



City Research Online

City St George's, University of London

Citation: Fayi, S., Ayaz, F. & Sheng, Z. (2025). A Blockchain-Based Reputation-Enhanced Vehicle Selection (REVS) for Computation Offloading. Paper presented at the 2025 IEEE 102nd Vehicular Technology Conference: VTC2025-Fall, 19-22 Oct 2025, Chengdu, China.

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/36128/>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

A Blockchain-Based Reputation-Enhanced Vehicle Selection (REVS) for Computation Offloading

Sharifah Fayi^{*†}, Ferheen Ayaz[‡], Zhengguo Sheng^{*}

^{*}School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom

[†]College of Computer Science, King Khalid University, Abha, Saudi Arabia

[‡]Department of Computer Science, City St. George's, University of London, London, United Kingdom

Email: s.fayi@sussex.ac.uk, ferheen.ayaz@city.ac.uk, z.sheng@sussex.ac.uk

Abstract—Secure and trustworthy computation offloading is essential in vehicular edge computing to ensure reliability and efficiency. Existing algorithms often emphasize efficiency over security, leaving systems exposed to malicious providers. This paper presents the Reputation-Enhanced Vehicle Selection (REVS) framework, which combines social trust-based initialization, direction alignment, and a weighted trust score based on provider reputation and stay time. To enhance provider selection reliability, REVS employs a lightweight consortium blockchain for decentralized and distributed reputation management, with a smart contract deployed at the edge RSU to automate the selection process. Simulations show that REVS improves task success rates by up to 40.85%, avoids 40.70% more malicious providers, and reduces latency by 20%, outperforming fixed-reputation and random selection methods that ignore trust.

Index Terms—Vehicular networks, computation offloading, provider selection, blockchain, social trust, reputation, security

I. INTRODUCTION

In vehicular networks, offloading computational tasks to edge servers or nearby vehicles is essential for low-latency, energy-efficient, and reliable applications by alleviating network congestion [1]–[4]. However, malicious providers can disrupt performance and complicate the selection of reliable offloading providers [5]–[7]. Many existing algorithms prioritize efficiency over security [3], while current centralized architectures introduce vulnerabilities such as single points of failure and limited traceability [8]. Centralized nodes become impractical for managing large-scale vehicle networks as intelligent transportation systems rapidly evolve, often causing latency, blocking, and reduced quality of service (QoS) [9].

Although cryptographic methods protect data confidentiality, they are insufficient for assessing the trustworthiness of authorized nodes [5]. Trust management remains under-prioritized in vehicular computation offloading (VCOff) [3] despite its importance in autonomous and cooperative systems [10]. In VCOff, trust represents the road side unit (RSU)'s current expectation of a provider's behavior, whereas reputation represents its cumulative reliability. Trust is updated after each task, acts as feedback, and contributes to reputation scores. Reputation in VCOff can be effectively managed using blockchain, which mitigates the risks of centralized systems and ensures tamper-resistant records, significantly enhancing the reliability and security of provider selection [8], [11], [12].

Blockchain has shown strong potential for enhancing trust and security of VCOff [13]. Blockchain offers a decentralized, tamper-resistant, and transparent infrastructure [9]. Recent work demonstrates blockchain's effectiveness in enabling secure offloading through distributed access control, automated trust verification, and smart contract-based behavior tracking [14]. However, gaps remain in securing provider selection. For example, Wang et al. [6] focus on incentivizing resource sharing but do not evaluate provider trustworthiness. [15] improves offloading decisions but overlooks secure provider selection while [16] introduces a social-score-based system, yet lacks concrete selection criteria definitions. These methods overlook critical factors like expected connection duration, which is essential for effective offloading in high-mobility vehicular networks. Table I summarizes key VCOff challenges and how blockchain addresses them through decentralization, transparency, and secure trust management.

To address these gaps, this paper proposes the Reputation-Enhanced Vehicle Selection (REVS) algorithm—a lightweight consortium blockchain-based framework for secure, efficient provider selection using trust scoring and mobility awareness. Blockchain in REVS replaces a single point of failure with distributed accountability and automated trust enforcement via smart contracts. Inspired by the Three-Valued Subjective Logic (3VSL) framework [5], which extends Jøsang's model [17] by integrating subjective and objective opinions and social trust (ST) levels of vehicle owners or drivers. Subjective logic offers a promising approach to trust management by evaluating reputation through “opinions” formed from individual experiences (subjective) and aggregated neighbor feedback using metrics (objective) [5], [18]. REVS incorporates: (i) ST-based reputation initialization from a trusted authority (TA), (ii) a weighted trust score combining reputation and connection duration, (iii) and smart contracts at RSUs equipped with Vehicular Edge Computing (VEC) for automated selection and reputation updates. Unlike fixed-reputation models [19], REVS initializes reputation and adapts to provider behavior using social trust inputs. It also improves fairness and scalability over fully vehicle-controlled models [18], [19] by delegates the decision process to edge smart contracts, enhancing fairness and scalability [20], which lays the foundation for more advanced trust and reputation modeling in future work.

Our main contributions are as follows:

- 1) We propose REVS, a trust-based provider selection algorithm automated by the smart contract.
- 2) We leverage ST levels–driven reputation initialization and consortium blockchain for reputation management.
- 3) We evaluate REVS under varying malicious node ratios and weight configurations, achieving up to a 40.85% improvement in success rates and a 20% reduction in latency over baseline approaches.

TABLE I: Blockchain Solutions to VCOff Challenges

VCOff Challenge	Blockchain-Based Solution
Centralized reputation management	Distributed ledger maintained by RSUs ensures decentralization
Single point of failure	Consortium blockchain minimizes dependence on central entities
Biased or unreliable provider evaluation	Immutable logs and SC-based provider selection ensure fair validation
Lack of transparency and traceability	Transparent, verifiable transactions allow traceable reputation updates
Unreliable behavior tracking	Tamper-proof and persistent provider history supports accountability

The rest of this paper is organized as follows: Section II introduces the system model. Section III explains the proposed blockchain-based solution REVS. Section IV presents the simulation setup and results, and Section V concludes it.

II. SYSTEM MODEL AND INITIALIZATION

A. System Overview and Network Entities

As illustrated in Fig.1, a typical VEC network is deployed along a two-way road and comprises three layers as follows:

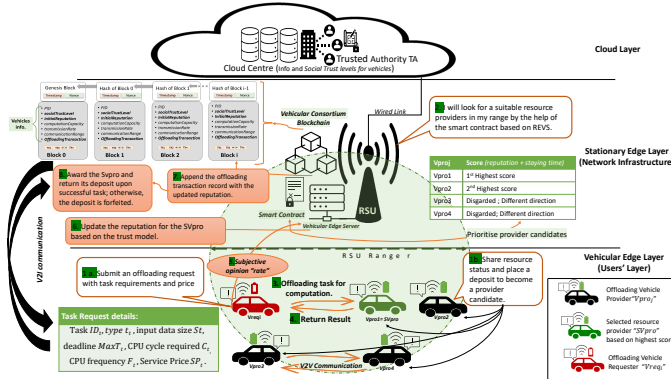


Fig. 1: Overview of the VCOff system applying REVS, showing task submission, provider selection based on trust score, and reputation update. The blockchain immutably stores reputation data and offloading outcomes.

1) *Vehicular Edge Layer:* This layer includes N vehicles operating as either offloading vehicle providers ($Vpro_j$) or requesters ($Vreq_i$), based on their available computational resources. Here, $N = N_{Vreq_i} \cup N_{Vpro_j}$ where $i, j < N$. $Vreq_i$ is task vehicles with computation-intensive tasks seek nearby providers for offloading, while $Vpro_j$ is service vehicles with idle resources capable of performing offloaded tasks. Vehicles must deposit digital coins, either as a service fee from $Vreq_i$ or

as collateral to qualify as $Vpro_j$. We assume $N_{Vpro_j} > N_{Vreq_i}$ to ensure sufficient resources.

2) *Stationary Edge Layer:* This layer includes a single RSU denoted as RSU_1 , equipped with VEC servers to handle large volumes of data and manage reputations in a distributed manner. The VEC servers act as managers and collectors, overseeing the blockchain and all VCOff processes, triggering SC to allocate providers, and updating reputations. It connects to the Cloud layer via high-speed backbone networks [18]. Key components include:

- **Consortium Vehicular Blockchain:** Stores VCOff transactions and reputation updates securely and immutably, serving as a distributed ledger, to maintain trustworthiness. Unlike public blockchains, this lightweight, permissioned blockchain is maintained by pre-selected authorized RSUs [18], reducing computational overhead while ensuring trusted control over reputation management. Leveraging blockchain's decentralization, RSUs can collaboratively manage reputation without relying on a central authority [9].
- **Smart Contract (SC):** Automates secure and fair provider selection by ranking $Vpro_j$ candidates based on weighted trust scores, calculated from reputation and estimated stay times. Leveraging blockchain-enabled SCs ensures transparent, tamper-resistant offloading decisions—ideal for mobile task offloading systems [14].

3) *Cloud Layer:* It consists of the TA, typically a government agency. The TA manages the registration and credentials of all network entities. Each vehicle registers using a verified identity (e.g., driver's license), and receives a digital wallet, and blockchain account. To preserve privacy, each real identity is mapped to a pseudonymous ID used throughout the network.

B. System Initialization

The consortium blockchain uses Elliptic Curve Digital Signature Algorithm (ECDSA) to generate key pairs and secure identity verification and digital signatures during vehicle registration and transaction validation [5], [18]. Given $Vpro_j$'s essential role in the success of VCOff, we assume RSU_1 and $Vreq_i$ act as honest participants, allowing the focus to remain on analyzing and securing the behavior of $Vpro_j$. Malicious actions from providers—such as delaying tasks or returning incorrect results—can severely disrupt the network and lead to VCOff failures [4]–[6]. To prevent flooding by fake requests, a deposit mechanism is used to discourage dishonest behavior from requesters [21]. Although RSUs may be susceptible to compromise, large-scale or long-duration attacks on multiple RSUs are considered highly unlikely due to their distributed deployment, limited attacker resources, and periodic audits by network operators [9]. Therefore, the consortium blockchain, maintained by a majority of semi-trusted RSUs, ensures consistent, tamper-resistant reputation management. Additionally, blockchain prevents malicious behavior from compromising majority of nodes [4], thus reducing risks associated with compromised RSU. This setup mitigates the risk of centralized

failure and enables secure, distributed trust evaluation without relying on a single point of control.

1) *Reputation Initialization*: To initialize the reputation of new participant vehicles, we adopt the ST-based evidence space model [5]. ST scores are assigned by the TA and derived from the verifiable records of the vehicle's owner or driver (e.g., driving history, violation reports, or financial dispute) based on credible sources such as government agencies (e.g., police departments or the Department of Motor Vehicles). This approach addresses the cold-start problem in trust management [5], where new vehicles lack historical interaction data, making it difficult to assign an initial reputation value. Each vehicle is categorized into one of three ST levels: *high*, *intermediate*, and *low*. Each level corresponds to a set of Dirichlet distribution parameters (α, β, γ) , representing counts of positive, negative, and uncertain behaviors, respectively [5]. These parameters determine the starting values of belief $b_{i,j}$, distrust $d_{i,j}$, and uncertainty $u_{i,j}$, as follows:

- **High Social Trust (ST^{high})**: Assigned to entities deemed reliable, typically unlikely to exhibit dishonest behavior. $ST_{i,j}^{high} \leftarrow \{\alpha_{i,j} = 3, \beta_{i,j} = 2, \gamma_{i,j} = 1\}$
- **Intermediate Social Trust (ST^{interm})**: Assigned to entities with no strong positive or negative behavioral history, indicating a neutral or average level of trustworthiness. $ST_{i,j}^{interm} \leftarrow \{\alpha_{i,j} = 2, \beta_{i,j} = 2, \gamma_{i,j} = 2\}$
- **Low Social Trust (ST^{low})**: Assigned to individuals considered more likely to behave dishonestly, resulting in a cautious reputation initialization with higher uncertainty. $ST_{i,j}^{low} \leftarrow \{\alpha_{i,j} = 1, \beta_{i,j} = 2, \gamma_{i,j} = 3\}$.

When a new vehicle joins the network, its initial trust opinion $w_{i,j} = [b_{i,j}, d_{i,j}, u_{i,j}]$ derived from Dirichlet parameters:

$$b_{i,j} = \frac{\alpha_{i,j}}{T}, \quad d_{i,j} = \frac{\beta_{i,j}}{T}, \quad u_{i,j} = \frac{\gamma_{i,j}}{T}, \quad T = \alpha_{i,j} + \beta_{i,j} + \gamma_{i,j}$$

The initial reputation score is then calculated as:

$$R_{i,j} = b_{i,j} + \gamma \cdot u_{i,j}$$

where γ is a predefined constant that controls the influence of uncertainty on the $R_{i,j}$ value.

2) *V2V Computation Offloading Initialization*: When a $Vreq_i$ exceeds its computational capacity, it offloads tasks to a selected provider $SVpro$ in V2V mode (Fig.1) involving:

- Upon entering the RSU range, authorized vehicles share encrypted beacon messages (e.g., resource status, trajectory). This data supports provider eligibility checks and trust score calculation (Algorithm 1).
- Vehicles with idle resources place a deposit to become $Vproj$ candidates. Simultaneously, $Vreq_i$ sends task offloading requests Req_t , includes: [Task ID_t , Type t_t , Input size S_t , completion Deadline $maxT_t$, CPU cycles C_t , CPU Frequency F_t , Service Price SP_t] where $maxT_t = \frac{C_t}{F_t}$.
- REVS is triggered by the SC to select the $SVpro$ with the highest calculated trust score (see Section III).

III. THE PROPOSED BLOCKCHAIN-BASED SECURE COMPUTATION OFFLOADING SOLUTION

Upon receiving Req_t , REVS (Algorithm 1) sorts and identifies the most suitable provider ($SVpro$) based on:

- 1) *Direction Alignment*: Ensures potential $Vproj$ moves in the same direction as the $Vreq_i$ to maintain connectivity.
- 2) *Trust Score*: A weighted sum of reputation and estimated stay time, the time a provider remains within V2V range, computed using relative speed and distance (Algorithm 1, lines 11–18). Weights w_1 and w_2 reflect their importance.

To optimize provider selection, the selection processes incorporate a mobility-aware trust mechanism that balances reputation with expected connection duration. By aligning directions and estimating stay time, providers are less likely to move out of range, enabling selected vehicles to remain within RSU range or V2V range long enough to complete the task successfully.

Algorithm 1 REVS Algorithm

```

1: Function SELECTCANDIDATES $Vproj(Vproj, Vreq_i)$ 
2: Candidates $Vproj \leftarrow Empty\_Array$ 
3: TrustScores  $\leftarrow Initialize\_Array\_of\_Size(Vproj)$ 
4: for each vehicle in  $Vproj$  do
5:   if vehicle.direction ==  $Vreq_i$ .direction and  $R_{i,j} \geq R\_threshold$  then
6:     ADD vehicle to Candidates $Vproj$ 
7:   end if
8: end for
9: SORT Candidates $Vproj$  BY reputation (Descending)
10: for each  $Vproj$  in Candidates $Vproj$  do
11:   Function CALCULATESTAYTIME( $Vreq_i, Vproj$ )
12:   distanceSquared  $\leftarrow (Vproj.x - Vreq_i.x)^2 + (Vproj.y - Vreq_i.y)^2$ 
13:   remainingDistance  $\leftarrow \sqrt{Vreq_i.radius^2 - distanceSquared}$ 
14:    $\triangleright$  Calculate remaining distance within  $Vreq_i$ 's radius as per (6) in [22].
15:   relativeSpeed  $\leftarrow |Vreq_i.speed - Vproj.speed|$ 
16:   V2V_stayTime  $\leftarrow remainingDistance / relativeSpeed$ 
17:   Return V2V_stayTime
18: End Function
19:  $Vproj.V2V\_stayTime \leftarrow CALCULATESTAYTIME(Vreq_i, Vproj)$ 
20:  $TrustScores[Vproj] \leftarrow \omega_1 Vproj.reputation + \omega_2 Vproj.V2V\_stayTime$ 
21: end for
22: ( $maxTrustScore, index$ )  $\leftarrow MAX(TrustScores)$ 
    $\triangleright$  Select provider with the highest trust score.
23:  $SVpro \leftarrow CandidatesVproj[index]$ 
24: Return  $SVpro$ 
25: End Function

```

After selecting $SVpro$, RSU initiates mutual authentication by sending a signed reply message with vehicle certificates. The V2V VCOff scenario proceeds in two main phases:

- 1) **Data Transmission and Task Computation**: Once $Vreq_i$ receives the selected provider's response, it transmits the necessary computation data to $SVpro$. The following parameters determine the success of V2V VCOff [22]:

- **Transmission Time:** is defined as $\text{transTime} = \frac{S_t}{SV_{\text{transRate}}}$, where $SV_{\text{transRate}}$ is SV_{pro} 's transmission rate.
- **Execution Time:** is defined as $\text{exeTime} = \frac{C_t}{SV_{\text{compCap}}}$, where SV_{compCap} is SV_{pro} 's computation capacity.
- **Total VCOFF Latency:** is given by $\text{VCOFFLatency} = \text{transTime} + \text{exeTime}$.
- **Success Criterion:** If the total offloading latency satisfies $\text{VCOFFLatency} \leq \max T_t$, the offloading is marked as successful; otherwise, it is unsuccessful.

2) **Result Feedback and Transactions Settlement:** Upon receiving the result, $Vreq_i$ provides a subjective rating for SV_{pro} . The RSU then updates the SV_{pro} 's reputation score on the blockchain. If the offloading is successfully completed, SV_{pro} is rewarded with reputation credit, claims the service charge and redeems its deposit. Otherwise, the deposit is forfeited, and its reputation is penalized, reducing future selection chances.

After each interaction, the provider's reputation is updated based on task outcome and integrated with existing belief, distrust, and uncertainty values to support future decisions.

IV. SIMULATION SETUP AND NUMERICAL RESULTS

We evaluated REVS using MATLAB simulations. For each run, vehicle reputations were reset and $Vpro_j$ attributes are randomly generated to test various initialization scenarios (Table II). A single $Vreq_i$ acted as the genesis node, assigned with high reputation and fixed direction within RSU_1 's coverage. Cryptographic keys were generated using MATLAB's `secp256k1` function for secure identification [23].

Task Req_t is randomly generated, specifying data size, CPU cycles, and CPU frequency fixed at 5 GHz, representing the lower bound of the [5–10 GHz] range for high-performance processors in smart vehicles [24]. Blockchain is implemented based on a MATLAB Blockchain Example in [25]. Smart Contract is implemented as structured functions following Algorithm 1. Key simulation variables are summarized in Table III, align with previous study [5], [18], [24].

To analyze VCOFF outcomes across reputation categories, initial reputations were grouped into Low, Intermediate, and High ST based on ST levels. Threshold set at intermediate value (≥ 0.5), excluding the ST^{low} providers.

Key findings (Table IV) include:

- **Variable Computation (100 runs):** A few ST^{high} and ST^{interm} failures occurred due to inadequate computation resources, highlighting resource availability's critical role.
- **Fixed Computation (10 GHz, 100 runs):** Improved success significantly, with only one ST^{interm} latency failure, demonstrating reliability of fixed resources.
- **Extended Fixed Computation (1,000 runs):** Maintain high success, confirmed reputation stability and provider reliability under sustained resource conditions.

A. REVS Algorithm Evaluation

1) *Simulation Parameter Setup:* $Vpro_j$'s number is increased to range from 6 to 20 and initialized with fixed

TABLE II: Vehicle Provider Attributes (Summary)

Attribute	Description
SocialTrustLevel	High (40%), Int. (50%), Low (10%)
Reputation	Based on ST level (e.g., 0.6)
Speed / Direction	$\pm V_{req_i}$, aligned or not
Computation / Bandwidth	5–10 GHz / 86 Mbps (high-speed)
Radius	250 m (comm. range)

TABLE III: Simulation Variables and Parameters

Parameter	Description / Value	Parameter	Description / Value
i	Requester vehicle number	j	Provider vehicle number
$R_{i,j}$	Reputation based social trust	$R_{\text{threshold}}$	$R_{i,j}$ cutoff ($\geq 0.4, 0.5$)
w_1	Reputation weight [0–1.0]	w_2	Stay time weight [0–1.0]
S_t	Task size (50–500 KB)	C_t	CPU cycles (0.2–3.2 GHz)
F_t	CPU freq. (5 GHz)	SP_t	Random service price
r	Comm. range (250 m)	γ	Uncertainty weight (0.5)

computation (10 GHz), motivated by earlier simulation insights (see Table IV). An additional attribute, `isMalicious`, which is a malicious behavior indicator, is introduced. The percentage of malicious $Vpro_j$ varied across five levels: {10%, 20%, 30%, 40%, 50%}. Malicious behaviors included delaying result feedback and were modeled with reduced capabilities in terms of *Computation Capacity* and *Transmission Rate*. In contrast, honest vehicles were assumed to possess adequate resources for efficient offloading task execution. Performance was analyzed across seven scenarios (Table V) and reputation threshold set to (> 0.40) to evaluate all candidates.

2) *Reputation Initialization Configurations:* Reputation initialization involved multiple scenarios for ST levels distribution (Table V.) in two configurations: **Proposed Dynamic ST-based Reputation**, initialized across three ST levels (Section II-B1) with maximum of probabilistic malicious assignment (ST^{high} : 10%, ST^{interm} : 30%, ST^{low} : 60%), and **Fixed Reputation**, uniformly initialized at 0.6 [19] (equivalent to ST^{interm}) with random malicious assignment.

Performance was evaluated through:

- Average Success Rate: Percentage of successful offloading transactions within ($\max T_t$).
- Average Latency: Task completion and response time.
- Selected Malicious Providers: Percentage of malicious providers chosen, reflecting resilience.
- Malicious Avoidance Rate (MAR): Effectiveness in avoiding malicious selections.

3) *Comparison of the Various Configurations:* Fig. 2(a), 2(b), 2(c), and 2(d) demonstrate the effectiveness of the proposed ST-based reputation in consistently achieving higher

TABLE IV: Reputation and Offloading Status

Simulation Setup	Social Trust Level	Success	Failure
Variable Computation Capacities (100 runs)	High	30	11
	Intermediate	16	3
	Low	0	40
Fixed Computation Capacity 10 GHz (100 runs)	High	44	0
	Intermediate	29	1
	Low	0	26
Fixed Computation Capacity 10 GHz (1000 runs)	High	398	11
	Intermediate	300	8
	Low	0	283

TABLE V: Scenarios and ST Distributions (%)

Scenario	Description	ST Levels (High/Int./Low)
1	Skewed High	40 / 50 / 10
2	Uniform	33 / 33 / 33
3	Skewed Low	30 / 30 / 40
4	Balanced	30 / 40 / 30
5	Random	34.24 / 33.16 / 32.60
6	Random Selection	40 / 50 / 10
7	Random Selection	30 / 40 / 30

success rates, lower latency, better MAR, and higher reliability. **Proposed Configurations** achieved high success rates averaging 96% at lower malicious percentages, with Scenario 1 dropping to 84.1% at higher malicious levels. Latency was lowest in Scenario 3 (204.26 ms) and highest in Scenario 1 (209.53 ms). Malicious selection remained under 10%, and MAR averaged 96.5%, exceeding success rates due to offloading failures from unavailable V_{proj} moving differently than V_{req} . **Fixed configurations** saw static initialization hinder adaptability, with success rates average dropping to 66.5%, latency between 256.96–259.22 ms, and malicious selection exceeding 50%, lowering MAR to 46%. **Random selection** in Scenarios 6 and 7 performed worst, with low success rates, high latency, and increased malicious interactions, emphasizing the importance of reputation-based systems like REVS for efficiency, security, and adaptability.

These results prove the 40.85% improvement in success rates, 40.70% in malicious avoidance rates (MAR) and a 20% latency reduction over fixed-reputation and random selection methods, not considering trust.

B. Discussion

The results highlight the superiority of REVS algorithm under varying malicious ratios, outperforming other approaches.

1) *Scenario 1 Performance:* Scenario 1, with a skewed ST^{high} distribution, showed reduced reliability at 40–50% malicious nodes. REVS favors ST^{high} providers due to their initial trust scores, but when some become malicious, over-reliance on this group leads to system instability. With only 10% ST^{low} vehicles available as fallback options, the system lacked diversity, pushing more malicious nodes into the ST^{interm} and ST^{high} categories, increasing the risk of poor selections and failure, raising the malicious count in ST^{high} to 7% at 50% malicious percentages, compared to about 4% in other scenarios with 50% malicious percentages, further undermining system reliability.

2) *Performance Across Other Scenarios:* Scenario 3, with a larger ST^{low} group, performed best by absorbing most malicious assignments and minimizing selection chances through REVS, maintaining high success and MAR at 40–50%. Scenario 5, with a random ST distribution, also performed well, slightly outperforming the uniform distribution in Scenario 2 due to different specific values (Table V). Scenario 4, while performing the lowest among these, still outperformed Scenario 1 due to a more balanced ST distribution, reducing the risks associated with over-reliance on compromised ST^{high}

nodes. Overall, Scenarios 2–5 kept malicious ST^{high} nodes below 4% and maintained MAR above 90%, which underscores the value of balanced or skewed ST^{low} distributions.

These findings emphasize the critical role of proper ST-level distributions and adaptive trust mechanisms in enhancing VCOFF reliability. Scenario 1’s skew toward ST^{high} created a single point of failure, as malicious in ST^{high} group disproportionately impacted performance at 40–50% malicious ratios. In contrast, distributing ST levels more evenly and using ST^{interm} and ST^{low} as fallback options helped mitigate risk. Ensuring malicious ST^{high} remains below 4% is key to maintaining success rates and MAR above 90%.

3) *Geographic Constraints and Provider Selection:* Geographic factors like provider location and direction alignment sometimes required selecting ST^{interm} or ST^{low} providers, even when ST^{high} options were available, resulting in minor failures under favorable conditions. For instance, isolating failures caused solely by malicious SV_{pro} in Scenario 3 improved the success rate to 93.11% at 50% malicious nodes, compared to overall 90.5%, highlighting REVS’s adaptability to geographic constraints and its ability to limit malicious impact.



Fig. 2: Performance comparison of the proposed dynamic ST-based reputation and fixed reputation initialization across multiple scenarios (Table V) in different evaluation metrics.

C. Refined Weighted Trust Score Analysis

We analyzed the impact of varying reputation and stay-time weights (w_1, w_2) on REVS performance under Scenario 1 (Fig. 3). Prioritizing reputation ($w_1 = 0.7, w_2 = 0.3$) provided consistently superior performance, achieving high success rates (above 84.1% at 50% malicious nodes), low latency (220 ms at 50%), minimal malicious provider selection (0.131 at 50%), and high malicious avoidance rates (MAR of 86.9% at 50% and above 90% at 40%). In contrast, prioritizing stay time ($w_1 = 0.3, w_2 = 0.7$) decreases the success rate to 73.5% with increasing latency up to (240 ms at 50%), due to increased selection of unreliable providers, highlighting their vulnerability in high-malicious scenarios. Balanced configurations

($w_1 = w_2 = 0.5$) provided moderate performance, while relying solely on reputation ($w_1 = 1.0$) reduced adaptability and overall performance (79.2% success rate at 50% malicious). Overall, the $w_1 = 0.7, w_2 = 0.3$ setting emerged as optimal, confirming the robustness and effectiveness of prioritizing reputation for secure, stable, and efficient provider selection in REVS. The proposed REVS proves more adaptable and effective than fixed configurations across all metrics.

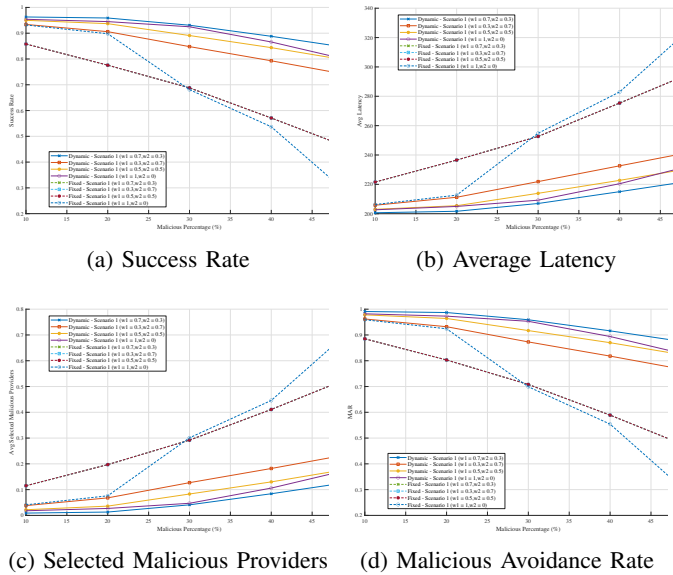


Fig. 3: Impact of weights configurations (w_1, w_2) on the performance of proposed Dynamic and Fixed Reputation Initialization under Scenario 1 in different evaluation metrics.

V. CONCLUSION

This paper has introduced Reputation-Enhanced Vehicular Selection (REVS) algorithm, a blockchain-based framework for secure and efficient computation offloading in vehicular networks. REVS achieved high success rates, and effective malicious provider avoidance, demonstrating superior performance and overall reliability. Although this study focuses on REVS, additional enhancements remain an area of interest. In addition to REVS, we aim to develop a comprehensive trust management model by integrating Reputation, Uncertainty, and Subjective Logic. Future work will focus on refining reputation updating processes by maintaining consistent vehicles across simulations to capture reputation evolution. Planned enhancements include reputation-based incentives, penalties for malicious providers, and AI-driven adaptability. These advancements aim to ensure secure vehicular computation offloading (VCOff) in high-mobility environments.

REFERENCES

- [1] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, 2021.
- [2] R. A. Dziyauddin, D. Niyato, N. C. Luong, A. A. A. M. Atan, M. A. M. Izhar, M. H. Azmi, and S. M. Daud, "Computation offloading and content caching and delivery in vehicular edge network: A survey," *Comput. Netw.*, vol. 197, p. 108228, 2021.

- [3] S. Y. Fayi and Z. Sheng, "A survey of security, privacy and trust issues in vehicular computation offloading and their solutions using blockchain." *Open Research Europe*, vol. 3, 2023.
- [4] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, 2021.
- [5] S. Xu, C. Guo, R. Q. Hu, and Y. Qian, "Blockchain inspired secure computation offloading in a vehicular cloud network," *IEEE Internet Things J.*, pp. 1–1, 2021.
- [6] S. Wang, D. Ye, X. Huang, R. Yu, Y. Wang, and Y. Zhang, "Consortium blockchain for secure resource sharing in vehicular edge computing: A contract-based approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1189–1201, 2021.
- [7] S. Iqbal, R. M. Noor, and A. W. Malik, "A review of blockchain empowered vehicular network: Performance evaluation of trusted task offloading scheme." IEEE, 2021, Conference Proceedings.
- [8] H. Zeyu, X. Geming, W. Zhaohang, and Y. Sen, "Survey on edge computing security." IEEE, 2020, Conference Proceedings.
- [9] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, 2019.
- [10] S. A. Siddiqui, A. Mahmood, Q. Z. Sheng, H. Suzuki, and W. Ni, "A survey of trust management in the internet of vehicles," *Electronics*, vol. 10, no. 18, p. 2223, 2021.
- [11] Y. Xu, X. Li, M. Jin, and Y. Lu, "A trusted distribution mechanism of tasks for the internet of vehicles based on blockchain." IEEE, 2021, Conference Proceedings.
- [12] C. Pu, "A novel blockchain-based trust management scheme for vehicular networks," in *2021 wireless telecommunications symposium (WTS)*. IEEE, 2021, Conference Proceedings, pp. 1–6.
- [13] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-based distributed trust and reputation management systems: A survey," *IEEE Access*, vol. 8, pp. 21 127–21 151, 2020.
- [14] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Secure computation offloading in blockchain based iot networks with deep reinforcement learning," *IEEE Transactions on Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3192–3208, 2021.
- [15] P. Lang, D. Tian, X. Duan, J. Zhou, Z. Sheng, and V. C. M. Leung, "Cooperative computation offloading in blockchain-based vehicular edge computing networks," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 783–798, 2022.
- [16] S. Iqbal, A. W. Malik, A. U. Rahman, and R. M. Noor, "Blockchain-based reputation management for task offloading in micro-level vehicular fog network," *IEEE Access*, vol. 8, pp. 52 968–52 980, 2020.
- [17] A. Jøsang, "The right type of trust for distributed systems," in *Proceedings of the 1996 workshop on New security paradigms*, 1996, Conference Proceedings, pp. 119–131.
- [18] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, 2019.
- [19] Q. Liu, J. Gong, and Q. Liu, "Blockchain-assisted reputation management scheme for internet of vehicles," *Sensors*, vol. 23, no. 10, p. 4624, 2023.
- [20] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25 408–25 420, 2017.
- [21] X. Huang, D. Ye, R. Yu, and L. Shu, "Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design," *IEEE/CAA J. Autom. Sinica*, pp. 426–441, 2020.
- [22] S. Raza, W. Liu, M. Ahmed, M. R. Anwar, M. A. Mirza, Q. Sun, and S. Wang, "An efficient task offloading scheme in vehicular edge computing," *Journal of Cloud Computing*, vol. 9, no. 1, 2020.
- [23] D. Hill, "secp256k1 elliptic curve shared-key generation gui," <https://www.mathworks.com/matlabcentral/fileexchange/73364-secp256k1-elliptic-curve-shared-key-generation-gui>, 2025.
- [24] J. Chen, X. Wang, and X. Shen, "Rte: Rapid and reliable trust evaluation for collaborator selection and time-sensitive task handling in internet of vehicles," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 12 278–12 291, 2024.
- [25] R. Aarenstrup, "Matlab blockchain example," <https://www.mathworks.com/matlabcentral/fileexchange/65419-matlab-blockchain-example>, 2018.