IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 October 2022, accepted 8 November 2022, date of publication 10 November 2022, date of current version 16 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221419

RESEARCH ARTICLE

An On-Chain Governance Model Based on Particle Swarm Optimization for Reducing Blockchain Forks

REZA NOURMOHAMMADI AND KAIWEN ZHANG, (Member, IEEE)
Department of Software and IT, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada

Corresponding author: Reza Nourmohammadi (rezanourmohammadi583@gmail.com)

ABSTRACT There is a significant drawback associated with blockchain networks in terms of their processing speed, which is one of the biggest. Due to the fact that sharding has the capability of solving this problem, the scalability of the network can be increased. One of the significant challenges in this study was determining how sharding would affect the probability of forks arising as a consequence of sharding. Towards this end, we have performed a number of experiments on the network EIP-1559 using 120 nodes in order to achieve this objective. During our analysis, we were able to determine that the number of orphan blocks on average decreases by 60% as a result of adding a shard to the system. The new on-chain governance model has also been implemented that utilizes Particle Swarm Optimization (PSO) in order to ensure that forks between different shards are reduced, as well as the probability of them occurring. The results obtained from our study give us the confidence that the proposed on-chain governance model reduces the risks associated with forking and maintains a positive user experience as a result of the results obtained.

INDEX TERMS Blockchain, sharding, on-chain governance fork, particle swarm optimization, EIP-1559.

I. INTRODUCTION

Blockchain technology has become increasingly popular since Bitcoin was introduced in 2008 [2]. Since then, this novel technology has been a hot topic among researchers due to its decentralized nature. This revolutionary technology is most notable for its decentralization, transparency, immutability, and security. Like other technologies, blockchain faces a number of challenges. The most significant challenge in blockchain networks is the fork. This occurs when two (or more) miners propagate their blocks simultaneously. A node in this situation will accept the block that arrived earlier as the tip of its chain. As a result, two (or more) branches of the chain will grow at the same time. By selecting the longest chain, the conflict will be resolved and the remaining blocks will be considered orphans.

As a result, forking can cause security issues for the network, destroy trust among users and consequently reduce the value of the coin. There are several factors that can lead to a fork, but the main one is network delay. Therefore, new blocks may arrive at the nodes of the network sooner or later than each other as a result of network delay and traffic. Accordingly, the last block of the nodes is determined by which block arrives first.

The low transaction processing rate of blockchain systems is another challenge. In comparison, Ethereum's network can process approximately 15 transactions per second, while other payment systems, such as VISA [4], [5], are able to process thousands of transactions per second. It becomes more difficult to solve this problem as the number of nodes increases. In a nutshell, scalability is another challenge associated with blockchain networks. In order to overcome this problem, Elastico [3] introduced a concept called sharding. By sharding, the nodes are divided into groups so that each group has its own chain running concurrently. The ability to process transactions more efficiently can be achieved through the use of sharding. There are generally two types of transactions in sharded networks:

The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mueen Uddin.

- Intra-shard transactions are transactions between nodes within a single shard.
- Cross-shard transactions are transactions that are made between nodes across different shards.

A transaction executed between two nodes in the same shard is confirmed in exactly the same manner as a transaction carried out between nodes in a non-sharded network. Nevertheless, for a cross-shard transaction in which the origin and destination are in different shards, both shards must approve the transaction for it to be included in the block. Therefore, it takes more time to confirm these types of transactions.

To ensure the security and stability of a network, governance is a mechanism used for controlling and governing it. There have been several governance models introduced up to this point, which are divided into two main groups: the on-chain governance model and the off-chain governance model. By utilizing an off-chain governance model, decisions are made by developers outside the chain, which is against the very nature of decentralization in a blockchain [16]. A disagreement in off-chain governance may result in a hard fork in the network. For example, the disagreement in the Bitcoin network in 2017 on updating block size split the Bitcoin network into bitcoin and bitcoin cash.

In contrast, in the on-chain governance model, decisions are made by users through a voting process. Consequently, these types of governance models are more secure and transparent [17]. To keep the network stable in terms of fork occurrence probability and user experience, a novel on-chain governance model based on the PSO algorithm is proposed in this research.

To illustrate this, we have added the ability to simulate a sharded network with the proposed on-chain governance model to the Blocksim [1] simulator. Then, we conducted several experiments in order to determine whether sharding affects the fork probability. Lastly, we have applied our proposed on-chain governance models to two sharded networks with two and four shards respectively, in order to determine how efficiently the proposed method reduces the fork probability and keeps the network in a satisfactory state for its users. Furthermore, we have selected the EIP-1559 network for our analysis as a case study.

A summary of the main contributions of this paper is as follows:

1) Analyzing the effects of sharding on the probability of forks occurring.
2) A novel on-chain governance model based on PSO is presented.
3) Implementing a cost function for optimizing the optimization algorithm based on fork probability and user experience.

The remainder of the paper is devoted to Section II, which reviews related works. Section III provides background information on our research and Section IV presents our proposed model. We conclude the paper with the results of the experiments and the conclusions in the final two sections.

II. RELATED WORKS

In most of the previous research, no one has investigated the effect of sharding or on-chain governance on the fork occurrence probability, which is considered one of the most critical challenges facing blockchain technology. Our primary objective is to overcome this disadvantage by evaluating the probability of forks occurring in sharded networks by utilizing a novel on-chain governance model. An explanation of the proposed on-chain governance model will follow in the next section.

A. SHARDING

In [9], the authors addressed the issue of malicious nodes taking over a shard and compromising the entire network. Because a shard computes at a fraction of the network, it is more susceptible to 51% attacks. To prevent collusion between malicious nodes, their proposal calculates the trust score of all nodes based on consensus results, followed by a genetic algorithm to calculate the distribution of nodes.

Paper [10] presents a four-objective model of shard validation validity that takes into consideration invalidation probability, delay, throughput, and the load of malicious nodes. Then they applied a dynamic reward and penalty mechanism to solve this many-objective optimization problem. The proposed method reduces the possibility of malicious nodes aggregation. According to their findings, the proposed method can handle the conflict between throughput and shard validity in order to improve the security of blockchain-enabled IoT applications.

The diagram in Figure 1 illustrates the general structure of blockchain sharding. Instead of maintaining a single blockchain for all transactions, network nodes maintain several blockchains called chains of shards. A shard consists of its own nodes. These nodes implement a consensus mechanism based on a PoW, staking, voting, or a combination of those [32].

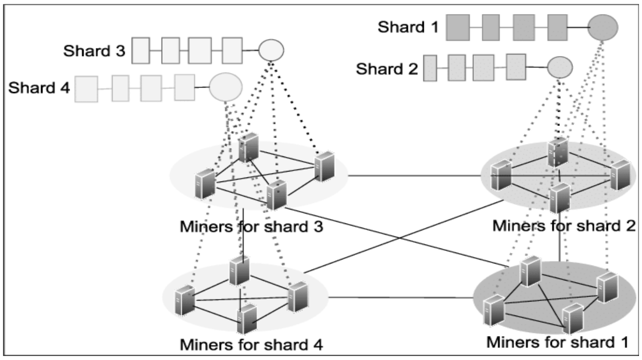


FIGURE 1. The general structure of blockchain sharding [32].

In [11], the authors employed deep reinforcement learning to design a sharded blockchain that autonomously determines its configuration for optimal throughput and security and AI(Artificial Intelligence) is used to analyze the network's performance. A number of parameters were considered in the simulation, including the maximum number of shards, the maximum block interval, the average transaction size, the number of nodes, etc. It is possible for the deep learning agent to select the block size, block interval, and shard number.

Based on a sharded blockchain, Yuan et al. Introduced a federated learning framework known as ChainsFL in [12]. The framework was designed using a Raft-based sharded blockchain architecture in order to increase its scalability. According to their results, ChainsFL has a higher convergence rate in training convolutional neural networks compared to other methods.

A blockchain framework was proposed by Yoo and Daejeon which is presented for the scalability and enhancement of domain-specific static sharding [13]. Their framework suggests that each shard is comprised of a committee that validates transactions using PBFT. As a result of their framework, the blockchain is supposed to be more effective.

Nguyen et al. Proposed a novel algorithm that analyzes the stream of network transactions in order to optimally place them in the shards in order to minimize cross-shard transactions in [14]. According to their results, OptChain reduces latency by 93% and increases throughput by 50% in comparison with OmniLedger.

To assign nodes to shards, a hybrid algorithm that combines a machine learning algorithm and a VRF (Verifiable Random Function) function is proposed in [15]. Its main objective is to reduce the likelihood that blockchains and shards will deteriorate.

B. GOVERNANCE METHODS

There is a creative approach called CLAUDIA, which is comprised of both on-chain and off-chain governance, proposed by Arribas et al. In [19]. Using this method, stakeholders will be able to discuss various issues and track issues on the blockchain using on-chain services such as Ethereum-based DAOs like WUDDER, along with an off-chain governance compliance desk. Through the use of both off-chain and on-chain discussions, stakeholders are able to resolve issues efficiently. As well as being more user friendly, CLUADIA offers a sandbox for testing upcoming features for limited users. This makes it easier to incorporate into existing business models and comply with regulatory requirements. Meanwhile, its reliance on the Ethereum network leaves it vulnerable to the inherent limitations of the Ethereum blockchain.

Due to the difficulty of protecting against a Sibyl attack, many current blockchain technologies rely on coin-based voting schemes. According to Chung et al [20], a voting system based on proof of participation would address these deficiencies. During a decentralized crypto game, the user's participation and level serve as a proxy for the user's identity. This helps to protect against sybil attacks while also spreading voting power among a variety of stakeholders.

According to [21], Li et al discussed not only the governance of blockchain technology, but also how it could be incorporated into a framework for blockchain governance. In this framework, blockchain technology influences blockchain regulation. The integration of governance of blockchains with governance on blockchains could pave the way for a dynamic blockchain ecosystem.

Singh and Vardhan used multi-objective PSO algorithms in [22] in order to maintain a balance between block composition time and transmission time. Based on their findings, the appropriate block size is 213 transactions per block. Even though they have utilized optimization methods to control block creation and transmission time, their method does not constitute an on-chain governance model.

In another study, Singh and Vardha utilized multi-objective PSO to solve a block size optimization problem [23]. Specifically, they have considered block building time and transaction selection time as the objective functions. The results indicate that a block size of 3.8 MB optimizes the time required for transaction selection and block building.

III. BACKGROUND

A. FORK ANALYSIS

Forks can be categorized into two main categories: intentional forks and accidental forks. The cause of intentional forks is the change in rules and protocols, as well as a lack of agreement on the modified rules. The purpose of these updates is usually to resolve technical problems, to recover lost resources, and to implement updated features [18]. In some cases, forks occur accidentally as a result of block mining occurring at the same time.

It is possible that different nodes will receive one block sooner than the other when two blocks are propagated at once. Therefore, their chain tips may differ. The result of this is that the main chain grows into different branches. By choosing the longest chain, the network resolves the conflict. Due to the potential security risks associated with accidental forks, as well as the potential destabilization of a cryptocurrency with significant losses, it is an imperative issue in the blockchain area. We have used the term *fork* in place of *accidental fork* for simplicity in the rest of the paper.

A fork is generally caused by a network delay, which is measured as the sum of the delays associated with the propagation of blocks. In order to propagate the newly mined block over the P2P network, a miner sends messages to his neighbors. Network delay is determined by the sum of the delays of these messages and the verification delay. Using the following equation, one can determine how long a network delay is [6]:

$$D = t\_v + t\_inv + t\_get\_header + t\_header + t\_get\_block\_body + t\_block\_body \quad (1)$$

where:

1) *t_v*: Verification time.
2) *t_inv*: The arrival time of the *inv* message in the destination node.
3) *t_get_header*: The arrival time of the message *get_header* through which the destination node asks

the origin node to send the header of the newly mined block.

4) *t_header*: The arrival time of the message *header* through which the origin node sends the header of the new block to the neighbor node.
5) *t_get_block_body*: The arrival time of the message *get_block_body* through which the receiver node asks the sender to send the block body.
6) *t_block_body*: The arrival time of the message *block_body* through which the origin node sends the body of the new block to the neighbor node.

B. EIP-1559

A proposed update, EIP-1559, was proposed by the founder of Ethereum in 2021 to address issues related to the first price auction mechanism [8]. To be included in a block, each user must pay a fee to the miner. Especially in case of network congestion, this mechanism is completely detrimental to users. This is because the miners have the ability to select transactions with higher fees and create a competitive environment among the users in order to increase their fees. There should be two fees for EIP-1559 users. As a requirement, *fee_cap* must be greater than *base_fee* in order to be valid. There is no payment to the miner for this *base_fee*. Then it is burned and removed from the network.

This second fee, called the *premium*, is directly credited to the miner. For a transaction to be included in a block, these two fees should meet the following conditions:

$$min\ \{fee\_cap\ \text{-}\ base\_fee,\ premium\} \geq miner's\ marginal\ cost \tag{2}$$

Equation 3 shows how *base fee* of a block is calculated:

$$b_{t+1} = b_t * (1 + d * \frac{block\_size - target\_load}{target\_load}) \tag{3}$$

The b_t refers to the base fee from the previous block, while *block size* corresponds to the block size in gas, d refers to the step size, and *target_load* refers to the old maximum size which can be doubled to *block_size*.

As a result of a smaller *block_size* compared to *target_load*, the output base fee will be smaller and vice versa. Step size regulates the maximum increase or decrease in the *base_fee* per step. Based on the step size of 12.5 %, the maximum increase and decrease of *base_fee* is 12.5 % in relation to the previous *base_fee*. As a matter of convenience, in all of our experiments, we generate the values of the miner's marginal cost and premium between 0 and 1. The following subsections review briefly the related works on sharding and on-chain governance.

C. PARTICLE SWARM OPTIMIZATION ALGORITHM

An iterative algorithm, PSO, is derived from mathematical modeling of the behaviors of species in groups, such as those observed in bird flocks [24]. In different fields, this optimization algorithm has been frequently used to solve challenging and complex optimization problems with swarm intelligence [25], [26], [27].

This algorithm employs a population of individuals called particles to search for solutions within the solution space. In terms of an objective function, each individual or particle can provide a possible solution. Through cooperation, the particles identify the optimal solution by exploring and exploiting different areas of the solution space. A particle is composed of a position vector (the likely solution), a velocity vector indicating its direction of movement, and a memory that saves the particle's optimized position from the beginning to the current iteration. The following equation illustrates how particles move within solution space:

$$\begin{aligned} v(t+1) &= w.v(t) + c_1.rand.(p\_best - x(t)) \\ &\quad + c_2.rand(p\_gbest - x(t)) \end{aligned} \tag{4}$$

$$x(t+1) = x(t) + v(t) \tag{5}$$

In these equations:

- v: The velocity of the ith particle
- x: The position of the particle
- t: the number of iterations
- c_1 *and* c_2: Learning factors designed to control the exploring and exploiting capability of PSO.
- *rand*: A positive random number between 0 and 1 under normal distribution
- w: The inertia weight coefficient
- *p_best*: The best position of the particle from the beginning to current iteration
- *p_gbest*: The position of the best particle in each iteration which has the best fitness value in the population.

At the beginning of the search process, particles are initialized randomly in the search space. A cost function is then used to calculate the fitness of each particle, and the most fit particle (which has the highest fitness value) is selected as the leader of the population (*p_gbest*). Each iteration thereafter involves the following steps until the termination criteria are met:

1) Equations for moving particles in space 4 and 5.
2) Using the cost function to calculate the fitness value of each particle.
3) Update of the leader's position and the particle's memory (*p_best*).

In the final step, the most optimal particle (or the leader) in the last iteration will be presented as the solution to the optimization problem. The following algorithm illustrates the pseudocode of the PSO algorithm.

As described by equations 4 and 5, particles tend to move toward the most suitable particle in the population (*p_gbest*) when c_2 is high and c_1 is low. In the event that c_1 and c_2 are high, particles will search around the most optimal position they observed at the beginning of the iteration to the present (*p_best*). As a result, it is necessary to select the appropriate amounts for these two factors in order to control the algorithm's exploratory and exploitative capabilities.

Algorithm 1 Pseduo Code of PSO Algorithm

```
Pop = Initialization
for $iteration = 1, 2, \ldots,$ Itmax do
    for $particle = 1, 2, \ldots,$ Popmax do
        velocity = w*velocity + cl*rand*(p_best — particle) + c2*rand*(p_gbest particle)
        particle = particle + velocity
        fitness = costfunction(particle)
        if $fitness_{particle} > costfunction(p\_gbest)$ then
            p_gbest = particle
        end if
        if $fitness_{particle} > costfunction(p\_best)$ then
            p_best = particle
        end if
    end for
end for
$Final\_Solution \leftarrow p\_gbest$
```

Different literatures indicate that fixing them at 2 will result in a reasonable convergence rate.

IV. PROPOSED MODEL

Using the PSO algorithm as an on-chain governance model is a highly efficient and effective way to maintain a balance between fork probability and users' experience on the chain. To resolve the fork problem, we propose a multi-objective PSO that maintains a positive user experience while also solving the on-chain governance problem.

To resolve the fork problem with contradictory objectives, it is appropriate to harness the capabilities of optimization algorithms that work with more than one objective. It has become more and more popular in recent decades to use metaheuristic algorithms to solve both single-objective optimization problems as well as multi-objective optimization problems [10], [22]. It is important to note that meta-heuristic algorithms are problem-independent, easy to use, and capable of solving a wide range of problems. The problem-independence of these algorithms means that they do not take advantage of the specificity of the problem.

The purpose of PSO optimization is to determine the most appropriate parameters for a sharded network in order to minimize the occurrence of orphan blocks and forks. By using a leading multi-objective optimization algorithm, this study attempts to solve this problem. Moreover, the problem is formulated for experimental evaluation in order to arrive at an optimized configuration that is suitable for sharded and non-sharded blockchains, as considered in the simulations.

At a high level, a new node has been added called learners. Based on the input parameters provided by their neighbors, this node is responsible for collecting the parameters from their neighbors, running PSO, and providing an optimal solution. The development of multi-objective optimization techniques for blockchain networks has been recently examined in the context of blockchain protocols [10], [22], [23]. Due to the voting process, the blockchain network will be able to stabilize in terms of the likelihood of forks arising and the user experience, unlike off-chain governance models. The following novelties are offered by our model:

1) Fork Reduction. Our model is the first blockchain protocol to be able to control the orphan block rate and consequently fork probability while still maintaining positive user experiences, which previously had not been possible (e.g. [7], [18], and [22]).
2) EIP-1559 Impact Analysis. A reduction in miners' marginal costs may reduce the likelihood of forks occurring; however, this could result in overpayments to miners and an increase in waiting times for users. The proposed model is the first sharding-based blockchain protocol that takes into account the impact of EIP-1559 on the Ethereum network. In contrast, none of the previous work provided an analysis of the user's perspective.
3) Learner Committee Consensus. Based on [21] and [31], we develop a committee consensus among learner nodes that they are responsible for executing the PSO optimization algorithm independently with local inputs in order to reduce fork probability and other objectives compared to previous solutions [13], [14], [22].
4) Sharding Analysis. This paper discusses a novel technique for designing a dynamic sharding method for partitioning blockchain transactions so that the fork probability will not be affected. This is an important property that has been lacking in previous sharding-based protocols [10], [11], [12], [13]. We have also designed a model that allows learner nodes to join the protocol without interruption or concern.

As a result of the proposed model, the following parameters are controlled: *validation degree*, *miner's minimal marginal cost*, *average block size*, and *average time between blocks*.

At the beginning of the learning process, each learner connects to a random node in the network. It obtains critical network parameters from the nodes that are connected to that node. After that, each learner runs a complete PSO independently to determine which parameters should be used to maintain the proper balance between fork occurrence probability and user experience. This is because each learner experiences the proper balance between fork occurrence probability and user experience.

In response to the random connection between the learners and the other nodes, each of them has a different value for the parameter. This is because of the random connection between them. Different solutions proposed by the learners are aggregated in a consensus mechanism. This enables us to arrive at a final decision about the entire network based on the variety of solutions proposed by the learners. The following subsections provide a thorough explanation of the updated simulator, the input parameters and their effects on the fork probability and users' experiences. In addition, they provide a thorough explanation of the structure of particles and the cost function in the PSO algorithms.

A. IMPROVED SIMULATOR

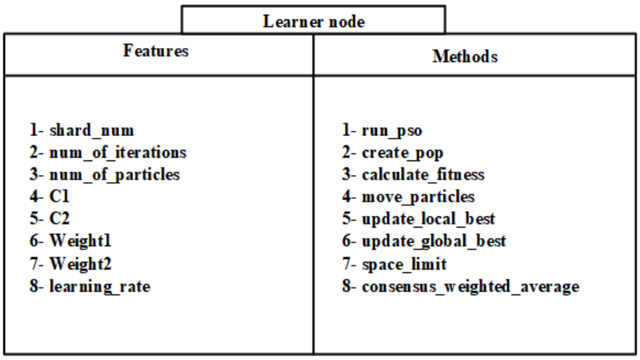
The effectiveness of our proposed on-chain governance model would have to be evaluated through the use of a simulator. Our work contributed to the improvement of Blocksim, a simulation tool developed by Faria et al. [1] by adding the capability of simulating sharded networks. In addition to this, we developed a PSO-based on-chain governance model and implemented it into Blocksim, and carried out our experiments in this simulation.

The simulator was developed using the Python programming language and the SimPy library. SimPy is a Python-based discrete-event simulation framework. Using SimPy, you can simulate asynchronous networking or implement multiagent systems using Python generator functions. Programmers can utilize generators to specify a function's exit point so that it can be re-entered at the point of its last exit, thus allowing two functions to run alternately. With the addition of cryptographic hash functions, we provide a more tailored framework for simulating blockchains compared with SimPy, which provides a framework for creating arbitrary models. The Ethereum blockchain network is simulated using an object-oriented program developed with SimPy without the need to install any blockchain platform. Due to the object-oriented nature of our model, we were able to adequately model several relevant blockchain properties.

In our simulator, a new node has been added called *learners*. This node is responsible for collecting the input parameters from their neighbors, running PSO, and providing an optimal solution based on the input parameters. It is critical to note that at the beginning of each shard, a predetermined number of blocks are created before these nodes become active. Based on the blocks that you have created, they provide you with the optimal solution. Finally, the suggestions are compiled using a consensus mechanism to reach consensus. To maintain control over the whole chain of solutions proposed by different learners in each shard, a consensus mechanism is used to aggregate the solutions proposed by different learners. In order to develop a consensus mechanism, there are several methods that have been proposed, such as [31] for example.

We have used a weighted average mechanism as part of this research in order to determine *validation degree*, *minimum marginal cost*, *average block size*, and *interval between blocks*. This procedure has the result of assigning weights to each solution based on the sum of the amount of hashes produced by all of the miners that connect to the solution. A learner node with the greatest weight is connected to the miners with the highest total hash rate. This is one of the learners that are linked to the miners with the highest total hash rate. As soon as the averaging process has been completed, the miners generate the *marginal cost* and *block size* randomly based on the average value that has been determined by the governance model after the process has been completed.

Furthermore, for each shard, it is determined by averaging the *validation degree* and *average time between blocks* values. In order to develop a governance model, the parameter *learning_frequency* had to be taken into consideration as part of the process. This factor will be considered when determining when to complete the tasks assigned to the governance model. Accordingly, if the governance model is *learning_frequency* $= 4$, then after four consecutive blocks, it should be able to determine the most optimal values. In all of the simulations we have run, *learning_frequency* has been set to 4 throughout the entire process. This can be seen in Figure 2, which shows the class diagram of the learner nodes.



Learner node	
Features	Methods
1- shard_num 2- num_of_iterations 3- num_of_particles 4- C1 5- C2 6- Weight1 7- Weight2 8- learning_rate	1- run_pso 2- create_pop 3- calculate_fitness 4- move_particles 5- update_local_best 6- update_global_best 7- space_limit 8- consensus_weighted_average

FIGURE 2. Class diagram of the learner nodes in the new simulator.

B. INPUT PARAMETERS

In this research, the focus is primarily on the occurrence of forks. Accordingly, it has been shown that the proposed governance model contributes to reducing the probability of a fork occurring as a result of this process. As a result of reducing the probability of forking, users may experience a negative user experience as a result of this.

A reduction in *miners' marginal cost* may decrease the probability of forking, however, this may lead to overpayments for miners and an increase in the amount of time users have to wait for their blocks. The following paragraphs provide an explanation of the network's behavior as a result of the selected parameters.

- *Miners' minimum marginal cost:* It is important to note that the *miner's marginal cost* for EIP-1559 refers to the amount that a user is supposed to pay to the miner in order to have his transaction included in the block. It is because of the higher marginal cost that users have to pay more for a single transaction. This decreases their chances of being included in the block as a result. The result of this could be an increase in the time it takes to create a block as a result. Over a specified period of time, this will result in fewer blocks being created. This in turn will reduce the possibility that a fork of the blockchain will occur.

Due to this effect, increasing the marginal cost of a product may adversely affect the user experience, which may adversely impact their satisfaction with the product. The result of the above is that the user's experience is negatively affected by this parameter. This parameter has been defined as our objective function that has a direct impact on both fork probability and users' experience. Based on our simulations, we are able to generate a value for the *miners' marginal cost* between 0 and 0.9 in our model. To determine the average marginal cost, it is necessary to use the governance model in order to do so. In other words, as a result of each run of a governance model that has been applied to the final solution, the average marginal cost can be calculated. The miners will determine their marginal cost by drawing random numbers from the average value of the marginal cost once the average value has been determined. In addition to this, the premium value of each transaction is also generated at random between 0 and 1.

- *Validation degree (v)*: In order for a new block to be approved as the new tip of the chain, the number of nodes that need to validate it is determined by this parameter. In the case of $v = 0.5$, half of the nodes in the chain approve it, and once it has been verified, they add it to their chain. The other half accepts it without verifying it. As a result, once the block has been validated, when $v = 1$, all nodes should accept the block once it has been validated. There is a strong correlation between this parameter and the number of nodes involved in the process of block verification, resulting in an increase in network delay (see equation 1) as the primary cause of forks in the chain. Therefore, the probability of a fork is higher when the *validation degree* of the text is high. As a result of the governance model, it has been found that the optimal value of *validation* lies between 0.5 and 1.
- *Average block size (number of transactions in the block):* According to the assumptions made in this paper, the size of each block will be determined in a way that is random and around an average block size, *average block size*. In general, this parameter determines how large the blocks will be based on the length of the parameter. The length of a block is directly related to the amount of time it takes to complete and propagate across the network. Therefore, it is likely that there will be a delay as well as a higher probability of a fork in the future. There is a maximum and minimum number of transactions that can be included in a block in the simulations, respectively, and these numbers are 10 and 100 respectively. According to the governance model, it will be the responsibility of the governance model to determine the optimal value of *average block size* between 10 and 100. It is the responsibility of the miners to decide the block size randomly somewhere around this average value.
- *Time between blocks:* This parameter determines the minimum amount of time that must elapse between consecutive blocks. Increasing the time interval between two consecutively created blocks may reduce the probability of a fork. However, it can also result in an increase in network latency, which will not be beneficial for users, since it increases the likelihood of forking. It is therefore crucial to determine the optimal value for this parameter so that the system can operate effectively. According to the results of the experiments, the governance model searches for the optimal value between 5 and 30 seconds by looking at the solution space based on the results of the experiments.

The use of learner nodes is a very effective way of controlling these parameters during the simulation. This is in order to make sure that there is an acceptable probability of forking while still maintaining a positive user experience.

C. PARTICLES' STRUCTURE AND COST FUNCTION

Based on our proposed model, the following steps described the on-chain governance process

- First, five learner nodes are considered per shard.
- The learning nodes are then randomly connected to other nodes, initiating their parameter values (Table 1) that will be used by the PSO algorithm.
- Next, the learning process will begin and will last until the end of the simulation after every four successive block generation. In this process, each learning node runs the PSO process independently and saves the optimal results.
- The fitness function described in 6 will be used by each learner node during the execution of the PSO process. The learners attempt to minimize the fitness function in order to obtain the most optimal possible solution. In order to compute the fitness values, they must simulate the network with the new parameters and calculate the objective functions. During this process, they are attempting to find the most optimal network parameters to reduce fork probability without compromising the quality of the user experience.

Once all the optimal solutions have been collected from the learners, the consensus process takes place, during which a voting-based consensus mechanism is employed and the optimal value vector for the input parameters is determined. As a result, the entire network is updated, and this procedure continues until the simulation is completed.

1) PARAMETERS

In this study, each particle consists of a vector containing the values of the network's parameters. A list of all parameters used in the experiment is specified in Table 1.

2) FITNESS FUNCTION

Another critical aspect of the design process is the development of a cost function capable of modeling both the

TABLE 1. List of parameters used for the experiment.

Parameters	*Description*
validating degree	the amount that a user is supposed to pay to the miner
miners' marginal cost	a certain number of nodes must validate a new block
average block size	the number of transactions in the block
time between blocks	the minimum time between consecutive blocks

increasing and decreasing patterns of forks. As well, it is critical to model the users' experience. As indicated in equation 6, the cost function used in this study is shown in equation:

$$\begin{aligned}&cost\ function\\&= w_1 * (1 - e^{-(\frac{validation\_degree}{minmarginalcost+avg\_block\_size+average\_time\_between\_blocks})})\\&\quad + w_2 * (minimum\_marginal\_cost + average\_block\_size\\&\qquad + average\_time\_between\_blocks)\end{aligned} \tag{6}$$

The fitness function is used as two objective functions (fork probability and user experience) in our proposed PSO model is represented in 6. The goal is to find the vector of variables such that the objective functions would be minimized. As a result, the optimal solution during a PSO iteration is the one with the minimum fitness value.

Based on the Poisson process, the first term of the function is used to model the increasing or decreasing movement pattern of the fork probability [28], [29]. To control the user's experience, especially overpayments, the second term is added. When these two terms are properly modeled, fork occurrence probability and user experience are mutually contradictory and can be traded off. As an example, if we increase the *average marginal cost*, fewer blocks will be generated in a specified period, resulting in a reduction in the amount of network latency. As a result, the likelihood of a fork will be reduced.

In equation 6 the second term is intended to prevent overpayments, so increasing it too much is not beneficial for users. Additionally, w_1 and w_2 are two coefficients that allow us to focus on the most critical objective, depending on the network's conditions. For example, if reducing the fork probability is critical to us, we should choose a larger value for w_1 than w_2. To emphasize the reduction of forks in our simulations, which are presented in the following section, we fixed w_1 and w_2 at 0.9 and 0.1, respectively. Figure 3 illustrates the flow chart of the proposed on-chain governance process.

V. EXPERIMENTS AND RESULTS

The purpose of this study is to investigate the effects of sharding and the proposed on-chain governance method on the fork probability issue as one of its major objectives. The following two scenarios have been used to simulate the network of EIP-1559 with 120 nodes in order to achieve this:

- Without an on-chain governance mechanism to monitor the effects of sharding on the probability of forking.
- An analysis of the impact of on-chain governance on fork occurrence probability is undertaken with regard to the proposed on-chain governance model.

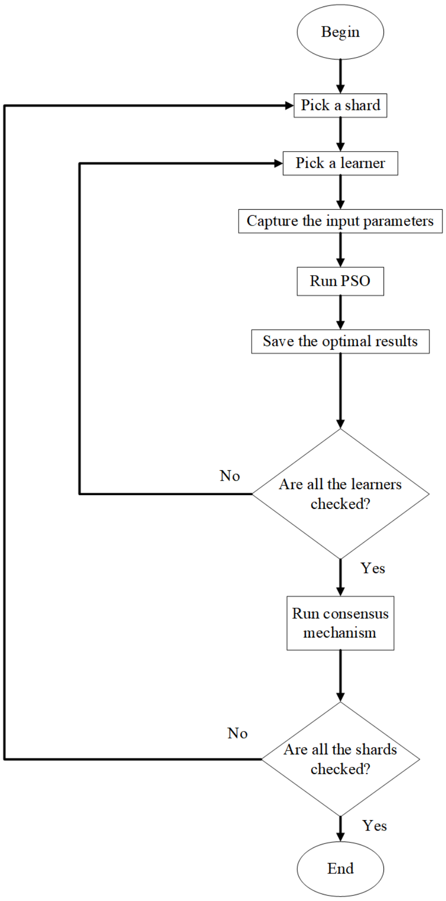


FIGURE 3. Flow chart of the on-chain governance process.

- A simulation of a non-sharded blockchain network without on-chain governance.

The results shown in the following section have been obtained running the simulator on a PC with an Intel Core i7-8665U 1.90GHz with 8GB of RAM and Windows 10 Enterprise (x64) as an operating system. Following this, the results for each case will be presented and discussed.

In addition to the simulator evaluation provided by [1], before beginning our simulation, we should also perform an evaluation of our simulator. In order to validate the proposed model further, Table 2 compares simulation results with actual values. Considering that the simulation was performed on average values for transactions in the Ethereum network, a comparison was also made between the averages. By the end of the simulation study, we have collected data both from the simulation and from the real Ethereum network. Based on the results obtained, the results are in reasonable agreement with the actual values.

A. SIMULATING A SHARDED NETWORK WITHOUT ON-CHAIN GOVERNANCE

In order to investigate the effect of sharding on the fork issue, four simulations of EIP-1559's network were performed using different numbers of shards, and the values of

TABLE 2. A comparison of the actual results with those simulated.

Parameters	*Actual*	*Simulated*
tx per block	93	93
mempool count	66523	65540
block propagation (ms)	634	735
transaction propagation (ms)	98	93
number of blocks	6129	6139

key parameters were measured. As can be seen in the table below, the average results were achieved Table 3.

TABLE 3. Average results achieved for a network with different number of shards without on-chain governance.

Number of Shards	*1*	*2*	*3*	*4*
average number of orphan blocks	227	29	13	7.25
maximum number of orphan blocks	227	39	17	11
total number of orphan blocks	227	58	39	29
average network delay (ms)	22495.15	3097.77	1379.67	779.81
maximum network delay (ms)	22495.15	3122.95	1443.22	826.91
average number of created blocks	458	62	28	17.5
maximum number of created blocks	458	84	37	24
total number of created blocks	458	124	88	70

According to this table, the total number of blocks created in a sharded network with four shards was 70. This means that each shard had an average of 17.50 blocks. Also, on average, there were 7.25 orphan blocks in each shard. The number of total blocks that were created in a non-sharded network was 458 and the number of orphan blocks was 217. As illustrated in figures 4 and 5, the graphs show the average number of orphan blocks and the average network delay.

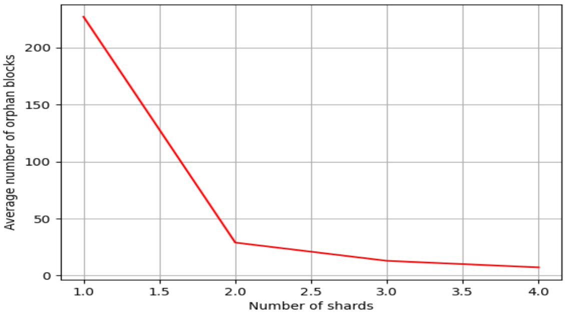


FIGURE 4. How number of orphan blocks changes by increasing the number of shards.

There is also an indication that by increasing the number of shards, the number of orphan blocks will decrease, as will network delay as well. It is important to note that we refer to high block latency, slow block propagation times, and high network delay as synonymous terms. In the paper [30], the relationship between the orphan block rate and latency is demonstrated. According to the study, orphan blocks are more likely to occur during periods of high network delays. This is in accordance with the intuition that slow block propagation leads to a large number of accidental chain forks. It is due to the increased probability of a new block being discovered before it has fully spread throughout the network that this occurs.

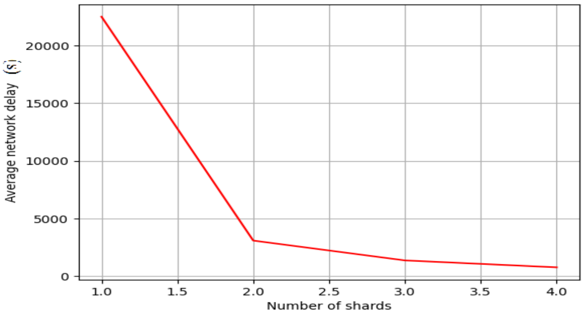


FIGURE 5. How network delay changes by increasing the number of shards.

The results shown in Table 3 also indicate that, on average, adding each shard results in a reduction of 62 % in the number of orphan blocks and a reduction of 61 % in the number of network delays when compared to the results without sharding. According to the results of the study, it can be said that increasing the number of shards reduces the possibility that forks will occur. Our simulations are based on the assumption that every node is divided uniformly into different shards based on its configuration.

In other words, in our simulations, all of the shards had the same number of nodes in them, regardless of the number of shards. There is a higher risk of forks occurring in shards with more nodes than in those with fewer nodes. This is due to the higher number of nodes present in a shard. In addition, we made the decision to fix *validation degre* and θ to one in order to obtain a reasonable result regarding the effect of sharding on the probability of fork occurrence.

B. SIMULATING A SHARDED NETWORK WITH ON-CHAIN GOVERNANCE

Based on the EIP-1559's network, we simulated four networks: four networks with 1 to 4 shards, respectively, to evaluate the performance of the proposed on-chain model. The number of learner nodes in each shard has been considered 5 in each case. The following subsections provide a summary of the results obtained.

It should be noted, due to the randomly determined transactions, the number of blocks created in each shard is outside of our control during the simulation. Consequently, some curves end before the simulation is completed. For each simulation, a table shows the number of blocks created for each shard.

1) ONE SHARD

We conducted this experiment to see only the effect of the proposed on-chain governance model on the fork occurrence. Table4 contains the achieved results for this experiment in addition to the results achieved for the case of simulating the network without on-chain governance.

Our on-chain governance model performs exceptionally well as indicated by this Table. In this table, it is shown

TABLE 4. **Results of simulating a network with and without on-chain governance.**

Parameters	*Orphan blocks*	*Created blocks*	*Network delay*
with on-chain governance	91	331	13169.88
without on-chain governance	227	458	22495

that there is a difference in the governance model between networks that have and do not have on-chain governance. As a result, the number of orphan blocks has increased by the factor of 2.49, while the number of all created blocks has increased by the factor of 1.3. Thus, the number of orphan blocks has been dramatically reduced using the proposed governance model. In Figure 6, we see how the proposed model operates.

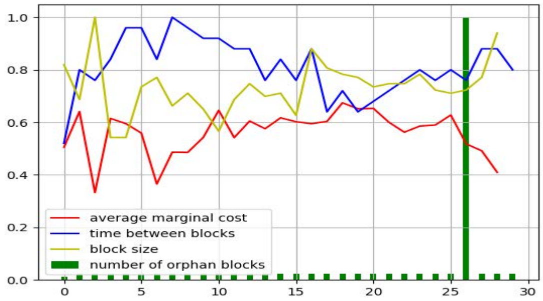


FIGURE 6. **Performance of the governance model in a non-sharded network.**

Figure 6 illustrates the normalized orphan block number between two consecutive runs of the proposed model in the form of a bar graph. In addition, the other graphs display the output of the model for *average marginal cost*, normalized *average time between blocks* and normalized *average block size*. It can be seen from this figure that there has been a significant increase in the number of orphan blocks.

As indicated by other graphs, this is the result of a reduction in both *average marginal cost* and *average time between blocks*. This has resulted in a reduction in the number of orphan blocks in the governance model. This has resulted in an increase of the *average time between blocks* as well as *average block size*, while *average marginal cost* has continued to decrease.

At the end, in Figure 7, we illustrate the convergence of the cost value for one shard of our proposed PSO model during the iteration process. As the simulation progresses, it is apparent that the cost value decreases and converges towards the optimal value.

2) TWO SHARDS

As shown in Table 5, the results of applying the proposed on-chain model have been presented.

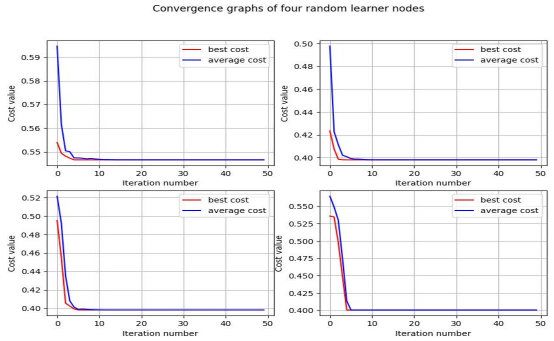


FIGURE 7. **Cost value convergence of the proposed PSO method for one shard.**

TABLE 5. **Average results of applying the on-chain governance model on a sharded network with two shards.**

Shards	*Shard 1*	*Shard 2*
validation degree	0.512	0.512
minimum marginal cost	0.69	0.63
block size	43.65	62.54
time between blocks	21.79	19.16
network delay (ms)	2198.85	1920.89
number of created blocks	68	41
number of orphan blocks	32	18

Table 5 indicates that the on-chain governance model has fixed the *validation degree* at 0.512, which is basically the minimum value considered. Accordingly, the proposed method selects the most appropriate value for *validation degree* in order to reduce the fork probability. In order to ensure a balance between fork occurrence probability and user experience, the proposed model has chosen a moderate value for *minimum marginal cost*. Since users will have to pay more for a straightforward transaction as the marginal cost increases, the probability of being included in a block will decrease. As a result, the time it takes to create a block increases.

As a result, the probability of forking is reduced. Although *marginal cost* may be beneficial to users, it may be detrimental to them. According to the study, the governance model has been developed in order to find a solution that benefits both users and miners.

According to the proposed model, the time between blocks for the first and second shards should be 21.79 seconds and 19.16 seconds, respectively. Once again, it can be seen clearly that the model has chosen a moderate value for this parameter. In addition to increasing the probability of forks occurrence, decreasing *time between blocks* is also more beneficial to the users. Furthermore, for *block size*, the on-chain model has determined that the average block size for the first and second shards is 46.65 and 62.54 transactions respectively.

It can be seen from the values selected for this parameter that the model has attempted to address the trade-off between two contradictory cost functions.

For the first and second shards, these values resulted in 32 and 18 orphan blocks, respectively. This means that there are a total of 50 orphan blocks generated in our simulations. This is fewer than the 58 orphan blocks created in the previous experiment without governance models.

Aside from that, the average network delay for a sharded network is less than that of a non-sharded network, in which the average network delay is 3097.77 (ms). As shown in Figures 8 to 10, the governance model performed well in determining the *average marginal cost*, *average time between blocks*, and *average block size*.

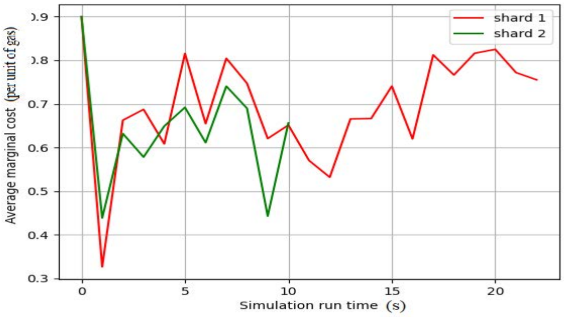


FIGURE 8. How the governance model determines the average marginal cost during the simulation of a network with 2 shards.

According to Figure 8, the red and green graphs clearly illustrate how the governance model balances the fork probability with the users' experience. This figure shows that at certain points, the graphs exhibit an ascending pattern, and at other points, they exhibit a decreasing pattern.

Accordingly, the proposed model sometimes increases *marginal cost* in order to reduce fork probability, and sometimes reduces it in order to improve users' experience. Increasing or decreasing is determined based on the observations made by the learner nodes from their connected nodes.

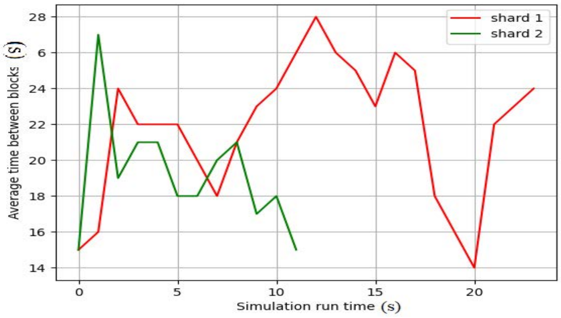


FIGURE 9. How the governance model determines the average time between blocks during the simulation of a network with 2 shards.

Both Figure 9 and Figure 10 are based on the same concept as *time between blocks* and *block size*. Depending on the results of monitoring the network from its connected nodes, the governance model may decide to increase or decrease these parameters. This is in order to reduce the probability of forks or to create a better experience for users. According to these figures, the green graphs are shorter than the red graphs. This is due to the lower number of blocks created on the second shard as compared to the first.

The final figure, Figure 11, illustrates the convergence of the cost value of two shards of our proposed PSO model during iteration. As the simulation progresses, it becomes evident that the cost value is decreasing and converges towards an optimal value.

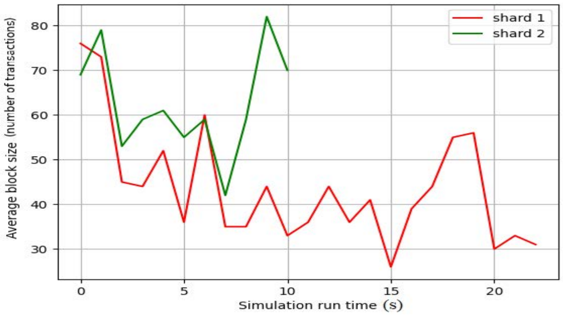


FIGURE 10. How the governance model determines the average block size during the simulation of a network with 2 shards.

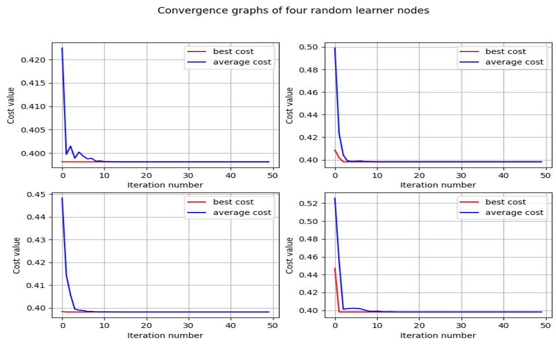


FIGURE 11. Cost value convergence of the proposed PSO method for two shards.

3) THREE SHARDS

Table 6 contains the results achieved in this experiment.

TABLE 6. Average results of applying the on-chain governance model on a sharded network with three shards.

Shards	*Shard 1*	*Shard 2*	*Shard 3*
validation degree	0.57	0.5	0.5
minimum marginal cost	0.65	0.415	0.383
block size	67.5	76.6	81.6
time between blocks	17	19.66	20.5
network delay (ms)	444.736	470.435	478.751
number of created blocks	19	22	27
number of orphan blocks	15	10	12

In accordance with this table, both shards have a minimum number of orphan blocks, while all of their parameters are within a moderate range. A relationship between average marginal cost, average time between blocks, and average block size can be seen in figures 12 to 14.

It is evident from Figure 12 that the marginal cost graphs for all shards have a decreasing pattern. Even though this is better for users, it can result in a higher probability of a fork occurring. Accordingly, the governance model has increased the *time between blocks* in order to maintain this trade-off and keep the chains in a reasonable condition in terms of forking, as illustrated in Figure 13.

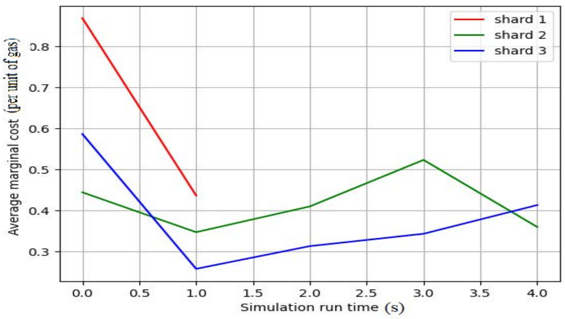


FIGURE 12. How the governance model determines the average marginal cost during the simulation of a network with 3 shards.

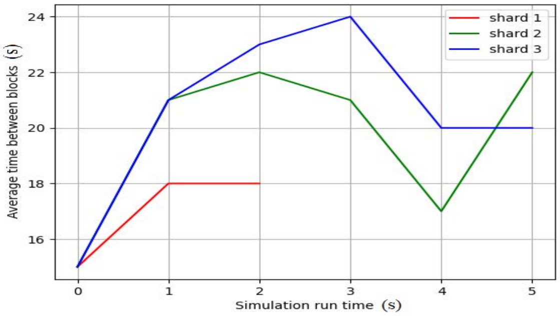


FIGURE 13. How the governance model determines the average time between blocks during the simulation of a network with 3 shards.

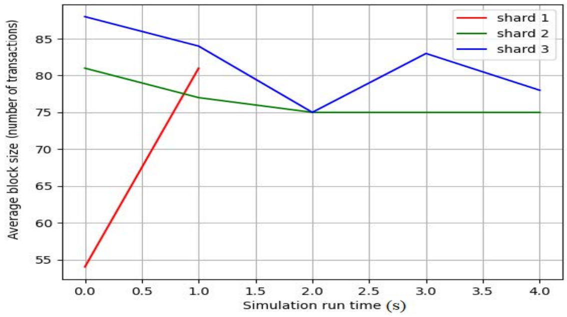


FIGURE 14. How the governance model determines the average block size during the simulation of a network with 3 shards.

Additionally, Figure 15 illustrates the convergence of the cost value of three shards of our proposed PSO model during iteration.

4) FOUR SHARDS

In Table 7, the results obtained following the implementation of the on-chain governance model on the network with four shards are presented. This table indicates that the first shard creates the least number of orphan blocks in comparison to the other shards. In this shard, the *average marginal cost* is the highest.

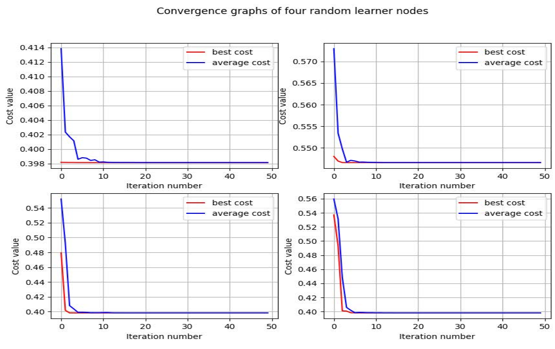


FIGURE 15. Cost value convergence of the proposed PSO method for three shards.

TABLE 7. Average results of applying the on-chain governance model on a sharded network with four shards.

Shards	*Shard 1*	*Shard 2*	*Shard 3*	*Shard 4*
validation degree	0.519	0.5	0.5	0.5
minimum marginal cost	0.623	0.535	0.485	0.579
block size	59.5	50	62.125	61.63
time between blocks	19.4	23.72	21.55	19.75
network delay (ms)	521.72	634.91	576.44	596.27
number of created blocks	11	20	16	22
number of orphan blocks	4	9	7	10

As a result, users will be required to pay a higher fee to have their transactions included in a block on this shard. In this instance, the chances of being included in a block are reduced. Due to this, there are fewer blocks created in this shard and there is a reduced network delay compared with other shards. Furthermore, this shard has a larger *validation degree* than other shards, resulting in a lower fork probability and a higher *marginal cost*. As a result, the forking probability for this shard is the lowest of all the shards.

The number of orphan blocks and blocks that have been created in the second and fourth shards are very close. There is an almost equal difference between the average marginal costs of these two shards. A major difference is observed in *average block size* and *average time between blocks*. While the *average block size* of the second shard is smaller than that of the fourth shard, the fourth shard has a longer *average time between blocks*. Fork occurrence probability and user experience are similarly affected by these parameters. If both of these factors are increased, it will result in longer block creation times. This will reduce the likelihood of a fork occurring.

As compared to the other shards, the third has a lower *average marginal cost*, but the *average block size* is the largest. In other words, even though the cost of users in this shard is lower than in the other shards, they have to wait a longer period of time before their transactions are included in a block. Compared to other shards, this shard has a longer time for block creation. Moreover, there was a more significant difference in *average time between blocks* between this shard and the first and fourth.

Comparatively to the first shard, this shard has a lower *average marginal cost*, which is more beneficial to users but increases the likelihood of a fork in the future. Meanwhile, it has a larger *average block size*, which is inversely proportional to the probability of a fork. Even though the third shard does not have the minimum fork probability, it is more cost-effective and provides a better waiting time and cost situation for users. The governance model's performance in determining *average marginal cost for four shards* and *average block size for four shards* for each shard during simulations is demonstrated in Figures 16 through 18.

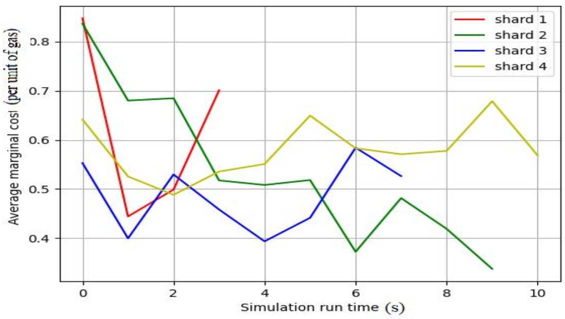


FIGURE 16. How the governance model determines the average marginal cost during the simulation of a network with 4 shards.

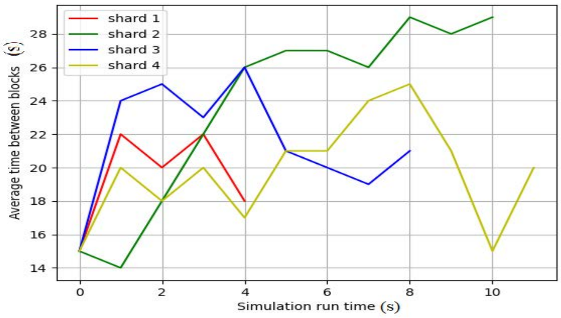


FIGURE 17. How the governance model determines the average time between blocks during the simulation of a network with 4 shards.

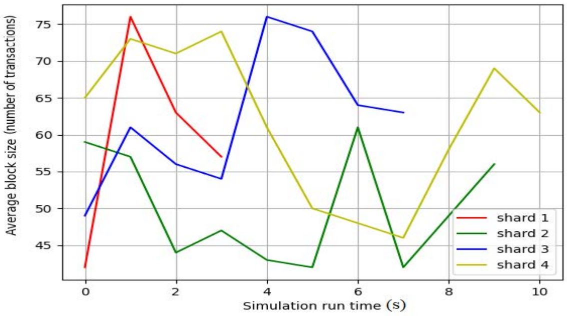


FIGURE 18. How the governance model determines the average block size during the simulation of a network with 4 shards.

Based on Figure 16, the *average marginal cost* of the first shard decreases initially and then increases. It has been observed that lowering the *marginal cost* will increase the fork probability and vice versa. It can be seen from 18 that the *average block size* graph of the first shard completely opposes its *marginal cost* graph. The pattern begins with an increase and then decreases.

As a result of the reduction of *average marginal cost*, the on-chain governance model increased *average block size* in order to manage the increase in fork probability while choosing a moderate range for *average block interval*.

Even though *average marginal cost* has some fluctuations, it generally has a downward trend, increasing the likelihood of a fork. As shown in 17, its *average time between blocks* shows an increasing pattern, which is inversely related to fork probability.

Alternatively, reducing the *average marginal cost* improves the experience of the users, but increases the likelihood of a fork. Therefore, to address this issue, the on-chain governance model has increased the *average time between blocks*. In the case of the *average block size*, according to Figure 18, the graph for the second shard does not exhibit any specific characteristics. In contrast, it falls within the moderate range of values.

It appears that the *average marginal cost* and the *average block size* are in conflict with each other for the third shard. As shown in the figures 16 and 18, when the *average marginal cost* increases, on the other hand, the *average block size* decreases and vice versa. Furthermore, when it comes to the *average time between blocks*, the third shard's graph does not appear to have a clear pattern.

Figures 16 and 18 depict the same concept for the fourth shard. The proposed method has attempted to keep the values of the parameters in a moderate range. This is in order to maintain the balance between the fork probability and the experience of users on this shard.

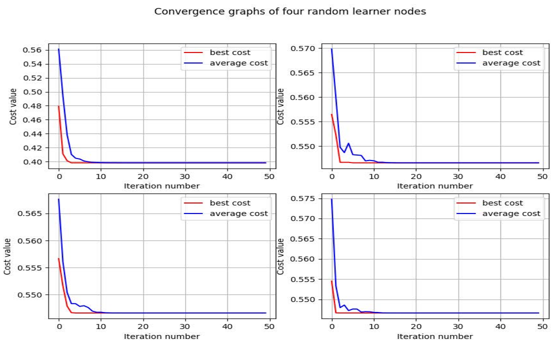


FIGURE 19. Cost value convergence of the proposed PSO method for four shards.

As a final illustration, Figure 19 illustrates the convergence of the cost value of four shards of our proposed PSO model during each iteration, indicating that the cost value is decreasing and is convergent towards an optimal value.

C. SIMULATING A NON-SHARDED NETWORK WITHOUT ON-CHAIN GOVERNANCE

Following are some results for the case in which we only have one shard and have chosen only to focus on the learning effect. A general description of our fitness function is provided in 7.

$$cost\ function = w_1 * (fork\_probability) + w_2 * (user\_experience) \quad (7)$$

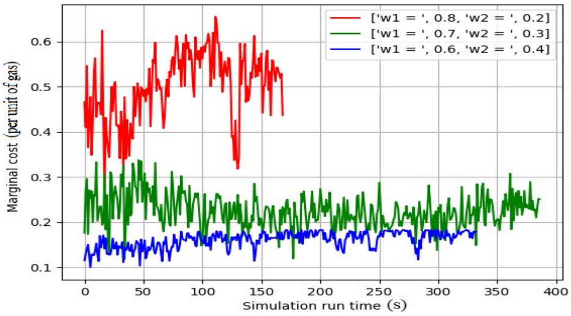


FIGURE 20. Average marginal cost for different learning parameters.

The three graphs shown in Figure 20 illustrate the results of the learning process for the marginal cost parameter in each run. To accomplish this, we performed three simulations and saved the output related to this parameter each time the learning process was completed. There is one significant difference between the three simulations in that each time the coefficients of the cost function were changed in order to evaluate the learning process. This figure shows that the overall marginal cost range also increases with an increase in w_1 (red diagram). When we increase w_1 (and therefore decrease w_2), the output value of the cost function is more influenced by the first term. The marginal cost has also been raised by the PSO method in order to minimize the likelihood of a fork. The probability of forks is reduced when fewer blocks are produced over a fixed period of time. By contrast, reducing w_1 (and naturally increasing w_2) reverses this process. As a result of setting w_1 to 6.0, the marginal cost range is lower than in the previous two scenarios. It was necessary to change w_1 and w_2 values to demonstrate that our method worked logically and perfectly.

Figures 21 and 22 illustrate the same scenario for *time_between_blocks* and *block_size*. These figures demonstrate that when w_1 increases, the algorithm tries to adjust the parameters so as to reduce the fork. In other words, the PSO algorithm places greater emphasis on reducing forks.

Figure 23 illustrates a similar concept to the previous ones, but with greater precision. As shown in this figure, w_1 has been increased from 5.0 to 1. In order to achieve this, we simulated six times and obtained the average size of the blocks for the entire simulation run (not for each learning run) and based on this result, we created the graph. As a result, our method prefers to increase the block size when w_1 is increased.

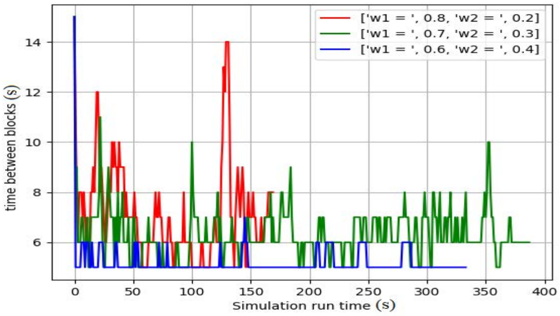


FIGURE 21. Time between blocks for different learning parameters.

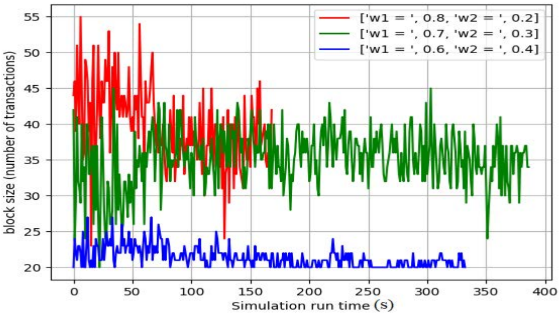


FIGURE 22. Block size for different learning parameters.

The result is a reduction in the number of blocks produced over a fixed period of time and a reduction in the number of forks.

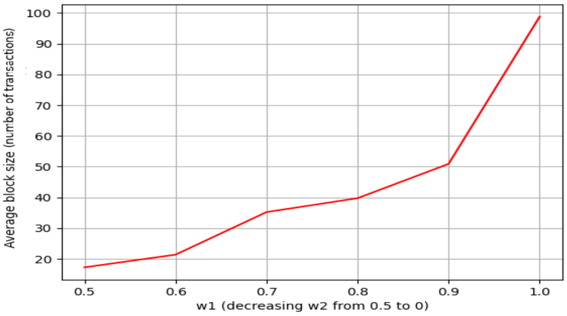


FIGURE 23. Different learning parameters impact on average block size.

It is important to note that in our previous figures, during the simulation, every time the learning was completed, we saved the output number of the parameters and at the end of the simulation I drew the graphs using those numbers. Therefore, each graph corresponds to a single simulation run. Here, we averaged the numbers at the end of each simulation and calculated this graph by plotting average points across multiple simulations.

Similarly, Figures 24 and 25 contain the same information for *time_between_blocks* and *marginal_cost* in order to illustrate the concept.

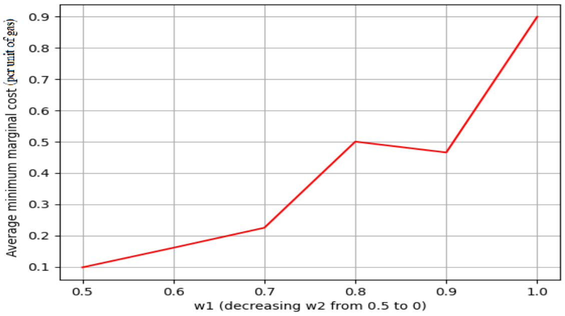


FIGURE 24. **Different learning parameters impact on average marginal cost.**

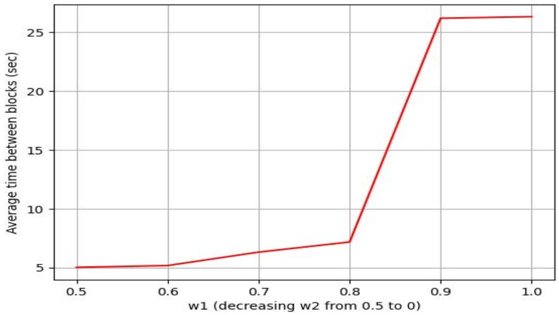


FIGURE 25. **Different learning parameters impact on average time between blocks.**

VI. CONCLUSION

In this paper, our main objective is to compare the probability of forking between the proposed on-chain governance model and the impact of sharding as well as the impact of sharding on the forking rate. Towards this goal, we have simulated a network with 120 nodes and different numbers of shards for EIP-1559 from the EIP-1559 specification. During our experiment, we increased the number of shards from one to four. This is to examine the impact of this on the number of orphan blocks and the delay in the network as a whole. It has been found that when a shard is added to a network, the number of orphan blocks is on average reduced by 60% on average for both networks.

It can be concluded from this that sharding has a positive impact on the forking process. It is critical to note, however, that sharding can increase the network's vulnerability by as much as 51%. Typically, a sharded network is characterized by having a lower hash power within each shard than a non-sharded network. Therefore, it makes it much easier for attackers to gain control over individual shards of a network than to gain control over the entire network, as a result.

A second step involved applying the proposed on-chain governance model to the sharded network in order to evaluate its efficiency and impact on the probability of fork occurrence within it. For this purpose, we simulated two networks consisting of two and four shards, respectively. This study showed that the proposed on-chain model was capable of solving the corresponding optimization problem and identifying the optimal parameter settings for the network as a means of maintaining a balance between the fork probability and the user experience, based on the results obtained.

In order to simplify the process, we currently assume that the size of each block will be determined randomly and on the basis of an average block size. However, in reality, it is much more likely that the block size is determined by the process by which the miner creates the block. Thus, the model could be improved by taking into account block size simulations based on real-world processes. As another limitation, our simulations assume that every node is divided uniformly into different shards based on its configuration. The work will be extended to include new methods of assigning nodes to different shards in the future. Another open challenge is to adapt our model for permissioned blockchains that do not use PoW, but use more conventional Byzantine fault-tolerant consensus algorithms. Last but not least, other consensus algorithms, such as proof of stake, could be included. As the design choices and processes of a platform have a profound impact on how a blockchain performs, the performance of the system is highly platform dependent. In addition to Ethereum, other blockchains can also be added.

Our model represents a major step forward in blockchain technology because it is the first protocol to control orphan block rates as well as fork probability in a positive manner while maintaining a positive user experience. Efficiency is one of the most important characteristics of a consensus protocol, such as Bitcoin. The orphan block rate is one of the most important indicators of Bitcoin's efficiency because it indicates how much hashing power is wasted on blocks that are not part of the main chain. An innovative method of dynamic sharding has been developed for partitioning blockchain transactions. It is possible to do this in a manner that will not have an effect on the fork probability. The previous sharding-based protocols lacked this critical property. Based on our committee consensus model, learner nodes should execute the PSO optimization algorithm independently with local inputs to reduce fork probability and other objectives.

REFERENCES

[1] C. Faria and M. Correia, "BlockSim: Blockchain simulator," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 439–446.

[2] S. Nakamoto, "Re: Bitcoin P2P e-Cash paper," Cryptogr. Mailing List, Nakamoto Inst., Austin, TX, USA, Tech. Rep. 1, 2008, pp. 1–2.

[3] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 17–30.

[4] E. Georgiadis, "How many transactions per second can Bitcoin really handle? Theoretically," Cryptol. ePrint Arch., Tech. Rep. 2019/416, Jul. 2020. [Online]. Available: https://eprint.iacr.org/2019/416

[5] Y. Gao, "Scalable Blockchain protocol based on proof of stake and sharding," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 23, no. 5, pp. 856–863, 2019.

[6] Y. Shahsavari, K. Zhang, and C. Talhi, "Performance modeling and analysis of the Bitcoin inventory protocol," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastruct. (DAPPCON)*, Apr. 2019, pp. 79–88.

[7] Y. Shahsavari, K. Zhang, and C. Talhi, "A theoretical model for fork analysis in the Bitcoin network," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 237–244.

[8] V. Buterin et al. (2019). *EIP-1559: Fee Market Change for ETH 1.0 Chain*. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1559

[9] J. Yun, "Trust-based shard distribution scheme for fault-tolerant shard blockchain networks," *IEEE Access*, vol. 7, pp. 135164–135175, 2019.

[10] X. Cai, S. Geng, J. Zhang, Di Wu, Z. Cui, W. Zhang, and J. Che, "A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7650–7658, Nov. 2021.

[11] J. Yun, Y. Goh, and J. M. Chung, "DQN-based optimization framework for secure sharded blockchain systems," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 708–722, Jan. 2020.

[12] S. Yuan, B. Cao, M. Peng, and Y. Sun, "ChainsFL: Blockchain-driven federated learning from design to realization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[13] H. Yoo, J. Yim, and S. Kim, "The blockchain for domain based static sharding," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1689–1692.

[14] L. N. Nguyen, T. D. T. Nguyen, T. N. Dinh, and M. T. Thai, "OptChain: Optimal transactions placement for scalable blockchain sharding," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 525–535.

[15] A. Bugday, A. Ozsoy, and H. Sever, "Securing blockchain shards by using learning based reputation and verifiable random functions," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Jun. 2019, pp. 1–4.

[16] S. Cao, T. Miller, M. Foth, W. Powell, X. Boyen, and C. Turner-Morris, "Integrating on-chain and off-chain governance for supply chain transparency and integrity," 2021, *arXiv:2111.08455*.

[17] K. Miyachi and T. K. Mackey, "HOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102535.

[18] S.-Y. Chang, "Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Cham, Switzerland: Springer, 2019, pp. 241–258.

[19] I. Arribas, "Sandbox for minimal viable governance of blockchain services and DAOs: CLAUDIA," in *Proc. Int. Congr. Blockchain Appl.* Cham, Switzerland: Springer, 2020, pp. 24–30.

[20] T. Chung et al., "Proof of participation voting for on-chain governance," Tech. Rep. 1, 2021.

[21] Y. Li and Y. Zhou, "Research on the reciprocal mechanism of hybrid governance in blockchain," *J. Econ. Manage., Res.*, vol. 121, no. 2, p. 3, 2021.

[22] N. Singh and M. Vardhan, "Computing optimal block size for blockchain based applications with contradictory objectives," *Proc. Comput. Sci.*, vol. 171, pp. 1389–1398, Jan. 2020.

[23] N. Singh and M. Vardhan, "Multi-objective optimization of block size based on CPU power and network bandwidth for blockchain applications," in *Proc. 4th Int. Conf. Microelectron., Comput. Commun. Syst.* Cham, Switzerland: Springer, 2021, pp. 69–78.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[25] A. Thakkar and K. Chaudhari, "A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 2133–2164, 2021.

[26] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, Sep. 2021.

[27] M. Haouari and M. Mhiri, "A particle swarm optimization approach for predicting the number of COVID-19 deaths," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, Dec. 2021.

[28] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on Bitcoin," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 195–209.

[29] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, "SimBlock: A blockchain network simulator," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 325–329.

[30] C. Schwarz-Schilling, S.-N. Li, and C. J. Tessone, "Stochastic modelling of selfish mining in proof-of-work protocols," *J. Cybersecur. Privacy*, vol. 2, no. 2, pp. 292–310, May 2022.

[31] P. Zhang, "Consensus mechanisms and information security technologies," *Adv. Comput.*, vol. 115, pp. 181–209, Jan. 2019.

[32] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "A survey on the adoption of blockchain in IoT: Challenges and solutions," *Blockchain, Res. Appl.*, vol. 2, no. 2, Jun. 2021, Art. no. 100006.

REZA NOURMOHAMMADI received the bachelor's degree in computer engineering, in 2011, and the dual master's degree in artificial intelligence from the K. N. T. U University of Technology University, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with ETS University, Montreal, Canada. His current research interests include machine learning, AI, and blockchain.

KAIWEN ZHANG (Member, IEEE) received the B.Sc. and M.Sc. degrees from McGill University, Montreal, and the Ph.D. degree from the University of Toronto. He was an Alexander von Humboldt Postdoctoral Fellow in computer science at TU Munich. He is currently a Professor with the Department of Software and IT Engineering, ÉTS. His research interests include blockchain technologies, publish/subscribe systems, massive multiplayer online games, performance modeling, and software-defined networking.