Evolved PoW: Integrating the Matrix Computation in Machine Learning Into Blockchain Mining

Yunkai Wei, Zixian An, Supeng Leng, *Member, IEEE*, and Kun Yang, *Senior Member, IEEE*

***Abstract*—Machine learning is an essential technology providing ubiquitous intelligence in Internet of Things (IoT). However, the model training in machine learning demands tremendous computing resource, bringing heavy burden to the IoT devices. Meanwhile, in the Proof-of-Work (PoW)-based blockchains, miners have to devote large amount of computing resource to compete for generating valid blocks, which is frequently disputed for tremendous computing resource waste. To address this dilemma, we propose an Evolved-PoW (E-PoW) consensus that can integrate the matrix computations in machine learning into the process of blockchain mining. The integrated architecture, the elaborated schemes of transferring matrix computations from machine learning to blockchain mining, and the reward adjustment scheme to affect the activity of the miners are, respectively, designed for E-PoW in detail. E-PoW can keep the advantages of PoW in blockchain and simultaneously salvage the computing power of the miners for the model training in machine learning. We conduct experiments to verify the availability and effect of E-PoW. The experimental results show that E-PoW can salvage by up to 80% computing power from pure blockchain mining for parallel model training in machine learning.**

***Index Terms*—Blockchain, consensus, evolved Proof of Work (E-PoW), machine learning, matrix computation.**

I. Introduction

MACHINE learning is an essential technology providing ubiquitous intelligence for the networks and applications in Internet of Things (IoT) [1], [2]. However, the model training in machine learning demands tremendous computing resource, which is usually limited in most IoT devices, and may substantially influence the application and development of machine learning in IoT.

Meanwhile, in Proof-of-Work (PoW) [3], [4]-based blockchains, each miner is equipped with generous computing resource to execute a brute-force search for the target hash value, and thereby competes to generate a valid block. Although widely adopted for better decentralization and security over alternative consensuses, such as Proof of Stake (PoS) or Proof of Activity (PoA) [5], [6], PoW is frequently disputed for tremendous waste in the computing resource. Some novel consensus with similar performance but much less computing resource consumption than PoW is urgent to be developed.

Basically, there are two kinds of consensuses, namely, lightweight consensus and task-based consensus, have been investigated to reduce such resource waste in blockchain and save more computing power for model training. Lightweight consensuses were proposed to replace PoW with the proof of other resources, e.g., PoS [7], Delegated-PoS [8], PoA [9], Practical Byzantine Fault Tolerance-based consensus [10], Proof-of-Luck [11], Raft [12], etc. A profit optimizing game between block mining and service providing was studied for the miners in [13]. However, these consensuses may considerably degrade the decentralization or security of the blockchain system. Instead of decreasing the computing resource consumed by the consensuses, task-based consensuses try to salvage the wasted computing resource in PoW with valuable task processing, e.g., Proof of Exercise (PoX) [14], useful-PoW (uPoW) [15], Proof of Prestige (PoP) [16], Primecoin [17], Proof of Evolution (PoE) [18], Proof of Deep Learning (PoDL) [19], Proof of Federated-Learning (PoFL) [20], etc. However, in these studies, the miners also face a challenge: when the miners solve the same problem in one period, the salvaged computing power will be very limited, otherwise, when they solve different problems in one period, the fairness cannot be guaranteed.

To address this dilemma in both machine learning and blockchain, we propose an Evolved-PoW (E-PoW) consensus to connect the computing resource by realizing the cooperative computation between machine learning and blockchain, and thereby, improve the overall computing efficiency. The underlying connection is the matrix multiplication calculation (MMC). MMC exists widely in machine learning. As an example, nearly 90% of the workload in Google's Tensor Processing Unit is due to multilayer perceptrons (MLPs) and recurrent neural networks (RNNs), which are both deep

Manuscript received 14 January 2022; accepted 27 March 2022. Date of publication 8 April 2022; date of current version 7 April 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0117500; in part by the Natural Science Foundation of China under Grant 62132004 and Grant 61871076; in part by the MOST Major Research and Development Project under Grant 2021YFB2900204; in part by the Sichuan Major Research and Development Project under Grant 22QYCX0168; and in part by the UESTC Yangtze Delta Region Research Institute, Quzhou, under Grant 2021D003 and Grant 2021D013. *(Corresponding author: Kun Yang.)*

Yunkai Wei is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324000, China (e-mail: ykwei@uestc.edu.cn).

Zixian An and Supeng Leng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zxan@std.uestc.edu.cn; spleng@uestc.edu.cn).

Kun Yang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, also with the Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324000, China, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: kunyang@uestc.edu.cn).

Digital Object Identifier 10.1109/JIOT.2022.3165973

learning algorithms based on MMC [21]. At the same time, MMC is feasible to be quantified and integrated into the process of the brute-force search for the target hash value. With integrated architecture and elaborated schemes, E-PoW can integrate the massive MMC of machine learning's model training into the block mining of PoW-based blockchains, keeping the advantages of PoW as well as making the computing power efficiently utilized. The contributions of this article are summarized as as follows.

1) We propose a novel consensus E-PoW to connect the computation of machine learning and blockchain in IoT, where MMC is integrated into the block mining process, and the miners conduct the target hash value search based not only on the traditional block header but also on the result of MMC.
2) We design detailed schemes to transform MMC tasks with different computing complexities into normalized matrix multiplications, which have identical computing complexity and can be easily integrated into the process of block mining through a uniform interface.
3) We put forward a reward adjustment scheme for E-PoW to adjust the miners' tendency of performing more MMC tasks or fewer MMC tasks. Extensive experiment results show the availability and effect of E-PoW.

The remainder of this article is organized as follows. The related works are introduced in Section II. In Section III, we put forward the architecture of E-PoW, and demonstrate the nodes, the decentralization, and security of E-PoW. In Section IV, we propose the elaborated schemes of transferring matrix multiplications from machine learning to blockchain mining. A reward adjustment scheme is presented in Section V to adjust the miners' interest on MMC tasks. Section VI shows the performance of E-PoW by extensive experiments. Finally, Section VII concludes this article and discusses the future work.

II. Related Work

Studies have been conducted to reduce the resource waste in blockchain and save more computing power for model training, which can be classified into two types: 1) lightweight consensuses and 2) task-based consensuses.

The lightweight consensuses dedicate to decrease the computing complexities and alleviate the computing resource waste. As a typical lightweight consensus, the practical-Byzantine-fault-tolerance (PBFT) [10] algorithm, which can well guarantee the distributed systems' strong consistency, has been applied to permissioned blockchain systems. King and Nadal [7] proposed a PoS consensus with a stochastic process similar to PoW, except that in PoS the hashing operation is done over a limited search space, and each miner can consume its *coin age* (the multiplication of the held coins and the corresponding holding time) to reduce the difficulty of generating a valid block. Based on PoS, Larimer proposed a Delegated-PoS (DPoS) consensus [8], in which 101 witnesses selected by stakeholders are responsible for generating new blocks. PoS-Velocity (PoSV) [22], Proof-of-Burn (PoB) [23], and PoA [9] are all consensuses based on the combination of PoW and PoS. The PoSV consensus modifies the linear function of *coin age* and time in PoS to an exponential decay function; thus, the *coin age* of the new coin grows faster than the old coin until it reaches the upper threshold, which can alleviate the phenomenon of hoarding coins. The PoB is featured by the design that the miners compete for generating new blocks by sending their Slimcoins to specific addresses that cannot be retrieved (burning). The more coins burnt can lead to higher probability of successfully mining new blocks. In a PoA-based blockchain, the generation of the empty blocks is still based on PoW, but there is a committee of N-ordered stakeholders responsible for the validity of the transactions. In summary, lightweight consensuses can reduce the resource waste in blockchain, usually with the price of weakened security or decentralization in the blockchain system.

The task-based consensuses try to salvage the wasted computing resource in PoW with valuable task processing, instead of decreasing the computing resource consumed by the consensuses. The PoX [14] consensus uses a pool of task proposals to replace the target hash value search by computing tasks, which however have different complexity levels and lead to unfair competition [24]. Ball *et al.* [15] proposed a new puzzle framework named uPo. In this framework, the primitive PoW puzzle is replaced with useful problems, e.g., k-Orthogonal Vectors, 3SUM [25], all-pairs shortest path [25], etc. Based on PoS, the PoP [16] consensus defines that the probability of generating a new valid block is determined by the miners' prestige, which can be obtained by performing useful work. However, this consensus lacks a measure mechanism of the useful work's complexity and a verification mechanism of the miners' work. Primecoin [17] replaces PoW's target hash value search with the search for two special chains of prime number. Unlike Primecoin, the PoE [18] consensus replaces the PoW puzzle with genetic algorithm problems. The PoDL [19] consensus requires the miners to participate in a competition of deep learning model training to generate valid blocks. Similarly, in PoFL [20], the miners need to join in the model training competition based on the miner pools, and the organizing structure is similar with federated learning. However, Primecoin, PoE, PoDL, and PoFL can only salvage limited computing resource, since all the miners are solving the same problem simultaneously to keep the fairness of the block generating competition.

III. Essentials of E-PoW

In the section, we will introduce the basic architecture of E-PoW as well as the nodes in it, and then demonstrate E-PoW's character on decentralization and security.

A. Basic Architecture

As shown in Fig. 1, the model training generated by the machine learning in IoT continuously produces massive computing requirements for matrix multiplication. These MMC requirements are injected as computing tasks into the blockchain, which is also widely used in IoT.

For simplicity, we assume there is an intermediate node, or so called the coordinator, that is responsible for dividing

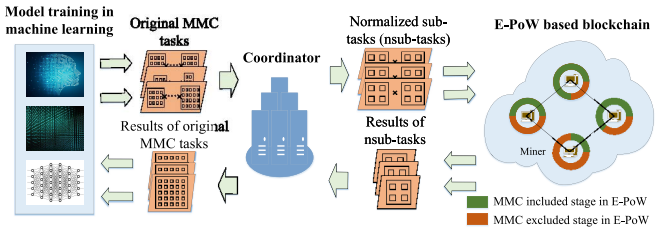


Fig. 1. Basic architecture of E-PoW.

and normalizing the computing tasks into normalized subtasks (nsub-tasks). The coordinator can be a physical device or a logical entity according to the features and requirements of the application scenario.

Then, the nsub-tasks are distributed to the miners in an E-PoW-based blockchain, which will subsequently finish these nsub-tasks. The result of each nsub-task will be verified, integrated, and returned to the MMC task source, i.e., the original training module in machine learning.

According to the E-PoW consensus, a miner has two stages in one block time, namely, MMC included stage and MMC excluded stage. In a given block time, when the MMC tasks have not been finished, the miner is in MMC included stage, otherwise, it is in the MMC excluded stage. In the MMC included stage, the miner conducts the target hash search with four elements: 1) the header in traditional PoW; 2) the task ID; 3) the hash of the matrices in the nsub-task; and 3) the hash of the nsub-task's result it calculated. The miner who first finds the target hash value will broadcast the block it generated, and win the mining reward when the block is accepted by most of the nodes. In the MMC excluded stage, the E-PoW-based miner acts as a traditional PoW-based miner.

B. Nodes in E-PoW

Although the computing tasks are finally finished by the miners in E-PoW, the whole architecture is based on the cooperation of the coordinator, the miners, and the validating nodes. The functions of them are demonstrated as follows.

1) Coordinator: The coordinator divides and normalizes the MMC tasks into nsubtasks, and distributes these nsubtasks to the miners. In addition, it needs to verify (in a way with low overhead) and merge the results from the miners before returning them to the MMC task sources. For possible inquiries and verifications, the coordinator also stores the information of the nsubtasks and their results for feasible verification periods.

The coordinator can be a physical device or a logical entity, according to the application scenario. In a centralized system, such as medical IoTs (medical IoTs) or 5G/6G, the coordinator can be provided by a trustful third party, which is easily integrated with such centralized system. As for a decentralized system such as vehicular ad hoc network (VANET), the coordinator can be a virtual server cooperatively realized with multiple devices. As an example, the miners in blockchain can cooperatively act as the coordinator based on a smart contract, or some distributed protocols, such as PBFT, speculative Byzantine fault tolerance [26], or redundant Byzantine fault tolerance [27]. The cooperating scheme can be further investigated according to the specific application scenario.

Since an original MMC task of calculating $\prod_{j=1}^{s} M_j$ ($s \in \mathbb{N}, s \geq 2$) may contain more than two matrices, a continuous multiplication processing scheme is needed to reduce the tasks' size and complexity. Specifically, an original task will be divided into $s-1$ subtasks, and each subtask is the multiplication of only two matrices. However, the sizes of the subtasks are still different, leading to different computing complexities and workloads. If we directly assign such subtasks to the miners, it will be hard to keep the miners having identical external computing workload. To solve this problem, we design a normalization scheme in Section IV, which provides a uniform pattern for the miners to complete the subtasks fairly. In the normalization scheme, a subtask is transformed into several nsubtasks, and each nsubtask is the multiplication of two normalized matrices, which have identical matrix order $k \times k$. In this way, the nsubtasks have the same computing complexity and workload.

After normalization, the nsubtasks with ID (including the original task ID, the subtask ID, and the nsubtask ID) will be distributed to the miners. When the miners send back the results of the nsubtasks, the coordinator will perform a low overhead verification, and those valid results will be merged and finally fetched to the MMC task sources. At the same time, the coordinator stores both the tasks and the results for possible inquiries and verifications.

One miner can get the rewards if its result matrices are verified. The coordinator will increase the credit of the miner after each successful calculation of a nsubtask. But there are still chances that one miner may have not completed its nsubtasks after a specified period, or be found cheating in completing the nsubtasks. Three methods are adopted to avoid such problem as follows.

1) The nsubtasks are allocated to the miners based on the nsubtasks' delay sensitivity. The nsubtasks with the highest delay requirement will be allocated to the miners with the highest credit on historical latency performance and so on.
2) Recovery of the interrupted machine learning. Based on the nsubtask allocation method, the interruption mainly occurs in those nsubtasks with lower delay requirement. In such occasion, the unfinished nsubtasks will be reassigned to other miners in next period, until all these nsubtasks have been correctly finished.
3) Punishment of the malicious miner. When a miner has malicious behaviors in finishing the nsubtasks, the coordinator will correspondingly decrease the credit of this miner and stop assigning new nsubtasks to this miner in a specified period. This will block the miner out of any profit from completing MMC tasks during this period, and more cheating times or fault times will lead to a longer forbidden period.

2) Miners: An E-PoW-based miner m_i ($i \in \mathbb{N}$) may conduct two types of loops to execute a brute-force search for the target hash value, namely, task-free loop in MMC excluded stage and task-involved loop in MMC included stage. In the task-free loop, m_i does not participate in MMC tasks and

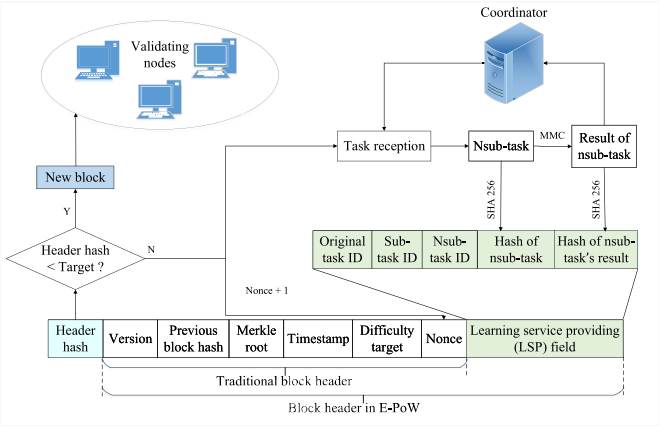


Fig. 2. Task-involved loop.

Algorithm 1 Task Execution in the Block Time of b_j

Input:
N_i^j, $\phi = 0$, $\omega = 0$.
Output:
The block b_j.

```
while ω = 0 do
  if φ < N_i^j then
    Execute one task-involved loop.
    φ = φ + 1.
  else
    Execute one task-free loop.
  end if
  Update ω, if the block b_j has been generated by the miner m_i or other miners, set ω = 1.
end while
```

acts as a traditional PoW-based miner. In the task-involved loop, m_i executes the brute-force search based not only on the traditional PoW's header (including the nonce) but also a learning-service-providing (LSP) field consisting of five elements: 1) original task ID; 2) subtask ID; 3) nsub-task ID; 4) hash of nsub-task; and 5) hash of nsub-task result, as shown in Fig. 2. To keep the fairness of the miners in generating valid blocks, a difficulty adjustment scheme is adopted to set feasible hash difficulties for the task-involved loops, which will be demonstrated in Section IV-D.

In the beginning of a block time for generating block b_j ($j \in \mathbb{N}$), the miner m_i first decides the number of the nsubtasks N_i^j it intends to complete during this block time. Then, m_i conducts N_i^j task-involved brute-force loops with each loop containing the computation of one nsubtask. As shown in Algorithm 1, one of the two cases may happen during this process.

1) The N_i^j task-involved loops are completed while b_j is not yet successfully generated. m_i will conduct task-free loops until some miner (or itself) generates and broadcasts a valid block b_j. In such case, m_i may increase the number of nsubtasks it claims in the next block time.
2) Block b_j is successfully generated before the N_i^j loops are completed. m_i will broadcast (or verify, when m_i is not the generator) this block, postpone the left nsubtasks to the next block time, and decrease the number of nsubtasks it claims in the next block time.

As shown in Fig. 2, in each task-involved loop, m_i calculates the nsubtask of this loop it fetches from the coordinator, fills in the LSP field based on the information and result matrix of this nsub-task, and submits the later to the coordinator. If the result matrix is verified, m_i will get the corresponding task reward. Otherwise, the nsubtasks of m_i in this block time will be shifted to other available miners and no new nsubtasks will be assigned to m_i in the next block time. The validation of the block b_j generated in a task-involved loop should include the LSP field, in which the task information can be inquired from the coordinator.

3) Validating Nodes: A validating node is a blockchain node that contains a full blockchain database. It verifies the validity of the received block, adds the valid block to its own blockchain, and passes the valid block along to other nodes, thus enabling only the valid blocks are propagated on the blockchain network.

After receiving a newly generated block broadcasted by a miner m_i, the validating nodes will reject it if the reputation value of m_i is less than the reputation threshold. Otherwise, the validating nodes will inquire the coordinator for the result data and the task data corresponding to this block. If the result matrix, result hash, and task hash are all correct, and the header hash satisfies the target hash difficulty, this block will be accepted by the validating nodes. A reputation value updating process will be triggered after the verification. The reputation maintaining schemes have been studied in many literature [28]–[31], and can be flexibly adopted to our proposed architecture.

C. Decentralization and Security of E-PoW

Besides the capability of connecting machine learning and blockchain, E-PoW can simultaneously keep the advantages of PoW in aspects of both decentralization and security, which are two major considerations in blockchain systems.

1) Decentralization: In an E-PoW-based blockchain, the possibility of generating a valid block for a miner is directly proportional to its computing power, based on the follows.

1) *Ensured Fairness:* Each task-free loop has identical computing complexity with other task-free loops, keeping the fairness of all the task-free loops, and so do the task-involved loops in one miner and one block time, since the nsub-task involved in each task-involved loop also has statistically identical computing complexity. For the fairness between task-free loops and task-involved loops, as well as the fairness among task-involved loops in different miners and different block times, a hash difficulty adjustment scheme is proposed and will be described in detail in Section IV-D.
2) *Independent Validation:* Although an intermediate node named coordinator is introduced for task assigning and result fetching, the independence of new block validation will not be influenced. From the validating nodes' perspective, the data and result of one task can be identified

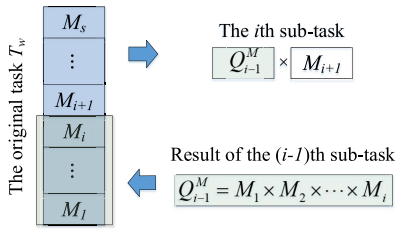


Fig. 3. Continuous multiplication processing.

with the Hash scheme. In addition, before verifying the correctness of a miner's result, the validating node will first verify whether the nsubtask and the result, which are both returned from the coordinator, have been tampered. This is executed based on the hash of the nsubtask and the hash of this nsubtask's result in the block submitted by the miner. Therefore, the validating nodes can ensure the integrity of the data fetched from the coordinator.

2) Security: In PoW-based blockchains, 51% attack is the major security threat [32], [33]. E-PoW inherits from PoW's basic idea of conducting the brute-force search for the target hash value, and maintains a similar reward framework. The essence of E-PoW is still computing power competition, and it is fundamentally a proof of work consensus. Consequently, the analysis on PoW in [32]–[35] is still applicable to E-PoW. Therefore, an E-PoW-based blockchain's tolerated power of the adversary is also no less than 51%.

Moreover, E-PoW-based blockchain is an endogenously increasing system since its interest producing scheme can continuously enroll new positive participants pursuing the conveniences or the profits. The additional revenue of completing an MMC task is deterministic and friendly to the miners with limited computing power, which, in PoW-based blockchains, would fall through for extremely low revenue from block mining [36] and thereby had more impulsion to take adversarial behaviors. All these factors can cooperatively increase the security of E-PoW-based blockchains.

IV. Processing of Matrix Multiplication Calculation

MMC is the underlying connection for E-PoW to integrate the model training into the block mining. In this section, we will present the detailed processing schemes for MMC according to the processing sequence, namely, the continuous multiplication processing scheme, which divides one original task into several subtasks, the normalization scheme, which transforms the subtasks into nsub-tasks with uniform matrix size, the result verifying and merging scheme, and the difficulty adjustment scheme, which can ensure the effective operation of E-PoW.

A. Continuous Multiplication Processing

An original MMC task may contain more than two matrices. In order to reduce the task size, we propose the continuous multiplication processing scheme as follows.

As shown in Fig. 3, for an original MMC task $T_w = \prod_{i=1}^{s} M_i$ $(w, s \in \mathbb{N}, s \geq 2)$ multiplying s matrices M_i $(i = 1, \ldots, s)$, the coordinator will divide it into $s-1$ subtasks, namely, $M_1 \times M_2$, $Q_1^M \times M_3, \ldots, Q_{i-1}^M \times M_{i+1}, \ldots, Q_{s-2}^M \times M_s$, where $Q_i^M = Q_{i-1}^M \times M_{i+1}$ $(i \in \mathbb{N}, i \leq s-1)$ and $Q_0^M = M_1$. Each subtask will be further divided into several normalized nsubtasks, so that each nsubtask has uniform computing complexity. Identified with $< task\ id, sub\_task\ id, nsub\_task\ id >$, these nsubtasks will be calculated in sequence by the miners, until the final result matrix Q_{s-1}^M is achieved.

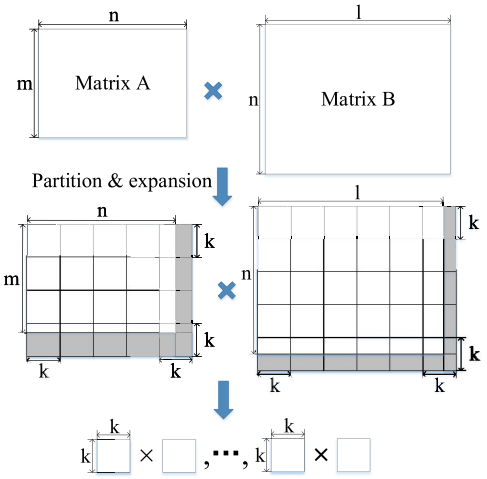


Fig. 4. Normalization of MMC.

B. Normalization of the Subtasks

The matrix sizes of the subtasks obtained through the continuous multiplication processing scheme are still different from each other, resulting in the difference of the computing workload. We propose a normalization scheme to transform the MMC subtasks into nsubtasks with uniform matrix size. Although matrices of the same size may have different sparsity and the computation workloads are not identical for all nsubtasks, each miner has the opportunity to be assigned matrices of various sparsity. Statistically, the computation workload of each miner to complete the nsubtasks tends to be identical after an appropriate time, especially when the nsubtasks are from correlative machine learning models belonging to the same application scenario.

Let $T_{w,u}$ $(u \in \{1, \ldots, s-1\})$ denote a subtask of T_w, and A and B denote the two matrices in $T_{w,u}$, i.e., $T_{w,u} = A \times B$. Suppose matrix A has m rows and n columns, and B has n rows l columns $(m, n, l \in \mathbb{N})$. Or in another word, the size of A is $m \times n$, and that of B is $n \times l$. As shown in Fig. 4, the subtask of calculating $A \times B$ is divided into several nsubtasks of calculating the multiplication between two k-order square matrices. The scheme contains two steps: 1) matrix expansion and 2) matrix partition.

1) Matrix Expansion: The matrix A can be expanded as

$$A' = \begin{bmatrix} A & Z_1 \\ Z_2 & Z_3 \end{bmatrix} \tag{1}$$

where the order of $A\prime$ is $m' \times n'$ $(m' = \lceil m/k \rceil k,\ n' = \lceil n/k \rceil k)$, and Z_1, Z_2, and Z_3 can be random matrices or random parts

from matrix A, with orders $m \times (n' - n)$, $(m' - m) \times n$, and $(m' - m) \times (n' - n)$, respectively.

Similarly, matrix B' can be expanded from matrix B as

$$B' = \begin{bmatrix} B & Z_4 \\ Z_5 & Z_6 \end{bmatrix} \tag{2}$$

where the order of $B\prime$ is $n' \times l'$ ($l' = \lceil l/k \rceil k$). Z_4 and Z_6 can also be random matrices or random parts from matrix B, and Z_5 satisfies $Z_1 \times Z_5 = O$ (O is null matrix). The orders of Z_4, Z_5, and Z_6 are $n \times (l' - l)$, $(n' - n) \times l$, and $(n' - n) \times (l' - l)$, respectively.

The multiplication of A' and B' is

$$A'B' = \begin{bmatrix} AB & AZ_4 + Z_1Z_6 \\ Z_2B + Z_3Z_5 & Z_2Z_4 + Z_3Z_6 \end{bmatrix}. \tag{3}$$

Consequently, the $m \times n$ order submatrix in the upper left corner of $A'B'$ is just the result matrix of the subtask $T_{w,u}$.

2) Matrix Partition: Through matrix expansion, we get matrices A' and B', whose orders are all integral multiples of k, and thereby, they can be partitioned as follows:

$$A' = \begin{bmatrix} A'_{11} & \cdots & A'_{1\lceil n/k \rceil} \\ \vdots & \ddots & \vdots \\ A'_{\lceil m/k \rceil 1} & \cdots & A'_{\lceil m/k \rceil \lceil n/k \rceil} \end{bmatrix} \tag{4}$$

$$B' = \begin{bmatrix} B'_{11} & \cdots & B'_{1\lceil l/k \rceil} \\ \vdots & \ddots & \vdots \\ B'_{\lceil n/k \rceil 1} & \cdots & B'_{\lceil n/k \rceil \lceil l/k \rceil} \end{bmatrix} \tag{5}$$

where A'_{iq} and B'_{qj} are both k-order square matrices ($i = \{1, \ldots, \lceil m/k \rceil\}$, $q = \{1, \ldots, \lceil n/k \rceil\}$, $j = \{1, \ldots, \lceil l/k \rceil\}$).

According to the multiplication principles of block matrix, we can get the result matrix of $A' \times B'$ by calculating $A'_{iq} \times B'_{qj}$. That is to say, the subtask $T_{w,u}$ can be divided into $N^{nsub}_{w,u}$ nsubtasks $T_{w,u,v}$ ($v \in \{1, \ldots, N^{nsub}_{w,u}\}$) of calculating $A'_{iq} \times B'_{qj}$, where $N^{nsub}_{w,u} = \lceil m/k \rceil \lceil n/k \rceil \lceil l/k \rceil$.

C. Result Merging and Verifying

1) Result Merging: The normalized nsubtasks are continuously assigned by the coordinator to the miners according to each miner's initiative on finishing MMC tasks. Specifically, a nsubtask $T_{w,u,v}$ will be assigned to a miner, calculated in this miner's task-involved loop, and the result will be returned to the coordinator. When all the results of the nsubtasks $T_{w,u,v}$ from subtask $T_{w,u}$ are collected, the coordinator will merge these result matrices to get the result matrix of $T_{w,u}$ as follows:

$$R_{w,u} = \begin{bmatrix} R_{11} & \cdots & R_{1\lceil l/k \rceil} \\ \vdots & \ddots & \vdots \\ R_{\lceil m/k \rceil 1} & \cdots & R_{\lceil m/k \rceil \lceil l/k \rceil} \end{bmatrix} \tag{6}$$

where R_{iqj} denotes the result matrix of $A'_{iq} \times B'_{qj}$, and $R_{ij} = \sum_{q=1}^{\lceil n/k \rceil} R_{iqj}$ ($i = \{1, \ldots, \lceil m/k \rceil\}$, $q = \{1, \ldots, \lceil n/k \rceil\}$, $j = \{1, \ldots, \lceil l/k \rceil\}$). Then, the $m \times n$ order submatrix in the upper left corner of $R_{w,u}$ is the result matrix of the subtask $T_{w,u}$. When $u < s - 1$, $R_{w,u}$ is an interim result of T_w. Otherwise, when $u = s - 1$, $R_{w,u}$ contains the final result matrix of T_w.

2) Result Verifying: The verification of the result for a nsubtask may be conducted by the coordinator or the validating nodes. In some extreme environment with adversaries aiming at destroying the model training instead of cheating for the reward, existing verification algorithms such as the Freivalds algorithm [37], [38] can be adopted. Otherwise, we can use a verification scheme with little computing overhead as follows.

Define the nsubtask to be verified is calculating $X \times Y$ (X and Y are two k-order square matrices), and the result matrix is S. Select two random integers s_r and s_c ($1 \le s_r, s_c \le k$), then use the s_rth row vector of X and the s_cth column vector of Y to calculate the verification value s, which can be expressed as

$$s = \sum_{i=1}^{k} x_{s_r i} y_{i s_c} \tag{7}$$

where $x_{s_r i}$ denotes an element from matrix X, and $y_{i s_c}$ denotes that from matrix Y. By judging whether s and $s_{s_r s_c}$ are equal, we can check the correctness of the result matrix S ($s_{s_r s_c}$ denotes the element of matrix S). If they are equal, the result matrix is right. Otherwise, it is malicious.

As an example, consider a decentralized system such as vehicular ad hoc network, the miners in blockchain cooperatively act as the coordinator based on a smart contract, then the miners will cooperatively verify the correctness of the nsubtask's result. The performance of the verification algorithm can be analyzed as follows.

1) *Probability of Successful Cheating in One Verification:* Let ρ denote the ratio of the correct elements to all elements in S. Since each miner randomly selects ϵ elements in S for inspection, the probability of the result matrix passing the verification of one miner can be expressed as

$$P^{\text{pass}}_{one\_verification} = \left(\frac{\rho k^2}{k^2}\right)^{\epsilon} = \rho^{\epsilon}. \tag{8}$$

We suppose that $\rho = 0.9$, which means that only one-tenth of the nsubtask has not been finished. When $\epsilon = 10$, the value of $P^{\text{pass}}_{one\_vefication}$ is 0.3487.

2) *Probability of Successful Cheating in One Block Time:* Define the number of the miners is N_{vld}. Generally, a blockchain system may have hundreds or thousands of miners. Once one miner finds the result manipulated, it will broadcast the manipulated element in this matrix to other validating nodes. Then, in one block time, the probability of the result matrix passing the verification of all miners can be expressed as

$$P^{\text{pass}}_{one\_blocktime} = \left(\frac{\rho k^2}{k^2}\right)^{\epsilon N_{vld}} = \rho^{\epsilon N_{vld}}. \tag{9}$$

When $\rho = 0.9$, $\epsilon = 10$, and $N_{vld} = 100$, there is $P^{pass}_{one\_blocktime} = 1.75 \times 10^{-46}$. This is a quite low probability.

3) *Probability of Successful Cheating in Multiple Block Times:* Suppose the cheater cheats in N_{cheat} block times (can be continuous or noncontinuous block times), then in these N_{cheat} block times, the probability of the result

matrix passing the verification of all miners can be expressed as

$$P_{N_{\text{cheat}}}^{\text{pass}} = \left(\frac{\rho k^2}{k^2}\right)^{\epsilon N_{vld} N_{\text{cheat}}} = \rho^{\epsilon N_{vld} N_{\text{cheat}}}. \quad (10)$$

When $\rho = 0.9$, $\epsilon = 10$, $N_{vld} = 100$, and $N_{\text{cheat}} = 10$, there is $P_{N_{\text{cheat}}}^{\text{pass}} = 2.66 \times 10^{-458}$. This probability can be neglected.

Moreover, in our verification scheme, once a node is found cheating, it will be removed out from the blockchain system. Although a rejoining scheme can be used, the cost of cheating can be designed much higher than the benefit from cheating.

D. Difficulty Adjustment

In an E-PoW-based blockchain system, the difficulty for a miner m_i to generate a new block b_j, which can be expressed as D_i^j and can decide the block generation rate (BGR), is influenced by the basic difficulty and a scale factor, as follows.

Suppose the system adjusts the basic difficulty every time when W blocks have been generated, then the length of the basic difficulty adjustment window is W. Let $D_{\mathbb{J}}^{\text{base}}$ denote the basic difficulty at block b_j, where $\mathbb{J} = \lceil j/W \rceil$. The basic difficulty adjustment algorithm can be expressed as

$$D_{\mathbb{J}}^{\text{base}} = D_{\mathbb{J}-1}^{\text{base}} * \frac{WT_{\text{tar}}}{T_{\text{span}}} \quad (11)$$

where $1/T_{\text{tar}}$ is the target average BGR, and T_{span} denotes the actual timespan for generating W blocks. According to (11), if the system's actual BGR is larger than the target BGR, which means that the system generates blocks faster than expected, the basic difficulty will be increased, and *vice versa*.

The scale factor for miner m_i in the block time of generating b_j can be expressed as follows:

$$F_i^j = \frac{1}{\alpha N_i^j + 1} \quad (12)$$

where α is an incentive factor ($\alpha \geq 0$), and N_i^j denotes the number of the nsubtasks that miner m_i wants to complete at the block time of generating b_j.

According to (12), F_i^j is inversely proportional to N_i^j, so the lower the enthusiasm of the miner to complete tasks, the larger the scale factor is. Furthermore, when the miner chooses not to participate in completing tasks, the scale factor is the maximum value 1. Then, the difficulty for the miner m_i to generate a new valid block b_j is

$$D_i^j = D_{\mathbb{J}}^{\text{base}} F_i^j. \quad (13)$$

According to (13), the basic difficulty is the same for all the miners, but the scale factor will change with the number of tasks that the miners want to complete. According to (12), F_i^j is inversely proportional to N_i^j, as a consequence, D_i^j is inversely proportional to N_i^j, which means that the more active the miners are involved in completing tasks, the lower their mining difficulty will be. Furthermore, when all the miners will not take any tasks, their scale factors will be 1, and the E-PoW-based system will degenerate into a traditional PoW-based system, where all the miners have the same hash difficulty, which equals the basic difficulty.

V. Reward Adjustment Scheme

In an E-PoW-based system, the reward should be adjusted according to the demand (from Machine learning) and the supply (from blockchain) on the computing power. In different application systems, the demand and the supply are both different. Consequently, different systems have different requirements for the miners to participate in MMC tasks. Specifically, when the demand is larger than the supply, we should encourage the miners to devote more computing resource in task-involved loops. When the demand is smaller than the supply, the miners should devote less computing resource in task-involved loops. That is to say, the miners should be encouraged to take more task-free loops.

To influence each miner's initiative in MMC tasks, we propose a reward adjustment scheme so that a balance between the number of MMC tasks in machine learning and the amount of computing service in blockchain can be achieved. According to this scheme, the reward can be preset for one system, or online adjusted when the demand and supply are dynamically changing.

Clock period (CP) can be used to measure the processing workload in E-PoW. For simplicity and without loss of generality, we suppose there are n miners in an E-PoW-based blockchain, and the integrated clock frequency of the CPUs in each Miner is f. Let C^{ti} denote the number of CPs required for a miner to conduct a task-involved loop, and C^{tx} denote that for a task-free loop. Then

$$C^{tx} = C_{\text{head}}^{\text{hash}} \quad (14)$$

$$C^{ti} = C^{nt} + C_{\text{task}}^{\text{hash}} + C_{rslt}^{\text{hash}} + C_{\text{ehead}}^{\text{hash}} \quad (15)$$

where C^{nt} denotes the number of CPs required for a miner to complete a nsubtask, and $C_{\text{head}}^{\text{hash}}$ and $C_{\text{ehead}}^{\text{hash}}$ denote that for computing the hash of a PoW header and an E-PoW header, respectively. $C_{\text{task}}^{\text{hash}}$ and C_{rslt}^{hash} represent the number of CPs required for a miner to calculate the task hash and result hash, respectively. Generally, there is $C^{nt} >> C_{\text{ehead}}^{\text{hash}} \approx C_{\text{head}}^{\text{hash}}$. That is, the number of CPs required by computers to complete a normalized subtask is much larger than the number of CPs required to calculate the E-PoW header hash, and the difference between $C_{\text{ehead}}^{\text{hash}}$ and $C_{\text{head}}^{\text{hash}}$ can be ignored considering it is trivial comparing to C^{nt}.

Theorem 1: Let r_t denote the reward for successfully completing a nsubtask, r_b denote the reward for generating a valid block, and $1/T_j$ denote current BGR of generating the valid block b_j. Suppose the number of nsubtasks completed by miner m_i is N_i^j in this block time. Then, in block time b_j, the total reward $e_i^{\text{sum}}(j)$ that m_i can obtain is

$$e_i^{\text{sum}}(j) = N_i^j r_t + \frac{r_b V_i^j}{I_i^j + V_i^j} \quad (16)$$

where

$$V_i^j = \left(T_j f - N_i^j \left(C^{ti} - C^{tx}\right)\right)\left(\alpha N_i^j + 1\right) \quad (17)$$

$$I_i^j = \sum_{l=1}^{i-1} V_l^j + \sum_{l=i+1}^{n} V_l^j. \quad (18)$$

Proof Idea: The total reward that the miner m_i can obtain during the competition of generating block b_j is composed of two parts: 1) the task reward and 2) the block reward. Based on the number of the nsubtasks completed by m_i in block time b_j and the reward for per nsub-task, we can get the task reward that the miner can obtain. By deriving the times m_i calculates the hash of block header in this block time, we can get the hash rate of m_i, and then get the probability of m_i winning the competition, which can be used to calculate the miner's expected block reward. Consequently, we can get the total reward of m_i in block time b_j as in (16)–(18). ■

Based on Theorem 1, we can change the variables to adjust the total reward function and hence influence the initiative of the miners to participate in conducting task-involved loops and task-free loops.

The miners always want to maximize their own interests. Consequently, when we need the miners to take more task-involved loops, there should be $e_i^{\text{sum}}(j) \propto N_i^j$. In this instance, the system parameters should satisfy the following Theorem 2.

Theorem 2: In the case of $\alpha \geq ([C^{ti}(C^{ti}-C^{tx})]/C^{tx}T_jf)$, $e_i^{\text{sum}}(j)$ is proportional to N_i^j when the following inequality holds:

$$\left(\frac{T_jf r_b(C^{ti}-2C^{tx})}{C^{ti}} - r_t C^{tx}\left(\frac{T_jf}{C^{ti}}\right)^2\right)\alpha < r_t T_jf\left(\frac{C^{tx}}{C^{ti}}+(n-1)\right) - r_b(C^{ti}-C^{tx}). \tag{19}$$

In the case of $0 < \alpha < ([C^{ti}(C^{ti}-C^{tx})]/[C^{tx}T_jf])$, $e_i^{\text{sum}}(j)$ is proportional to N_i^j when the following inequality holds:

$$\left(\frac{T_jf r_b\left(C^{ti}-2C^{tx}\right)}{C^{ti}} - nr_t C^{tx}\left(\frac{T_jf}{C^{ti}}\right)^2\right)\alpha < \frac{nr_t T_jf C^{tx}}{C^{ti}} - r_b\left(C^{ti}-C^{tx}\right). \tag{20}$$

Proof Idea: By analyzing the partial derivative of $e_i^{\text{sum}}(j)$ with respect to N_i^j, we find that in the definitional domain of N_i^j, $([\partial^2 e_i^{\text{sum}}(j)]/[\partial(N_i^j)^2])$ is always smaller than 0, which means that $([\partial e_i^{\text{sum}}(j)]/[\partial N_i^j])$ is inversely proportional to N_i^j. According to this conclusion, we find the minimum value of $([\partial e_i^{\text{sum}}(j)]/[\partial(N_i^j)])$ in the domain of N_i^j, then we use the scaling method to derive the conditions that make the inequality $\min\{([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)\} > 0$ always hold, which ensures that the inequality $[(\partial e_i^{\text{sum}}(j))/\partial(N_i^j)] > 0$ is true, and thereby, $e_i^{\text{sum}}(j)$ is proportional to N_i^j. ■

On the contrary, when we want the miners to take more task-free loops, there should be $e_i^{\text{sum}}(j)$ inversely proportional to N_i^j. In this instance, the system should satisfy the following Theorem 3.

Theorem 3: In the case of $0 < \alpha < [(C^{ti}-C^{tx})/T_jf]$, $e_i^{\text{sum}}(j)$ is inversely proportional to N_i^j when the following inequality holds:

$$C^{tx}(n-1)\left(T_jf\right)^2\alpha^2 + T_jfC^{tx}(n-1)\left(2C^{ti}-C^{tx}\right)\alpha + T_jf\left(nC^{ti}\right)^2\frac{r_t}{r_b} - C^{tx}C^{ti}(n-1)\left(C^{ti}-C^{tx}\right) < 0. \tag{21}$$

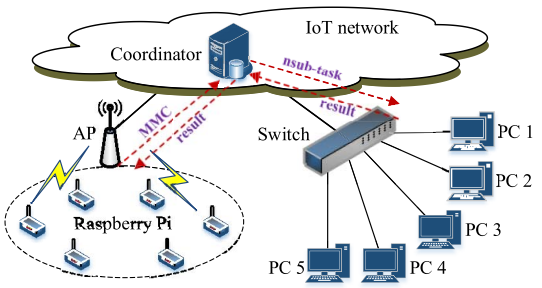


Fig. 5. Experiment environment.

Proof Idea: Similar to Theorem 2, as $([\partial e_i^{\text{sum}}(j)]/[\partial(N_i^j)])$ is inversely proportional to N_i^j, we use the scaling method to get the upper bound of $([\partial e_i^{\text{sum}}(j)]/[\partial(N_i^j)])$, and then find the conditions that make the upper bound smaller than 0, which can ensure $([\partial e_i^{\text{sum}}(j)]/[\partial(N_i^j)]) < 0$, and thereby, $e_i^{\text{sum}}(j)$ is inversely proportional to N_i^j. ■

VI. Experiment and Results Analyses

In this section, we conduct experiments to evaluate the availability and performance of the E-PoW consensus. We will first introduce the experimental environment, and then demonstrate the experiment results and analyses in detail.

A. Experiment Environment

We build the experiment platform in a simple IoT network. As shown in Fig. 5, six devices based on Raspberry Pi (Model 3B) act as lightweight IoT devices, which need to train the machine learning model, and continuously generate MMC tasks. For simplicity in the experiment, we use a physical coordinator instead of a logical coordinator, i.e., we use a trusted ThinkSystem SR860 server to act as the coordinator, which transforms the MMC tasks into nsubtasks for the miners and returns the results back to the machine learning devices. Five personal computers with Intel Core i5-6500 CPU @ 3.20 GHz, which are connected with an Ethernet switch, play the part of five miners in an E-PoW-based blockchain in IoT.

According to the hardware of our experiment platform, it takes about 0.48 s for the miners to execute one task-involved loop, and the target average BGR is set to 1/300 block per second. Consequently, the maximum number of nsubtasks that each miner can complete in a block time is set to 600.

In practice, an E-PoW-based blockchain network may be composed of one or two kinds of miners, namely, miners do not participate in MMC tasks, and miners participate in MMC tasks. In order to ensure the representativeness of our experiment, we run the E-PoW consensus in five types of blockchain networks. Type I network only contains the miners do not participate in MMC tasks. In Type II–IV networks, the miners all participate in the MMC tasks, but with different degrees of participation. Type V network is a hybrid network containing all kinds miners in Type I–IV networks. The detailed parameters in our experiments are demonstrated in Table I.

TABLE I
EXPERIMENT PARAMETERS

Parameters	Value
k	500
r_b	1
Initial difficulty	0.0025
Target average BGR	1/300 block/s
Type I miner	Claims 0 tasks in each block time
Type II miner	Claims 200 tasks in each block time
Type III miner	Claims 400 tasks in each block time
Type IV miner	Claims 600 tasks in each block time
Type I Network	Contains only Type I miners
Type II Network	Contains only Type II miners
Type III Network	Contains only Type III miners
Type IV Network	Contains only Type IV miners
Type V Network	Contains all miners from Type I to Type IV

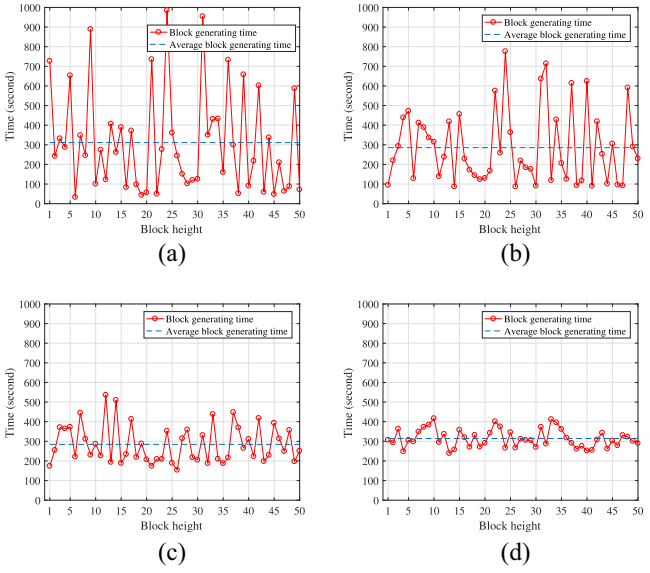


Fig. 6. Block height versus block generating time. (a) Type I network. (b) Type II network. (c) Type III network. (d) Type IV network.

B. Experiment Results and Analyses

We conduct the experiment for 50 times, and show the average results and analyses in three aspects, namely, the time of generating a block, the efficiency of the miners' computing power, and the rewards of the miners.

1) Time of Generating Block: Fig. 6(a)–(d) shows the time to generate a valid block in Type I, II, III, and IV network, respectively, where each type of network contains the maximum number of miners. As shown in Fig. 6, the average block generating time in each network is around 300 s, which just matches the target average BGR, and indicates that the difficulty adjustment scheme works well. In addition, comparing Fig. 6(a)–(d), the fluctuation range of the block generating time is gradually reduced. This fluctuation comes from the randomness of the process in finding the target hash value of the E-PoW block header. This is because that when the miners take more computing tasks, the blockchain system will be more stable in the time needed to generate a valid new block. The transaction delay can consequently be more stable and the number of extreme long block time can be efficiently reduced.

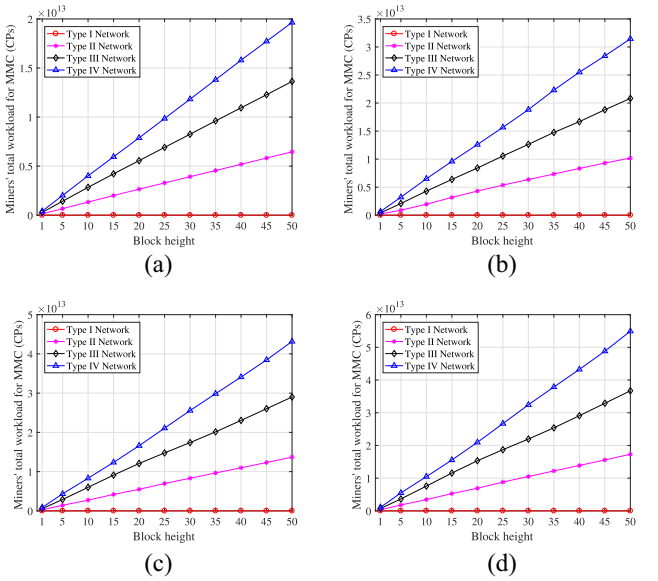


Fig. 7. Block height versus the miners' total MMC workload. (a) Two miners in each network. (b) Three miners in each network. (c) Four miners in each network. (d) Five miners in each network.

2) Efficiency of the Miners' Computing Power: Fig. 7(a)–(d) illustrates the impact of the blockchain's height on the miners' total workload in MMC. Specifically, in order to show the impact of the network scale, Fig. 7(a)–(d) provides the trend of the miners' total workload when the number of miners varies from 2 to 5 in each type of network.

As shown in Fig. 7(a), the total computing power used for MMC tasks in the miners increases when the block height gets larger. That is to say, the amount of salvaged computing power in the miners will be continuously increasing with the operation of the blockchain. In addition, compared with Type I network, during the same time span, Type II, III, and IV network not only generate 50 valid blocks but also provide MMC services for machine learning. Specifically, among the four types of blockchain networks, Type IV network salvages the most amount of computing power for MMC tasks from machine learning, followed by Type III network, Type II network, and Type I network in sequence. The reason is that the types of miners in these networks are different. The miners in Type IV network are most active in conducting task-involved loops, which means that they complete the most tasks in each block time. Thus, compared with other networks, the amount of effectively utilized computing power in Type IV network is the largest. Therefore, the computing power of E-PoW-based network can be utilized more effectively than the computing power of the traditional PoW-based network. The more active the miners are involved in MMC tasks, the more computing power is effectively salvaged.

Comparing Fig. 7(a)–(d), we can find the amount of the salvaged computing power for MMC tasks also increases with the expansion of the network scale. Take Type IV network as an example, when the network has only two miners and the block height is 50, the miners' total computing resource salvaged for MMC tasks is about 2×10^{13} CPs. When the number of miners grows to 5, the salvaged computing resource

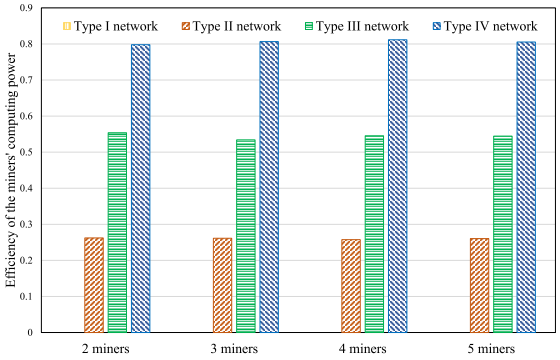


Fig. 8. Impact of network scale on the efficiency of the miners' computing power.

increases to 5.5×10^{13} CPs. Consequently, unlike traditional PoW-based blockchains, the expansion of network scale will lead to more computing resource salvaged, or in another word, less computing resource wasted.

Fig. 8 illustrates the impact of the network scale on the efficiency of the miners' computing power. For each network type, although the number of miners is different, the efficiency change of the computing power is negligible. That is to say, the difference of the network scale has little influence on the efficiency of the computing power in the miners. At the same time, the efficiency of the computing power is proportional to the initiative of the miners in completing MMC tasks. In Type I network, which is a traditional PoW network, the miners utilize all their computing power for target hash value search. That is to say, besides the block mining, no computing power is salvaged and effectively used for MMC tasks. As a comparison, in Type II, III, and IV network, the number of nsubtasks completed by each miner during one block time reaches their corresponding maximum value, and the efficiency of their computing power improves considerably by up to 80% compared with that in PoW-based blockchains.

3) Rewards of the Miners: To verify the effect of the reward adjustment scheme, we build the so called Type V network in Table I, and conduct the experiment with all four kinds of miners, i.e., Type I–IV miners. Specifically, the number of tasks that one Type I miner can take in a block time is slightly modified into a range of [0, 100], and [0, 200], [0, 400], [0, 600] for Type II–IV miners, respectively, so that the miners can flexibly adjust the tasks they want to take according to the reward adjustment scheme. Moreover, in our experiment, the system scale is extremely limited by the hardware platform with only four miners, and is very sensitive to the number changing of MMC tasks. To prevent a severe fluctuation in the system, we set a factor of 2% to the optimal task number, so as to limit the adjusting scope that one miner can make in one block time. In practical blockchain systems, the factor can be much larger, or even ignored since these systems are more stable for their much larger scales.

Fig. 9(a) and (b) shows the scenario that the reward adjustment scheme encourages the miners to take more tasks in the process of the block mining. In this scenario, the slightly modified Type I–IV miners can flexibly and gradually adjust

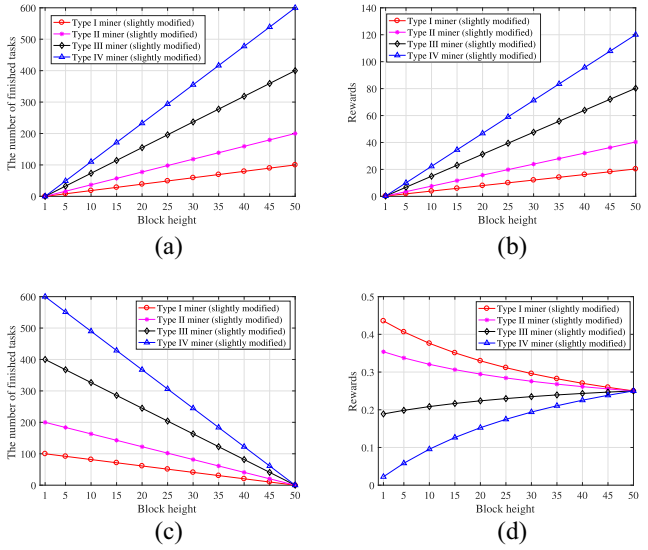


Fig. 9. Influence of the reward adjustment scheme on the miners. (a) Tasks finished by the miners under task-encouraging reward. (b) Rewards of the miners under task-encouraging reward. (c) Tasks finished by the miners under task-discouraging reward. (d) Rewards of the miners under task-discouraging reward.

their resource allocation to take more tasks in one block time, and therefore, achieve more total rewards. Fig. 9(c) and (d) shows the scenario that the miners are discouraged to participate in MMC tasks from machine learning. In this scenario, the Type I–IV miners gradually reduce the number of tasks they take in one block time, and their total rewards will converge to a common number since each miner only has the reward from block mining in the final state. Note that there is a big gap between the rewards in Fig. 9(b) and (d), this is because that too small system scale brings an abnormal ratio of *(task reward)/(block reward)*. Such gap will vanish in a blockchain system with normal scale such as Bitcoin or Ethereum.

VII. Conclusion and Future Work

This article has proposed an E-PoW consensus to integrate vast MMC tasks of machine learning's model training into the block mining in blockchain systems, as well as the detailed schemes transforming MMC tasks in machine learning to nsubtasks computable in the miners. A reward adjustment scheme has also been presented for E-PoW to adjust the initiative of the miners in taking nsubtasks. Based on the elaborated schemes, E-PoW can keep the fairness, independence, and security of generating valid blocks in the integrated system connecting machine learning and blockchain. Extensive experiments show that E-PoW can salvage by up to 80% computing power from pure block mining for parallel model training in the next-generation intelligent applications, and the proposed reward adjustment scheme can lead the miners' behaviors and total rewards effectively as expected.

For future work, the cooperation scheme of forming a logical coordinator with multiple distributed devices can be further investigated according to the application scenario. The latency, which is introduced by transferring computing tasks from

machine learning to blockchain mining, should be thoroughly studied and optimized. The energy consumption and security challenge of the entire system can be analyzed theoretically.

APPENDIX A
PROOF OF THEOREM 1

Proof: According to the difficulty adjustment scheme, the miner m_i needs to declare N_i^j to the coordinator before generating block b_j, then it will receive N_i^j nsubtasks. So during the period when block b_j is generated (which can be expressed as T_j), the time that miner m_i spends on conducting task-involved loops can be expressed as

$$t_i^{\text{epow}}(j) = \frac{N_i^j C^{ti}}{f}. \tag{22}$$

As $t_i^{\text{epow}}(j)$ should be smaller than T_j, so we can get N_i^j's domain of definition as follows:

$$\left\{ N_i^j \in \mathbb{R} | 0 \le N_i^j \le \frac{T_j f}{C^{ti}} \right\}. \tag{23}$$

In fact, N_i^j should be a discrete integer rather than a continuous real number. Since we will analyze the monotonicity of the series with N_i^j as the independent variable later, and the monotonicity of the series is consistent with the monotonicity of its corresponding continuous function. Therefore, in order to facilitate the derivative analysis, we set N_i^j as a continuous real number, which does not affect the correctness of the analysis.

The remaining time of T_j, which is spent on conducting task-free loops, can be expressed as

$$t_i^{\text{pow}}(j) = T_j - t_i^{\text{epow}}(j). \tag{24}$$

During generating block b_j, the miner m_i calculates the head hash $N_i^{\text{hash}}(j)$ times, which can be expressed as

$$N_i^{\text{hash}}(j) = N_i^j + \frac{t_i^{\text{pow}}(j) f}{C^{tx}}. \tag{25}$$

Therefore, the average hash rate of miner m_i at block b_j can be expressed as

$$H_i^j = \frac{N_i^{\text{hash}}(j)}{T_j}. \tag{26}$$

At block b_j, the reward of miner m_i includes two parts: 1) the reward from completing tasks and 2) the reward from generating a valid block. We define the reward for completing a nsubtask is r_t, and that for generating a block is r_b.

The reward that miner m_i can get from completing tasks at block b_j can be expressed as

$$e_i^t(j) = N_i^j r_t. \tag{27}$$

Lemma 1: The probability of miner m_i generating the block b_j can be expressed as

$$P_i^j = \frac{H_i^j / D_i^j}{\sum_{q=1}^{n} H_q^j / D_q^j}. \tag{28}$$

Proof: According to [4, Th. 1], the probability of generating a block is proportional to the ratio of computing power and the assigned difficulty, that is, $P_i^j \propto (H_i^j / D_i^j)$. As a consequence, we can calculate the probability of miner m_i generating the block b_j by dividing (H_i^j / D_i^j) with the summation of all ratio of the hash rate and the assigned difficulty. ■

According to Lemma 1, the miner m_i has the probability of P_i^j to win the competition of generating block b_j, so the expectation reward that miner m_i can get from generating a block at block b_j can be expressed as

$$e_i^b(j) = P_i^j r_b. \tag{29}$$

Finally, during competing for generating block b_j, the total reward of miner m_i can be expressed as

$$\begin{aligned} e_i^{\text{sum}}(j) &= e_i^t(j) + e_i^b(j) \\ &= N_i^j r_t + \frac{r_b V_i^j}{I_i^j + V_i^j} \end{aligned} \tag{30}$$

where V_i^j and I_i^j can be expressed as

$$V_i^j = \left(T_j f - N_i^j (C^{ti} - C^{tx}) \right) \left(\alpha N_i^j + 1 \right) \tag{31}$$

$$I_i^j = \sum_{l=1}^{i-1} V_l^j + \sum_{l=i+1}^{n} V_l^j. \tag{32}$$

■

APPENDIX B
PROOF OF THEOREM 2

Proof: The partial derivative of $e_i^{\text{sum}}(j)$ with respect to N_i^j can be expressed as follows:

$$\frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j} = r_t + \frac{I \left(V_i^j \right)' r_b}{\left(I + V_i^j \right)^2} \tag{33}$$

where $(V_i^j)'$ can be expressed as

$$\left(V_i^j \right)' = \frac{\partial V_i^j}{\partial N_i^j} = -2\alpha \left(C^{ti} - C^{tx} \right) N_i^j + T_j f \alpha - \left(C^{ti} - C^{tx} \right). \tag{34}$$

Our goal is to find some conditions so that the function in (33) is always larger than zero in the domain of N_i^j. In order to analyze the positive and negative of (33), we calculate the second partial derivative of $e_i^{\text{sum}}(j)$ with respect to N_i^j as follows:

$$\frac{\partial^2 e_i^{\text{sum}}(j)}{\partial \left(N_i^j \right)^2} = r_b I_i^j \frac{\left(V_i^j \right)'' \left(I_i^j + V_i^j \right) - 2 \left(\left(V_i^j \right)' \right)^2}{\left(I_i^j + V_i^j \right)^3} \tag{35}$$

where $(V_i^j)''$ can be expressed as

$$\left(V_i^j \right)'' = \frac{\partial^2 V_i^j}{\partial \left(N_i^j \right)^2} = -2\alpha \left(C^{ti} - C^{tx} \right). \tag{36}$$

Lemma 2: The workload of the task-involved loop and the workload of the task-free loop satisfy the following relationship: $C^{ti} > 2C^{tx}$.

Proof: C^{tx} and C^{ti} can be expressed as

$$C^{tx} = C^{\text{hash}}_{\text{head}} \tag{37}$$

$$C^{ti} = C^{nt} + C^{\text{hash}}_{\text{task}} + C^{\text{hash}}_{rslt} + C^{\text{hash}}_{\text{ehead}} \tag{38}$$

where C^{nt} denotes the number of CPs required for a miner to complete a nsubtask, and $C^{\text{hash}}_{\text{head}}$ and $C^{\text{hash}}_{\text{ehead}}$ denote that for computing the hash of a PoW header and an E-PoW header, respectively. $C^{\text{hash}}_{\text{task}}$ and C^{hash}_{rslt} represent the number of CPs required for a miner to calculate the task hash and result hash, respectively. Generally, there is $C^{nt} >> C^{\text{hash}}_{\text{ehead}} \approx C^{\text{hash}}_{\text{head}}$. Therefore, $C^{ti} > 2C^{tx}$. ■

Lemma 3: V_i^j in Theorem 1 (17) satisfies the relationship that $V_i^j > 0$.

Proof: According to Lemma 2, C^{ti} is larger than $2C^{tx}$, so $C^{ti} - C^{tx} > 0$. According to the definition domain of N_i^j, $0 \le N_i^j \le (T_j f / C^{ti})$, so we can get that

$$T_j f - N_i^j (C^{ti} - C^{tx}) > \frac{T_j f}{C^{tx}} > 0. \tag{39}$$

Furthermore, α is positive, hence

$$\alpha N_i^j + 1 > 0. \tag{40}$$

Finally, the product of $T_j f - N_i^j (C^{ti} - C^{tx})$ and $\alpha N_i^j + 1$ should be larger than 0. ■

Lemma 4: I_i^j in Theorem 1 (18) satisfies the relationship that $I_i^j > 0$.

Proof: According to Lemma 3, V_i^j is positive. Obviously, the sum of positive numbers should also be positive. ■

According to Lemma 2 and the definition of α, $C^{ti} > 2C^{tx}$, $\alpha > 0$, so $(V_i^j)''$ is smaller than 0. According to Lemmas 3 and 4, V_i^j and I_i^j are both positive, hence, we can find that $[(\partial^2 e_i^{\text{sum}}(j))/\partial (N_i^j)^2]$ is smaller than 0, which means that $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ is inversely proportional to N_i^j. Consequently, we can get the minimum value of $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ at $N_i^j = (T_j f / C^{ti})$, which can be expressed as

$$\begin{aligned} \min\left\{\frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\right\} &= \frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\} \\ &= r_t + \frac{r_b I_i^j \left(V_i^j\right)'\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}{\left(I_i^j + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}\right)^2}. \end{aligned} \tag{41}$$

By using the scaling method, we can get the inequality relation about $\min\{([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)\}$ as follows:

$$\begin{aligned} &\min\left\{\frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\right\} \\ &> r_t + \frac{r_b\left(I_i^j + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}\right)\left(V_i^j\right)'\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}{\left(I_i^j + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}\right)^2}. \end{aligned} \tag{42}$$

We apply the scaling method on the right-hand side of the inequality in (42) to get the following inequality:

$$\begin{aligned} &r_t + \frac{r_b\left(V_i^j\right)'\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}{I_i^j + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}} \\ &> r_t + \frac{r_b\left(V_i^j\right)'\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}{\min\left\{I_i^j\right\} + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}} \end{aligned} \tag{43}$$

where the left-hand side of the inequality is the simplification of the right-hand side of the inequality in (42).

Consequently, we can get the lower bound of $\min\{([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)\}$, which can be expressed as

$$\min\left\{\frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\right\} > r_t + \frac{r_b\left(V_i^j\right)'\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}{\min\left\{I_i^j\right\} + V_i^j\left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}}. \tag{44}$$

If the lower bound of $\min\{([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)\}$ is larger than 0, we can achieve the goal of making $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ always positive in the domain of N_i^j.

The minimum value of I_i^j can be expressed as follows:

$$\min\left\{I_i^j\right\} = (n-1)\min\left\{V_i^j\right\} \tag{45}$$

where V_i^j is the quadratic function of N_i^j, its minimum value can be expressed as

$$\min\left\{V_i^j\right\} = \begin{cases} V_i^j\{N_i^j = \frac{T_j f}{C^{ti}}\} & 0 < \alpha < \frac{C^{ti}(C^{ti} - C^{tx})}{C^{tx} T_j f} \\ V_i^j\{\ N_i^j = 0\} & \alpha \ge \frac{C^{ti}(C^{ti} - C^{tx})}{C^{tx} T_j f} \end{cases}. \tag{46}$$

Let L denote the lower bound of $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$, then it can be expressed as follows:

$$L = \begin{cases} r_t + \frac{r_b (V_i^j)'\{N_i^j = \frac{T_j f}{C^{ti}}\}}{n V_i^j\{N_i^j = \frac{T_j f}{C^{ti}}\}} & 0 < \alpha < \frac{C^{ti}(C^{ti} - C^{tx})}{C^{tx} T_j f} \\ r_t + \frac{r_b (V_i^j)'\{N_i^j = \frac{T_j f}{C^{ti}}\}}{(n-1) V_i^j\{\ N_i^j = 0\} + V_i^j\{N_i^j = \frac{T_j f}{C^{ti}}\}} & \alpha \ge \frac{C^{ti}(C^{ti} - C^{tx})}{C^{tx} T_j f}. \end{cases} \tag{47}$$

When L is positive, the inequality $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j) > 0$ is true, which means that $e_i^{\text{sum}}(j)$ is proportional to N_i^j. Consequently, calculate $L > 0$ and simplify, we can get the conclusion in Theorem 2. ■

APPENDIX C
PROOF OF THEOREM 3

Proof: Our goal is to find the conditions that make the inequality $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j) < 0$ true within the domain of N_i^j. In the proof of Theorem 2, we have proved that $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ is inversely proportional to N_i^j. Therefore, we can get the maximum value of $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ at $N_i^j = 0$, which can be expressed as

$$\begin{aligned} \max\left\{\frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\right\} &= \frac{\partial e_i^{\text{sum}}(j)}{\partial N_i^j}\left\{N_i^j = 0\right\} \\ &= r_t + \frac{r_b I_i^j \left(V_i^j\right)'\left\{N_i^j = 0\right\}}{\left(I_i^j + V_i^j\left\{N_i^j = 0\right\}\right)^2}. \end{aligned} \tag{48}$$

If we want to make $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ smaller than 0, its maximum should be negative. In the proof of Theorem 2, we have proved that I_i^j is positive. Therefore, the only way to

make $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j) < 0$ is to make $(V_i^j)'\{N_i^j = 0\} < 0$, which means that the following inequality should be true:

$$\alpha < \frac{C^{ti} - C^{tx}}{T_j f}. \tag{49}$$

In this situation, we can apply the scaling method on $\max\{([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)\}$ as follows:

$$r_t + \frac{r_b I_i^j \left(V_i^j\right)' \left\{N_i^j = 0\right\}}{\left(I_i^j + V_i^j \left\{N_i^j = 0\right\}\right)^2} < r_t + \frac{r_b \min\left\{I_i^j\right\} \left(V_i^j\right)' \left\{N_i^j = 0\right\}}{\left(\max\left\{I_i^j\right\} + V_i^j \left\{N_i^j = 0\right\}\right)^2}. \tag{50}$$

The minimum value of I_i^j has been shown in (45), and the maximum value of I_i^j can be expressed as

$$\max\left\{I_i^j\right\} = (n-1)\max\left\{V_i^j\right\}. \tag{51}$$

Consequently, we need to calculate the maximum and minimum value of V_i^j. When the inequality in (49) is true, they can be expressed as follows:

$$\max\left\{V_i^j\right\} = V_i^j \left\{N_i^j = 0\right\} \tag{52}$$

$$\min\left\{V_i^j\right\} = V_i^j \left\{N_i^j = \frac{T_j f}{C^{ti}}\right\}. \tag{53}$$

Then, we can get U, which is the upper bound of $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$. It can be expressed as

$$U = r_t + \frac{r_b (n-1) V_i^j \left\{N_i^j = \frac{T_j f}{C^{ti}}\right\} \left(V_i^j\right)' \left\{N_i^j = 0\right\}}{\left(n V_i^j \left\{N_i^j = 0\right\}\right)^2}. \tag{54}$$

When U is negative, the inequality $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j) < 0$ is true, which means that $([\partial e_i^{\text{sum}}(j)]/\partial N_i^j)$ is inversely proportional to N_i^j. Therefore, calculate $U < 0$ and simplify the result, then we can get the conclusion in Theorem 3. ■

REFERENCES

[1] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.

[2] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020.

[3] S. Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System." 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] C.-N. Chou, Y.-J. Lint, R. Chen, H.-Y. Chang, I.-P. Tu, and S.-W. Liao, "Personalized difficulty adjustment for countering the double-spending attack in proof-of-work consensus protocols," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2018, pp. 1456–1462.

[5] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Convent. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.

[6] "Top 100 Cryptocurrencies by Market Capitalization." 2021. [Online]. Avaliable: https://coinmarketcap.com

[7] S. King and S. Nadal. "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake." Aug. 2012. [Online]. Available: http://peercoin.net/assets/paper/peercoin-paper.pdf

[8] D. Larimer. "Delegated Proof-of-Stake (DPOS)." 2014. [Online]. Available: https://bitshares.org/technology/delegated-proof-of-stake-consensus

[9] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," IACR, Lyon, France, Rep. 2014/452, 2014.

[10] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, 1999, pp. 173–186.

[11] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proc. 1st Workshop Syst. Softw. Trusted Execution*, 2016, pp. 1–6.

[12] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. 2014 USENIX Annu. Tech. Conf. (USENIX ATC)*, 2014, pp. 305–319.

[13] Y. Wei, M. Xiao, N. Yang, and S. Leng, "Block mining or service providing: A profit optimizing game of the PoW-based miners," *IEEE Access*, vol. 8, pp. 134800–134816, 2020.

[14] A. Shoker, "Sustainable blockchain through proof of exercise," in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, 2017, pp. 1–9.

[15] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Proofs of useful work," *IACR Cryptol. ePrint Arch.*, IACR, Lyon, France, Rep. 2017/203, 2017.

[16] M. Król, A. Sonnino, M. Al-Bassam, A. Tasiopoulos, and I. Psaras, "Proof-of-prestige: A useful work reward system for unverifiable tasks," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, 2019, pp. 293–301.

[17] S. King. "Primecoin: Cryptocurrency with Prime Number Proof-of-Work." 2013. [Online]. Available: http://primecoin.io/bin/primecoin-paper.pdf

[18] F. Bizzaro, M. Conti, and M. S. Pini, "Proof of evolution: Leveraging blockchain mining for a cooperative execution of genetic algorithms," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, 2020, pp. 450–455.

[19] C. Chenli, B. Li, Y. Shi, and T. Jung, "Energy-recycling blockchain with proof-of-deep-learning," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 19–23.

[20] X. Qu, S. Wang, Q. Hu, and X. Cheng, "Proof of federated learning: A novel energy-recycling consensus algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 8, pp. 2074–2085, Aug. 2021.

[21] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2017, pp. 1–12.

[22] L. Ren. "Proof of Stake Velocity: Building the Social Currency of the Digital Age." 2014. [Online]. Available: https://coss.io/documents/whitepapers/reddcoin.pdf

[23] "Slimcoin: A Peer-to-Peer Crypto-Currency With Proof-of-Burn." P4Titan. 2014. [Online]. Available: https://github.com/slimcoin-project/slimcoin-project.github.io/blob/master/whitepaperSLM.pdf

[24] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[25] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Average-case fine-grained hardness," in *Proc. 49th Annu. ACM SIGACT Symp. Theory Comput.*, 2017, pp. 483–496.

[26] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," in *Proc. 21st ACM SIGOPS Symp. Oper. Syst. Principles*, 2007, pp. 45–58.

[27] P.-L. Aublin, S. B. Mokhtar, and V. Quéma, "RBFT: Redundant byzantine fault tolerance," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, 2013, pp. 297–306.

[28] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[29] H. Khelifi, S. Luo, B. Nour, H. Moungla, and S. H. Ahmed, "Reputation-based blockchain for secure ndn caching in vehicular networks," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, Oct. 2018, pp. 1–6.

[30] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based byzantine fault-tolerance for consortium blockchain," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2018, pp. 604–611.

[31] T. Do, T. Nguyen, and H. Pham, "Delegated proof of reputation: A novel blockchain consensus," in *Proc. Int. Electron. Commun. Conf.*, 2019, pp. 90–98. [Online]. Available: http://doi.acm.org/10.1145/3343147.3343160

[32] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[33] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018.
[34] D. Bradbury, "The problem with bitcoin," *Comput. Fraud Security*, vol. 2013, no. 11, pp. 5–8, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1361372313701015
[35] N. Anita and M. Vijayalakshmi., "Blockchain security attack: A brief survey," in *Proc. 10th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2019, pp. 1–6.
[36] S. Wu, Y. Chen, M. Li, X. Luo, Z. Liu, and L. Liu, "Survive and thrive: A stochastic game for DDoS attacks in bitcoin mining pools," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 874–887, Apr. 2020.
[37] R. Freivalds, "Probabilistic machines can use less running time," in *Proc. IFIP Congr.*, vol. 839, 1977, p. 842.
[38] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

Yunkai Wei received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016.

He is currently an Associate Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. He is also an Associate Professor with the Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou, China. His research interests include blockchain and machine learning, wireless communications and networks, and the Internet of Things.

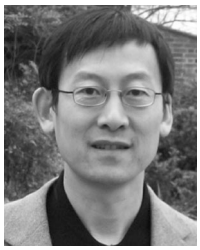
Zixian An received the bachelor's degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2019, where he is currently pursuing the master's degree with the School of Information and Communication Engineering.

His research interests include blockchain and wireless networks.

Supeng Leng (Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2004.

He is a Full Professor and a Vice Dean of the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. He is also the Leader of the Research Group of Ubiquitous Wireless Networks. He has been working as a Research Fellow with the Network Technology Research Center, NTU. He published over 200 research papers and four books/book chapters in recent years. His research focuses on resource, spectrum, energy, routing and networking in Internet of Things, vehicular networks, broadband wireless access networks, and the next-generation intelligent mobile networks.

Dr. Leng got the best paper awards at four IEEE international conferences. He serves as an organizing committee chair and a TPC member for many international conferences, as well as a reviewer for over 20 international research journals.



Kun Yang (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London (UCL), London, U.K, in 2006.

He is a Chair Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the Network Convergence Laboratory, U.K. He is also an Affiliated Professor with the University of Electronic Science and Technology of China, Chengdu, China. Before joining the University of Essex in 2003, he worked with UCL on several European Union (EU) research projects for several years. He manages research projects funded by various sources, such as U.K. EPSRC, EU FP7/H2020, and industries. He has published 200+ journal papers and filed 20 patents. His main research interests include wireless networks and communications, IoT networking, data and energy integrated networks, and mobile computing.

Dr. Yang serves on the editorial boards of IEEE, such as IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE WIRELESS COMMUNICATIONS LETTERS, and *IEEE Communications Magazine* and non-IEEE journals. He was an IEEE ComSoC Distinguished Lecturer from 2020 to 2021 and is a member of Academia Europaea.