electronics

MDPI

Article

A Proof-of-Multiple-State Consensus Mechanism for Mobile Nodes in Internet of Vehicles

Feng Zhao [1], Ruimin Cheng [2], Chunhai Li [1,*], Zhaoyu Su [2], Guoling Liang [2] and Changsong Yang [3]

[1] Guangxi Engineering Research Center of Industrial Internet Security and Blockchain, Guilin University of Electronic Technology, Guilin 541004, China; zhaofeng@guet.edu.cn
[2] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China; ruimincheng@mails.guet.edu.cn (R.C.);19022201001@mails.guet.edu.cn (Z.S.); guolliang@mails.guet.edu.cn (G.L.)
[3] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; csyang@guet.edu.cn
* Correspondence: chunhaili@guet.edu.cn

Abstract: Blockchain technology provides a reliable information access environment for the Internet of Vehicles, but the high latency and complex computing consensus mechanism in blockchain make it difficult to port to onboard devices. Recently, there are many methods to reduce the time cost of consensus by optimizing node grouping or reducing redundant calculations, but this would lower the security level of the blockchain. To address these issues and reduce the adverse effects of frequently changing channel quality on consensus results, a consensus mechanism based on vehicle comprehensive state factors for nodes selection (PoMS) is proposed. Firstly, the vehicle nodes utilize the machine learning model to predict local driving parameters and broadcast the predicted results to the other nodes. Secondly, each node uses interactive data to calculate the state values, and the leader comprehensively evaluates the nodes participating in the consensus and selects the nodes as relays. Finally, we also adopted a double-layer blockchain structure to accelerate the selection process of relay nodes. In order to verify the performance of the proposed consensus algorithm, we conducted tests on transmission time and communication quality. The experimental results show that compared to traditional consensus mechanisms, the algorithm proposed in this paper can reduce time overhead by an average of 12.7% and maintain a good transmission rates under a certain number of malicious nodes.

Keywords: Internet of Vehicles; blockchain; consensus mechanism; node selection

Citation: Zhao, F.; Cheng, R.; Li, C.; Su, Z.; Liang, G.; Yang, C. A Proof-of-Multiple-State Consensus Mechanism for Mobile Nodes in Internet of Vehicles. *Electronics* **2024**, *13*, 1553. https://doi.org/10.3390/electronics13081553

Academic Editor: Mehdi Sookhak

Received: 26 March 2024
Revised: 13 April 2024
Accepted: 17 April 2024
Published: 19 April 2024

1. Introduction

The rapid development of the automotive industry and the Internet of Vehicles (IoV) has escalated concerns regarding the security of vehicle networks [1]. A primary benefit of the Internet of Vehicles (IoV) is the enhancement of traffic safety and efficiency, notably through real-time vehicle information sharing. While most vehicle information requires sharing only among nearby vehicles, certain types (e.g., announcements) demand broader dissemination. However, due to the extensive user privacy contained in these data, they are easily leaked during storage and distribution, posing significant risks to users. The presence of data security and privacy protection issues [2] presents numerous difficulties and challenges for the IoV messaging process. Given the emergence of blockchain technology, with its benefits of decentralization, immutability, and transparency [3], well-suited to the IoV complex scenarios, it can alleviate various security issues in the transmission and distribution of in-vehicle data, thereby transforming the IoV traditional centralized architecture and gaining widespread application within IoV.

In recent years, blockchain technology, particularly its consensus mechanism [4], has emerged as a significant research focus, attracting considerable scholarly interest.

Blockchain technology integrates various technologies, including P2P networks, smart contracts, consensus mechanisms, and cryptography, culminating in an innovative form of distributed ledger technology. Its decentralized, tamper-proof, traceable, and transparent characteristics afford it vast application potential in the Internet of Vehicles (IoV), particularly in addressing issues of centralization, data security, and privacy protection. Initially, blockchains employ a Proof-of-Work (PoW) mechanism, which necessitated extensive computations among nodes for transaction verification.

Despite guaranteeing network security, the mechanism's high computational resource consumption makes it difficult to implement in actual scenarios with a range of computational requirements. Later suggestions focus on the PBFT algorithm to address Byzantine fault tolerance problems and Proof-of-Stake (PoS) consensus mechanisms and their improvements [5,6]. In addition, the IoV requirement for low latency and the arithmetic bottlenecks present in vehicles limit the wider application of traditional consensus algorithms within the IoV. These traditional algorithms, despite their advancements, have limitations. First, they are designed for cryptocurrency protection and the impact of mobile nodes' dynamic characteristics is ignored in the process of consensus.

Building on the existing Proof-of-Work consensus algorithms, this paper presents an improved message transmission model for vehicular networks. The goal of the model is to ensure transmission security and stability while addressing the deterioration of communication quality brought on by fast nodes mobility in vehicular networks. It makes use of a consensus mechanism that is based on Quality of Service (PoMS). The network is initially divided into different service areas based on communication states, and nodes are first loosely clustered together. Then, using the obtained data to determine its future travel conditions, a trained machine learning model forecasts the vehicle's movement state. This method improves the efficiency and dependability of network communication by having the model determine the best transmission path. The main contributions of this article can be summarized as follows:

(1) We achieve consensus in different partitions and manage it uniformly through a two-layer chain to speed up the consensus process.
(2) We have developed a consensus on speed change factors for mobile nodes in the network so that they can adapt to dynamic networks.
(3) Based on this, we propose the PoMS consensus algorithm that takes multiple factors into account to evaluate vehicle levels and filter out better relay nodes based on this. The experiment shows that our method improves efficiency and security.

2. Related Work

In recent years, there has been an increasing integration of blockchain into the Internet of Vehicles (IoV), with the aim of considering complex vehicle scenarios and improving the provision of blockchain services in the IoV. This has led researchers to propose new consensus algorithms [7–9]. Currently, blockchain-based IoV models [10] are on the rise, leveraging decentralization, immutability, and transparency to transform traditional centralized IoV architectures. This widespread application offers significant development prospects for the IoV. Currently, significant research efforts are aimed at refining the blockchain's consensus mechanism to eliminate inefficiencies and resource waste of traditional consensus methods. Various consensus algorithms are developed that are tailored to different scenarios [11,12] and improve communication efficiency over classic consensus mechanisms.

Existing consensus mechanisms mostly refine and improve classic schemes such as Proof-of-Work (PoW) [13] and Byzantine Fault Tolerance (PBFT) [14]. The PoW algorithm requires users to perform time-consuming and complex calculations, which incurs significant costs in terms of time, equipment, and energy. The PBFT algorithm resolves node breakdown in asynchronous environments caused by malicious attacks and software errors. Current algorithms achieve a fault tolerance of $(n - 1)/3$, ensuring liveliness and security. In [15], an improved DPoS algorithm is proposed that provides higher throughput and a two-layer blockchain structure for improved consensus efficiency and scalability. In

addition, there are also many similar approaches to [16], which combine multiple traditional consensus mechanisms to create new consensus mechanisms. However, these classic improvement schemes tied to the static node environment have problems in systems with frequently changing node communication quality. In [17], a novel lightweight PoBT (Proof of Block and Transaction) algorithm and integration framework for the IoT blockchain is proposed, which can verify transactions and blocks with reduced computation time. Literature [18] introduces Proof of Credit (PoC), a blockchain protocol that focuses on fairness and includes a self-verification and hybrid incentive mechanism that is resistant to double-spend and self-serving mining attacks. However, the protocol oversimplifies the Proof-of-Work process to effectively select the leader nodes. Ref. [19] minimizes consensus size and network overhead by dividing nodes into communication groups and assigning node roles based on trust factors, thereby optimizing communication resources. An efficient consensus algorithm is proposed in paper [20] to reduce the communication overhead, where the leader dynamically adjusts the reputation values of nodes according to their behavior and allows high-performing nodes to transmit data efficiently. This approach significantly improves the responsiveness of the blockchain, especially in large node environments. While these consensus mechanisms optimize the dynamic attributes of the network, they primarily address the communication overhead between nodes without considering the continuous mobility of nodes. The EPoW consensus introduced in [21] shortens the waiting time for arithmetic mining by executing the algorithm regularly. It also improves consensus efficiency by managing a specific set of miners and assigning multiple accounting rights. These methods prioritize changing reputation scores and aim to limit malicious nodes within the network. However, they do not sufficiently consider the communication process between nodes.

The environments assumed by existing consensus algorithms are predominantly static and immutable. Therefore, this paper focuses on the specific requirements of IoV scenarios, especially with regard to mobile node communication. Based on various consensus schemes from previous research, we improve these models and propose a special consensus algorithm that aims to improve data quality and communication efficiency with a focus on data security during the transmission process.

3. Proposed System Model

3.1. Architecture

The mobile vehicles in the system form an in vehicle self-organizing network (VANET), which uses relays to complete message transmission. This decentralized network structure is very compatible with blockchain. Considering the dynamic characteristics of the network, communication between vehicles is achieved through the Dedicated Short Range Communication (DSRC) protocol.

This paper models the rapid mobility of numerous vehicles within the IoV scenario, with the comprehensive structure of the proposed vehicle messaging model shown in Figure 1. The model includes an edge layer with mobile vehicles and roadside units (RSUs), a blockchain layer with selected RSUs and a trusted third party (Third Party Trusted Authority (TA).

Mobile Vehicles: In this model, mobile vehicles act as primary data generators, continuously producing and transmitting significant amounts of data. Every vehicle has an on-board unit (OBU) with different computing power and storage capacity. Vehicles continuously collect data about their position, speed and direction of travel and send it to nearby vehicles and RSUs, creating the basic communication infrastructure of the IoV network. This information is not only fundamental to communication, but also serves as a crucial parameter in the consensus mechanism that assigns points to vehicles for leader node selection based on their travel status. Additionally, each vehicle is equipped with a Transient Protection Device (TPD) to store important key and identity information provided by a trusted third party.

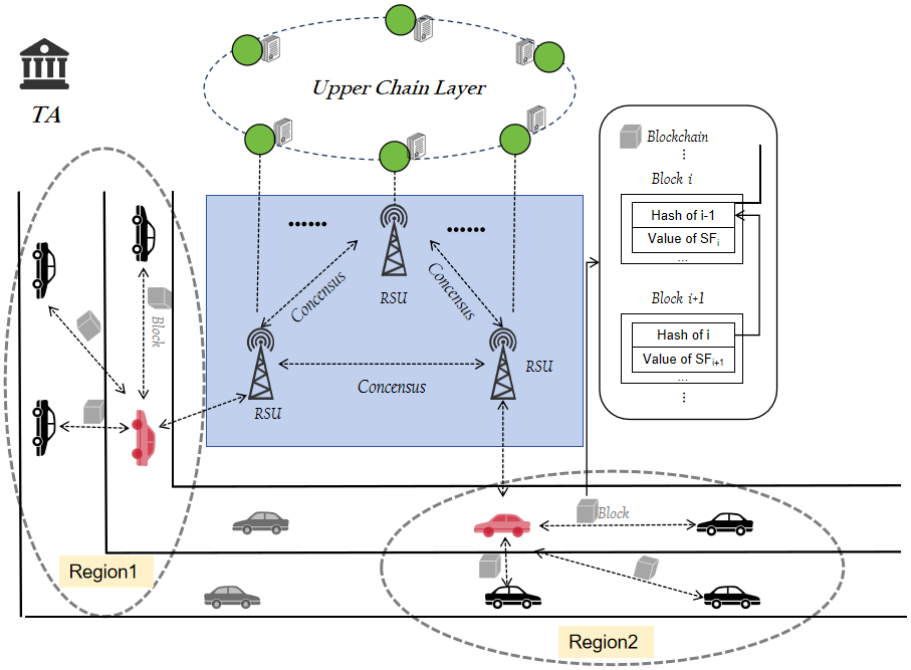

Figure 1. System Architecture.

Roadside Unit (RSU): It is arranged at a certain density on both sides of the road and maintains communication with the vehicle via the DSRC protocol commonly used in the IoV. The RSU receives and checks various information sent by the vehicle. Once validated, RSUs forward the information to the leading node, package it and upload it to the blockchain ledger. Meanwhile, RSUs communicate with each other via a wired connection.

Trusted Third Party: The Trusted Third Party (TA) is responsible for the identity and key management of the entire blockchain and is usually considered completely trustworthy and free from any malicious behavior. TA has powerful computing and storage capacities and takes over the daily work. Today's operation and maintenance of the blockchain. TA is also responsible for initializing the blockchain network, generating the system parameters, and requesting all joining vehicles to complete the registration of their identity information. Vehicles wishing to join the blockchain must first submit a registration application to TA, and only those vehicles that pass the application can join the blockchain.

Blockchain: We use a public blockchain as the decentralized underlying architecture to instantiate VANET in our scheme. The RSU validates messages based on a consensus algorithm and registered vehicles can access the blockchain via RSUS to obtain the necessary traffic flow data [22].

3.2. Implementation Process

The vehicles in the model travel on the road at different speeds depending on the predetermined density of the area. The driving process of the vehicles regularly sends their own driving data, including driving direction, driving speed, coordinate position, etc., to the RSU in the sub-region, where the RSU has larger storage space and stronger computing power of the collected vehicle data to calculate In the next action, the vehicles, which are still within the communication range of the RSU within a certain period of time, are included in the group. The schematic diagram of regional division is shown in Figure 2, where vehicles of the same color represent being within the jurisdiction of the same RSU.

In order to improve message forwarding speed, this article chooses a special grouping method based on relative position information and travel speed prediction. This method means that vehicles with similar driving status in a specific area are classified as the same group. After completing the first round of message transmission, RSU nodes collect position information and speed information for all vehicles involved in communication,

and each individual RSU selects a set of vehicle nodes that meet the conditions for sending the grouping identifier in close proximity and the vehicles with the same identifier are recognized by the network as the same group in the subsequent consensus process. The execution process of the consensus algorithm in each round is shown in Figure 3.

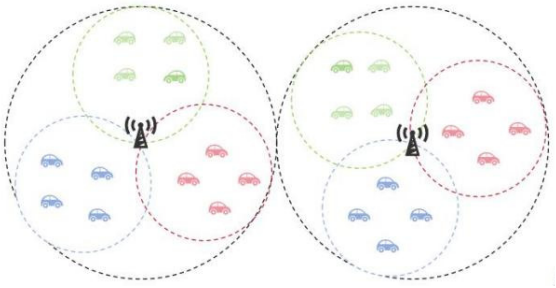

Figure 2. Schematic diagram of grouping.

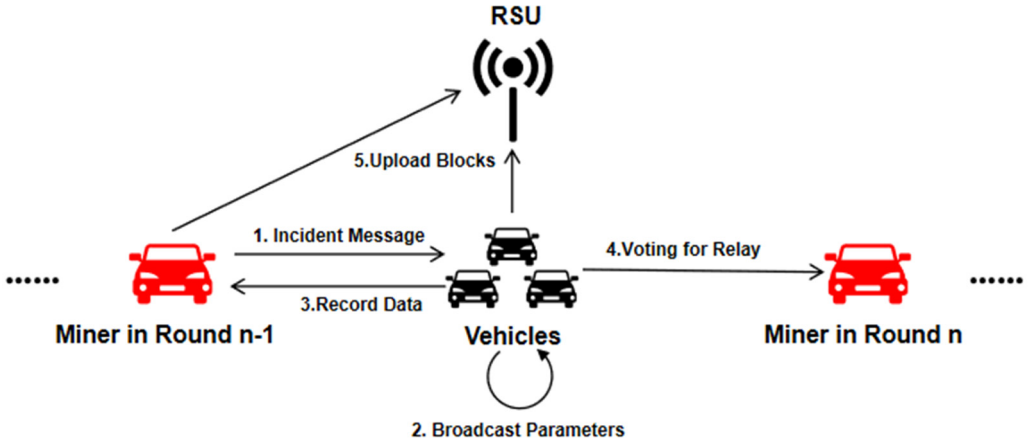

Figure 3. Consensus implementation.

After completing a certain number of rounds of consensus, the state of vehicle movement has changed significantly compared to that before consensus, and at this point the RSU needs to regroup the nodes again using the collected vehicle information. The regrouping process is similar to the initial grouping, in which the RSU removes the vehicles that have left or are about to leave a certain area from the previous group and the newly entered vehicle nodes that meet the entry criteria for the group into the Partitioned network involves participating in the subsequent consensus.

4. Consensus Mechanism Based on Integrated State

4.1. Speed Prediction Models

The vehicle speed prediction method used in this paper is based on the Long Short-Term Memory (LSTM) network model [23], which aims to predict short-term vehicle speed. The model's training data comes from the simulation environment, which includes the state data of several vehicle nodes at different points in time, such as vehicle speed, acceleration and road vehicle density information at any time. The model focuses on short-term speed predictions rather than long-term macro predictions because the simulation environment cannot fully reproduce the special factors in real-world conditions such as weather changes and holidays. This short-term prediction method can more accurately reflect the real situation in the simulation environment. Vehicle the driving speeds over a historical period are $\left[v_i^1, v_i^2, \cdots, v_i^n\right]$, and the prediction model obtains the speed in the future Δt speed after a certain period of time, which will be the basis for subsequent state value calculations $\left[v_i^{1+\Delta t}, v_i^{1+\Delta t}, \cdots, v_i^{1+\Delta t}\right]$. The outputs of these models describe the possible driving state of a specific vehicle i in the future, rather than the average speed of all vehicles in the

network. This is due to the relatively short monitoring time for changes in vehicle status. We are concerned about the movement mode of each vehicle node rather than the overall state changes of traffic flow. This will be an important parameter for subsequent state value calculations.

4.2. PoMS Consensus Algorithm

The flow of this consensus algorithm is as follows: Given the inefficiencies inherent in the traditional proof-of-work consensus algorithm, especially its wasteful use of system resources, this study introduces a modified proof-based consensus mechanism. This adjustment replaces the computationally intensive hash value verification with a scenario-adapted, comprehensive quality score verification process.

(1) Initialization: This study establishes a blockchain-based vehicular mobility network, described as an open, distributed peer-to-peer network in the broader vehicular network context. It denotes the total number of vehicle nodes in the network as n, where each node potentially becomes a ledger-maintaining node through competitive rotation. The total number of Roadside Unit (RSU) nodes is represented by m, where each RSU is part of a smaller network and maintains a local blockchain record.

(2) Mobility Process: Once the vehicle nodes enter the mobile stat, RSU nodes start sending messages continuously. Vehicles that receive broadcasts from RSUs respond by initiating mining activities. Using reliable on-board sensors, the vehicle collects position, speed and additional data, uses a pre-trained network model and inputs travel information to predict its state over an upcoming period, and then logs the predicted state data. The vehicle calculates its value, summarizes these findings in a block, documents essential parameters and forwards this to the mining node to complete the uplink and storage processes.

(3) Relay Nodes Selection: After receiving the position information and state prediction information of each vehicle, the RSU nodes begin to analyze all values for analysis. The time taken by each node to submit the block and the calculated value show a reverse trend, such as:

$$t_w = \frac{\tau}{SF} \tag{1}$$

where the SF represents a special value, State-Factors, which will be explained in the next section, and the parameters τ are used to ensure that the waiting time is within reasonable limits. Nodes with better channel states can transmit blocks earlier, reducing the waiting time for information transmission. At the same time, differentiate the period during which different nodes transmit blocks to prevent multiple nodes from initiating communication requests at the same time, causing transmission conflicts, and to avoid the calculations becoming untrustworthy.

(4) Messages Delivery: After the relay node is selected, the message is delivered through the chosen node, repeating the selection process throughout the delivery phase until the message count reaches the system-defined threshold. The threshold specifies both the required number of messages for delivery and the maximum hop count for message forwarding. It is determined by the road department based on the different communication requirements.

4.3. State-Factors Calculations

The SF value indicates the channel state and relative position of vehicle node i at time t, signifying its capability to relay messages. Nodes periodically send beacon messages to collect mutual position and velocity data. A relative condition value is calculated from this data, which serves as a crucial reference for the integrated quality factor. The prediction interval's time point is set with the current time denoted as t. Vehicle nodes employ a pre-trained learning model to predict future states. Each node inputs its coordinates, speed, and direction into the model, which then calculates the state information for the moment $t2$. The node forwards the result to the RSU for storage and management. The nodes forward the result to the RSU for storage and management. The RSU evaluates the status

of the vehicle nodes at $t2$, regroups those within its jurisdiction, and excludes nodes out of communication range. Within each blockchain network group, nodes calculate their values and record them in blocks for transmission to the mining node.

The calculation is divided into three parts: (1) the difference between the current driving state of the vehicle and the predicted driving state; (2) the probability that the vehicle successfully sends a message to a neighboring node, the channel status; (3) the computing power of the integrated CPU which that their ability to process the messages and status data received. The calculation of these three factors is described below:

(1) $P(diff_i)$: This value indicates the probability that the message delivery at time $t2$ exceeds the range threshold, effectively preventing delivery errors due to longer distances caused by nonisotropic vehicle movements, which can lead to excessive separation of relay nodes. Assuming that the position of vehicle node i when in $t2$ is $P_i^{t2} = (X_i^{t2}, Y_i^{t2})$, for the distance difference between node i and node j after the time interval Δt is expressed as:

$$D_{i,j}^{\Delta t} = \left((X_i^{t2} - X_j^{t2})^2 + (Y_i^{t2} - Y_j^{t2})^2 \right)^{1/2} - D_{i,j}^{t1} \tag{2}$$

This study focuses on short-term forecasting, particularly on predicting vehicle movement within a few seconds. Consequently, we approximate the speed change of a single vehicle node over a short period of time as a uniform acceleration, a model that is generally consistent with the scenario. By expressing the velocity of node i at the $t2$ moment in time is denoted as v_i^{t2} and the velocity of node j is denoted as v_j^{t2}, then the vehicle's relative movement in the horizontal direction during the time interval Δt. The relative distance travelled in the horizontal direction during the time interval can be expressed as:

$$D_{ijx}^{\Delta t} = \left(\left(v_{ix}^{t2} - v_{jx}^{t2} \right) - \left(v_{ix}^{t1} - v_{jx}^{t1} \right) \right) \cdot \frac{\Delta t}{2} \tag{3}$$

The longitudinal axes are similar, so by synthesising them the relative distances between vehicle i and vehicle j can be varied as follows:

$$D_{i,j}^{\Delta t} = \sqrt{(D_{ijx}^{\Delta t})^2 + (D_{ijy}^{\Delta t})^2} \tag{4}$$

However, the relative distance does not directly reflect the probability of successful communication between vehicles, and the following function is used to reflect the relative relationship between distance and communication range:

$$P_{dis} = 1 - \frac{D_{i,j}^{\Delta t}}{R} \tag{5}$$

As the proximity between two vehicles increases, P_{dis} approaches 1, suggesting a higher likelihood of successful single communication. Conversely, a greater distance results in a negative P_{dis}, diminishing the competitiveness of the nodes that are further away.

Considering the difference in the vehicle's travel direction, we add an additional speed factor to the range factor, which ensures that the more vehicle nodes there are that maintain the same travel direction as node i, the higher the probability that the node will win the competition and become a relay node. In this article we define the speed factor as:

$$P_{velocity}^{i,j} = 1 - \alpha \frac{|v1 - v2|}{|v1| + |v2|} \tag{6}$$

where α is the weight of the adjustment factor weights, and when this value is larger, the greater the effect of the speed factor on $P(diff_i)$ produces, the greater the effect it has, and vice versa, the weaker it is. For each vehicle's communication range we define as R, so the probability that vehicle i and vehicle j are still in communication range after time period Δt

can be expressed as $P_{range}^{i,j} = \frac{\Delta D_{i,j}^{t2}}{R}$. It can be obtained that the value of $P\ (diff_i\ )$ between node i and node j at the moment $t2$ is expressed as:

$$P\left(diff_{i,j}\right) = P_{dis} \cdot P_{velocity}^{i,j} \tag{7}$$

And for a single vehicle node i, there is

$$P(diff_i) = \sum_{j=1,j\neq i}^{n_{range}} P\left(diff_{i,j}\right) \tag{8}$$

n_{range} indicates the number of communicable mobile vehicles within the management area of the RSU to which the vehicle nodes belong.

(2) $P(SINR_i)$: If node i is selected as a relay node, its channel quality at the moment $t2$ is expressed as:

$$SINR_{i,j} = \frac{\left(D_{i,j}\right)^{-\gamma}}{\sum_{k=1,k\neq i}^{n_{range}} \left(D_{j,k}\right)^{-\gamma} + P_n} \tag{9}$$

where $D_{i,j}$ is the same as above, $D_{j,k}$ denotes the distance between node j and the remaining interfering node k, while γ is the path loss exponent and P_n denotes the noise power. To ensure that the channel quality meets the message delivery requirements, we set a threshold for the SINR value. If it is higher than this threshold, it means that the message can be transmitted smoothly. That is, the term can be written as:

$$(SINR_i > \beta) = \left(\frac{\left(D_{i,j}\right)^{-\gamma}}{\sum_{k=1,k\neq i}^{n_{range}} \left(D_{j,k}\right)^{-\gamma} + P_n} \geq \beta \right) \tag{10}$$

A transformation is performed to obtain:

$$(SINR_i > \beta) = \left(D_{i,j} \leq \left(\beta \left(\sum_{k=1,k\neq i}^{n_{range}} \left(D_{j,k}\right)^{-\gamma} + P_n \right) \right)^{-\frac{1}{\gamma}} \right) \tag{11}$$

Within a packet network, every time after exchanging information about all vehicles in a group through broadcast messages, vehicle i that wants to compete for a relay node calculates its own SINR value based on different vehicle position differences and finally aggregates them. This process can be represented as follows:

$$f\left(SINR_{i,j}\right) = \begin{cases} 1,\ if\ D_{i,j} \leq \left(\beta \left(\sum_{k=1,k\neq i}^{n_{range}} \left(D_{j,k}\right)^{-\gamma} + P_n \right) \right)^{-\frac{1}{\gamma}} \\ 0,\ otherwise \end{cases} \tag{12}$$

Eventually, there is:

$$P(SINR_i) = \frac{\sum_{j=1,i\neq j}^{n_{range}} f\left(SINR_{i,j}\right)}{n_{range}} \tag{13}$$

(3) $P(Com_i)$: The value in question is dictated by the computational capability of the onboard processor, with those possessing higher arithmetic capacities enjoying a significant advantage. Given the different instruction set architectures and hardware components of CPUs, and given that a significant portion of a processor's computing capacity is dedicated to performing prediction tasks, the experiments aim to use program execution time as a metric to measure performance differences among CPUs. The main frequency of the processor, denoted by f, is a critical parameter that significantly influences its computing performance. In addition, the average number of instruction cycles is written as N_{avg} and

the number of instructions to be executed under different architectures is denoted as N_{num}, we can get the time spent by the CPU to execute the programme:

$$T_{exc} = \frac{N_{avg} \cdot N_{num}}{f} \tag{14}$$

The time taken by different vehicle nodes to run the programme is rightfully different, and nodes that can perform tasks faster will be more likely to be selected as relay nodes. In order to facilitate a subsequent more intuitive comparison of the difference in SF_i values between different nodes, this paper sets an acceptable maximum value of the running time, T_{max}, and when a node is unable to end the procedure within this time limit, it is considered that the node's computational capability does not satisfy the requirements, and it is out of the running for the miner node. Therefore, there are:

$$P(Com_i) = \frac{T_{exc}}{T_{max}} \tag{15}$$

Thus, the final SF_i value can be obtained:

$$SF_i = P(diff_i) \cdot P(SINR_i) \cdot P(Com_i) \tag{16}$$

Due to the constraints imposed on the parameters, both the first and third terms clearly fall within the range (0, 1). The second term, which represents a summation, spans the interval [0, nrange]. Consequently, the overall value range for is set to [0, nrange).

4.4. Voting Options

After calculating the value, the node initiates a local vote among the remaining nodes. Only when the number of votes inclined towards "agree" exceeds the threshold of votes, will the node be considered an honest node. Verified vehicles can package messages and send them to the miner node to participate in the next round of relay node competition, and such messages are marked as valid messages. Conversely, for vehicles below this threshold, the nodes vote "reject", meaning communication is not possible or denied. A node is considered honest and can only forward its packet to the miner node for the next relay node election if it receives a majority of "acceptance" votes, marking its messages as valid. Conversely, nodes that do not receive enough approval votes are considered malicious for that consensus round, their messages are considered invalid, and ignored by the miner nodes as harmful to the network. After enough valid messages are collected in a consensus round, miner nodes rank these messages according to their values. The node with the highest value is selected as the next relay node. If two nodes i and j have identical SF values, the miner node compares their acceptance votes. The node with more acceptance votes is considered more capable of message delivery and is therefore more suitable as a forwarding node. If the miner node fails to collect enough valid messages within the allotted time, it ends the current round of consensus and requests other nodes to begin the next round.

5. Security Analysis

5.1. Message Validation

After each message is received, it needs to be verified by the miner node [24]. The inability to verify can be classified into two cases: (a) for correct messages verified as error messages; (b) for error messages verified as correct messages. Suppose the percentage of malicious nodes among all nodes is p, and the malicious nodes always take the behaviour of initiating an error message or giving a malicious voting tendency. Thus the probability of generating an error message is p, the probability of generating a correct message is $1 - p$, the probability that a message cannot be verified in the system is denoted as:

$$F = p_m \cdot Pr(n_m \geq n_{vote}) + [1 - (1 - p_m) \cdot Pr(n_h \geq n_{vote})] \tag{17}$$

The former represents the n_m the probability that a malicious node casts enough votes for the wrong message to be authenticated as correct, and the latter denotes the probability that a miner node does not receive a sufficient number of votes such that the correct message cannot be successfully authenticated. And from Chernoff's inequality:

$$Pr(n \geq n_{vote}) \leq e^{\frac{-(n_{vote}-\mu_n)^2}{(\mu_n+n_{vote})}} \tag{18}$$

where n can denote the number of malicious or honest nodes, and μ_n denotes the expected value of the corresponding variable. It can be seen that with the same percentage of malicious nodes, due to the monotonically decreasing nature of the function, we are able to continuously reduce the probability of verification failure by increasing the voting threshold.

5.2. Attack Methods

The consensus mechanism works within a federated blockchain and requires a license from all participating nodes. In this framework, the vehicles communicate exclusively with other nodes and vehicles in the federated blockchain. Despite these precautions, potential threats [25–27] that can disrupt the blockchain network still exist. The next section explains how the PoMS algorithm counteracts such attacks.

Double Spending Attack: The double spending attack is a widespread form of attack in blockchain technology. The PoMS consensus mechanism verifies every transaction message. For a double-spending attack to be successful, the initiator of the fraudulent message must be verified as legitimate, which means that the initiator has a number of votes that exceeds the threshold set by the system, as follows. By adjusting the value threshold for successful message validation, we can change the difficulty of successful message validation, challenging attackers who want to perform a double-spending attack. At the same time, manipulating a malicious node to obtain voting rights is not possible because the value calculation relies on predicting relative speed and position. Given the frequent changes in a vehicle's motion state within a consensus round, it is difficult to ensure that the state factor of a malicious node exceeds the state of normal nodes, so the algorithm can effectively prevent double-spending attacks.

False message attack: Malicious nodes can significantly disrupt network communications by sending false messages, including false status factors or numerous invalid message frames. The algorithm presented here encodes the sender's identity in each broadcast message frame along with a unique sequence number to identify the message, allowing RSUs to easily locate the sender vehicle. Penalties are imposed on malicious activities by reducing the network weight of the node, thereby curbing further adverse actions.

Malicious rating attack: Malicious nodes can negatively rate other nodes to reduce their weight and increase their chances of becoming a miner node. However, the consensus algorithm presented in this article includes a voting mechanism. The presence of sufficient honest nodes participating in the voting ensures the integrity of the algorithm and renders the voting of a single malicious node ineffective in changing the outcome.

Sybil attack: A hex attack is an attack in which an attacker using multiple virtual identities sends false messages in the system, compromising the security of the system. However, no matter how the malicious nodes forge the identity information and message IDs, the status factor of the message sender is calculated in the same way, and a large number of similar SF values can be easily detected by the RSU and then recorded, and the conflict mechanism is triggered, when the miner node compares the received values. When the miner nodes compare the received values, the conflict mechanism is triggered. If the relay node cannot be selected after frequent triggering of this mechanism, the consensus in this round will be reset. This effectively prevents switching.

6. Simulation Results

The simulation experiments in this paper are conducted based on the OMNET platform, incorporating the urban traffic simulation software SUMO (https://sumo.dlr.de/docs/index.html (accessed on 16 April 2024)) to complete the visualisation function of the network, to simulate the process of message passing in a complex scenario of multi-vehicle movement at a certain speed, and to analyse the performance of the proposed blockchain and PoMS consensus.

The vehicle mobility model used in this simulation is the Freeway model proposed in [28], where the vehicle is limited to the driving road and the vehicle speed relationship can be expressed as:

$$v_i(t+1) = v_i(t) + random(\,) * a_i(t) \tag{19}$$

where $a_i(t)$ represents the acceleration constant of vehicle i at time t, which is consistent with our analysis above. In our initial setup, the vehicle sends message packets every 0.1 s, and a total of approximately 2000–4000 message data will be sent during the simulation duration. Each message contains a data volume of 512 KB. For vehicle nodes that frequently send information, the simulation uses the MAC protocol to control the forwarding of message packets at the Data Link Layer, which prevents message conflicts between vehicles.

In the network simulation environment, the value of the neighbourhood node threshold nrange of the vehicles is kept below 40 in the experiments and the density of the vehicles within the whole map is kept at 100 nodes/km^2, which is set according to the derivation in the literature [29] to ensure that the vehicles can remain free to drive during the simulation without being blocked due to the over-dense traffic. The results obtained are the average of 100 simulation runs.

Figure 4 shows the probability that vehicle-generated message packets successfully reach their destination at different average movement speeds. The results show that the success rate of data transmission continues to decrease as vehicle speed increases, especially in high-speed scenarios where transmission becomes significantly more difficult. The main reason is that a vehicle traveling at high speed may leave the reachable area of the message before communication is established, resulting in packet loss. For this situation, the relative speed factor is defined in this document. Vehicles traveling at high speeds in the opposite direction receive lower campaign weights to reduce the likelihood of communication equipment failure. In low-speed scenarios, the consensus mechanism proposed in this study performs slightly better than other approaches. Especially in the high-speed scenario, the relay nodes using the PoMS algorithm perform better than PoET in literature [27] and TIA in literature [30], although the packet loss rate still inevitably increases. This is attributed to the fact that the algorithm takes into account the speed fluctuations and effectively avoids the problem that the distance between nodes exceeds the communication range due to the speed difference, thereby performing well in terms of transmission efficiency.

Figure 5 shows the effect of vehicle speed on node consensus delay. Higher speed increases the difficulty of selecting the relay node, which in turn increases the time required for the consensus process. With the same number of nodes, the PoMS algorithm is able to complete the consensus process in less time and with higher quality at different average speeds. This is due to the algorithm's prediction of speed fluctuations, which effectively resists the effects of driving state changes. The effect of speed fluctuations of mobile nodes is not considered in any of the schemes [27,30], where the consensus delay increases rapidly with higher node mobility. On the other hand, in the literature [29], a probabilistic model is used in the PoQF to simulate the speed change while combining the vehicle location information for consensus between nodes, which effectively reduces the consensus time of the vehicle in the mobile state, but the algorithm's effectiveness decreases, when the speed reaches a higher threshold. The algorithm proposed in this article takes the speed factor into account. The algorithm proposed in this article attenuates the impact of relative speed changes on consensus by taking the speed factor into account, ensuring that the future driving speeds predicted by the model are more accurate than the probabilistic predictions.

This allows PoMS to filter out better relay nodes within a fixed communication distance to ensure a more stable and reliable connection. At high driving speeds, the consensus time of PoMS is reduced by an average of 13.3% compared to the method in literature [26].

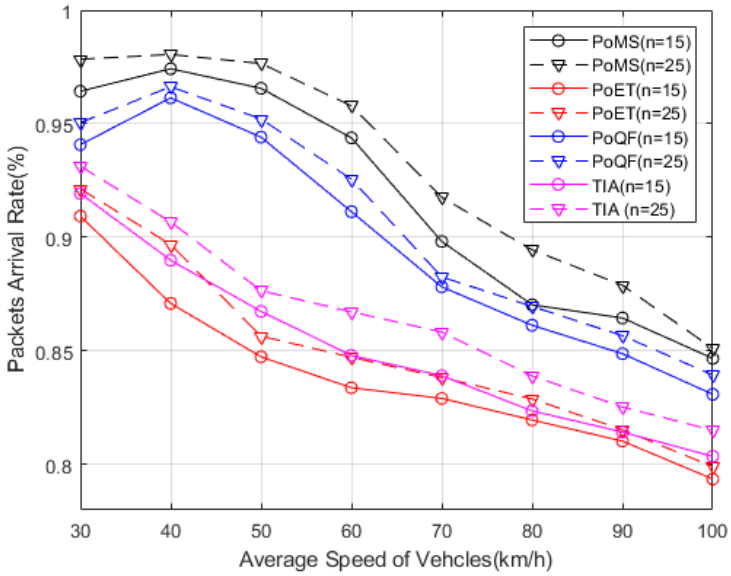

Figure 4. Success rate of message transmission at different speeds.

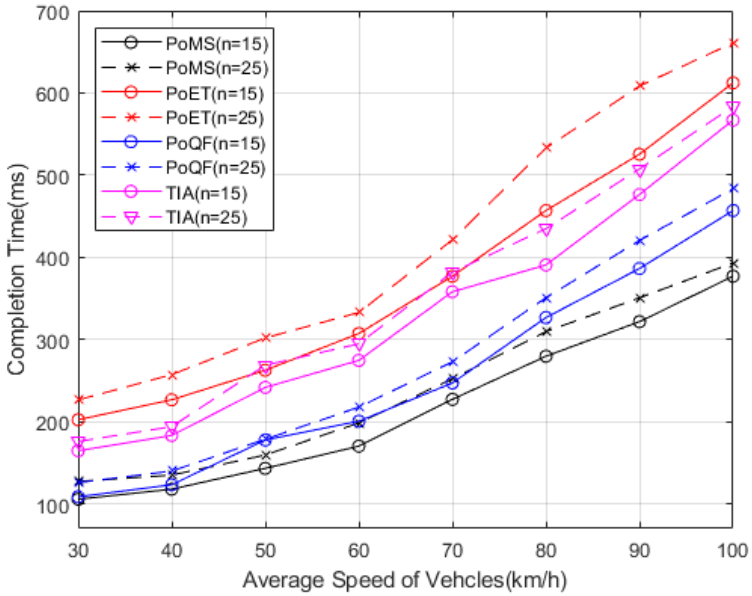

Figure 5. Consensus time at different speeds.

Figure 6 illustrates the variation in the time required for the vehicle to complete consensus for different probabilities of malicious node generation. Malicious nodes disrupt the network by generating useless packets with no information load, consuming resources, or interrupting message transmission. Therefore, the question of how to effectively identify malicious nodes as relay nodes and reduce their probability has become an important research topic. The PoMS algorithm proposed in this study is able to verify all node behaviors in the next consensus round by recording node behaviors and reporting them back to the blockchain, thereby reducing the weight of malicious nodes during the selection process. As the probability of malicious nodes increases, the increase of invalid messages in the network has a significant impact on the speed of consensus execution. In networks with less malicious nodes, the time cost in [27] is mainly the baseline waiting time, while [29]

and PoMS perform similarly; in networks containing more malicious nodes, [30] it spends more rounds to check credible nodes for the election period and the waiting time increases; The PoQF determines the normality of behavior based on the node's votes. The more malicious nodes there are, the longer it takes to get enough votes, the larger the number of malicious nodes, which increases the time consumption. and the PoMS consensus shows more efficient resistance by assigning weights based on the historical behavior of nodes. As the number of malicious nodes increases, RSU nodes can collect node behavior information and report back to miner nodes more quickly, thereby improving the anti-interference ability of the network. In a high probability malicious node environment, the consensus time required by PoMS is reduced by 9.8% on average.

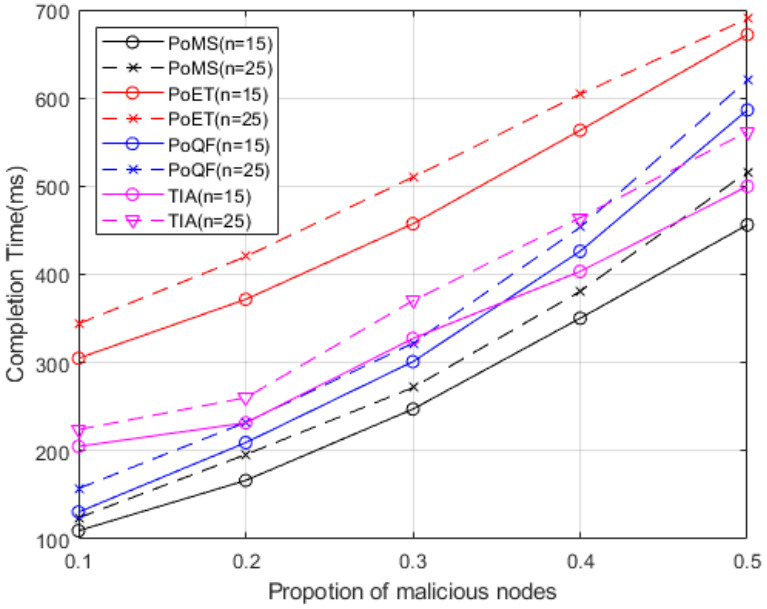

Figure 6. Consensus time for different number of malicious nodes.

Figure 7 shows the resilience of PoMS to different proportions of malicious nodes in the network. The upper blockchain of the algorithm records the behavior of vehicles in the system and dynamically adjusts the weights of nodes in the consensus process according to their performance to resist the impact of malicious nodes on the system. As the proportion of malicious nodes increases, more invalid blocks are generated in the network and the validity of miner nodes' collected values and the number of votes decrease, resulting in slower blockout. However, it can still be seen that the throughput of the blockchain proposed in this paper decreases more slowly, and in the presence of malicious nodes, PoMS consensus is still the more powerful algorithm, maintaining a more efficient blockout efficiency in all cases.

Figure 8 shows the average time spent for each additional hop of a message in the network. The number of message hops in PoMS reaches the specified threshold. The number of message hops in PoMS reaches the set threshold and it is proven that the message has been successfully transmitted to the destination and the entire network establishes a communication link. Malicious nodes increase the time it takes for miner nodes to collect enough blocks, which leads to an increase in the selection time of each relay node, thus affecting the time to complete the final round of consensus. The PoMS algorithm has a relative speed and a relative distance The calculation in the PoMS algorithm causes the miner nodes to select more distant vehicles as relay nodes as much as possible and to connect within the communication range in fewer hops during the transmission process final destination, effectively reducing the average time spent per hop.

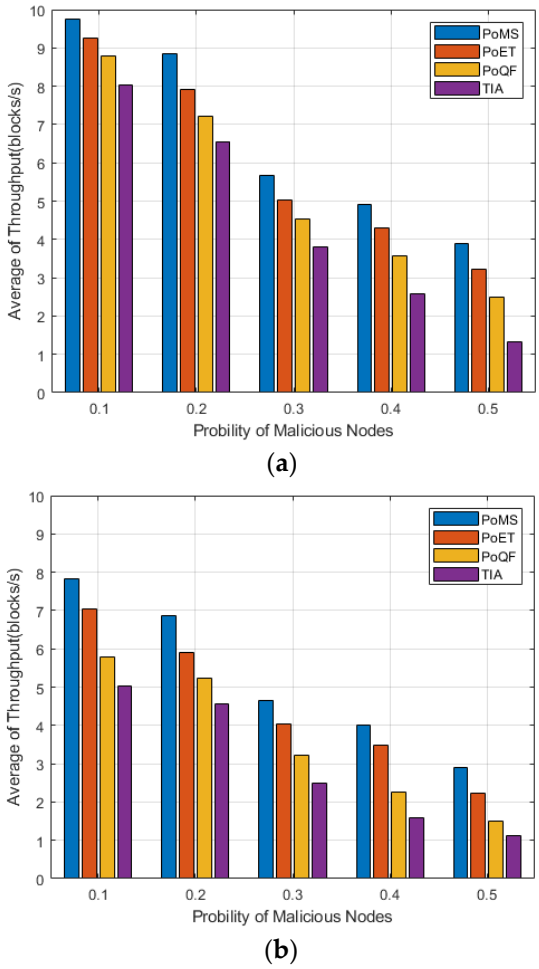

Figure 7. Variation of throughput with different occupancy of malicious nodes: n = 15 in (**a**) and n = 25 in (**b**).

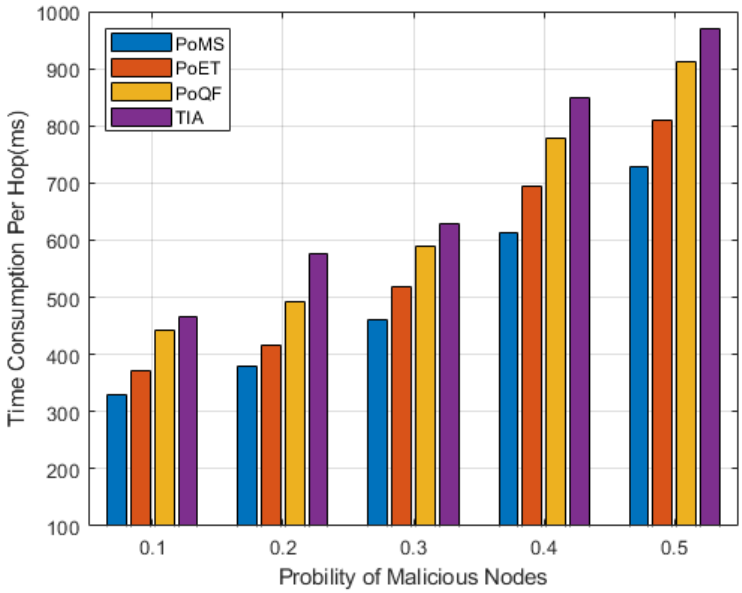

Figure 8. Single hop time with different occupancy of malicious nodes.

While Figure 9 shows the effect of vehicle speed on the time per hop. the PoET consensus time overhead has nothing to do with the vehicle speed itself, and the reason for its time increase lies more in the increased difficulty of establishing communication

connections at high speeds; SF_i Compared with [29], the consensus proposed in this paper increases the weight of the speed parameter in the state factor, and distinguishes between same-direction and opposite-direction vehicles. $SF_i > SF_j$, whereupon the miner nodes are able to complete the construction of forward transmission links more easily, reducing the time spent on each message hop.

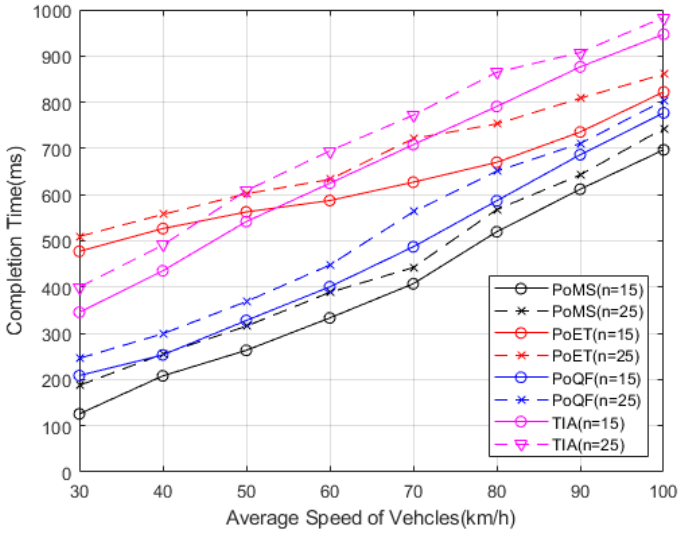

Figure 9. Effect of vehicle speed on single hop time.

Figure 10 shows the ability of other normal nodes to successfully participate in consensus when there are different percentages of malicious nodes in the network. A specific reputation mechanism in the literature [30] ensures that honest nodes can be effectively verified to participate in the consensus when malicious nodes exist in the system; The algorithm proposed in this article is able to receive fewer blocks of valid votes as the number of malicious nodes increases, and the number of nodes participating in the consensus gradually decreases. However, with the function of RSU to record the malicious behavior of nodes, the algorithm can still allow the remaining nodes to successfully complete consensus if half of the nodes are malicious nodes and the network is able to resist the attack of malicious nodes.

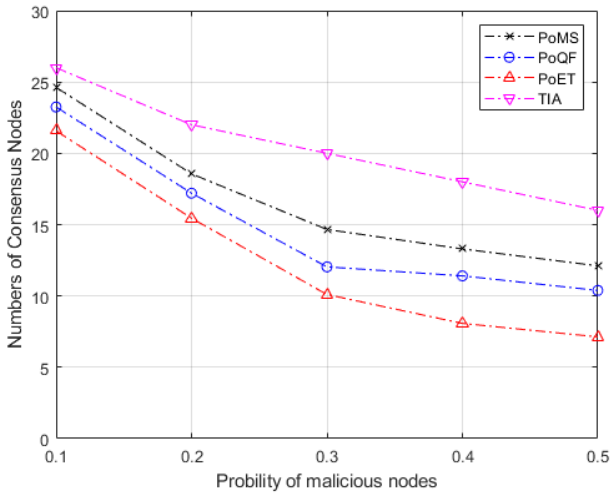

Figure 10. Number of nodes involved in mining.

In summary, the experimental tests conducted in this article demonstrate that the proposed consensus algorithm has less time overhead and better transmission rate compared to the selected consensus. This is due to the algorithm optimizing the computational difficulty

required for nodes to obtain accounting rights, and dynamically adjusting the weights of different nodes based on multiple state factors ($P(diff_i)$, $P(SINR_i)$ and $P(Com_i)$). Nodes with excellent communication conditions in the network are more likely to become relays, and the voting verification mechanism of nodes limits the harm that malicious nodes can cause to system security.

7. Conclusions

This paper analyzes the current research landscape on blockchain technology in the Internet of Vehicles (IoV) and presents a blockchain solution that uses the PoMS consensus algorithm. This solution is designed to optimize the selection of relay nodes for message dissemination within the IoV. Using a machine learning model, we predict the upcoming driving states of mobile vehicles and, based on these predictions, calculate the state values of all nodes participating in the consensus. In addition to evaluating node performance, the algorithm takes into account vehicles' tuning tendencies to strategically select high-quality relay nodes to improve wireless communication between vehicles. We evaluate the theoretical performance of the proposed consensus algorithm by analyzing transmission reception rate, delay, throughput and security. Simulation results show that our consensus algorithm has significantly lower packet loss rates and consensus delays compared to similar current proposals for vehicular networks in high-speed scenarios, highlighting its effectiveness in rapidly changing and complex environments.

Future research directions include improving vehicle node behavior evaluation by integrating more specific indicators into consensus selection processes that go beyond historical records, or using methods similar to those in [31] to improve vehicle speed prediction models with other factors to improve prediction stability and accuracy, while enhancing system safety to best adapt to driving conditions.

Author Contributions: Conceptualization, R.C. and G.L.; Data curation, R.C. and G.L.; Formal analysis, R.C. and Z.S.; Funding acquisition, F.Z., C.L. and C.Y.; Investigation, R.C., F.Z., Z.S., G.L. and C.Y.; Methodology, R.C.; Project administration, F.Z., C.L. and C.Y.; Resources, F.Z. and C.L.; Software, R.C.; Supervision, F.Z., C.L. and C.Y.; Validation, R.C., C.L. and C.Y.; Visualization, R.C.; Writing-original draft, R.C.; Writing-review & editing, R.C., C.L. and C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grants from the Guangxi Natural Science Foundation (2023GXNSFAA026294) and the National Natural Science Foundation of China (62362013).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, Y.; Li, Y.; Chen, S. A survey of Internet of vehicles/vehicles to everything security based on Blockchain. *Sci. Sin. Inform.* **2023**, *53*, 841. (In Chinese) [CrossRef]
2. Islam, M.R.; Rashid, M.M. A Survey on Blockchain Security and Its Impact Analysis. In Proceedings of the 2023 9th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 15–16 August 2023; pp. 317–321.
3. Zhang, C.; Zhao, M.; Zhu, L.; Zhang, W.; Wu, T.; Ni, J. FRUIT: A Blockchain-Based Efficient and Privacy-Preserving Quality-Aware Incentive Scheme. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3343–3357. [CrossRef]
4. Wen, X.; Guan, Z.; Li, D.; Lyu, H.; Li, H. A Blockchain-Based Framework for Information Management in Internet of Vehicles. In Proceedings of the 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Washington, DC, USA, 26–28 June 2021; pp. 18–23.
5. Badertscher, C.; Gaži, P.; Kiayias, A.; Russell, A.; Zikas, V. Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 913–930.
6. Tong, W.; Dong, X.; Zheng, J. Trust-PBFT: A PeerTrust-Based Practical Byzantine Consensus Algorithm. In Proceedings of the 2019 International Conference on Networking and Network Applications (NaNA), Daegu, Republic of Korea, 10–13 October 2019; pp. 344–349.

7. Ur Rehman, M.H.; Salah, K.; Damiani, E.; Svetinovic, D. Towards Blockchain-Based Reputation-Aware Federated Learning. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual, 6–9 July 2020; pp. 183–188.
8. Xu, M.; Zou, Z.; Cheng, Y.; Hu, Q.; Yu, D.; Cheng, X. SPDL: A Blockchain-Enabled Secure and Privacy-Preserving Decentralized Learning System. *IEEE Trans. Comput.* **2023**, *72*, 548–558. [CrossRef]
9. Li, W.; Zhao, Z.; Ma, P.; Xie, Z.; Palade, V.; Liu, H. Graphical Consensus-Based Sharding for Efficient and Secure Sharings in Blockchain-Enabled Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2024**, *73*, 1991–2002. [CrossRef]
10. Du, J.; Cheng, W.; Lu, G.; Cao, H.; Chu, X.; Zhang, Z.; Wang, J. Resource Pricing and Allocation in MEC Enabled Blockchain Systems: An A3C Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 33–44. [CrossRef]
11. Liu, L.; Feng, J.; Mu, X.; Pei, Q.; Lan, D.; Xiao, M. Asynchronous Deep Reinforcement Learning for Collaborative Task Computing and On-Demand Resource Allocation in Vehicular Edge Computing. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15513–15526. [CrossRef]
12. Zhang, J.; Liu, Y.; Qin, X.; Xu, X.; Zhang, P. Adaptive Resource Allocation for Blockchain-Based Federated Learning in Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 10621–10635. [CrossRef]
13. Ul Abadin, Z.; Syed, Z. A Pattern for Proof of Work Consensus Algorithm in Blockchain. In Proceedings of the 26th European Conference on Pattern Languages of Programs, Graz, Austria, 7–11 July 2021; pp. 1–6.
14. Xu, H.; Yu, L.; Liu, Z.; Liu, Z.; Gu, D. Dynamic Practical Byzantine Fault Tolerance. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Beijing, China, 30 May–1 June 2018; pp. 1–8.
15. Bachani, V.; Bhattacharjya, A. Preferential Delegated Proof of Stake (PDPoS)—Modified DPoS with Two Layers towards Scalability and Higher TPS. *Symmetry* **2022**, *15*, 4. [CrossRef]
16. Fitzi, M.; Wang, X.; Kannan, S.; Kiayias, A.; Leonardos, N.; Viswanath, P.; Wang, G. Minotaur: Multi-Resource Blockchain Consensus. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 1095–1108.
17. Biswas, S.; Sharif, K.; Li, F.; Maharjan, S.; Mohanty, S.; Wang, Y. PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain. *IEEE Internet Things J.* **2020**, *7*, 2343–2355. [CrossRef]
18. Han, X.; Yuan, Y.; Wang, F.Y. A Fair Blockchain Based on Proof of Credit. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 922–931. [CrossRef]
19. Yu, J.; Shen, T.; Bai, F.; Yu, Z.; Luo, J. A Blockchain Communication Resource Optimisation Consensus Method. In Proceedings of the 2022 4th Blockchain and Internet of Things Conference, Tokyo, Japan, 8–10 July 2022; pp. 107–114.
20. Chen, C.; Liu, M.; Mo, P.; Yuan, C.; Dai, P. LBLCO: A Lightweight Blockchain with Low Communication Overhead for Internet of Things. In Proceedings of the 2022 4th Blockchain and Internet of Things Conference, Tokyo, Japan, 8–10 July 2022; pp. 92–99.
21. 21. Du, G; Cao, Y.; Li, J.; Yan, Z.; Chen, X.; Li, Y.; Chen, J. A Blockchain-Based Trust-Value Management Approach for Secure Information Sharing in Internet of Vehicles. *IEEE Internet Things J.* **2024**, *11*, 333–344. [CrossRef]
22. Zhang, C.; Luo, X.; Liang, J.; Liu, X.; Zhu, L.; Guo, S. POTA: Privacy-Preserving Online Multi-Task Assignment with Path Planning. *IEEE Trans. Mob. Comput.* **2024**, *23*, 5999–6011. [CrossRef]
23. Cao, M.; Li, V.O.K.; Chan, V.W.S. A CNN-LSTM Model for Traffic Speed Prediction. In Proceedings of the IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.
24. Zhang, C.; Zhao, M.; Liang, J.; Fan, Q.; Zhu, L.; Guo, S. NANO: Cryptographic Enforcement of Readability and Editability Governance in Blockchain Databases. *IEEE Trans. Dependable Secur. Comput.* **2023**. *early access*. [CrossRef]
25. Chao, D.; Xu, D.; Gao, F.; Zhang, C.; Zhang, W.; Zhu, L. A Systematic Survey On Security in Anonymity Networks: Vulnerabilities, Attacks, Defenses, and Formalization. *IEEE Commun. Surv. Tutor.* **2021**. *early access*. [CrossRef]
26. Yuan, Y.; Wang, B.; Zhang, C.; Xiong, Z.; Li, C.; Zhu, L. Towards Efficient and Robust Federated Unlearning in IoT Networks. *IEEE Internet Things J.* **2024**, accepted. [CrossRef]
27. Liu, Y.; Ma, Y.; Tia, D.; Li, G. Research on consensus optimization of trusted incentive algorithm for blockchain mobile nodes. *Sichuan Univ. Nat. Sci. Ed.* **2022**, *59*, 062004.
28. Bai, F.; Sadagopan, N.; Helmy, A. IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks. In Proceedings of the IEEE INFOCOM—Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA, 30 March–3 April 2023; pp. 825–835.
29. Salimitari, M.; Joneidi, M.; Fallah, Y. BATS: A Blockchain-based Authentication and Trust Management System in Vehicular Networks. In Proceedings of the IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 6–8 December 2021; pp. 333–340.
30. Ayaz, F.; Sheng, Z.; Tian, D.; Guan, Y. A Proof-of-Quality-Factor (PoQF)-Based Blockchain and Edge Computing for Vehicular Message Dissemination. *IEEE Internet Things J.* **2021**, *8*, 2468–2482. [CrossRef]
31. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 4245–4257. [CrossRef]