A Blockchain Framework for Incentivized Data Sharing in Autonomous Vehicle Networks*

Giuseppe Olivieri[1], Agostino Marcello Mangini[1], *Senior, IEEE*, Maria Pia Fanti[1], *Fellow, IEEE*

***Abstract*— Autonomous vehicles (AVs) continuously generate high-resolution sensor data on road conditions, infrastructure updates, and traffic dynamics. Despite their critical relevance for real-time navigation and urban planning, these datasets remain siloed within manufacturer-specific platforms. Motivated by the necessity to overcome such fragmentation, this paper introduces a novel decentralized, blockchain-based framework whose key innovation is a dynamic voting threshold integrated into a modular smart contract architecture. In our model, AVs can submit and validate road events –such as newly detected closures or construction sites– through a modular smart contract system employing dynamic voting thresholds that adapt acceptance criteria based on different factors. This allows urgent changes to achieve consensus while quickly minimizing malicious or erroneous reporting. Upon reaching a consensus regarding the specific event, the proposer is granted token-based incentives redeemable for operational cost reductions (e.g., charging or parking discounts). The proposed approach is validated via a Hardhat simulation on an Ethereum Virtual Machine compatible test network, demonstrating our design's feasibility, robustness, and responsiveness under diverse scenarios.**

***Index Terms*— Autonomous Vehicles, Blockchain, Decentralized Architecture, Token-based Incentives, Distributed Validation**

I. Introduction

Autonomous vehicles (AVs) generate vast, high-resolution datasets through advanced sensor networks (including LiDAR, camera arrays, and radar systems), capturing everything from evolving road conditions to near-instantaneous traffic updates. Although these data are essential for enhancing road safety, optimizing route planning, and guiding infrastructure development, most information remains locked within brand-specific platforms. As a result, AVs from different manufacturers do not benefit from each other's real-time observations, and public authorities, service providers, and end users miss out on crucial insights for city-wide traffic management, road maintenance, and urban planning. Additionally, modern proprietary navigation platforms typically operate under centralized, closed-source paradigms, offering no direct economic incentives to users for sharing high-precision data and severely limiting data ownership transparency within the provider's ecosystem.

Recently, blockchain-based architectures have emerged as promising approaches for addressing information sharing, interoperability, and trust issues in Intelligent Transportation Systems (ITS) [1]. Rajkumar et al. [2] proposed vehicular data architectures leveraging trusted execution environments, and Cui et al. [3] introduced consortium blockchains for secure V2V communications. Both works, however, primarily focus on privacy/security without considering economic incentives or reputation-driven consensus mechanisms, which are essential factors pursued in our approach. Moreover, discrete event system techniques have also been considered for enhancing cyber-attack detection, security, and fault diagnosis in decentralized AV data-sharing systems [4], [5].

Recent systematic reviews, such as those by Sarwatt et al. [6], Alherimi et al. [7], Kim and Vong [8], and Vairam et al. [9], catalog diverse blockchain applications spanning automotive, IoT, and ITS domains. These contributions generally overlook barriers associated with cross-manufacturer collaboration, open participation, and economically-driven data-sharing ecosystems, which constitute core challenges our work explicitly tackles.

Additionally, Yang et al. [10] and Rasool et al. [11] explored Decentralized Autonomous Organizations (DAOs) as governance frameworks, enabling decentralized decision-making processes. Nevertheless, these contributions do not directly support dynamic sensor-data monetization or cross-brand interoperability. Similarly, Qin et al. [12] introduced a multi-blockchain (”TriBoDeS”) architecture suited to hazard dissemination in vehicular environments. Despite the innovation, the tri-blockchain configuration brings significant operational complexity, impeding applicability to realistic AV scenarios involving heterogeneous manufacturers.

Motivated by these limitations, this paper proposes a lightweight, decentralized framework for real-time AV data exchange, integrating modular smart contracts, dynamic consensus threshold mechanisms, vehicle reputation-based incentivization, and blockchain tokenomics. Unlike existing solutions, our approach uniquely combines cross-brand interoperability, real-time validation, economic incentivization, and high scalability, ultimately fostering open, collaborative vehicular data marketplaces. Verified contributions directly translate into incentives (tokens) redeemable as practical benefits (charging discounts, parking fees, toll reductions), thereby aligning individual vehicle incentives with collective intelligent mobility objectives.

The remainder of this paper proceeds as follows: Section II describes the problem, and Section III outlines the System Architecture. Section IV mathematically outlines the dynamic threshold of the considered system, Section V presents the blockchain simulation and Section VI draws the conclusions.

II. Problem Description

Autonomous vehicles can collect both static information, such as road infrastructure characteristics and classification, and dynamic phenomena, such as unexpected traffic incidents, temporary road closures, or routine maintenance work. Despite the critical importance of these insights for real-time decision-making and broader traffic management, the data collected often remains confined within proprietary ecosystems belonging to individual manufacturers.

*This research is part of the IN2CCAM project that has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101076791.

[1] Department of Electrical and Information Engineering, Polytechnic University of Bari, 70126, Bari, Italy g.olivieri@phd.poliba.it; (agostinomarcello.mangini, mariapia.fanti)@poliba.it.

The isolation of data within brand-specific platforms severely limits interoperability and stifles the potential benefits of a more comprehensive, cross-manufacturer data repository. For instance, consider a scenario in which a vehicle navigates what was initially marked as a secondary, paved route in its onboard navigation system yet encounters an unpaved, rough terrain in reality. The AV's sensors detect this mismatch, generating valuable information about the true state of the roadway. However, because this information remains locked within a single manufacturer's proprietary system, other AVs and stakeholders do not benefit from these real-time observations.

A key challenge in addressing these limitations lies in establishing a secure, trust-based mechanism that incentivizes the seamless exchange of vehicular data across multiple stakeholders, specifically addressing several key elements. Firstly, ensuring robust data provenance and integrity is imperative to guarantee confidence in the origin and authenticity of shared information and effectively mitigate risks associated with falsification or unauthorized tampering. Secondly, owner privacy must be meticulously preserved, especially in contexts where vehicular datasets encompass sensitive or personally identifiable information, necessitating advanced anonymization and privacy-preserving strategies. Thirdly, a carefully structured monetization and incentive system –potentially incorporating operational or economic rewards, such as credits redeemable for charging discounts– is essential to motivate extensive and willing participation in data-sharing initiatives. Finally, the capability to support real-time processing and scalability is vital, enabling rapid, near-instantaneous transactions within distributed environments and adequately managing the substantial throughput demands typical of contemporary sensor networks deployed in autonomous vehicles.

In light of these considerations, the need for a distributed data framework becomes evident. By leveraging blockchain-based architectures, smart contracts, and advanced cryptographic techniques, AV data can be securely exchanged and monetized across brand and platform boundaries.

III. System Architecture and Modular Smart Contracts

The system architecture, depicted in Fig. 1, enables autonomous vehicles from different manufacturers to detect, broadcast, and validate critical environmental changes through a public, permissionless blockchain infrastructure. The core components and interactions are summarized in the following subsections.

A. *Overview of the Proposed Workflow*

Step 1: Detection of Discrepancy. Vehicle **A** (shown in orange in Fig. 1) identifies a mismatch between the real-world state of an infrastructure element (e.g., a newly unpaved road or a temporary closure) and the existing information in its onboard navigation system. This discrepancy can arise from *ad hoc* changes (e.g., ongoing construction) or more permanent infrastructural modifications.

Step 2: Broadcast of New Information. Once Vehicle **A** detects this discrepancy, it encodes the updated information (e.g., the exact location, time, and nature of the change) in a structured JSON object. Our earlier research [14] indicates that implementing a hash-based system for uniquely identifying intersections is feasible. This JSON payload is then submitted to the *smart contract system* (further elaborated in Algorithm 1), triggering a blockchain transaction. This transaction records key metadata such as a timestamp, vehicle pseudonym (to preserve privacy), and event category.

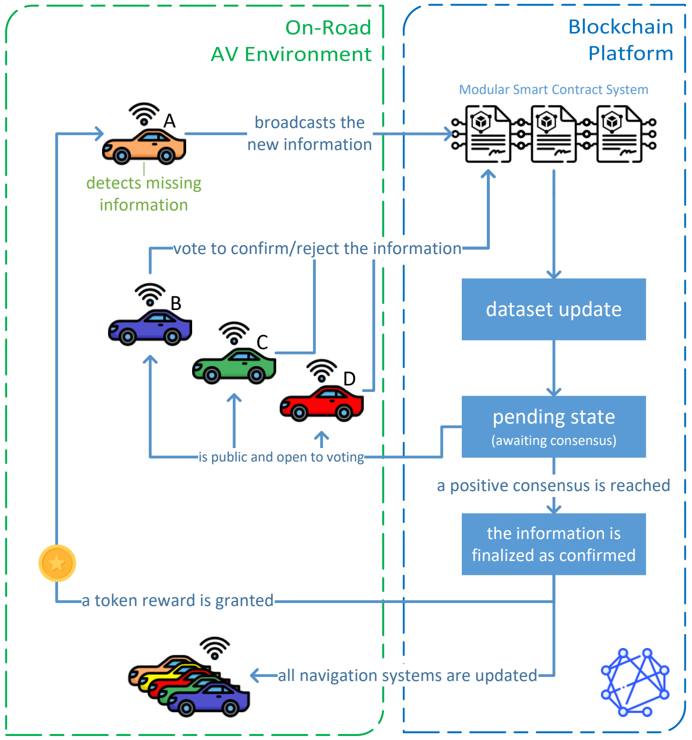

On-Road
AV Environment
Blockchain
Platform
Modular Smart Contract System
A
broadcasts the
new information
detects missing
information
vote to confirm/reject the information
B
C
D
dataset update
pending state
(awaiting consensus)
is public and open to voting
a positive consensus is reached
the information is
finalized as confirmed
a token reward is granted
all navigation systems are updated

Fig. 1. System architecture [13]

Step 3: Pending State Creation. Upon receiving the transaction, the blockchain-based smart contract appends the new data to a global dataset in a *pending state*. At this stage, the reported change has no immediate effect on navigation systems; rather, it awaits consensus from other vehicles.

Step 4: Crowd-Sourced Validation. As vehicles **B**, **C**, and **D** traverse or otherwise sense the same region, they submit *confirmation* or *rejection* transactions to the smart contract. These vehicles also rely on their own sensor inputs to verify the reported information. Multiple confirmation transactions strengthen the claim, while multiple rejection transactions may overturn it.

Step 5: Consensus Threshold. Once the threshold discussed in Section IV is reached, the pending state is resolved. The threshold is dynamically adjusted based on road type, event severity, and vehicle reputation. If consensus deems the newly reported data to be accurate, it is marked as *confirmed*. Otherwise, the entry is *rejected*.

Step 6: Data Finalization and Reward. In the event of a successful confirmation, the updated information is promoted to the *structured data layer* and disseminated to all vehicles' navigation systems. This ensures that other AVs immediately benefit from the new information, improving their route planning and operational safety. Vehicle **A** –the original proposer of the valid data– is awarded a token as an incentive.

B. *Modular Smart Contracts System*

The proposed framework achieves flexibility and scalability by deploying three closely integrated smart contracts, each with a distinct but interlocking responsibility. As detailed in Algorithm 1, the *FilterContract* first receives every new data submission transmitted by a vehicle, performing checks on the JSON-encoded payload and verifying that no identical *pending* entry exists for the same event. When a new submission is found to be unique, the contract relays it to the *VotingContract* with *pending* status;

Algorithm 1 Workflow for Distributed AV Data Sharing via Modular Smart Contracts

```
Step 1: FilterContract - Initial Validation and Duplicate Checking
Receive txData (e.g., infrastructure mismatch) and $v_{\text{prop}}$ (vehicle pseudonym).
Perform basic checks on txData (required fields in JSON format).
Verify whether an identical pending entry already exists for the same issue.
if no matching entry is found then
    Forward txData to VotingContract with pending status.
else
    Convert this submission into a CONFIRM vote for the existing pending entry in VotingContract.
    Do not create a new pending record.
end if
Step 2: VotingContract - Consensus Mechanism
Initialize vote counts for each pending proposal: yesVotes = 0, noVotes = 0.
for each vote received from vehicles do
    if vote is CONFIRM then
        Increment yesVotes.
    else if vote is REJECT then
        Increment noVotes.
    end if
    Dynamic recalculation of $T_{\text{dyn}}(i)$ and immediate check
    Check if yesVotes and noVotes trigger immediate approval or rejection.
end for
At the end of the voting window or upon reaching the minimum required number of votes
Check final consensus according to Section IV.
if proposal is approved then
    Forward the approval result to TokenContract.
else
    Mark the proposal as rejected.
end if
Emit an event indicating the outcome ("CONFIRMED" or "REJECTED").
Update rep(v) and f(v) in accordance with Section IV-C.
Step 3: TokenContract - Token Management
if proposal is approved then
    Mint a reward token to the proposer $v_{\text{prop}}$.
    Store the validated data in the confirmed repository.
end if
Handle all token lifecycle operations (e.g., emission, balance tracking, burn logic).
```

Once a proposal is marked as *pending*, the *VotingContract* takes over to manage crowd-sourced validation. Each vehicle with relevant sensor data can cast a `CONFIRM` or `REJECT` vote. The contract keeps a running tally of both counts and, after each vote, recalculates a dynamic acceptance threshold according to the methodology presented in Section IV. This threshold, denoted $T_{\text{dyn}}(i)$, varies based on different factors such as the severity of the event, the criticality of the road segment, the elapsed time since the proposal was submitted, and the reputation of the proposer. In this way, the *VotingContract* can either finalize an outcome before the voting window concludes –triggered by immediate confirmation or rejection criteria– or wait until a predefined time limit or a minimum vote count is reached. Once a decision has been reached, the result is irrevocably recorded on-chain, ensuring transparency and accountability in the consensus process. The final module, the *TokenContract*, governs the system's incentive layer. Suppose the *VotingContract* signals that a proposal has been confirmed. In that case, the TokenContract mints one or more tokens in the name of the original proposer, thereby acknowledging a valid and beneficial data contribution to the network. Note that all reputation updates and flag management –incorporating the double-hit logic (discussed further below in Section IV-C) for handling consecutive negative proposals– are performed within the *VotingContract*, which decouples these operations from token management. The *TokenContract* further handles an array of token lifecycle operations, including transfer, balance tracking, and burn logic. This modular design, wherein the tokenomics are confined to a dedicated contract, allows for tailored governance and dynamic adjustments to reward policies without disrupting the consensus or filtering processes.

Together, these three smart contracts operate in unison to secure real-time data, validate its authenticity, and incentivize honest reporting. By cleanly separating the *filtering*, *consensus*, and *reward* functionalities, the architecture remains flexible to evolving requirements while preserving core trust assumptions.

C. Blockchain Platform

In order to maximize transparency, immutability, and cross-brand participation, the proposed system leverages a blockchain **public**, one in which every transaction on the ledger is openly readable, enabling stakeholders to audit data flows and verify event integrity, and **permissionless**, meaning that any entity may participate in block validation or production (in accordance with the network's consensus rules).

Since vehicles remain pseudonymized via cryptographic addresses, privacy concerns are mitigated while retaining the benefits of a trustless, decentralized architecture. This permissionless approach fosters an open ecosystem where entities from multiple automotive manufacturers can seamlessly contribute to and benefit from the shared dataset by obviating the need for a central authority.

Although the proposed system already integrates a reputation model and dynamic thresholding, additional mechanisms could further refine both the consensus process and the reward structure. Under such a *dynamically weighted* scheme, submissions from reputable entities could reach an accelerated consensus on critical events while mitigating the influence of malicious actors. Similarly, the token issuance could be tied to contextual factors such as the severity or criticality of the event, effectively rewarding contributors based on the *quality* or *importance* of their reports.

Furthermore, practical considerations for real-time, large-scale deployments underscore the need to minimize transaction costs and ensure network scalability. One viable solution is to adopt a high-throughput platform or a specialized *subnet* on public permissionless blockchains like Avalanche [15], or to employ a layer-two scaling solution on Ethereum [16]. By leveraging existing validator infrastructures, the system can dramatically reduce per-transaction fees and latencies, thereby enhancing both cost-effectiveness and responsiveness. Such an approach also preserves the trustless, decentralized nature of the network, which is essential for an open, multi-vendor ecosystem of autonomous vehicles.

D. Real-World Applications of the Reward Mechanism

The token-based reward mechanism introduced in our framework offers practical utilities beyond data contribution incentives, encouraging continuous engagement and fostering an ecosystem of mutual value creation among vehicles, users, and service providers. Specifically, tokens can be practically redeemed in multiple key scenarios within the autonomous vehicle ecosystem, such as:

- **Premium Data Access:** Vehicles holding tokens can utilize them to reduce or waive fees for accessing specialized or high-value data within the system, thus incentivizing continued participation.
- **Charging Discounts:** Electric and hybrid vehicles may redeem tokens to obtain discounts at charging stations, thereby reducing operational expenditures.
- **Parking Fee Management:** Integration of tokens into urban parking systems allows users to conveniently pay parking fees or receive price reductions.
- **Toll Payment Reduction:** Toll authorities can offer preferential toll rates or facilitate smoother billing transactions through token adoption, rewarding vehicles that actively contribute reliable road-condition data.

These practical applications, which can be later implemented, establish a continuous cycle of economic incentives, fostering widespread adoption by aligning individual benefits with collective improvements in road safety and efficiency.

IV. Mathematical Formulation for Determining the Dynamic Threshold

This section proposes a mathematical formulation for a dynamic threshold to efficiently manage the second smart contract described in Section III-B and Algorithm 1. The goal is to parameterize the various factors involved to allow urgent information to be accepted and transmitted to navigation systems in a timely manner. At the same time, it is crucial to prevent system manipulation by entities that repeatedly provide incorrect or fraudulent information, disfavoring them in the long term while accounting for possible detection and transmission errors.

A reputation system was introduced for vehicles transmitting new information to the system; this parameter will play a role in modeling the dynamic acceptance threshold. However, we have chosen not to have the reputation affect the reaching of consensus (so individual vehicles will not have different weights when voting on whether a piece of information is correct or not). This ensures that the system is as distributed as possible and avoids single points of failure.

A. Dynamic Acceptance Threshold: Formulation and Notation

Parameters and variables employed throughout this section are summarized in Table I.

TABLE I
Summary of Symbols and Notation

Symbol	**Description**
$\mathcal{V} = \{v_1, \ldots, v_N\}$	Set of vehicles involved.
$v_{\text{prop}}(i) \in \mathcal{V}$	Vehicle (proposer) that submits proposal i.
$\text{sev}(i) \in [0, 1]$	Severity of the event reported by proposal i.
$\text{road}(i) \in [0, 1]$	Criticality of the road segment for proposal i.
$\Delta t_i \geq 0$	Time elapsed since the initial submission of proposal i, i.e., $\Delta t_i = \text{timestamp}_{\text{current}} - \text{timestamp}_{\text{start}}^{(i)}$.
$\text{rep}(v) \in \mathbb{R}_{\geq 0}$	Reputation score of vehicle v.
$f(v) \in \{0, 1\}$	Alert flag: $f(v) = 1$ if vehicle v is flagged in an alert state due to a recently rejected proposal
$\tau_{\text{base}} \geq 0$	Base threshold for approving a proposal with minimum severity and criticality.
λ	Weighting coefficient for the proposer's reputation term $\text{rep}(v_{\text{prop}}(i))$.
α	Weighting coefficient for the event severity term $\text{sev}(i)$.
β	Weighting coefficient for the road criticality term $\text{road}(i)$.
γ	Weighting coefficient for the elapsed time term $\ln\bigl(1 + \Delta t_i\bigr)$, ensuring diminishing returns.
$T_{\text{dyn}}(i)$	Dynamic acceptance threshold for proposal i
$T_{\min} \geq 0, \quad T_{\max} \geq T_{\min}$	Global minimum and maximum bounds for the dynamic threshold $T_{\text{dyn}}(i)$.
$\text{timeWindow} > 0$	Maximum time window within which a proposal must be confirmed or rejected.
$k_{\min} \geq 1$	Minimum number of votes required to reach a decision.
$\text{Y}(i) \in \mathbb{N}_{\geq 0}$	Number of `CONFIRM/YES` votes for proposal i.
$\text{N}(i) \in \mathbb{N}_{\geq 0}$	Number of `REJECT/NO` votes for proposal i.

The dynamic acceptance threshold $T_{\text{dyn}}(i)$ for proposal i is defined as:

$$T_{\text{dyn}}(i) \;=\; \text{clamp}\Bigl[\,\tau_{\text{base}} \;+\; \alpha\,\text{sev}(i) \;+\; \beta\,\text{road}(i) \;-\; \gamma\,\ln\bigl(1 + \Delta t_i\bigr) \;-\; \lambda\,\text{rep}\bigl(v_{\text{prop}}(i)\bigr),\; T_{\min},\; T_{\max}\Bigr] \tag{1}$$

where $\text{clamp}[x, a, b] = \min\bigl(\max(x,\, a),\, b\bigr)$ ensures that x is limited between a and b, preventing the threshold from becoming negative or excessively high and the parameters τ_{base}, α, β, γ, and λ are weighting coefficients that determine the influence of each factor on the dynamic threshold.

In this initial application scenario, we opted to initially set all coefficients equal to 1 ($\alpha = \beta = \gamma = \lambda = \tau_{\text{base}} = 1$), deferring a more detailed parameter tuning to subsequent analyses. This allows us to assess the behavior of the normalized variables before applying different weights and to lay the groundwork for future multi-objective optimization as part of a simulation. Retaining the coefficients in the formula provides flexibility for potential adjustments based on empirical data or changing requirements.

Variables $\text{sev}(i)$ and $\text{road}(i)$ are normalized within the range $[0, 1]$ for consistency. The term Δt_i is measured in terms of the *number of blocks* emitted by the network, as it is the basic unit within the blockchain. Its scaling will depend on the selected blockchain platform.

The reputation $\mathrm{rep}(v)$ is updated according to the rules specified in Section IV-C. The global thresholds $T_{\min}$ and $T_{\max}$ ensure that $T_{\mathrm{dyn}}(i)$ stays within acceptable bounds. With this formulation, if the proposer's reputation is very high and/or a significant amount of time has passed (large Δt_i), the threshold lowers but never goes below $T_{\min}$. Conversely, if the event's severity or the road segment's criticality is very high, the threshold cannot rise indefinitely due to $T_{\max}$. Defining $T_{\min}$ and $T_{\max}$ based on preliminary tests or simulations to reflect real voting conditions and vehicle participation is advisable.

B. Voting Rules and Final Decision

Let $\mathrm{Y}(i) \in \mathbb{N}_{\geq 0}$ and $\mathrm{N}(i) \in \mathbb{N}_{\geq 0}$ respectively denote the cumulative counts of approval (`CONFIRM`) and rejection (`REJECT`) votes for a pending proposal i. Let $T_{\mathrm{dyn}}(i)$ represent the dynamic validation threshold for proposal i, calculated according to the formula presented in Sec. IV-A. Additionally, let $k_{\min}$ represent the minimum number of votes necessary and $\mathrm{timeWindow}$ the maximum allowed voting duration.

The decision-making follows four phases:

1) **Immediate Confirmation Criterion:** A submitted data item is immediately accepted and finalized as **confirmed** if:

$$\mathrm{Y}(i) \geq T_{\mathrm{dyn}}(i) \quad \text{and} \quad \mathrm{Y}(i) > \mathrm{N}(i). \tag{2}$$

2) **Immediate Rejection Criterion:** Conversely, a proposal is immediately marked as **rejected** if either of the following conditions holds:

$$\mathrm{N}(i) \geq T_{\mathrm{dyn}}(i) \quad \text{or} \quad \mathrm{N}(i) > \mathrm{Y}(i). \tag{3}$$

3) **Final Resolution at Voting Closure:** If neither immediate confirmation nor immediate rejection criteria have been triggered by the end of the voting window (i.e., upon completion of time $\mathrm{timeWindow}$ or upon reaching the minimum vote requirement $k_{\min}$), the final state of the proposal is decided according to the majority criterion:

$$\mathrm{status}(i) = \begin{cases} \textbf{confirmed}, & \mathrm{Y}(i) > \mathrm{N}(i), \\ \textbf{rejected}, & \mathrm{Y}(i) \leq \mathrm{N}(i). \end{cases} \tag{4}$$

4) **Resolving Simultaneous Threshold Exceedance:** In scenarios where both approvals votes $\mathrm{Y}(i)$ and rejection votes $\mathrm{N}(i)$ concurrently cross the dynamic threshold $T_{\mathrm{dyn}}(i)$ within the same time interval (e.g., within the same block of blockchain transactions), we deterministically resolve the ambiguity by evaluating the vote difference:

$$\mathrm{status}(i) = \begin{cases} \textbf{confirmed} & \text{if } \mathrm{Y}(i) - \mathrm{N}(i) > 0, \\ \textbf{rejected}, & \text{otherwise.} \end{cases} \tag{5}$$

This deterministic voting scheme effectively prevents ambiguity and mitigates potential manipulation or exploitation of the voting system. The combination of dynamic thresholding, minimum vote counts, and defined timing windows guarantees reliable and timely decision-making, thereby enhancing overall data accuracy and consistency across automotive platforms.

C. Vehicle Reputation and Incentive Mechanism

A simple reputation model inspired by repeated-game frameworks is implemented to incentivize truthful data submission and build long-term vehicle collaboration [17]. Each vehicle $v \in \mathcal{V}$ maintains a reputation score $\mathrm{rep}(v)$, initialized at a positive default value $\mathrm{rep}_{\mathrm{default}}$.

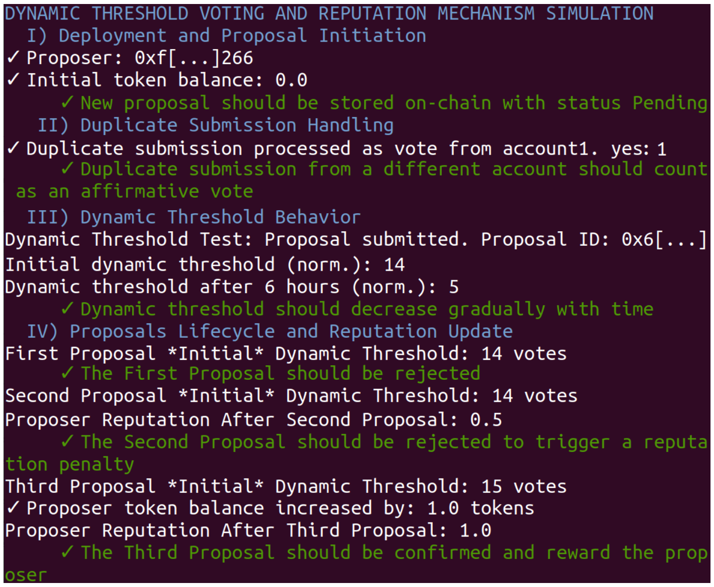
Our system adopts a *double-hit penalty* policy to balance fairness and robustness to errors. Specifically, when a vehicle proposes information that is subsequently rejected through the consensus procedure, it does not suffer an immediate penalty, rather receiving a one-time binary warning flag $f(v)$. Only upon submitting two consecutive rejected proposals does the vehicle incur a reputation penalty. Conversely, if previously decreased, a successfully confirmed contribution resets the flagged status and increases the vehicle's reputation score.

These reputation adjustments have a twofold benefit: on the one hand, they discourage malicious and careless behavior while avoiding excessively penalizing occasional sensor inaccuracies or unavoidable false positives. This straightforward yet effective logic creates incentives for sustained collaborative participation by rewarding accurate, timely inputs and gently discouraging unreliable submissions. In our implementation, the reputation updating logic remains encapsulated within the VotingContract (Section III-B), thus clearly separating incentive management from foundational blockchain transactions.

V. Blockchain Simulation and Experimental Validation

To preliminarily validate the proposed blockchain architecture, the modular smart contract system outlined in Algorithm 1 was implemented and tested, including the *FilterContract*, *VotingContract*, and *TokenContract*. Experiments were conducted using the Hardhat Ethereum development environment, deploying Solidity-based smart contracts on an Ethereum Virtual Machine (EVM) emulator. All simulations were executed under a modest computer environment hosted by Ubuntu Linux 20.04 LTS system, powered by an Intel Core i3 processor and 8 GB of RAM. Additionally, to thoroughly emulate multiple interacting vehicles, the Hardhat environment was configured to feature an increased number of default accounts to carry out an extensive simulation. The simulation involved scripting in JavaScript to automate testing, thus ensuring repeatability and reproducibility. Notably, deploying the simulation on an EVM-based emulator allowed the immediate transferability of validated smart contracts to any compatible Ethereum-like blockchain network, such as an Ethereum public testnet or Avalanche Fuji public testnet.

Specifically, our simulations successfully demonstrated on-chain dynamic threshold updates in accordance with the mathematical formulation defined in Section IV. As illustrated in the Hardhat simulation output in Fig. 2, test `I` verifies the correct deployment of all smart contracts and confirms that the initial reward token balance for the proposer account `0xf[...]226` is zero. Test `II` focuses on duplicate submission handling: the same submission –e.g., an identical JSON payload– is sent from a different proposer account. In this scenario, the smart contract identifies the duplicate event, but (since the submission originates from a distinct account) it automatically counts this as an affirmative vote for the initial proposal. Test `III` evaluates the dynamic acceptance threshold, confirming that it properly



```
DYNAMIC THRESHOLD VOTING AND REPUTATION MECHANISM SIMULATION
  I) Deployment and Proposal Initiation
✓ Proposer: 0xf[...]266
✓ Initial token balance: 0.0
     ✓ New proposal should be stored on-chain with status Pending
   II) Duplicate Submission Handling
✓ Duplicate submission processed as vote from account1. yes: 1
     ✓ Duplicate submission from a different account should count
 as an affirmative vote
  III) Dynamic Threshold Behavior
Dynamic Threshold Test: Proposal submitted. Proposal ID: 0x6[...]
Initial dynamic threshold (norm.): 14
Dynamic threshold after 6 hours (norm.): 5
     ✓ Dynamic threshold should decrease gradually with time
  IV) Proposals Lifecycle and Reputation Update
First Proposal *Initial* Dynamic Threshold: 14 votes
     ✓ The First Proposal should be rejected
Second Proposal *Initial* Dynamic Threshold: 14 votes
Proposer Reputation After Second Proposal: 0.5
     ✓ The Second Proposal should be rejected to trigger a reputa
tion penalty
Third Proposal *Initial* Dynamic Threshold: 15 votes
✓ Proposer token balance increased by: 1.0 tokens
Proposer Reputation After Third Proposal: 1.0
     ✓ The Third Proposal should be confirmed and reward the prop
oser
```

Fig. 2. Excerpt from the Hardhat simulation logs

decreases over time as expected, and test IV demonstrates the dynamic threshold's sensitivity to the proposer's reputation. Here, the same proposer submits two proposals characterized by different parameters, each exerting the same weight on the dynamic threshold, both of which are rejected. Notably, the threshold value for both proposals remains unchanged, as the flag mechanism described in Section IV-C is activated after the first rejection. However, the proposer's reputation score is decreased upon the second consecutive rejection. As a result, when the same proposer submits a third proposal, the initial dynamic threshold increases from 14 to 15 votes. Following the acceptance of this third proposal, the reward token balance for the proposer is incremented by one unit, and the proposer's reputation is restored to its default value. The code behind these tests is publicly available in the GitHub repository [18]. All experimentations and validations confirmed the practical feasibility and functionality of the designed system architecture.

VI. Conclusions

This paper presents a novel decentralized, blockchain-based framework designed to facilitate the secure and trustworthy exchange of sensor data among autonomous vehicles produced by diverse manufacturers. By employing a carefully designed modular smart contract architecture –comprising distinct filtering, consensus voting, and reward modules– long-standing challenges related to interoperability, validation, and incentivization in Intelligent Transportation Systems are effectively addressed. Specifically, our proposed approach leverages smart contracts, reputation-based voting mechanisms, and blockchain-based tokenization for economically incentivizing authenticated real-time data exchanges, thus significantly enhancing cross-brand cooperation.

The key of the proposed contribution is the *dynamic threshold model*, formulated to support real-time decision-making processes. The model incorporates multiple parameters, including event severity, road segment criticality, elapsed time, and submitter reputation, thereby enabling agile responses to critical infrastructural changes while mitigating adversarial input from malicious entities. A simulation using Hardhat, a widely adopted smart contract testing platform, was conducted to evaluate the approach's effectiveness and practical performance rigorously.

Simulation results validate the efficacy and robustness of our proposed framework. Specifically, the implemented voting mechanism efficiently adapted the consensus conditions to varying contextual parameters, accurately adjusting acceptance thresholds. Furthermore, the smart contracts adequately managed proposal processing, validation, and reward token issuing, underscoring the system's operational feasibility and potential scalability.

Future work will implement realistic traffic simulations and optimize threshold coefficients through adaptive multi-objective optimization approaches.

References

[1] D. Das, S. Banerjee, P. Chatterjee, U. Ghosh, and U. Biswas, "Blockchain for intelligent transportation systems: Applications, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 10, pp. 18961–18970, Nov 2023.

[2] V. Rajkumar, E. Kavitha, E. Ranjith, and R. Aruna Kirithika, "Apco-blockchain integration for data trust and congestion control in vehicular networks," *Telecommunication Systems*, vol. 88, p. 15, jan 2025.

[3] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 8857–8867, July 2022.

[4] R. Liu, W. Duan, A. M. Mangini, and M. P. Fanti, "K-protection of global secret in discrete event systems using supervisor control," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2832–2837, Oct 2023.

[5] R. Liu, Y. Hu, A. M. Mangini, and M. P. Fanti, "K-corruption intermittent attacks for violating the codiagnosability," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, pp. 159–172, January 2025.

[6] D. S. Sarwatt, Y. Lin, J. Ding, Y. Sun, and H. Ning, "Metaverse for intelligent transportation systems (its): A comprehensive review of technologies, applications, implications, challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, pp. 6290–6308, July 2024.

[7] N. Alherimi, A. Saihi, and M. Ben-Daya, "A systematic review of optimization approaches employed in digital warehousing transformation," *IEEE Access*, vol. 12, pp. 145809–145831, 2024.

[8] S.-K. Kim and H. C. Vong, "Secured network architectures based on blockchain technologies: A systematic review," *ACM Comput. Surv.*, vol. 57, Feb. 2025.

[9] T. Vairam and M. Srijeimathy, "Investigation of blockchain for security and transparency in intelligent transportation systems," *Procedia Computer Science*, vol. 252, pp. 851–861, 2025. 4th International Conference on Evolutionary Computing and Mobile Sustainable Networks.

[10] J. Yang, Q. Ni, G. Luo, Q. Cheng, L. Oukhellou, and S. Han, "A trustworthy internet of vehicles: The dao to safe, secure, and collaborative autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, pp. 4678–4681, Dec 2023.

[11] J. Rasool and S. Gupta, "Decentralised autonomous organisation based ecosystem structure for commercial companies and organisations," in *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)*, pp. 212–220, June 2024.

[12] H. Qin, Y. Tan, Y. Chen, W. Ren, and K.-K. R. Choo, "Tribodes: A tri-blockchain-based detection and sharing scheme for dangerous road condition information in internet of vehicles," *IEEE Internet of Things Journal*, vol. 11, pp. 3563–3577, Jan 2024.

[13] Freepik Company, S.L., "Icons from Flaticon used in this work." https://www.flaticon.com, 2025. Accessed: 22 January 2025.

[14] G. Olivieri, G. Volpe, A. M. Mangini, and M. Pia Fanti, "Enhancing intersection identification for autonomous vehicles: A hash-based approach," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 700–705, July 2024.

[15] K. Sekniqi, D. Laine, S. Buttolph, and E. Gün Sirer, "Avalanche platform," tech. rep., avalabs.org, 2020.

[16] P. Robinson and R. Ramesh, "Layer 2 atomic cross-blockchain function calls," *CoRR*, vol. abs/2005.09790, 2020.

[17] N. Case, "The evolution of trust," 2017. Accessed: 22 January 2025.

[18] G. Olivieri, "Hardhat simulation repository." https://github.com/GSEPE/BC-AV-dataSharing-hardhat.git. Accessed: 5 March 2025.