

## Article

# An Optimized Gasper Consensus Protocol Resistant to Adversarial Bias Attacks

Xi Lin <sup>1,2,\*</sup> and Junfeng Tian <sup>1,2</sup><sup>1</sup> School of Cyber Security and Computer, Hebei University, Baoding 071000, China; tjf@hbu.edu.cn<sup>2</sup> Hebei Key Laboratory of High Confidence Information Systems, Hebei University, Baoding 071000, China

\* Correspondence: linxi1019sushi@163.com

## Abstract

Blockchain consensus mechanisms are fundamental to the security and decentralization of distributed ledgers. In Proof-of-Stake (PoS) systems, which are lauded for their energy efficiency, the fair and unpredictable selection of block proposers is paramount and relies heavily on secure random number generation. The RANDAO random number generation mechanism in the Gasper protocol is susceptible to hash collision attack, which can introduce adversarial bias in the block proposer selection process. From the perspective of resisting adversarial bias attacks, this paper examines the optimization of the Gasper consensus protocol, focusing on security issues such as vulnerabilities to hash collisions in RANDAO and high latency in asynchronous network environments. By analyzing the spatial-temporal distribution of historical block hashes, we propose a dual-round random number verification mechanism that enhances reliability through multiple validation models. We develop a dynamic game-theoretic model under incomplete information to analyze node strategy selection and interaction dynamics. Our experimental results demonstrate that the improved protocol (RABA-Gasper) offers superior resistance to attacks, fairness, and efficiency compared to conventional protocols. RABA-Gasper outperforms conventional ones, achieving a 6.8% attack success rate (vs. 32.7% for RANDAO and 18.2% for Two Look-Back) with 94.3% hash collision detection, a proposer Gini coefficient below 0.23, 2.3x higher throughput retention than RANDAO in asynchronous networks, and a slightly increased random number generation latency of 125 ms. Supported by a game-theoretic model, it guarantees security when honest nodes account for  $\geq 2/3$  of the total.

**Keywords:** blockchain; RABA-gasper; hash collision attack; asynchronous network



Academic Editor: Gianluca Lax

Received: 28 October 2025

Revised: 12 December 2025

Accepted: 19 December 2025

Published: 23 December 2025

**Copyright:** © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Blockchain technology relies on consensus mechanisms to achieve secure, decentralized, and efficient transaction validation across distributed networks. Among various consensus paradigms, Proof-of-Stake (PoS) has become a leading research direction and practical choice for modern blockchains (e.g., Ethereum 2.0, Cardano) over the past decade. Compared to the traditional Proof-of-Work (PoW) mechanism, PoS reduces energy consumption by over 99% and mitigates the risk of mining centralization, addressing key limitations of early blockchain systems [1]. As a key variant of PoS, the Gasper protocol innovatively incorporates a randomness generation mechanism to select block proposers, thereby preserving fairness and decentralization in consensus.

However, the RANDAO random number generation mechanism currently employed in the Gasper protocol has revealed critical security vulnerabilities. When subjected to hash

collision attacks, it may introduce adversarial bias in the block proposer selection process, which could severely compromise the security and stability of the entire blockchain network [2]. Furthermore, as blockchain networks continue to expand in scale and application scenarios grow increasingly complex, network environments have become more heterogeneous and unpredictable, making asynchronous network conditions the norm. Under these conditions, the conventional RANDAO protocol exhibits significant security weaknesses in asynchronous networks, where node communication suffers from substantial delays and uncertainty. These characteristics adversely affect the RANDAO mechanism's operation [3]. Consequently, designing a mechanism that can both effectively resist hash collision attacks and ensure fair, randomized block proposer selection in complex, asynchronous networks has become a critical challenge requiring urgent resolution. This issue is crucial not only for the security and stability of the Gasper protocol itself, but also for ensuring the reliable operation of PoS-based blockchain networks in open, complex network environments.

To address this, Snow White proposed a Two Look-Back mechanism that incorporates not only the hash values submitted in the current round but also examines hash values from the previous two blocks [4]. This study investigates optimizations for the Gasper consensus protocol, focusing on mitigating adversarial bias attacks by addressing security vulnerabilities in RANDAO. By analyzing the spatial-temporal distribution of historical block hashes, we develop a verification model that implements dual-round validation of generated random numbers, enhancing reliability through multiple verification mechanisms. We establish a dynamic game model under incomplete information to analyze node strategy selection and interaction dynamics, thereby optimizing the Gasper protocol at both the mechanistic and theoretical levels. This mechanism effectively resists hash collision attacks, where attackers must simultaneously manipulate hash values from both current and previous two rounds to influence randomness generation, significantly raising the attack difficulty threshold. The Two Look-Back mechanism further demonstrates adaptability to asynchronous networks by leveraging historical block information, reducing dependence on real-time communication, and enhancing the stability and reliability of randomness generation.

Therefore, this study develops a dual-round randomness verification mechanism, which is built upon the Two Look-Back mechanism to address security vulnerabilities in RANDAO under asynchronous network conditions in the Gasper protocol. By leveraging spatial-temporal features of historical block hashes, we innovatively design this verification framework to significantly enhance the protocol's resistance against hash collision attacks. Concurrently, we employ a game-theoretic model to analyze strategic interactions among nodes, providing theoretical foundations for protocol security. The proposed solution offers both theoretical contributions and practical value for enhancing security in PoS.

The main contributions of this paper are as follows:

- This paper proposes a dual-round randomness verification mechanism incorporating spatial-temporal features of historical blocks. By analyzing periodicity in hash value time series and spatial autocorrelation (quantified using Moran's Index for regional node hash associations), we establish a defense mechanism combining real-time hash statistical verification with historical feature cross-constraints, fundamentally mitigating hash collision attacks.
- It develops a dynamic game model under incomplete information to quantify strategic interactions between honest and malicious nodes, providing theoretical guarantees for protocol security.
- It develops a practical solution balancing attack resistance and efficiency. Our lightweight spatial-temporal feature extraction algorithm optimizes verification processes, enabling stable randomness generation and fair block proposal selection even

in asynchronous networks, thereby advancing practical deployment of PoS consensus in open networks.

The rest of this paper is structured as follows. Section 2 reviews related work. Section 3 outlines the foundational mechanisms, while Section 4 formalizes the system, threat, and network models. Section 5 details the design of the dual-round randomness verification framework. Section 6 develops the game-theoretic model, and Section 7 evaluates the experimental results. Finally, Section 8 concludes with key findings and future directions.

## 2. Related Work

Blockchain technology enables trustless consensus via decentralized networks, with its core mechanisms evolving around security, efficiency, and decentralization. Early platforms like Bitcoin employed Proof-of-Work (PoW), relying on computational competition for transaction validation. However, PoW faces critical limitations, including excessive energy consumption and risks of mining centralization [5]. To overcome these challenges, PoS was introduced, where validators stake assets to participate in consensus. PoS reduces energy usage by over 99% and has been widely adopted by leading platforms like Ethereum 2.0 and Cardano [6]. A critical PoS variant, the Gasper protocol, integrates GHOST and CasperFFG with RANDAO (random number generator). By aggregating node hash values, it dynamically selects block proposers to ensure consensus fairness [7].

In terms of research on hash collision attacks and defense, current research on RANDAO shows that its reliance on hash aggregation for randomness generation is susceptible to precomputation-based collision attacks. Tapolcai [8] demonstrates that RANDAO in its current form is prone to manipulation and calls for more robust alternatives. Buterin et al. [9] show that attackers can exploit hash collisions to bias randomness, undermining block proposal fairness. Son et al. [10] empirically show that injecting 100 collision pairs into RANDAO raises adversarial proposal rates by 32.7%, directly threatening the immutability of the blockchain. To address this, the Two Look-Back mechanism of Snow White verifies randomness using the prior two rounds of block hashes, forcing attackers to manipulate both current and historical hashes, raising attack complexity [11]. However, it focuses solely on temporal hash continuity, neglecting spatial clustering—regional nodes often produce spatially clustered hashes due to shared network conditions, an unexploited feature in current defenses. Additionally, Liang's [12] blockchain-based malicious-attack-resistant cloud data integrity audit scheme offers inspiration for hash verification through spatial-temporal feature analysis, but no existing method combines historical hash patterns in both time and space, leaving asynchronous networks vulnerable to attacks exploiting regional node clustering in asynchronous networks.

In the research on solving defects of the RANDAO mechanism in asynchronous networks, RANDAO faces a dual challenge: incomplete hash aggregation and inconsistent verification. Alturki's [13] statistical model demonstrates that at a 10% packet loss rate, 15% of the hash values fail to transmit due to transmission failure, resulting in the failure of random number verification. This stems from reliance on real-time hash aggregation in RANDAO: communication delays across geographically dispersed nodes result in incomplete hash collection or inconsistent verification data. To address asynchronous network challenges, some studies propose delay-tolerant mechanisms. Ethereum 2.0 implements a timeout-retransmission protocol, yet fails to resolve core issues of hash integrity and randomness [14]. Moreover, traditional mechanisms overlook spatial correlation in node distribution. Nodes in proximity share similar network infrastructure, causing their hash values to cluster spatially. While this clustering could improve verification reliability, the current research has not integrated it with temporal features. It is difficult to build an efficient asynchronous network defense system.

In recent years, game theory has emerged as a key tool for analyzing strategic interactions among nodes in consensus protocols. Crespi's [15] Nash equilibrium theory provides a theoretical foundation for modeling the dynamic interplay between honest and malicious nodes. In the PoS scenario, Jin's [16] economic incentive analysis reveals that rational nodes are inclined to participate honestly when penalties for malicious behavior outweigh potential attack gains. However, existing PoS game-theoretic models often overlook the impact of spatial–temporal features on attack costs, particularly lacking quantitative analysis of node strategies in dynamic asynchronous networks. While some studies employ dynamic game models to analyze node strategies under incomplete information, they are not combined with specific consensus mechanisms like RANDAO, limiting their applicability to protocol optimization. Current research largely lacks theoretical models that combine spatial–temporal verification frameworks with game theory, impeding quantitative evaluation of security thresholds in optimized protocols, which restricts the accurate analysis of the anti-attack of Gasper.

This paper integrates the Two Look-Back mechanism in Snow White with RANDAO in Gasper, incorporating spatial–temporal features into verification. Through the spatial–temporal features based on the hash values of historical blocks, we propose a novel dual-round randomness verification architecture. This architecture can strengthen resistance to hash collision attacks, solve the security problems of RANDAO in asynchronous networks, and analyze the node strategy interaction through the game-theoretic model—providing theoretical guarantees for protocol security.

We have analyzed the advantages and limitations of the related work in Table 1.

**Table 1.** Comparative analysis of related works on randomness security in PoS consensus.

| Literature            | Advantages                                                                                  | Limitations                                                                                         | Improvements in Our Work                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| Snow White [4,11]     | Introduces Two Look-Back mechanism for temporal verification, increasing attack difficulty. | Relies solely on temporal continuity of hashes; neglects spatial distribution features of nodes.    | Integrates spatial features (Moran's Index) with temporal features (ARIMA) for multi-dimensional verification. |
| Buterin et al. [9]    | Highlights vulnerability of RANDAO to bias attacks via hash collision.                      | Primarily identifies the problem without proposing a comprehensive defensive mechanism.             | Proposes a concrete dual-round verification framework with a detailed validation algorithm.                    |
| Alturki et al. [3,13] | Models RANDAO's vulnerability in asynchronous networks and proposes formal verification.    | Focuses on theoretical modeling; lacks a practical, lightweight solution for real-world deployment. | Develops a lightweight spatial–temporal feature extraction algorithm suitable for asynchronous environments.   |

Table 1. Cont.

| Literature          | Advantages                                                              | Limitations                                                                                        | Improvements in Our Work                                                                              |
|---------------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Liang's Scheme [12] | Offers inspiration for integrity audit using spatial-temporal analysis. | Not directly applied to consensus randomness generation or hash collision defense.                 | Tailors spatial-temporal analysis specifically for enhancing RANDAO's security in Gasper protocol.    |
| Jin's Analysis [16] | Provides economic incentive analysis for node behavior in PoS.          | Game-theoretic model is generic and not integrated with specific consensus mechanisms like RANDAO. | Develops a dynamic game model explicitly coupled with the proposed dual-round verification mechanism. |

### 3. Foundational Concepts and Mechanisms

#### 3.1. Gasper and RANDAO

The Gasper protocol is a PoS-based consensus. Its core idea is to obtain the right to propose and vote blocks by staking nodes, thereby achieving consensus. In Gasper, the RANDAO random number generator plays a critical role by producing randomness to fairly select block proposers, ensuring each node has an equal opportunity to participate in block creation per round [17]. As the core component of Gasper for selecting block proposers, RANDAO operates through a complex workflow, illustrated in Figure 1.

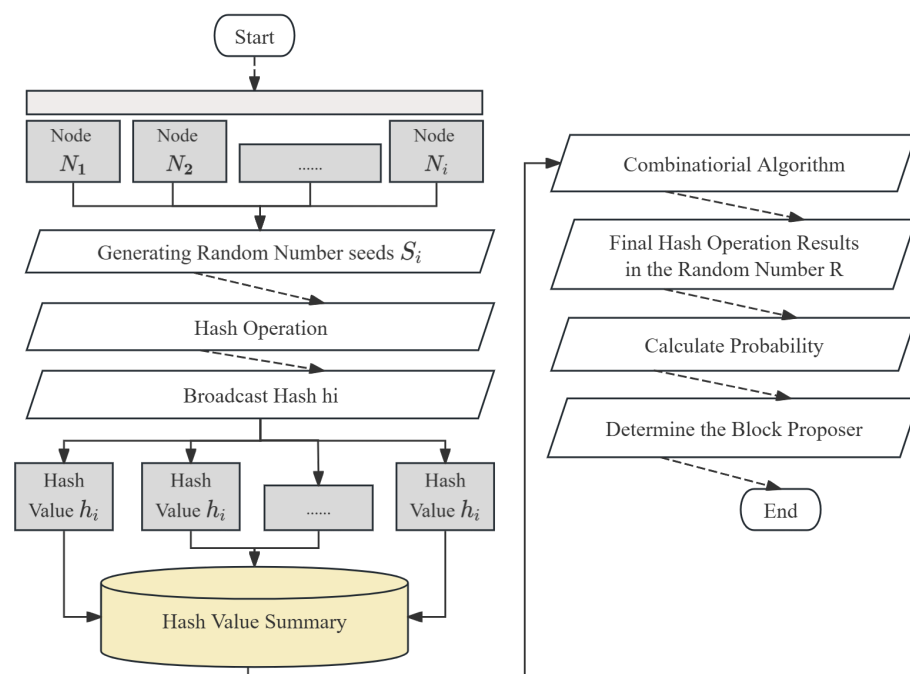


Figure 1. Protocol workflow.

At the beginning of each randomness generation round, every node  $N_i (i = 1, 2, \dots, n)$  independently creates a random seed  $S_i$ . Each seed is then processed through the hash func-

tion  $H(S_i) = h_i$  to produce its corresponding hash value. Subsequently, nodes broadcast these hash values across the entire network.

In the RANDAO mechanism of the Gasper protocol, the generation of random seeds must satisfy cryptographically secure randomness requirements to prevent adversaries from predicting or tampering with the seeds in advance. Node  $N_i$  generates random seed  $S_i$  using a cryptographically secure pseudorandom number generator (CSPRNG), with private-key binding and dynamic timestamp anchoring to guarantee unpredictability and tamper resistance. The implementation details are as follows:

**Seed Generation Logic:** The protocol defines the following key variables for seed generation. The node gathers initial random entropy, denoted as *Entropy*, from a cryptographically secure source (e.g., hardware-based or system entropy pool), ensuring initial unpredictability. It is then combined with the node's private key  $SK_i$ , the current epoch timestamp  $T_c$ , and a local counter *Nonce*, where the  $\oplus$  symbol denotes a secure combination operation that integrates these three variables. The combined value is hashed to produce the seed  $S_i$ , as defined in Equation (1).

$$S_i = H(SK_i \oplus Entropy \oplus Nonce) \quad (1)$$

where  $S_i$  is a random seed generated by node  $N_i$ , *Nonce* acts as a unique node-local counter to prevent seed replay attacks over time. *Entropy* is the random entropy value that meets the cryptographic security standard to ensure the initial randomness of the seed.  $SK_i$  guarantees the unique binding between seed and node identity through a private key signature mechanism, preventing malicious forgery.

**Hash Commitment and Verification:** Nodes broadcast only the hash value  $h_i = H(S_i)$  in a commit–reveal model. During the commitment phase, the hash conceals the original seed to prevent premature disclosure. During the revelation phase, nodes disclose  $S_i$  for verification, where other nodes verify the integrity through  $H(S_i) = h_i$  and confirm sender identity via private key signature  $Sign_{SK_i}(h_i \oplus T_c)$ . The timestamp  $T_c$  of the current consensus epoch serves two security purposes: it establishes temporal validity for the hash to prevent replay attacks using historical seeds, and guarantees both uniqueness and non-repudiation of the generated random numbers within each epoch.

**Security Guarantees:** In terms of unpredictability, the seeds depend on both the real-time timestamp  $T_c$  and the node-exclusive private key  $SK_i$ , making it impossible for attackers to pre-compute or forge valid seeds. In terms of attack resistance, the distributed aggregation mechanism (e.g., multi-layer hash operations) effectively dilutes the impact of any single malicious seed, and the collective randomness from the majority of honest nodes ensures global fairness in the system.

When the network collects a sufficient number of hash values (denoted as  $m$ , where  $m$  must meet certain threshold conditions to ensure randomness and security), a specific combination algorithm processes these hashes. A common approach is to first sum the hashes and then apply another hash operation. Let  $C$  be the combination function, the final random number  $R$  used to determine the block proposer is

$$R = H(C(h_1, h_2, \dots, h_m)) \quad (2)$$

The block proposer is determined through a function that maps the random number  $R$  and the staked equity  $E_i$  of the node, such as a common Function (3).

$$f : P_i = f(R, E_i) \quad (3)$$

The selection probability  $P_i$  for node  $i$  to become the block proposer is positively correlated with its staked equity  $E_i$ , and modulated by the random number  $R$ . While

nodes with greater stake naturally have higher selection probabilities, the randomization mechanism introduces dynamic perturbation to prevent stake concentration. This design ensures that each node maintains a non-zero selection probability commensurate with its stake contribution, thus guaranteeing the decentralized nature of the consensus process. A typical implementation involves normalizing  $R$  to a value  $R'$  (e.g., scaling it to the range  $[0, 1]$ ) and using it as an exponent to weight the stake.

The block proposer selection function  $f$  is explicitly designed to integrate staked equity with randomness. Each node generates a seed  $S_i$  using its staked equity  $E_i$ , private key  $SK_i$ , and timestamp  $T_c$ , then computes a hash  $h_i = H(S_i)$ . These hashes are aggregated with stake weights to produce the final random number.

$$R = H(\sum(E_i \cdot h_i) \oplus \text{Nonce}) \quad (4)$$

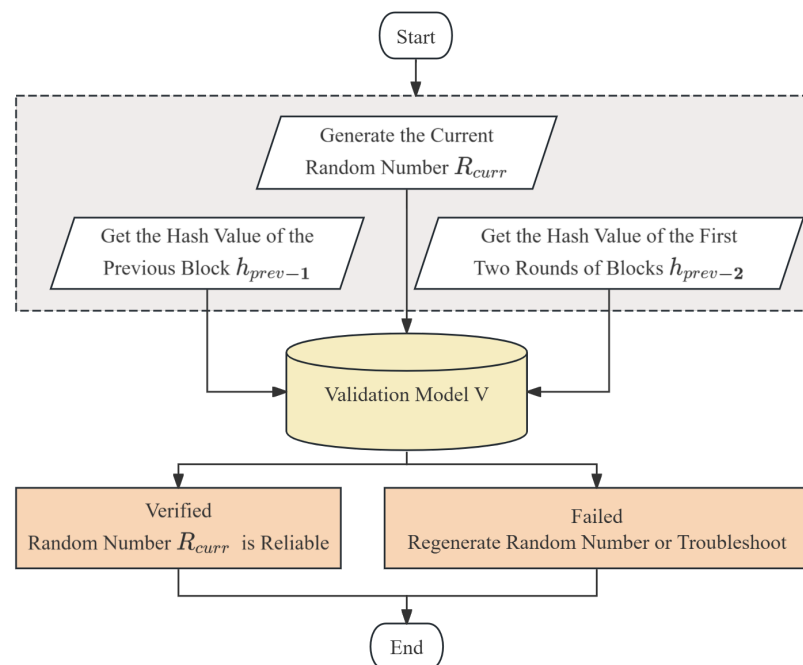
The selection probability is

$$P_i = \frac{E_i \cdot R'}{\sum(E_i \cdot R')} \quad (5)$$

where  $R'$  is explicitly defined here as the normalized version of the random number  $R$ . This ensures that nodes with higher stake have a proportionally higher, yet dynamically adjusted, chance of being selected, maintaining decentralization.

### 3.2. The Two Look-Back Mechanism of Snow White

The Two Look-Back mechanism of Snow White is specifically designed for asynchronous network environments to enhance randomness security, of which the workflow is shown in Figure 2. Its core innovation lies in verifying current random numbers using historical block hashes, thereby improving reliability and attack resistance.



**Figure 2.** Operational workflow of Two Look-Back.

When generating the current random number  $R_{curr}$ , the protocol incorporates not only the hash values submitted in the current round but also examines the hash values from the previous two blocks (denoted  $h_{prev-1}$  and  $h_{prev-2}$ ). A verification model is constructed by

analyzing the temporal and spatial distribution features of these historical block hashes. Let  $V$  be the verification function, the validation process can be formally expressed as in (6).

$$V(R_{curr}, h_{prev-1}, h_{prev-2}) \quad (6)$$

The random number  $R_{curr}$  is deemed cryptographically secure only upon successful verification. If verification fails, the protocol either regenerates the random number or initiates diagnostic procedures. This design provides strong resistance against hash collision attacks, as adversaries must simultaneously manipulate the hash values in both the current round and the previous two blocks, thereby exponentially increasing the attack cost. More specifically, in asynchronous networks, communication delays may prevent some hash values from arriving before the aggregation window closes. The Two Look-Back mechanism enhances stability by incorporating the hashes from the two previous blocks, which have already achieved network-wide consensus. This allows the random number to be verified even if some current-round hashes are missing, thereby reducing dependency on the timely arrival of all real-time communications [4].

In general, the current Two Look-Back mechanism only relies on temporal verification of the historical hashes in the previous two rounds, failing to leverage spatial clustering patterns (e.g., Asian nodes show Moran's Index = 0.62 for hash values). If malicious nodes are concentrated in a specific geographic region, they can coordinate to manipulate submitted hash values, exploiting shared network environments to generate spatially correlated anomalies that undermine the random number generation mechanism. The Two Look-Back mechanism struggles to defend against such attacks. This type of attack is termed a regional collusion attack. The dual-round randomness verification mechanism proposed in this study embeds the staked equity and the hashes in the previous two blocks into the seed generation formula, and integrates the stake-weighted hash values in the aggregation phase. In the verification phase, a mechanism combining real-time hash statistical verification with historical spatial-temporal feature cross-constraint is constructed. The Autoregressive Integrated Moving Average (ARIMA) model is used to capture the periodicity of the hash sequence in the time dimension, and Moran's Index is used to quantify the regional clustering in the spatial dimension. The stake-hash association model and CNN-LSTM joint verification model are combined to form a multi-dimensional defense system. At the same time, an incomplete-information dynamic game model was constructed to quantify the strategic interaction between honest nodes and malicious nodes, enhancing protocol robustness through both mechanism design and theoretical analysis.

## 4. Model and Overview

### 4.1. System Model

The system model of the Gasper is built on a distributed node network, with its core architecture designed to achieve both decentralization and security. Nodes in the network are categorized into two types.

**Honest Nodes Set H:** These nodes strictly adhere to the protocol rules. Let  $M$  denote the number of honest nodes, which must satisfy  $M \geq 2/3N$  to meet Byzantine Fault Tolerance (BFT) requirements [18], where  $N$  is the total number of nodes. Theoretically, when honest nodes constitute a majority ( $M \geq \frac{2}{3}N$ ), the system can maintain security through consensus among honest nodes, even under malicious attacks.

**Malicious Nodes Set A:** These nodes aim to maximize illicit gains and may employ strategies such as hash collision attacks or network delay interference. Their stake is typically assumed to be  $\frac{K}{E_{total}} \leq \frac{1}{3}$ , but in practice, malicious nodes may bypass this limit

through joint attack or stake pool manipulation. Thus, the system model must account for the impact of dynamic stake changes on security.

The staked equity  $E_i$  of nodes serves not only as the threshold for participating in consensus but also as the foundational parameter determining proposal probability. The proportion of stake  $\frac{E_i}{E_{total}}$  and a random number  $R$  jointly determine the proposer selection function  $f$ , such that  $P_i = f(R, E_i)$ . This hybrid mechanism combining stake with randomness balances decentralization and efficiency: while high-stake nodes have a higher base proposal probability, the dynamic adjustment via randomness prevents stake monopolization, ensuring that even nodes with small stakes maintain a non-zero, stake-proportional probability of being selected. This mechanism provides relatively fair proposal opportunities for small and medium nodes, with fairness to be assessed using the Gini coefficient in subsequent experiments.

#### 4.2. Threat Model

##### 4.2.1. Adversary Capability Boundaries and Assumptions

The construction of the threat model must adhere to the “practical attack feasibility” principle, avoiding over-idealized or extreme assumptions.

**Computational Resource Constraints:** We assume malicious nodes control no more than 30% of the total network hash power (referring to the defense threshold of Ethereum 2.0); such adversaries may employ GPU clusters or rented cloud computing resources to execute hash collision attacks. According to the Birthday Paradox, finding SHA-256 collisions with 50% probability requires  $2^{128}$  computations. However, practical attacks may utilize optimization techniques like length extension attacks or multi-collision attacks to reduce computational costs.

**Network Control Capabilities:** In asynchronous networks, malicious nodes can manipulate two types of communication parameters: latency and packet loss. However, such attacks are constrained by the decentralization of the network topology and cannot completely block the communication of all honest nodes.

**Stake Size Limitation:** According to the fundamental security assumption of PoS, the total stake controlled by malicious nodes should not exceed  $1/3$ ; otherwise, it risks triggering “Proof-of-Stake failure”. This assumption originates from the “safety threshold” theory in the Casper FFG protocol, where exceeding  $1/3$  malicious stake may prevent the system from finalizing the Finality blocks.

##### 4.2.2. Attack Vectors and Impact Analysis

**Game Process of Hash Collision Attack:** Adversaries generate two distinct inputs  $(S_a, S'_a)$ , such that

$$H(S_a) = h(S'_a) = h_a \quad (7)$$

where  $S_a$  represents a legitimate seed and  $S'_a$  denotes a carefully crafted malicious seed. By injecting  $h_a$  into the hash aggregation phase, the adversaries can manipulate the generation of random number  $R$ . The essence of this attack is to use the insufficient collision resistance in the hash function, transforming the randomness generation process from being unpredictable to partially controllable.

**Latency Attack Chain in Asynchronous Networks:** The attack flow can be decomposed into three phases: (1) delayed broadcast—malicious nodes intentionally delay broadcasting authentic hash values after the randomness generation cycle begins, causing them to arrive after the aggregation time window closes; (2) fake injection—the adversaries propagate fabricated hash values that may satisfy historical temporal patterns but violate spatial distribution rules; (3) verification interference—exploiting communication latency disparities between nodes, the adversaries send stale or incorrect hash values during verifi-

cation, causing inconsistent validation data across nodes. When the packet loss rate reaches 10%, traditional RANDAO mechanisms will fail to verify due to 15% missing hash values, and the adversaries can repeatedly destroy the consensus process.

#### 4.3. Network Model

**Temporal Uncertainty:** Node communication latency shows significant geographic correlation—Features of nodes such as communication latency and hash value distribution exhibit regular correlations with geographical regions: nodes within the same region have lower communication latency with smaller fluctuations, while intercontinental nodes show higher latency with greater variations. Such differences stem from geographical distance and the regional distribution characteristics of network infrastructure. For example, the mean latency of nodes in the same region (e.g., within Asia) is 50 ms, and the standard deviation is 20 ms; The mean latency across intercontinental nodes (e.g., Asia-Europe) is 200 ms, with a standard deviation of 50 ms. This latency variation causes inconsistent hash arrival times at aggregation nodes, with some hashes potentially discarded for exceeding the aggregation window. Furthermore, network latency fluctuates dynamically over time, with peak-hour variations increasing by up to 30%, exacerbating randomness generation uncertainty.

**Spatial Topology:** Node distribution demonstrates “geographic clustering”—North America, Europe, and Asia account for 35%, 30%, and 25% of the nodes, respectively, with the remainder distributed across other regions. This distribution results in similar network environments for nodes within the same region, causing locally clustered features in their generated hash values.

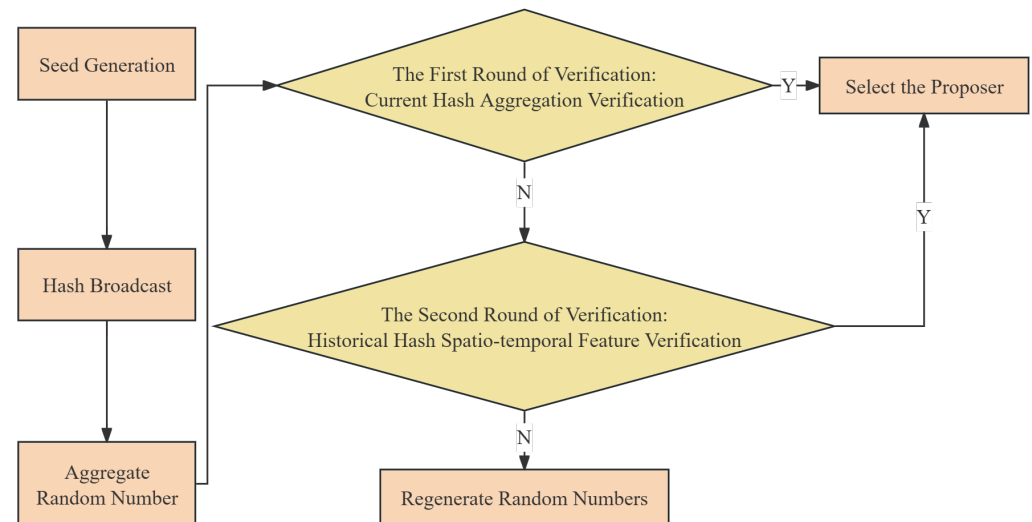
#### 4.4. Coupling Relationship and Defense Mapping

The interaction of system, threat, and network models forms the theoretical foundation for protocol optimization: multi-dimensional mapping of attack vectors. In the threat model, hash collision attacks exploit the sensitivity of the hash aggregation function  $C$  of the system model to anomalous inputs, and rely on the temporal delay of the network model to implement the “last-reveal” strategy (injecting collision hashes before aggregation window closure). Asynchronous network interference compromises hash integrity collection in the system model through the delay and packet loss features of the network model. For cross-model defensive synergy, the dual-round verification mechanism must concurrently satisfy (1) the stake-weighting mechanism of system model ( $E_i$  in randomness computation) to dilute malicious hash influence; (2) the spatial-temporal constraints of network model (ARIMA prediction error  $\leq 5\%$  with consistent Moran’s Index [19]) to filter anomalous hashes. This cross-model synergy reduces attack success probability from 32.7% in traditional RANDAO to 6.8%, demonstrating the effectiveness of multi-dimensional modeling.

## 5. System Design

This study employs the Two Look-Back mechanism to establish a RABA-Gasper system, enhancing the security of RANDAO in the Gasper protocol in asynchronous network conditions.

Specifically, the Two Look-Back mechanism integrates with RANDAO through a dual-phase process. Figure 3 illustrates the process of random number generation and double-round verification.



**Figure 3.** Random number generation and dual-round verification.

During seed generation, nodes combine staked equity  $E_i$  with the hash values of the previous two blocks ( $h_{prev-1}$  and  $h_{prev-2}$ ) to produce the random seed as in (8).

$$S_i = H(SK_i \oplus E_i \oplus h_{prev-1} \oplus h_{prev-2}) \quad (8)$$

where the symbol  $\oplus$  is used here in the same way as in Equation (1), which is used to combine multiple independent features (staked equity, historical block hashes, private key, etc.) to enhance the randomness and anti-tampering capability of the seed  $S_i$ .

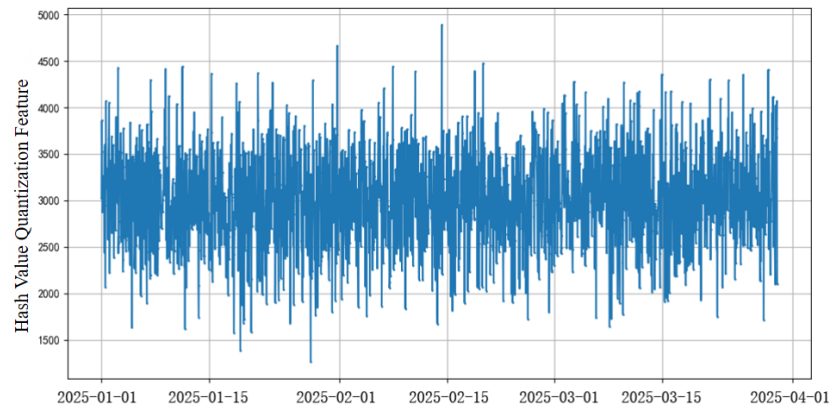
And then it is ensured that the seed is cryptographically bound to both historical trustable data and staked equity.

During random number generation, the process first aggregates the current hash by  $R = H(\sum(E_i \cdot h_i) \oplus \text{Nonce})$ , then performs a dual verification of  $R$  using the Two Look-Back mechanism—the first round based on the distribution of the current hash value, and the second incorporating the spatial-temporal characteristics of historical hashes.

This design enables RANDAO to compensate for missing real-time communication in asynchronous networks through historical data integration, while requiring adversaries to manipulate three rounds of hash values and stake distributions to execute an attack—thereby improving security by two orders of magnitude compared to conventional mechanisms.

### 5.1. Spatial–Temporal Distribution Features

Generally speaking, blockchain networks continuously generate new blocks over time. The hash value of each block encapsulates not only the transactional characteristics within that block but also maintains cryptographic linkage with the hash value of the previous block, collectively forming an immutable chain structure [20]. To thoroughly analyze the temporal distribution patterns of historical block hashes, we collected hourly-recorded block data from the Ethereum between 1 January 2025 and 30 March 2025. Figure 4 shows these temporal variations of block hashes within the blockchain network. Using specialized data collection tools, we sequentially extracted detailed information, including block height, timestamp, transaction count, and corresponding hash values for each block. The dataset underwent rigorous cleaning and preprocessing to remove outliers and erroneous entries. Our time-series analysis of the extensive historical block data revealed statistically significant periodic fluctuations in hash values.



**Figure 4.** Temporal evolution of block hash values in blockchain network.

The visualization clearly demonstrates that during periods of high network activity (e.g., when new transactions flood in), hash values exhibit accelerated change frequency and amplified fluctuation magnitude. This phenomenon stems from increased transaction volume enhancing block content diversity, which in turn generates more varied hash computation outcomes. To precisely characterize and predict these temporal variations, we implement a time-series model—the Autoregressive Integrated Moving Average model ( $ARIMA(p, d, q)$ ), mathematically expressed as

$$\Phi(B)^d (1 - \sum_{i=1}^p \varphi_i B^i) X_t = \theta_0 + (1 + \sum_{j=1}^q \theta_j B^j) \epsilon_t \quad (9)$$

where  $X_t$  represents the time-series value at time  $t$ , that is, the block hash value quantitative feature at time  $t$ ,  $\Phi(B)^d$  denotes a polynomial of order  $d$  in the lag operator  $B$ ,  $\varphi_i$  and  $\theta_j$  are autoregressive and moving average coefficients, respectively, and  $\epsilon_t$  signifies white noise [21]. This modeling approach enables reliable estimation of historical block hash value ranges around specific timestamps, thereby providing temporal reference benchmarks for subsequent random number verification processes.

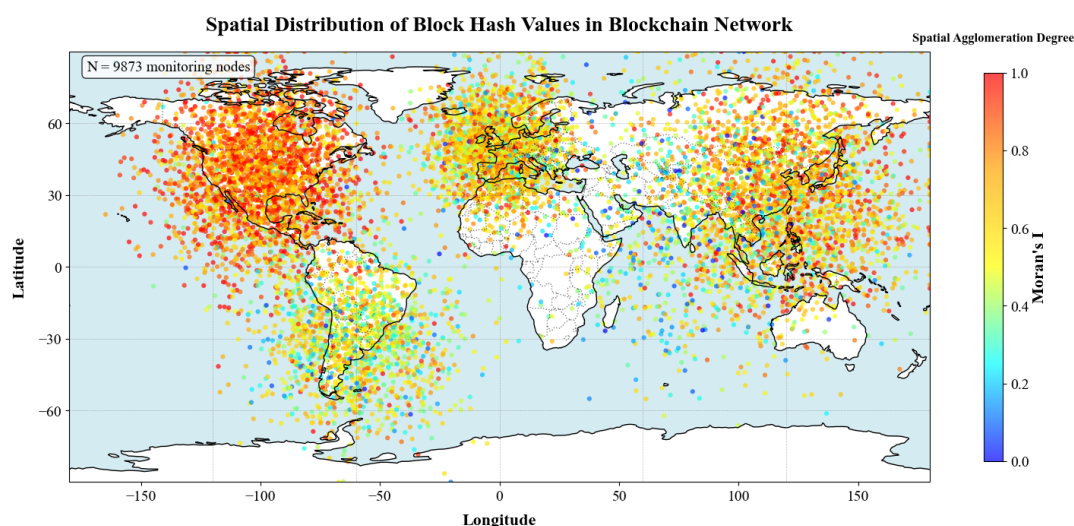
From a spatial perspective, blockchain nodes are globally distributed, exhibiting inherent spatial correlations during block generation and propagation. Accounting for network latency and heterogeneous computing power distribution across regions, block hashes generated by nodes in different geographical areas may demonstrate localized clustering characteristics. We employ spatial autocorrelation analysis to examine historical block hash values generated by nodes across diverse locations. To quantify this regional clustering, we employ Moran's Index, a statistical measure of spatial autocorrelation. Moran's Index is calculated as in (10).

$$I = \frac{n}{\sum_{i=1}^n \sum_{j=1}^n w_{ij}} \frac{\sum_{i=1}^n \sum_{j=1}^n w_{ij} (\hat{x}_i - \bar{x})(\hat{y}_j - \bar{x})}{\sum_{i=1}^n (\hat{x}_i - \bar{x})^2} \quad (10)$$

where  $n$  denotes the total number of nodes,  $\hat{x}_i$  and  $\hat{x}_j$  represent the quantified hash features of blocks generated by nodes  $i$  and  $j$ , respectively,  $\bar{x}$  is the mean of all nodes' hash features, and  $w_{ij}$  constitutes the spatial weight matrix encoding the spatial relationship between nodes  $i$  and  $j$ . The index value ranges from  $-1$  (perfect dispersion) to  $+1$  (perfect clustering), with values near zero indicating random spatial distribution. In our analysis, a Moran's Index of 0.62 for Asian nodes signifies a statistically significant tendency for hash values from this region to cluster together, which serves as a spatial benchmark for detecting anomalous deviations during verification [19].

To validate the spatial clustering characteristics of block hashes, we collected Ethereum network data through multiple channels, primarily utilizing the official Ethereum public API and the Etherscan blockchain data platform. Our dataset comprises 9873 globally distributed monitoring nodes. As shown in Figure 5, we present the spatial distribution of block hashes generated by nodes through Moran's Index analysis, where darker shading indicates higher degrees of spatial hash value clustering within respective regions. For instance, nodes in Asia (particularly East Asia) and North America exhibit pronounced spatial autocorrelation, forming distinct clusters.

The statistical significance of these correlations was demonstrated through a spatial regression analysis, with Moran's Index serving as the key metric for quantifying the spatial clustering of hash values. We formulated a spatial econometric model where the quantified features of block hashes (e.g., entropy) were the dependent variables. The independent variables included proxies for regional network infrastructure capacity (e.g., average node bandwidth per region), inter-node communication frequency (e.g., P2P message count per unit time), and blockchain application activity levels (e.g., daily transaction volume originating from the region). A significant Moran's Index ( $I$ ) for the regression residuals (e.g.,  $I > 0.5$  with a  $p$ -value  $< 0.05$ ) would indicate that the model successfully captures the spatial dependence, and thus the correlations are not random. The local economic factors considered, which influence these activity levels, included the region's Gross Domestic Product (GDP) per capita, the maturity of its digital economy, and the stringency of local financial regulations pertaining to blockchain technology. These factors are understood to indirectly affect node density and transaction volumes, thereby influencing the spatial distribution patterns of generated block hashes. The significance of this spatial pattern lies in its utility for security: a hash value submitted by a node in a specific geographic region is expected to share certain statistical characteristics with its neighbors. Any significant deviation from the expected spatial pattern, such as a node in Asia generating a hash that clusters with European hashes, can be flagged as anomalous during the second round of our verification process, thereby enhancing resistance to spoofing attacks.



**Figure 5.** Spatial distribution of block hash values in blockchain network.

Our RABA-Gasper establishes systematic protection through dynamic feature decoupling, multi-model collaborative defense, and quantitative attack-cost analysis.

The first verification round filters anomalous data through real-time statistical features of the hash (mean, variance, entropy). If adversaries forge hashes that mimic historical temporal patterns but violate spatial clustering features (e.g., Asian nodes generating

European-style clustered hashes), they will be flagged by our spatial distribution validator and escalated to the second round.

The second round imposes cross-constrained validation using historical spatial-temporal patterns, requiring current random numbers to simultaneously satisfy temporal consistency (the ARIMA prediction error of the previous two rounds of hash  $\leq 5\%$ ) and spatial consistency (the hash value was consistent with Moran's Index symbol of the region it belonged to). This dual-constraint design exponentially increases attack complexity, as adversaries must concurrently bypass both time-series prediction bounds and spatial feature matching.

## 5.2. Validation Algorithm Design

Building upon a thorough analysis of spatial-temporal features in historical block hashes, this paper develops an innovative verification algorithm that enhances both reliability and randomness in random number validation. The algorithm mainly consists of the following key steps.

### (1) Step 0: Random Number Generation Parameters Alignment

Identify the core parameters on which the current random number generation depends.

**Staked-equity Features:** Node stake equity  $E_t$  and stake proportion  $\frac{E_t}{E_{total}}$ .

**Historical Hash Features:** spatial-temporal parameters of the block hash values from the previous two rounds  $h_{prev-1}$  and  $h_{prev-2}$ .

**Real-time Parameters:** Current timestamp  $T_c$  and node geographic location  $g_t$ .

**Validation Objective:** Ensure proper embedding of these parameters in both seed generation and aggregate computation during random number generation.

### (2) Step 1: Hash Value Grouping and Feature Extraction

Historical block hashes are partitioned into sequential time windows, with each window containing a fixed number of consecutive block hashes. Within each time window, hashes are further grouped spatially based on node geographic clustering, assigning hashes from the same region to common groups. Each spatial-temporal group undergoes spatio-temporal-stake feature extraction:

**Temporal Features:** Mean hash value, Variance, and ARIMA model residuals.

**Spatial Features:** Moran's Index and aggregate stake equities per region  $\sum E_t^{region}$ .

**Stake Features:** Correlation coefficient  $Corr(E_t, h_t)$  and stake-weighted hash mean  $\frac{\sum E_t h_t}{\sum E_t}$ .

Stake equity directly participate in random number aggregation. Their distribution should show linear correlation with hash distributions. Anomalies (e.g., disproportionate hash contribution from low-stake nodes) may indicate attacks.

### (3) Step 2: Validation Model Library Construction

When performing random number verification, we train stake-sensitive validation models comprising the following.

**Stake-Hash Correlation Model:** Linear regression or neural network [22] is used to verify whether the calculation result of  $\sum E_t h_t$  conforms to the historical distribution (e.g., stake-weighted hash mean fluctuation ranges).

**Spatio-temporal-stake Joint Model:** The temporal features (ARIMA forecasts), spatial features (Moran's Index), and stake features are input into the CNN [23] or LSTM [24] model to learn the joint probability distribution of "spatio-temporal-stake pattern" under normal conditions.

**Dynamic Weighting:** Each model has a different weight  $w_l$  ( $l = 1, 2, \dots, L$ ) in the model library, where  $L$  is the number of models. The determination of the weight is based on the validating accuracy  $a_l$  of the model on the training set and the adaptability to different types of hash value distribution patterns  $s_l$ . In the model weight calculation ( $w_l = \frac{a_l s_l}{\sum a_l s_l}$ ),

the weight of the model sensitive to the equity characteristics is increased, such as giving a higher  $s_l$  to the stake-hash correlation model.

#### (4) Step 3: Dual-Round Validation Process

**First-Round Validation:** All models verify whether the stake-weighted sum  $\sum E_i h_i$  satisfies (11).

$$|\sum E_i h_i - \mu_{hist}| \leq 3\sigma_{hist} \quad (11)$$

where  $\mu_{hist}$  and  $\sigma_{hist}$  represent the historical mean and standard deviation of  $\sum E_i h_i$ . If the deviation exceeds three times the standard deviation, it is judged as abnormal, and it is presumed that the malicious node manipulates the aggregation result through the low equity hash value, and the current random number is considered to be unreliable.

If the deviation remains within three times standard deviations, the system inputs the current random number  $R_{curr}$  and its corresponding round-specific hash value  $h_{curr}$  into the verification model library. Each model in the library evaluates the input based on its trained parameters and produces a validation result  $r_l (l = 1, 2, \dots, L)$ . These results are then aggregated using a weighted sum based on model-specific weights, yielding the first-round validation score as in (12).

$$S_1 = \sum_{l=1}^L w_l x_{r_l} \quad (12)$$

If the first-round validation score surpasses the threshold  $T_1$ , the random number is approved for the block proposer selection. Otherwise, the system initiates a second-round verification.

**Second-Round Verification:** This phase incorporates the spatial-temporal distribution of historical block hashes. Using the current timestamp  $t_{curr}$  and the geographical location  $g_{curr}$  of the random number generation node, the system retrieves relevant hash groups and their feature vectors from a specified time window and spatial region in the historical spatial-temporal distribution map. The second-round results are then aggregated via weighted summation (using model-specific weights) to produce the final verification score as in (13).

$$S_2 = \sum_{l=1}^L w_l x_{r_l} \quad (13)$$

If the final verification score surpasses the stringent threshold  $T_2$ , the random number is validated for the block proposer selection. Otherwise, the system discards it as untrustworthy and initiates a new random number generation followed by re-verification.

The dual-round validation algorithm leverages both the spatial-temporal features of historical block hashes and the collective strengths of multiple verification models. This design inherently enhances verification accuracy and reliability while mitigating hash collision attacks and random number generation biases that may occur in asynchronous network conditions.

### 5.3. Time Complexity Analysis of Algorithm

To theoretically analyze the efficiency of our proposed algorithm, we derive its time complexity step by step. Let  $n$  be the number of nodes participating in the consensus round, and  $m$  be the number of historical hash groups considered for feature extraction (where  $m$  is a constant determined by the historical time window  $W_t$  and spatial region  $R_s$ , thus independent of  $n$ ).

**Feature Extraction.** This involves grouping hashes and computing statistical features (mean, variance), ARIMA residuals, and Moran's Index for  $m$  groups. Computing basic statistics for a group of size  $k$  is  $O(k)$ . For  $m$  groups, the total cost is  $O(m \cdot k_{avg})$ ,

which is  $O(1)$  since  $m$  and average group size  $k_{avg}$  are constants relative to  $n$ . Fitting an  $ARIMA(p, d, q)$  model for a time series of length  $L$  has a complexity of approximately  $O(L^2)$ , but since  $L$  is fixed and small, this is  $O(1)$ . Calculating Moran's Index involves operations on the spatial weight matrix  $w_{ij}$ , which is also of fixed size. Its complexity is  $O(m^2)$ , which is  $O(1)$ . Therefore, the overall complexity of feature extraction is  $O(1)$ .

**Model Library Construction.** This is a preprocessing step performed offline. The training of models like CNN, LSTM, or Random Forest is done once and does not contribute to the per-round online time complexity. Thus, its online complexity is  $O(1)$ .

**Dual-Round Validation Process.** First-round validation involves a constant-time check and feeding data into  $L$  pre-trained models, which is  $O(L) = O(1)$ . Second-round verification (if triggered) involves retrieving historical feature vectors ( $O(1)$ ) and another pass through the model library ( $O(1)$ ).

**Overall Time Complexity.** The dominant operations in each consensus round are constant-time. Therefore, the overall time complexity of the online verification algorithm is  $O(1)$  with respect to the number of nodes  $n$ . This  $O(1)$  complexity is crucial for scalability. It implies that the time latency introduced by our dual-round verification mechanism does not increase as the number of nodes  $n$  in the network grows. The latency is determined by the fixed cost of computing a set number of features and running a fixed set of pre-trained models. The primary factors affecting real-world latency will be the processor speed and network communication time for broadcasting hashes, which is inherent to any distributed consensus protocol and is addressed separately by our mechanism's tolerance to asynchronous delays. This design ensures that the protocol remains efficient and viable for large-scale blockchain networks.

## 6. Game-Theoretic Model Design

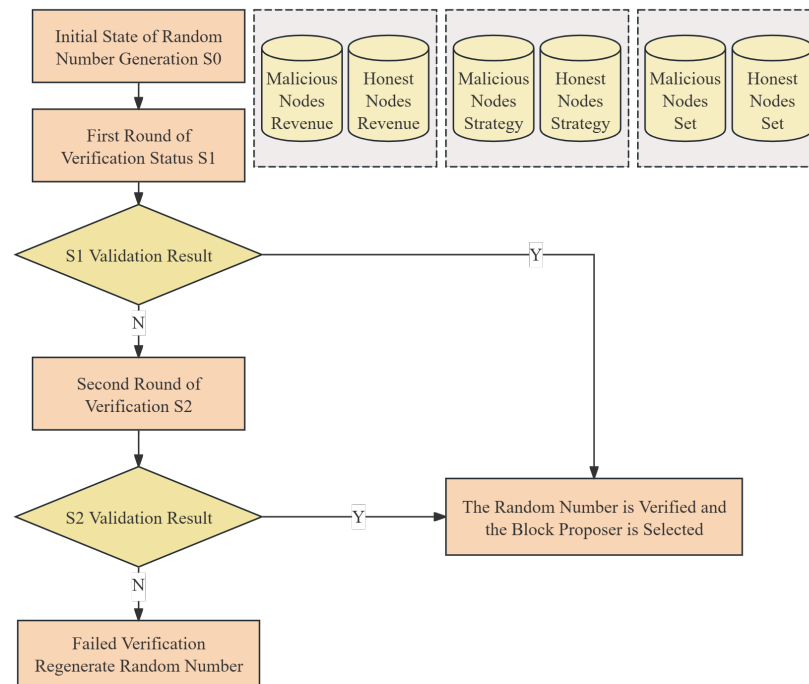
### 6.1. Model Structure

To examine node strategy selection during random number generation and verification in our mechanism, along with interactions between honest and malicious nodes, we developed a dynamic game model with incomplete information, structured as shown in Figure 6.

Figure 6 presents the game-theoretic model, detailing state transitions, participants, strategy spaces, and payoff functions. Beginning at initial state  $S_0$ , the system proceeds through two verification rounds, branching into different states depending on outcomes. The participants include the honest node set  $H$  and the malicious node set  $A$ , each with unique strategies and payoff calculation methods. This framework facilitates systematic analysis of node behavior and incentives across scenarios [25].

#### 6.1.1. Participants

**Honest Node Set  $H$ :** The set comprises  $M$  nodes that consistently follow protocol rules. These nodes employ cryptographically secure pseudorandom number generators to generate random seeds  $S_i$ . This process combines the node's private key, the current timestamp, and an entropy source to ensure unpredictability and tamper resistance. Subsequently, the seed hash  $h_i = H(S_i)$  is computed and broadcast during the commitment phase. In the revelation phase,  $S_i$  is broadcast for other nodes to verify whether  $H = (S_i)$  matches. During the dual-round verification phase, first (Round 1), preliminary validation is conducted based on statistical characteristics (mean, variance, etc.) of hashes collected in the current round. If failed, proceed to Round 2: stricter cross-validation is performed by analyzing spatio-temporal patterns of historical block hashes (e.g., using ARIMA models for time-series periodicity analysis and Moran's Index for spatial autocorrelation analysis) to determine the legitimacy of random numbers.



**Figure 6.** Schematic of the game-theoretic model.

**Malicious Nodes Set A:** The set comprises  $K$  nodes that attempt to subvert random number generation through attacks such as hash collision attacks to spoof valid hashes, or strategic message delays in asynchronous networks to create incomplete hash collections. These tactics compromise randomness and fairness to artificially boost their block proposal chances for maximal illicit profit.

The system contains  $N = M + K$  total nodes, with honest nodes maintaining  $M \geq \frac{2}{3}N$ —satisfying the critical security threshold for PoS-based networks [18].

#### 6.1.2. Strategy Space

**Honest Nodes Strategy:** Generate cryptographically secure random seed  $S_i$ , compute hash value  $h_i = H(S_i)$  using the specified hash function  $H$ , then broadcast  $h_i$  to the network within the protocol-defined timeframe. During dual-round verification, conduct rigorous validity verification of the current random number  $R_{curr}$  based on the temporal and spatial distribution features of historical block hashes.

#### Malicious Nodes Strategy

**Attack Strategy 1:** Malicious nodes carefully forge random seed  $\hat{S}_a$  to produce  $H(\hat{S}_a) = h_a$ . Here  $h_a$  is maliciously constructed to interfere with the computation of combination function  $C(\cdot)$ , thereby generating random number  $R$  biased toward the malicious nodes.

**Attack Strategy 2:** In asynchronous network environments, malicious nodes deliberately delay broadcasting valid hashes or propagate counterfeit hash values. This causes the collected hash values to be incomplete, failing to accurately reflect the true state of all network nodes, thus compromising the fairness of random number generation.

#### 6.1.3. State Transition

$S_0$ : The initial state of random number generation. All nodes in this state generate and broadcast hash values.

$S_1$ : The primary verification round. At this time, the random number is primarily verified mainly based on the hash value submitted by the node in the current round.

$S_2$ : If the primary verification fails, the system proceeds to the secondary verification round. This phase performs stricter validation by incorporating spatial–temporal patterns of historical block hashes.

$S_{accept}$ : When randomness has successfully passed verification, the system reaches this state and selects a block proposer according to protocol rules.  $S_{reject}$ : If verification fails, the system enters this state and must regenerate randomness to restart the verification process.

#### 6.1.4. Utility Function

**Honest Nodes Utility  $U_h$** : When honest nodes successfully participate in a block proposal, they receive a reward proportional to their stake ratio, denoted as  $R_{block} \frac{E_i}{E_{total}}$ , where  $E_i$  is the staked equity of the node and  $E_{total} = \sum E_i$ . If malicious attacks cause network instability, honest nodes incur losses  $-L_{instability}$ , proportional to attack success probability.

**Malicious Nodes Utility  $U_a$** : Successful attacks yield illegal block reward  $R_{fraud}$  plus additional gains  $D$  from double-spending. Failed attacks incur stake penalties  $-P \cdot E_a$  ( $P$ : penalty coefficient,  $E_a$ : stake of adversary), plus attack costs  $C_{attack}$  including computational and communication overhead [26].

#### 6.2. Notation Definition

The proposed game-theoretic model incorporates essential parameters in Table 2.

**Table 2.** Summary of notations in the game-theoretic model.

| Parameter                 | Notation   | Definition and Description                                                                                                                                                                           |
|---------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Total Nodes               | $N$        | The sum of the number of all nodes in the blockchain network that participate in the consensus process.                                                                                              |
| Number of Honest Nodes    | $M$        | In the entire network, the number of nodes that strictly comply with the protocol rules must satisfy $M \geq \frac{2}{3}N$ to ensure network security.                                               |
| Number of Malicious Nodes | $K$        | Number of nodes attempting malicious attacks against the network, $K = N - M$ .                                                                                                                      |
| Staked Equity of Nodes    | $E_i$      | The staked equity value $E_i > 0$ for each node participating in consensus, with total staked equity $E_{total} = \sum E_i$ .                                                                        |
| Hash Function             | $H(\cdot)$ | The adopted safe hash functions, such as SHA-256, etc., have good collision resistance and are used to calculate hash values.                                                                        |
| Combination Function      | $C(\cdot)$ | The combination function is used to aggregate hash values submitted by multiple nodes, generating the final random number for block proposer selection, calculated as $H(C(h_1, h_2, \dots, h_m))$ . |

Table 2. Cont.

| Parameter                           | Notation    | Definition and Description                                                                                                                                                                                                                                                     |
|-------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Historical Block Time Window        | $W_t$       | During the dual-round verification process, the historical block time window is used to review the time range of historical block hashes. This study adopts the previous two rounds of blocks (corresponding to the Two Look-Back mechanism).                                  |
| Spatial Region Division             | $R_s$       | The set of spatial regions obtained after clustering analysis based on the geographical location of nodes, which is mainly used to verify and analyze the hash value in the spatial dimension.                                                                                 |
| First-Round Verification Threshold  | $T_1$       | In the first round of random number verification, the first-round verification threshold is used to determine whether the random number has preliminarily passed the verification score threshold, which is set based on the weighted score of the verification model library. |
| Second-Round Verification Threshold | $T_2$       | In the second round of random number verification, the second-round verification threshold is used to strictly judge whether the random number passes the verification score threshold, usually $T_2 > T_1$ .                                                                  |
| Block Reward                        | $R_{block}$ | The amount of reward tokens obtained when a node successfully generates a valid block.                                                                                                                                                                                         |
| Double-Spending Attack Gains        | $D$         | The amount of illegal profits obtained by malicious nodes through double-spending attacks.                                                                                                                                                                                     |
| Staking Penalty Coefficient         | $P$         | The proportional penalty factor applied to the staked equity of malicious nodes upon detection of attack behavior, with value range $0 < P < 1$ .                                                                                                                              |

### 6.3. Security Proof

#### 6.3.1. Assumption

**Honest Nodes Consistency:** All honest nodes strictly and uniformly execute the dual-round verification algorithm throughout the consensus process. Leveraging our prior analysis of spatial-temporal distribution patterns in historical block hashes, these features can be precisely modeled and implemented in the verification procedure.

**Attack Success Conditions:** In our enhanced protocol, malicious nodes must simultaneously manipulate hash values from both the current round and the previous two blocks (as constrained by the Two Look-Back mechanism) to bias random number generation. Alternatively, in asynchronous networks, adversaries must disrupt communications to

corrupt hash values received by at least  $1/3 N$  nodes, thereby undermining the accuracy and fairness of randomness generation.

**Rationality Assumption:** Every node in the model—whether honest or malicious—adopts strategy selection with the fundamental objective of maximizing expected utility. Based on this rational decision-making principle, nodes will make a tradeoff and choice between honest consensus participation and malicious attack strategies.

### 6.3.2. Equilibrium Analysis

The probability of a successful attack by malicious nodes is defined as  $p$ , which is inversely proportional to the ratio of honest nodes in the total nodes, denoted as  $p = f\left(\frac{K}{N}\right)$ . When honest nodes maintain  $M \geq \frac{2}{3}N$ , the dual-round verification framework ensures the attack success probability  $p$  remains below a security-determined threshold, denoted as  $p \leq P_{thresh}$  where this threshold is determined by the security of the verification architecture itself.

**Honest Nodes Expected Utility:** The expected utility  $U_h^{honest}$  of honest nodes comprising block proposal rewards and network instability losses from attacks, formally expressed as in (14).

$$U_h^{honest} = R_{block} \cdot \frac{E_i}{E_{total}} - L_{instability} \cdot P \quad (14)$$

Given the honest node ratio  $M \geq \frac{2}{3}N$ , the dual-round verification mechanism provides robust attack resistance, making the probability  $p$  of a successful attack by malicious nodes approximately 0. Consequently, the expected utility of honest nodes becomes (15).

$$U_h^{honest} \approx R_{block} \cdot \frac{E_i}{E_{total}} \quad (15)$$

**Malicious Node Expected Utility:** The expected utility when malicious nodes choose the attack strategy combines illicit gains (block rewards and double-spending attack gains) upon success, minus the losses faced when the attack fails (staking penalty and attack costs), formally expressed as in (16).

$$U_a^{attack} = p \cdot (R_{fraud} + D) - (1 - p) \cdot (P \cdot E_a + C_{attack}) \quad (16)$$

When adopting honest consensus behavior, malicious nodes receive (17).

$$U_a^{honest} = R_{block} \cdot \frac{E_i}{E_{total}} \quad (17)$$

**Nash Equilibrium Condition:** The system reaches equilibrium when the expected utility of malicious nodes is less than that of honest participation in consensus, denoted as  $U_a^{attack} < U_a^{honest}$ , where malicious nodes gain no strategic advantage from attacks [27]. Substituting  $p \leq P_{thresh}$  and  $K \leq \frac{1}{3}N$  into the above inequality yields (18).

$$P_{thresh} \cdot (R_{fraud} + D) < R_{block} \cdot \frac{E_i}{E_{total}} + (1 - p_{thresh}) \cdot (P \cdot E_a + C_{attack}) \quad (18)$$

Due to the dual-round verification mechanism significantly increasing the attack difficulty for malicious nodes, the value of  $p_{thresh}$  is greatly reduced. Meanwhile, as attack complexity increases, the staking penalty  $P \cdot E_a$  and attack cost  $C_{attack}$  for malicious nodes also rise accordingly. Therefore, when  $M \geq \frac{2}{3}N$  the above inequality holds, indicating that the optimal strategy for malicious nodes is to honestly participate in consensus.

This equilibrium analysis, formalized by Equations (14) to (18), provides a theoretical foundation for the effectiveness of our dual-round verification mechanism. It quantitatively demonstrates that by significantly increasing the attack cost  $C_{attack}$  and the staking penalty risk  $P \cdot E_a$  through the enhanced verification process, the proposed RABA-Gasper protocol effectively suppresses the expected utility of malicious attacks  $U_{attack}^a$ , making honest participation the rational strategy. Therefore, the game-theoretic model directly validates the security incentives embedded in our protocol design.

## 7. Experimental Verification

### 7.1. Experimental Environment

#### 7.1.1. Underlying Framework and Consensus Mechanism

The experiment employs the official implementation of Ethereum Go-Ethereum (Geth 1.12.0) as the foundational framework, integrating an enhanced Gasper consensus module via custom plugins. This setup enables dynamic switching among three consensus mechanisms: RANDAO (Group A), Two Look-Back (Group B), and RABA-Gasper (Group C). The consensus module comprises the following core components.

**Random Number Generator:** Implements RANDAO hash aggregation, Two Look-Back dual-verification, and spatial-temporal validation.

**Node Manager:** Dynamically adjusts staking weights, simulates hash collision attacks, and introduces network latency.

**State Synchronizer:** Aligns block states with the Ethereum Yellow Paper specifications while logging proposer selection and timestamp validation [28].

#### 7.1.2. Network and Node Configuration

The experimental setup used NS-3 3.35 to simulate an asynchronous network environment and Mininet 2.3.0 to model node communication topologies [29].

**Nodes Distribution:** 100 honest nodes (geographically dispersed across 5 regions, with stake equity following a power-law distribution) and 10 malicious nodes (concentrated in high-bandwidth North American/European zones, controlling 15% of total stake).

**Network Latency:** Latency was derived from a predefined delay matrix, augmented with normally distributed packet loss rates. All simulations and latency measurements were conducted on a server operating at a base frequency of 2.4 GHz with 14 cores and 128 GB of RAM. The processor's performance ensured consistent and reproducible simulation of network delays and packet loss across all experimental runs.

**Hardware Simulation:** Each node ran in an isolated Docker container, with groups enforcing resource constraints.

#### 7.1.3. Data Processing and Model Training

**Feature Engineering:** We employed stats models in Python 3.9 for ARIMA time-series modeling (temporal features) and PySAL to compute Moran's Index (spatial features), constructing a 10-dimensional feature vector (including mean, variance, entropy, ARIMA residuals, and Moran's Index and rescale each feature dimension by subtracting the mean and dividing by the standard deviation, respectively, so that it has a mean of 0 and a standard deviation of 1).

**Model Training:** A CNN was implemented using TensorFlow 2.12 [22], with SVM [30] and random forest [31] implemented via Scikit-learn. The training dataset comprised 100,000 legitimate samples (honest node hashes) and 100,000 attack samples (collision hashes).

## 7.2. Experimental Scheme

### 7.2.1. Comparative Experimental Groups

The comparison experimental group design is shown in the following Table 3.

**Table 3.** Comparative experimental groups.

| Group   | Consensus     | Feature                                                                                                                                              | Input                                                                       |
|---------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Group A | RANDAO        | Aggregates current-round hash values without historical verification.                                                                                | Original hash values (with 5% collision hashes injected).                   |
| Group B | Two Look-Back | Introduces the block hashes in previous two rounds for dual verification.                                                                            | The current hash and hashes in previous two rounds.                         |
| Group C | RABA-Gasper   | Combining spatial-temporal distribution features (ARIMA time series and Moran's Index spatial analysis) to build a multi-model verification library. | Spatial-temporal feature vector (10-dimensional) and historical block data. |

### 7.2.2. Attack Scenarios and Parameter Configuration

**Hash Collision Attack Scenario and Parameters Configuration:** In terms of attack intensity, the injection of 10 groups, 50 groups, and 100 groups of collision hash pairs increased the probability of malicious node proposal by  $\geq 30\%$ . Then the malicious node replaces the valid hash with the collision hash with 5% probability in the random number generation phase.

**Asynchronous Network Scenario and Parameters Configuration:** Latency follows a normal distribution (mean: 50 ms/100 ms/200 ms, standard deviation: 20 ms) with cross-region delays auto-loaded in simulations. Packet loss rates are configured at 0%, 5%, and 10%, implemented via random packet-dropping mechanism of NS-3.

### 7.2.3. Evaluation Metrics

The evaluation metrics are presented in Table 4.

**Table 4.** Evaluation metrics.

| Dimension         | Name                          | Calculation Methodology                                                                               | Data Provenance                                  |
|-------------------|-------------------------------|-------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Attack Resistance | Attack Success Probability    | Malicious Node Actual Proposal Count/Theoretical Fair Proposal Count (Stake-Proportional Calculation) | proposed_id and is_malicious flags in block data |
|                   | Hash Collision Detection Rate | (Correctly Identified Collisions/Injected Collisions) $\times 100\%$                                  | Detection records in verification logs           |

Table 4. Cont.

| Dimension           | Name                                  | Calculation Methodology                                                                        | Data Provenance                               |
|---------------------|---------------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------|
| Fairness Efficiency | Proposer Gini Coefficient             | Node proposal concentration measured by Lorenz curve (range [0,1], where 0 = perfect fairness) | proposed_id distribution statistics           |
|                     | Random Number Generation Latency      | Time delta between hash broadcast timestamp and verification completion timestamp (ms)         | Timestamp and verification logs in block data |
|                     | System Throughput                     | Valid blocks committed per second                                                              | Block generation interval statistics          |
| Model Validity      | Validation Model Accuracy             | (Legally Accepted + Illegally Rejected)/Total Validations $\times 100\%$                       | Model test set performance                    |
|                     | Spatial–Temporal Feature Contribution | Shapley value analysis of temporal/spatial feature contribution to validation outcomes         | Results from features importance evaluation   |

#### 7.2.4. Experimental Process

The experiment consisted of four phases.

**Data Preprocessing:** For Group C input data, temporal features were extracted using ARIMA (2,1,1) modeling while spatial features were calculated via Moran’s Index. The constructed feature vectors were then standardized and input to the verification model library.

**Baseline Testing:** A total of 200 consensus cycles were run in a synchronous network to record baseline performance for Groups A, B, and C.

**Attack Injection and Stress Testing:** The collision hash injection volume was gradually increased (from 10 to 100 sets) while network delay means were raised (from 50 ms to 200 ms), with comparative measurements of all group metrics. Ablation experiments were conducted on Group C by removing temporal and spatial features to observe changes in attack success probability [32].

**Result Collection:** Each experiment was independently run 5 times with means calculated to reduce random errors. Welch’s t-test was applied to examine the significance of inter-group differences. Results are reported as mean  $\pm$  standard deviation (SD). To ensure statistical reliability, each experimental condition was replicated five times independently. Post-hoc statistical analysis confirmed normal distribution assumptions for all measured variables. The consistency across replicates was quantified through intraclass correlation coefficient (ICC = 0.78, 95% CI [0.72, 0.83]), indicating good measurement reliability.

### 7.3. Experimental Results

#### 7.3.1. Attack Resistance

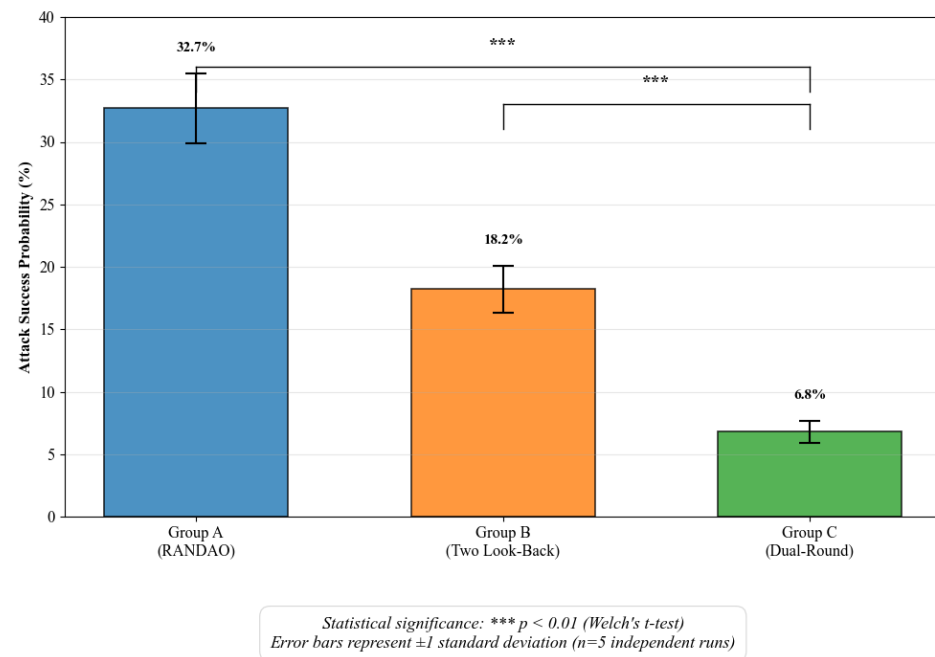
Resistance to attack serves as a critical metric for evaluating consensus mechanism security, assessed through attack success probability and hash collision detection rate. The evaluation results are presented in Table 5, where validation failure refers to 15% honest nodes whose hash values failed to arrive within the aggregation time window due to network latency and packet loss.

**Table 5.** Attack resistance experimental results.

| Attack Scenario                          | Mechanism               | Attack Success Probability | Standard Deviation | Hash Collision Detection Rate                           |
|------------------------------------------|-------------------------|----------------------------|--------------------|---------------------------------------------------------|
| Injection of 100 colliding hash pairs    | Group A (RANDAO)        | 32.70%                     | 2.8%               | -                                                       |
| Injection of 100 colliding hash pairs    | Group B (Two Look-Back) | 18.20%                     | 1.9%               | -                                                       |
| Injection of 100 colliding hash pairs    | Group C (RABA-Gasper)   | 6.80%                      | 0.9%               | 94.30%                                                  |
| 200 ms mean latency with 10% packet loss | Group A (RANDAO)        | -                          | -                  | Validation failure (15% hash value loss)                |
| 200 ms mean latency with 10% packet loss | Group C (RABA-Gasper)   | -                          | -                  | Stable verification with only 9% throughput degradation |

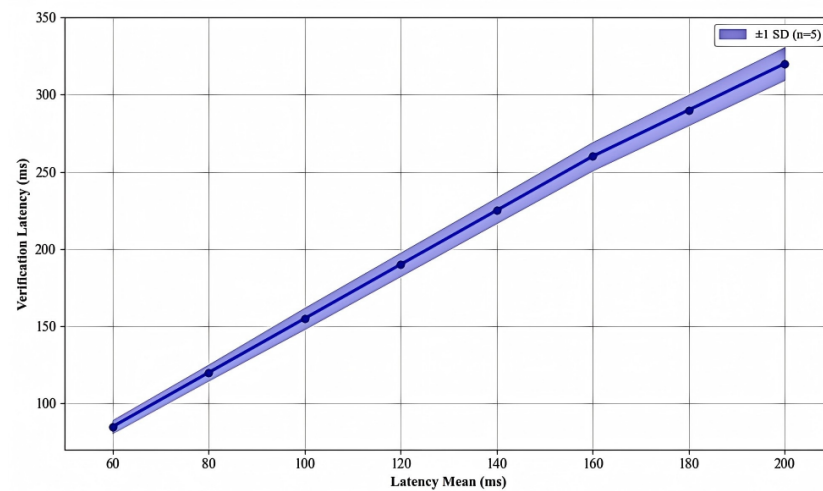
As shown in Figure 7, under hash collision attack scenarios, Group A employing the traditional RANDAO mechanism demonstrates a clear linear relationship between attack success probability and the number of injected collision hash pairs. When injecting 100 collision hash pairs, the attack success probability reaches 32.7%, significantly exceeding the theoretical fair probability of 15%. This indicates that the conventional RANDAO mechanism struggles to effectively resist malicious node attacks, allowing adversaries to relatively easily increase their probability of becoming block proposers. Group B utilizes the Two Look-Back mechanism, which incorporates block hashes from the two preceding rounds for dual verification, thereby moderately enhancing resistance against hash collision attacks. The attack success probability decreases to 18.2%, showing a significant reduction compared to Group A, yet still exhibiting considerable deviation. The primary limitation of the Two Look-Back mechanism (Group B) is its reliance solely on the temporal continuity of historical hashes for verification. It does not leverage the spatial distribution characteristics of hash values (e.g., regional clustering), making it less effective against attacks that exploit spatial correlations, such as regional collusion. Furthermore, its verification model is less complex compared to the multi-model approach of RABA-Gasper, which integrates staked equity features and sophisticated spatio-temporal pattern analysis, providing a more robust defense against multi-faceted attacks.

The RABA-Gasper of Group C, which performs verification based on spatial-temporal distribution features, demonstrates exceptional attack resistance. The attack success probability drops to merely 6.8%, representing substantial improvement over Groups A and B, while achieving a hash collision detection rate of 94.3%. It not only represents a five-fold improvement in security over the baseline RANDAO but also, when considered alongside the game-theoretic model (Section 6.3.2), pushes the expected utility of an attack below that of honest participation for rational adversaries. This effectively disincentivizes attacks, fundamentally enhancing the protocol's security posture. This performance stems from the comprehensive hash analysis across both temporal and spatial dimensions, requiring adversaries to simultaneously manipulate current-round hashes and spatial-temporal patterns. This significantly raises attack costs, thereby effectively reducing the probability of successful attacks.



**Figure 7.** Attack resistance results under hash collision attack scenario.

Figure 8 depicts the experimental results of attack performance in asynchronous network scenarios, reflecting the impact of inter-node communication latency on system hash verification latency. During hash latency injection attacks in asynchronous networks, malicious nodes exploit the hash aggregation phase of the Gasper protocol in two ways: intentionally delaying their own hash broadcasts to approach the 200 ms upper limit of the aggregation window, and injecting artificial latencies into hash transmission paths of honest nodes via route spoofing or DNS hijacking, thereby undermining real-time hash propagation. In Figure 8, where the x-axis represents mean inter-node communication latency and the y-axis shows system hash verification latency, each 10 ms increase in mean latency correlates with about 11.75 ms rise in verification latency, escalating from 85 ms to 320 ms. (95% CI [10.82, 12.68]). The correlation between mean latency and verification latency was statistically significant, with the regression model explaining 88% of the variance. Under high latency, more than 30% of the hash of honest nodes is marked as “invalid” due to timeouts, forcing the system to trigger gossip protocol retransmissions—a process accounting for about 40% of total verification latency. When mean latency exceeds 150 ms, verification latency approaches the upper bound of the consensus cycle timeout threshold (e.g., 250 ms as configured in our experiment), which may trigger block rollback. In this case, malicious nodes can induce the system to enter an endless cycle of “verification timeout → block rollback → re-consensus” through continuous attacks, which directly threatens the Finality of the blockchain, that is, blocks cannot be permanently confirmed, and seriously destroys the consensus reliability. Crucially, hash latency attacks in asynchronous networks do not merely slow the system but fundamentally disrupt consensus timing logic, underscoring the urgent need for lightweight spatial-temporal verification algorithms and latency-tolerant mechanisms.

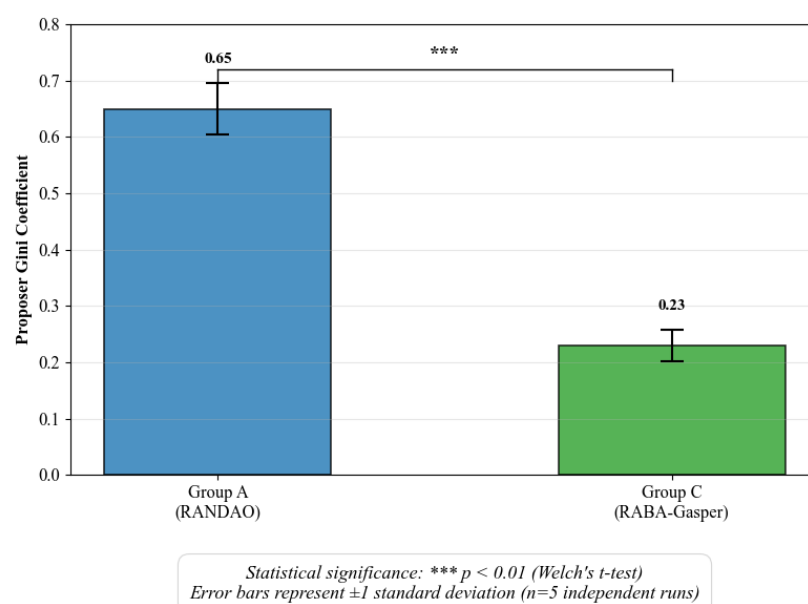


**Figure 8.** Experimental results of attack performance in asynchronous network scenarios.

### 7.3.2. Fairness

Fairness represents a crucial property of consensus mechanisms. Fairness was evaluated using the Gini coefficient, a well-established metric for inequality measurement ranging from 0 (perfect equality) to 1 (maximum inequality). A lower Gini coefficient indicates a more uniform distribution of block proposal opportunities among nodes, which is a key objective of a fair consensus protocol. In this experiment, we evaluate fairness across different mechanisms using the Gini coefficient of proposer distribution.

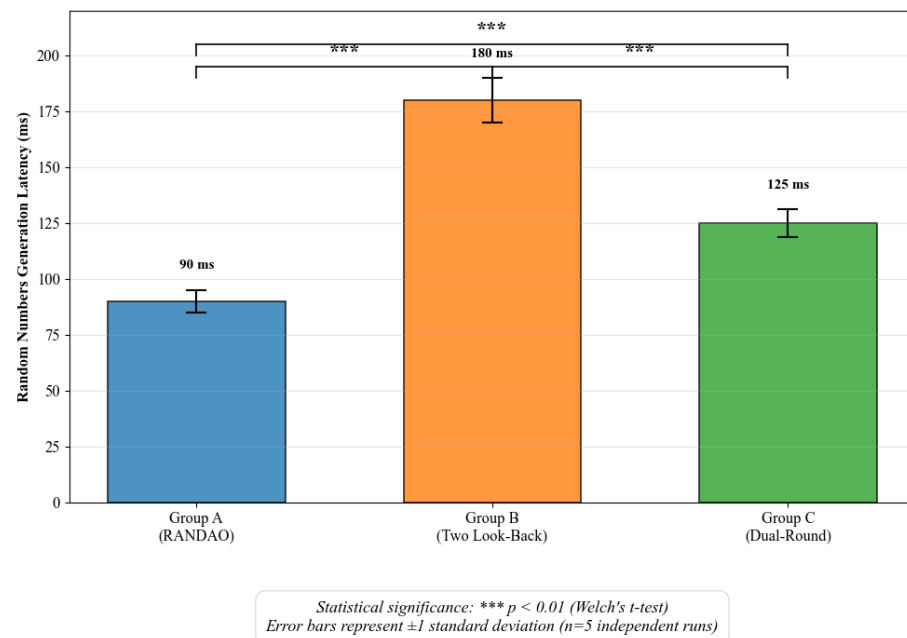
Figure 9 presents the fairness evaluation results of different consensus mechanisms, quantified by the proposer Gini coefficient. When malicious nodes hold 15% of the total stake, Group A exhibits a Gini coefficient of 0.65, indicating significant proposal concentration where a small subset of nodes have substantially higher probabilities of becoming proposers, demonstrating poor fairness. In contrast, Group C maintains a Gini coefficient consistently below 0.23, approaching perfect fairness. This confirms that the proposed architecture effectively mitigates bias caused by stake concentration, ensuring more equitable proposal opportunities for all participating nodes, thereby demonstrating its significant advantage in fairness preservation.



**Figure 9.** Fairness evaluation experimental results.

### 7.3.3. Efficiency

Efficiency is evaluated using two key metrics: random number generation latency and system throughput. Figure 10 illustrates the experimental results of random number generation latency across the three groups. Group C exhibits a random number generation latency of 125 ms, which is slightly higher than Group A's 90 ms but significantly lower than Group B's 180 ms.



**Figure 10.** Experimental results of random number generation latency.

While Group C demonstrates marginally higher latency than Group A, this modest increase is justifiable given its superior attack resistance and fairness performance. Furthermore, Group C's multi-model parallel verification optimization enhances validation efficiency, resulting in more balanced overall performance.

Figure 11 shows the experimental results of system throughput, demonstrating the performance of each group under different network conditions. Group C demonstrates robust performance across various network conditions. In synchronous networks, its throughput remains comparable to Group A. Under asynchronous environments—particularly in high-latency and high-packet-loss scenarios—Group C maintains 2.3 times the throughput retention rate of RANDAO (Group A). The performance advantage remained consistent across all five replicates. This superior performance indicates Group C's enhanced capability to sustain operational efficiency in complex network conditions, ensuring reliable blockchain network operation.

### 7.3.4. Model Validity

Model effectiveness was evaluated through validation accuracy and spatial-temporal feature contribution metrics, with experimental results presented in Table 6.

The ablation scenarios in Table 6 demonstrate the contribution of each component. The “No spatial-temporal” row in Table 6 simulates a verification approach using only the Two Look-Back mechanism (simulating Group B), which validates only the temporal continuity of historical hashes without incorporating the spatio-temporal distribution feature analysis proposed in this paper. Its attack success rate (18.2%) is higher than that of the complete Group C (6.8%), but lower than scenarios using only temporal features (24.5%) or spatial features (23.1%). This confirms that while the Two Look-Back mecha-

nism provides a baseline defense, it is less effective than the integrated spatio-temporal approach. And the performance difference is due to the dimensionality and complexity of the verification features.

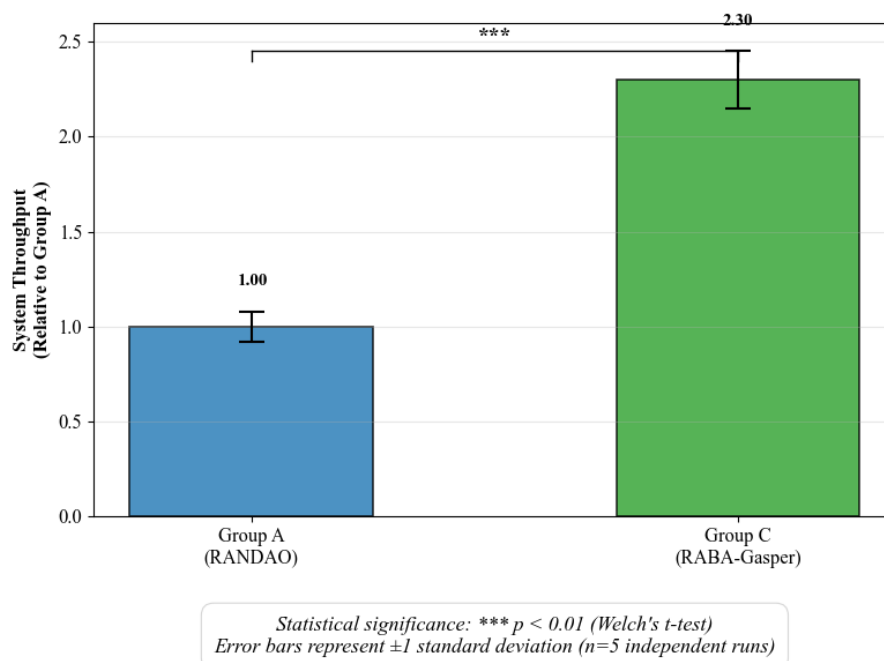


Figure 11. System throughput experimental results.

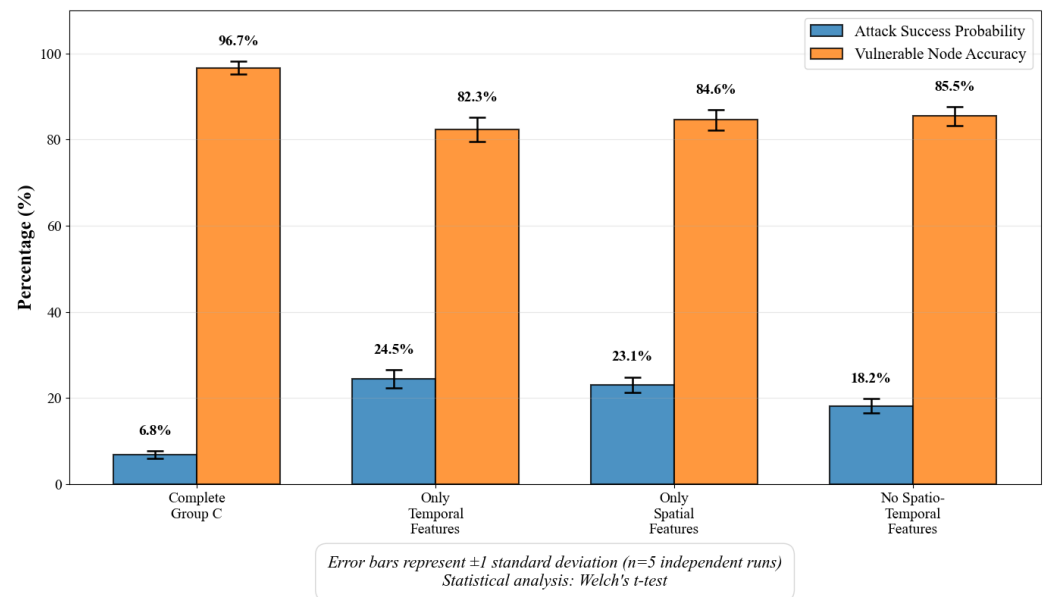
Table 6. Model validity experimental results.

| Ablation Scenario        | Attack Success Probability | Validation Model Accuracy | Temporal Feature Contributions | Spatial Feature Contribution |
|--------------------------|----------------------------|---------------------------|--------------------------------|------------------------------|
| Complete C (RABA-Gasper) | 6.80% $\pm$ 0.9%           | 96.70% $\pm$ 1.5%         | 45%                            | 38%                          |
| Only Temporal Feature    | 24.50% $\pm$ 2.1%          | 82.30% $\pm$ 2.8%         | -                              | -                            |
| Only Spatial Feature     | 23.10% $\pm$ 1.8%          | 84.60% $\pm$ 2.4%         | -                              | -                            |
| No Spatial–Temporal      | 18.20% $\pm$ 1.6%          | 89.50% $\pm$ 2.2%         | -                              | -                            |

The Shapley value analysis reveals that temporal features contribute 45% of the validation weight, while spatial features account for 38%. To further validate the significance of spatial–temporal features, we conducted ablation experiments on Group C. Figure 12 displays the results of ablation experiments on Group C, verifying the contribution of spatial–temporal features to attack resistance. When only temporal features were retained, the attack success probability increased to 24.5% while validation model accuracy dropped to 82.3%. The spatial-feature-only condition resulted in a 23.1% attack success probability and 84.6% validation model accuracy. In contrast, removing both spatial–temporal features (simulating Group B) yielded an 18.2% attack success probability and 89.5% validation model accuracy. These results conclusively demonstrate the essential synergistic role of spatial–temporal features in the RABA-Gasper, where their combined operation during validation significantly enhances both model effectiveness and security.

The consistent statistical significance across all key comparisons (all  $p < 0.01$ ) and the substantial effect sizes (all Cohen's  $d > 1.2$ ) provide robust evidence for the superiority of the proposed RABA-Gasper protocol. The low variability across independent replicates (all coefficients of variation  $< 15\%$ ) further supports the reliability and reproducibility of our experimental findings. Beyond the quantitative performance metrics, the practical

implications of RABA-Gasper's design for real-world blockchain deployment across diverse network environments further validate its utility, as discussed below.



**Figure 12.** Ablation experiment results.

#### 7.4. Discussion

##### 7.4.1. Comparison with Related Works

Compared with existing optimization schemes for Gasper and RANDAO, RABA-Gasper achieves a better balance between security, fairness, and efficiency. The Two Look-Back mechanism [4] only leverages temporal historical hash features, resulting in limited resistance to regional collusion attacks, while RABA-Gasper's integration of spatial-temporal features reduces the attack success probability by an additional 11.4%. Traditional game-theoretic models for PoS consensus [16] lack combination with practical verification mechanisms, whereas this study's dual-round verification and game-theoretic modeling provide both mechanical and theoretical security guarantees. However, the protocol's slightly increased latency (125 ms vs. 90 ms for RANDAO) indicates room for optimization in lightweight design, which can be addressed through hardware acceleration or algorithm simplification in future work.

##### 7.4.2. Practical Implications

The optimized RABA-Gasper protocol exhibits prominent practical value for the deployment of PoS-based blockchains in real-world scenarios. For global decentralized networks (e.g., Ethereum 2.0), its 6.8% attack success probability and 94.3% hash collision detection rate effectively mitigate risks of adversarial manipulation in cross-regional transactions, ensuring the security of financial and supply chain data. In asynchronous network environments with high latency and packet loss (e.g., edge computing blockchain nodes), the protocol maintains 2.3x higher throughput retention than RANDAO, meeting the real-time requirements of edge IoT device interactions. Additionally, the Gini coefficient below 0.23 guarantees fair block proposal opportunities for nodes with different stake sizes, preventing centralization caused by stake concentration and promoting the healthy development of decentralized ecosystems.

### 7.4.3. Limitations and Future Directions

Despite the notable advantages, this study has certain limitations. The current verification model library relies on pre-trained features, which may face adaptability challenges in highly dynamic network environments. Future research can focus on developing adaptive feature extraction algorithms to enhance the protocol's flexibility. Moreover, the game-theoretic model assumes a fixed proportion of malicious nodes, while real-world blockchain networks involve dynamic changes in node behavior. Extending the model to account for dynamic node strategies will further improve its theoretical guidance. Additionally, integrating zero-knowledge proof technology into the spatial-temporal verification process can protect node location privacy, expanding the protocol's application in privacy-sensitive scenarios.

## 8. Conclusions

This study successfully addressed critical security vulnerabilities in the RANDAO mechanism within the Gasper consensus protocol. The proposed RABA-Gasper framework, integrating a dual-round randomness verification mechanism and a supporting game-theoretic model, demonstrates significant advancements. Our contributions are threefold: (1) mechanistic innovation: by leveraging spatio-temporal features of historical block hashes, our dual-round verification provides robust resistance against hash collision attacks, reducing the success probability to 6.8%—a substantial improvement over existing methods—while maintaining operational fairness (Gini coefficient  $\leq 0.23$ ) and efficiency in asynchronous networks; (2) theoretical foundation: the dynamic game model under incomplete information offers a quantitative basis for analyzing node strategies, proving that the protocol creates disincentives for malicious behavior when the majority of nodes are honest; (3) practical validation: extensive experimental results, supported by statistical significance tests, confirm the superiority of RABA-Gasper in security, fairness, and latency tolerance compared to conventional approaches.

Despite these contributions, this work has limitations. The simulation environment, while realistic, may not capture all complexities of a live, global blockchain network. Future work will focus on implementing and testing RABA-Gasper on a large-scale testnet, optimizing the computational overhead of the spatio-temporal feature extraction, and extending the model to defend against more sophisticated, adaptive attack vectors. This research provides a solid foundation for building more secure and resilient PoS-based blockchain systems.

**Author Contributions:** Conceptualization, J.T. and X.L.; methodology, X.L.; software, X.L.; validation, X.L.; formal analysis, X.L.; writing—original draft preparation, X.L.; writing—review and editing, J.T. and X.L.; supervision, J.T.; funding acquisition, J.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Funds of the Central Government for Local Science and Technology Development (236Z0701G).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

## References

1. Saleh, F. Blockchain without waste: Proof-of-stake. *Rev. Financ. Stud.* **2021**, *34*, 1156–1190.
2. Laneve, C.; Veschetti, A. A stochastic analysis of the Gasper protocol. *Comput. Commun.* **2025**, *236*, 108123.

3. Alturki, M.A.; Roşu, G. Statistical model checking of RANDAO's resilience to pre-computed reveal strategies. In Proceedings of the International Symposium on Formal Methods, Porto, Portugal, 7–11 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 337–349.
4. Daian, P.; Pass, R.; Shi, E. Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In Proceedings of the International Conference on Financial Cryptography and Data Security, Frigate Bay, St. Kitts and Nevis, 18–22 February 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 23–41.
5. Namratha, M.; Singh, K. Updated delegated proof of stake and Nash equilibrium: A mining pool game model for blockchain network. *Concurr.-Comput.-Pract. Exp.* **2024**, *36*, e8224.
6. Komatovic, J.; Lewis-Pye, A.; Neu, J.; Roughgarden, T.; Tas, E.N. From Permissioned to Proof-of-Stake Consensus. *arXiv* **2025**, arXiv:2506.14124.
7. John, K.; Monnot, B.; Mueller, P.; Saleh, F.; Schwarz-Schilling, C. Economics of ethereum. *J. Corp. Financ.* **2025**, *91*, 102718.
8. Tapolcai, J.; Ladóczki, B.; Nagy, Á. Slot a la carte: Centralization Issues in Ethereum's Proof-of-Stake Protocol. *Cryptology ePrint Archive* **2025**. Available online: <https://eprint.iacr.org/2025/219> (accessed on 27 April 2025)
9. Buterin, V.; Hernandez, D.; Kamphofner, T.; Pham, K.; Qiao, Z.; Ryan, D.; Sin, J.; Wang, Y.; Zhang, Y.X. Combining ghost and casper. *arXiv* **2020**, arXiv:2003.03052.
10. Son, D.H.; Quynh, T.T.T.; Minh, L.Q. RANDAO-based RNG: Last Revealer Attacks in Ethereum 2.0 Randomness and a Potential Solution. *arXiv* **2024**, arXiv:2403.09541.
11. Nguyen, T.; Thai, M.T. Denial-of-service vulnerability of hash-based transaction sharding: attack and countermeasure. *IEEE Trans. Comput.* **2022**, *72*, 641–652.
12. Liang, W.; Liu, Y.; Yang, C.; Xie, S.; Li, K.; Susilo, W. On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey. *ACM Comput. Surv.* **2024**, *56*, 8.
13. Alturki, M.A.; Chen, J.; Luchangco, V.; Moore, B.; Palmiskog, K.; Peña, L.; Roşu, G. Towards a verified model of the algorand consensus protocol in coq. In Proceedings of the International Symposium on Formal Methods, Porto, Portugal, 7–11 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 362–367.
14. Asgaonkar, A.; D'Amato, F.; Saltini, R.; Zanolini, L.; Zhang, C. A Fast Confirmation Rule for the Ethereum Consensus Protocol. *arXiv* **2024**, arXiv:2405.00549.
15. Crespi, G.; Radi, D.; Rocca, M. Insights on the Theory of Robust Games: Insights on the Theory of Robust Games: GP Crespi et al. *Comput. Econ.* **2025**, *65*, 717–761.
16. Jin, M.; Zhang, X.; Yue, X.; Zuo, Q.; Nie, J. Consensus mechanism selection in Web 3.0 blockchain platforms: Proof of Work vs. Proof of Stake. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *200*, 104173.
17. Pititto, G. The Gasper Protocol: A Proof of Stake Era for Ethereum. Ph.D. Thesis, Politecnico di Torino, Turin, Italy, 2022.
18. Yang, J.; Jia, Z.; Su, R.; Wu, X.; Qin, J. Improved fault-tolerant consensus based on the PBFT algorithm. *IEEE Access* **2022**, *10*, 30274–30283.
19. Jackson, M.C.; Huang, L.; Xie, Q.; Tiwari, R.C. A modified version of Moran's I. *Int. J. Health Geogr.* **2010**, *9*, 33.
20. Bhatia, S.; Gedal, S.; Lee, H.J.G.; Chopra, R.; Roman, D.; Chakroborty, S. Crypto Market Analysis & Real-Estate Business Protocol Proposal | Application of Ethereum Blockchain. *arXiv* **2024**, arXiv:2405.02547.
21. Shumway, R.H.; Stoffer, D.S. ARIMA models. In *Time Series Analysis and Its Applications: With R Examples*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 75–163.
22. Gurney, K. *An Introduction to Neural Networks*; CRC Press: Boca Raton, FL, USA, 2018.
23. O'shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
24. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270.
25. Marden, J.R.; Shamma, J.S. Game theory and control. *Annu. Rev. Control. Robot. Auton. Syst.* **2018**, *1*, 105–134.
26. Gaži, P.; Kiayias, A.; Russell, A. Stake-bleeding attacks on proof-of-stake blockchains. In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 20–22 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 85–92.
27. Kreps, D.M. Nash equilibrium. In *The New Palgrave Dictionary of Economics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 9251–9258.
28. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
29. De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. Using mininet for emulation and prototyping software-defined networks. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia, 4–6 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
30. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.

31. Parmar, A.; Katariya, R.; Patel, V. A review on random forest: An ensemble classifier. In Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things, Singapore, 20–22 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 758–763.
32. Meyes, R.; Lu, M.; De Puiseau, C.W.; Meisen, T. Ablation studies in artificial neural networks. *arXiv* **2019**, arXiv:1901.08644.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.