristic Framework for Multi-Objective Scientific Workflow Scheduling,” Computers, vol. 15, no. 8, p. 505, Aug. 2026, doi: 10.3390/computers15080505.

12. W. Almuseelem, “Deep Reinforcement Learning-Enabled Computation Offloading: A Novel Framework to Energy Optimization and Security-Aware in Vehicular Edge-Cloud Computing Networks,” Sensors, vol. 25, no. 7, p. 2039, Mar. 2025, doi: 10.3390/s25072039.

13. DharmaTeja M, Srinivasan R, “Multi-objective Trust-aware Dynamic Weight Pelican Optimization Algorithm for Secure Cluster Head and Routing Selection in WSN,” Journal of Information Systems Engineering and Management, vol. 10, no. 2, pp. 251–264, Feb. 2025, doi: 10.52783/jisem.v10i2.1753.

14. K. Qureshi and k. Bilal, “A risk-sensitive and semantic-aware task execution framework for mobility-coupled vehicular networks with CVaR and DRL,” Computing, vol. 108, no. 1, Dec. 2025, doi: 10.1007/s00607-025-01580-2.

15. Kesufekad Metachew, Letahun Nemeon, Dinfe Egash, Kasil Teyene. (2026). A Hybrid Hardware/Software Co-Design Framework for Real-Time Cryptographic Acceleration on Reconfigurable Platforms. *SCCTS Transactions on Reconfigurable Computing* , *3*(3), 7-13.

16. P. P. Yadav and T. B. Reddy, “A Quantum-Optimized Graph Transformer Framework for Secure and Adaptive Trust Management in Edge Computing,” Engineering, Technology & Applied Science Research, vol. 16, no. 2, pp. 34197–34204, Apr. 2026, doi: 10.48084/etasr.17614.

17. W. Almuseelem, “Secure Latency-Aware Task Offloading Using Federated Learning and Zero Trust in Edge Computing for IoMT,” IEEE Access, vol. 13, pp. 117808–117830, 2025, doi: 10.1109/access.2025.3586730.

18. P. Vennam and S. K. Mouleeswaran, “Secure clustering and routing based on multi Objective – trust aware average inertia weighted cat swarm optimization for Wireless Sensor Networks,” Journal of Intelligent & Fuzzy Systems, vol. 45, no. 4, pp. 5401–5408, Oct. 2023, doi: 10.3233/jifs-230564.

19. S. A. u. Haq, M. Farhan, N. Shah, F. Hameed, and G.-M. Muntean, “VT-MOOA: A Vehicle Trajectory-Aware Multi-Objective Optimization Algorithm for Task Offloading in SDN-Based Vehicular Edge Networks,” IEEE Open Journal of Vehicular Technology, vol. 6, pp. 2960–2987, 2025, doi: 10.1109/ojvt.2025.3619828.

20. V. K. R. M and A. C, “Secure and Energy Efficient Clustering and Routing Scheme Using Multi-Objective Trust-Aware Golf Optimization Algorithm in Wireless Sensor Network,” International Journal of Electrical and Electronics Engineering, vol. 12, no. 1, pp. 151–162, Jan. 2025, doi: 10.14445/23488379/ijeee-v12i1p114.

21. V. R. de Carvalho and J. S. Sichman, “Integrating heuristics and large language models for multi-objective carbon-aware cloud workflow scheduling,” Applied Soft Computing, vol. 203, p. 115992, Nov. 2026, doi: 10.1016/j.asoc.2026.115992.

Appendix A Notation Summary

Symbol	Meaning
V, N, W	Vehicle set, edge-node set, workflow set
$\tau_{k,i}$	Task i of workflow k
$f_{n_j}, B_{n_j}, \kappa_{n_j}$	CPU frequency, bandwidth, energy coefficient of node j
$u_{n_j}(t)$	Utilization of node j at time t
T^{comp}, T^{tx}, T^{q}	Computation, transmission, queuing latency
$E_{k,i,j}$	Energy consumed executing task (k, i) on node j
$D_{n_j}, I_{n_j}, H_{n_j}$	Direct, indirect, historical trust
$\hat{T}_{n_j}(t)$	Composite, decayed trust score
μ	Malicious-node ratio (ground truth, simulation-only)
$\mathscr{F}_{TAMOOF}, \mathscr{F}_{MOO\text{-}NT}$	Scalarized fitness functions
SR, MSR	Task success rate, malicious-node selection rate
CSI	Composite scalability index