DLBC: A Deep Learning-Based Consensus in Blockchains for Deep Learning Services

Boyang Li, *Member, IEEE*, Changhao Chenli, Xiaowei Xu, *Member, IEEE*,
Yiyu Shi, *Senior Member, IEEE*, and Taeho Jung, *Member, IEEE*

With the increasing artificial intelligence application, deep neural network (DNN) has become an emerging task. However, to train a good deep learning model will suffer from enormous computation cost and energy consumption. Recently, blockchain has been widely used, and during its operation, a huge amount of computation resources are wasted for the Proof of Work (PoW) consensus. In this paper, we propose DLBC to exploit the computation power of miners for deep learning training as proof of useful work instead of calculating hash values. it distinguishes itself from recent proof of useful work mechanisms by addressing various limitations of them. Specifically, DLBC handles multiple tasks, larger model and training datasets, and introduces a comprehensive ranking mechanism that considers tasks difficulty(e.g., model complexity, network burden, data size, queue length). We also applied DNN-watermark [1] to improve the robustness.

In Section V, the average overhead of digital signature is 1.25, 0.001, 0.002 and 0.98 seconds, respectively, and the average overhead of network is 3.77, 3.01, 0.37 and 0.41 seconds, respectively. Embedding a watermark takes 3 epochs and removing a watermark takes 30 epochs. This penalty of removing watermark will prevent attackers from stealing, improving, and resubmitting DL models from honest miners.

Index Terms—**Blockchain, Deep Learning, Proof-of-Useful-Work**

I. INTRODUCTION

BITCOIN [2] is the most popular blockchain technology-based application. Besides countless cryptocurrencies, blockchain technology has been successfully applied in different fields. However, the traditional Proof-of-Work (PoW) consensus mechanism demands an immense amount of energy for computation to maintain the blockchain. According to Digiconomist [3], the estimated power consumption of Bitcoin "mining" reaches around 70 TWh per year during the second half of the year 2018. As a result, there are the concerns and warnings about energy wasting of cryptocurrencies [4], for instance, Camilo Mora published a paper in Nature Climate Change with the title of "Bitcoin emissions alone could push global warming above 2 centigrade" [5].

To maintain the consistency of transactions, the traditional Proof of Work (PoW) consensus mechanism utilizes the brute-force algorithms to host a competition of hardware and energy source, and this is the major component that leads to the energy wasting issue. A series of solutions have been proposed to address this issue, such as ASIC machine [6], Proof of Stake (PoS) [7], Proof of Capacity (PoC) [8] and Proof of Useful Work (PoUW) [6]. ASIC machines compute hash efficiently, but this type of machine is only able to calculate on a certain type of brute-force algorithms and it is relatively inflexible. PoC significantly wastes disk space instead of electricity. PoS mechanism cannot provide solid security as PoW, because determination of the block creators involves efficient computation only, and it does not consume much energy [9]. On contrary, PoUW exploits computation power of "miners" for useful tasks, therefore the energy consumed by the miners is not wasted.

The authors are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA. E-mail: {bli1, cchenli, xxu8, yshi4, tjung}@nd.edu).
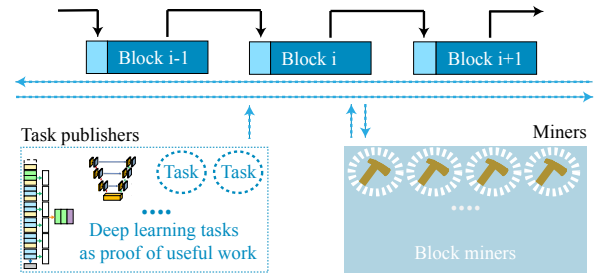

Fig. 1. A blockchain maintained with image Segmentation algorithm as PoUW.

Deep learning plays a crucial role in many fields, for instance, supporting clinic diagnosis [10]–[16]. By taking advantage of rapid increase of computing power and machine learning research interests [17]–[25], the performance of deep learning has been improved significantly.

However, it does require huge computation power to achieve high performance model [12], [26]. In addition, an accurate model requires machine learning experts to tune it by training and evaluating with different hyper-parameters multiple times. Therefore, a good model comes at the cost of very high computation consumption.

Primecoin [27], PoX [28], PrivacyPreserving Blockchain Mining [29], Coin.AI [30], WekaCoin [31], and PoDL [32] are PoUW mechanisms that ask miners to perform valuable tasks. This paper proposes a deep learning based consensus (DLBC) as PoUW, as demonstrated in Fig. 1, to exploit the computation power of "miners" for training deep learning tasks while addressing the limitations of existing PoUW mechanisms. This work solved three challenges: (1) Because the "useful work" in term of training deep learning model is different from hash algorithm, the original consensus will not fit our case. This novel mechanism has to be fair for all miners which

is challenging. (2) Because the DL models and source code are accessible to public, it is necessary and challenging to protect the ownership of all submitted DL models. (3) Because sustainability of this blockchain depends on whether there are enough training tasks, we suggested an mechanism to balance the reward and create incentive for task publisher when the system needs more tasks to train.

In order to address these challenges, we augmented existing memory pool of full nodes to keep all unconfirmed tasks and unselected tasks, a novel task scheduler in each block interval so that miners will work on the same job, and augmented the block data structure to keep the current task and unselected tasks record. To protect the ownership of the DL models, we leveraged DNN-watermark [1] such that the winner miner can prove and protect the ownership with the embedded watermark in the model. Our major contributions are: (1) our mechanism can accept and handle multiple tasks; (2) similar to the state-of-the-art consensus mechanisms based on deep learning, our mechanism can also handle large models and training datasets; (3) we proposed a difficulty score for each submitted task that miners will select a training task based on the scores as the guideline and the mechanism also provide incentive for task publishers; (4) we adopt the DNN-watermark [1] to protect submitted DL models and evaluate the reliability of the embedded watermark method.

As the overhead evaluation shown in Section V indicates, the average overhead of digital signature is 1.25, 0.001, 0.002 and 0.98 seconds, respectively, and the average overhead of network is 3.77, 3.01, 0.37 and 0.41 seconds, respectively. In the watermark evaluation, it demonstrates that embedding a watermark takes 3 epochs and successful removing a watermark takes 30 epochs. The penalty of removing watermark will prevent attackers steeling, improving, and resubmitting DL models from honest miners.

II. BACKGROUND AND RELATED WORK

Multiple machine learning tasks have adopted deep neural networks as solutions and achieved state-of-art results.

Application of deep neural networks in image classification: Image classification is a classic computer vision task that has been existing for decades. Classification can be as simple as recognition of handwritten digits such as MNIST dataset. On the other hand, complex classification problems can have thousands of categories involved which sets high accuracy obstacles even for human [33]. As deep neural networks triumphed the ImageNet [34] challenge since 2012, neural networks have been going deeper and deeper, from AlexNet [35] with 8 layers to as much as ResNet [36] with 152 layers. A Major component of these networks is convolutional layers with several filters. As a result, the stack of dozens of convolutional layers usually requires significant amount of computation power.

Application of deep neural networks in biomedical image segmentation: Biomedical image segmentation is another task which has been boosted tremendously by the development of deep neural networks. The task usually outputs a pixel-wise classification result in the original size of the high-resolution input image. Fully convolutional networks (FCN) is
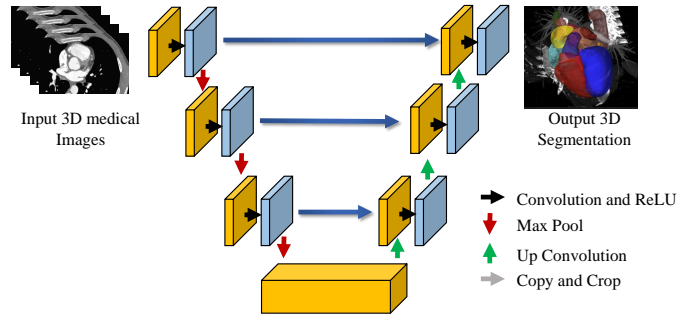

Fig. 2. 3D U-Net [41]: a widely used framework in fully convolutional networks for medical image segmentation.

Fig. 3. Hand written digit samples for images classifier tasks (MNIST) [45]

a special category of DNN, which is widely used for medical image segmentation. Compared with general DNNs, FCNs only consist of convolutional layers, up convolutional layer, and pooling layers as shown in Fig. 2. With this characteristic, FCNs can efficiently output images with the same size as the input images as shown in Fig. 2. Almost all the DNN-based methods for 3D image segmentation adopt FCN as the backbone network structure, and add some special structures and improve training strategies [13], [37]–[44]. For example, 3D U-Net [38] adds more connections between the first several layers and the last several layers as shown in Fig. 2 to better extract features. With all the details implemented in these convolution operation based deep neural networks and the large image size in this specific problem (square of thousands of pixels for single medical image), training for these models is undoubtedly computationally expensive.

Application of deep neural networks in speech recognition: Besides the computer vision area, deep neural networks also dominate the research in speech recognition. Speech recognition aims at translating an input sound wave signal into matching text output. A common approach used to solve the problem would be preprocessing the sound wave into a sequence of slices, then a Recurrent neural network can be applied to generate a corresponding sequence of characters recognized from the input sequence. By training the model with CTC loss, which removes repetition in the output text sequence, the output text can reach a satisfying accuracy.

Consensus mechanisms for blockchain: Proof of Stake (PoS) is a consensus mechanism in cryptocurrencies to decide the creator of the next block based on the amount of cryptocurrencies the creator owns or other weights that can prove the authority of the creator. Determination of the block creators involves efficient computation only, and it does not consume
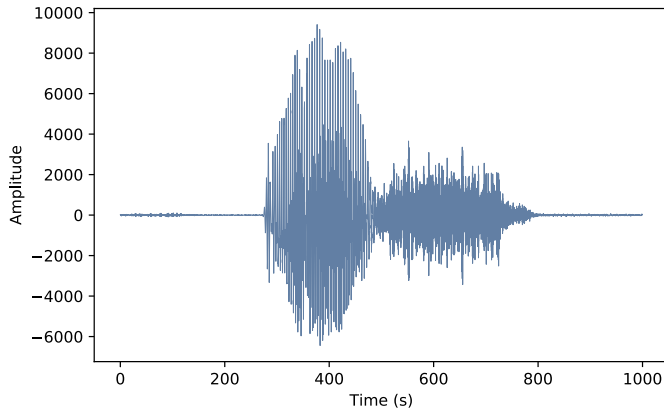
Fig. 4. Single word command audio sample for automatic speech recognition tasks [46]

much energy. However, it is unknown whether PoS mechanism is a robust distributed consensus mechanism owing to various limitations [47]–[49]. Existing cryptocurrencies adopting PoS all have extra rules to make the consensus mechanism more robust, however, the necessity of extra rules implies the PoS is inherently unstable, and the benefit of energy efficiency is diluted.

Proof of Work (PoW) does not suffer from the instability of PoS. Its stability is supported by the enormous computation resources contributed to hard computation problems, and its criticism has been on its large amount of wasted energy. The idea of PoW was proposed to prevent from the denial-of-service attacking [50]. In PoW consensus, all participants are required to solve problems before they send messages. Such problems are challenging to solve and easy to validate. In Bitcoin, miner nodes are required to calculate hash values as PoW.

Proof-of-Deep-Learning (PoDL) In this paper, we inherit the block acceptance policy of PoDL [32], [51] to substitute the hash calculation with segmentation model training. We briefly explain the mechanism before introducing our novel protocols that addresses PoDL's limitations.

Each block interval is divided into two phases, Phase 1 and Phase 2. At the beginning of Phase 1, a task publisher releases the training dataset (with labels) as well as the hyperparameters of the deep learning model to miners and full nodes. During Phase 1, miners train their own deep learning model, and commit their model by the end of Phase 1. The commit process is completed through submitting the hash of their trained models as well as their IDs. At the beginning of Phase 2, the task publisher releases the test dataset to miners and full nodes, and each miner submits (1) the block header and the block that contains information describing the trained model on top of existing attributes, (2) the trained model, and (3) the accuracy of the trained model, to full nodes. Note that the hash of the block header does not need to be smaller than the threshold because the hard computation is replaced with the model training. Full nodes, during Phase 2, validate the submitted models to check whether they have the claimed accuracy, and this happens on top of existing validation in the blockchain (*e.g.,* validation of the correctness of transactions, Merkle tree, hash). To avoid miners over-fitting their models on the disclosed test dataset or stealing others

models (published during Phase 2), full nodes discard any block whose model was not committed during Phase 1 (*i.e.,* hash of the model or ID have not been received in Phase 1). Full nodes will accept the block that is submitted with the highest-accuracy model that claimed its accuracy correctly. They choose and validate the models in decreasing order of the claimed accuracy for that. Such a block acceptance policy yields a robust consensus and is secure against double-spending attack as long as no more than 51% of computation power is owned by the attackers. However, the PoDL is limited in that they can only handle one task at once, and there is insufficient details about how to handle training model.

To address these drawbacks, we present an alternative PoW mechanism that asks miners to perform biomedical image segmentation tasks and present a trained segmentation model as the proof. The major contributions of this paper are: (1) our blockchain allows submissions of multiple tasks, (2) our blockchain can handle large models with large training datasets. These contributions are significant since they make the idea of Proof-of-Useful-Work behind the PoDL more practical and applicable in the real world by supporting multiple tasks and larger predictive models, (3) our blockchain suggested a sustainable task scheduling mechanism which provides incentive for task publishers, thus the system will be sustainable.

There exists other consensus mechanisms based on deep learning as well. PrivacyPreserving Blockchain Mining [29] proposed a Sybil-resistance scheme based on privacy-preserving machine learning as PoUW consensus. In their design, they applied the hybrid consensus protocols [52] which dynamically selects flexible amount of full nodes as committee members and the participants include data providers, miners, the committee and non-committee nodes. In addition, Privacy-Preserving Blockchain Mining [29] introduced their two parallel chains that long-interval for useful work and short-interval for transaction. In the simulation, they especially evaluated the submission conflict scenario. Coin.AI [30] introduced a frame which requires miners to train DL models as PoUW and a proof-of-storage scheme for rewarding users. WekaCoin [31] presented a new distributed consensus protocol which alleviates the computational waste in PoW brute-force algorithms and creates a public distributed and verifiable database of DL models.

All three related works address the first challenge which supports miners to perform DL training as PoUW. PrivacyPreserving Blockchain Mining and WekaCoin may not address the seconnd challenge where an attacker could steal and improve the pre-trained model which submitted from honest miner, and submit improved model before the block is conformed. The Coin.AI addressed the second challenge which required miners to train deep learning models with hyperparameter achieved from the hash value of the previous block. With a pre-fixed hypermarameter, attackers will have less opportunities to improve the honest miner's DL model. Here, DLBC applied embedded watermark in the DL model, therefore, the honest miner can claim the ownership of a model and full nodes will detect the DL model from attackers.

In general, all three related works described similar partic-

ipants with different names. They all give sufficient incentive for miners. But, it is also important to collect tasks for miners to train and all three related works did not provide a proper strategy to encourage task publishers when the system needs more training task. In DLBC, we introduced an algorithm to provide reward for task publishers when the task queue is relative short. In addition, we introduced ranking score which considered the training difficult, data size, model size, network, and task queue.

Other Proof-of-Useful-Work mechanisms: Primecoin [27] is an altcoin that asks the miners to find a special sequence of prime numbers (Cunningham chain) instead. Although the outcome of miners' computation has mathematical and research meaning, *i.e.,* discovering the Cunningham chain. The application of Cunningham chain in the real world is unclear.

Proof of Exercise (PoX) is a design proposed in [28], which is another PoUW mechanism that lets miners perform certain exercises and present the outcome as a proof. In PoX, *employers* publish their tasks onto a board and the miners will randomly fetch tasks from it. The limitation of PoX is that they rely on this centralized board maintained by a third party, which significantly dilutes the decentralization property of the blockchain.

There are similar but orthogonal approaches as well. Hybird Mining [53] and Conquering Generals [54] described a similar mechanism which solves NP-complete problem [55] as the proof in their PoUW consensus. Proof-of-Search [56] addresses the energy waste issue in PoW by solving optimization problems as PoUW.

III. DEFINITIONS AND ASSUMPTIONS

In this section, we define the entities involved in our blockchain, which will be used to support DL training as PoUW.

A. Participants

Miners are the machines of individual or small organizations who wish to contribute their computation power for maintaining a blockchain and may receive rewards as the exchange. In our case, we only consider a standard computer (not ASIC machine) with one or more dedicated graphic cards as a miner, for instance, the gaming machines and deep learning machines. Miners train the DL models as PoW with GPUs, maintain a max heap of submitted tasks based on task rewards and validate the result of potential block owner.

Full nodes record all blocks and transactions, maintain a min heap of submitted tasks based on task reward, validate the submitted tasks and check the checkpoint of miners and validate the result of potential block owner.

Task publishers release biomedical image segmentation training tasks and training data. After a training task is selected and performed by the miners, the corresponding publisher will pay certain amount of reward to the miner presenting the best image segmentation model in the form of the cryptocurrency that is maintained by the blockchain.

B. Assumptions

There are three assumptions that our design relies on. Some of them hold naturally in existing blockchains while others do not.

Assumption 1: We assume task publishers' best interest is to achieve the image segmentation model with the best performance. Therefore, we assume no collusion happens between miners and task publishers, because colluding with miners (*e.g.,* disclosing test datasets to specific miners) will degrade the accuracy of the model only. However, it is true that miners are well motivated to collude with task publishers (even though task publishers are not motivated to do so) since winning miners gain block rewards. It is our future work to achieve a robust consensus mechanism that does not rely on this assumption.

Besides, we also assume task publishers will pay the task reward honestly once their tasks are performed by the miners. However we introduce how to relax this assumption via smart contract by the end of this paper.

Assumption 2: We assume the training tasks can be interrupted and stopped at any time by the miners. We make this assumption because training tasks may be complex and time consuming, but we need to guarantee certain block generation rate. The gap between the length of training time and the short block interval will be handled by allowing the miners to stop the training tasks at any time and submit the saved checkpoint as their proof of work during the block interval. Note that this assumption holds for optimization algorithms that are based on gradient descendent.

Assumption 3: The full nodes' network condition is stable and reliable enough such that all full nodes have the same view on their memory pool and that miners and task publishers can access such view without significant network delay. In addition, we assume the full nodes' clocks are synchronized up to the difference of 5 seconds. These assumptions are necessary for achieving security properties in our blockchain.

IV. DESIGN

We propose the DLBC for maintaining a blockchain which is DL-based consensus. Most of the computation power of miners will be spent on training DL models instead of calculating useless hash values as in existing PoW mechanisms. Each new block is generated by the miner who submits the DL model with the best performance, which will be validated by the full nodes. Once the model is confirmed to be the best, the miner will generate the block and receive both of the task reward and the block reward. The block reward distribution is suggested by Equations 5,7. The major novelty of this paper is to overcome the limitation of a prior work that cannot handle more than one task, large deep learning models and large training datasets.

A. Ranking Mechanism for Multiple Tasks

As aforementioned challenges about sustainability, it is important to collect sufficient training tasks from honest publishers if we can provide rewards when the task list is
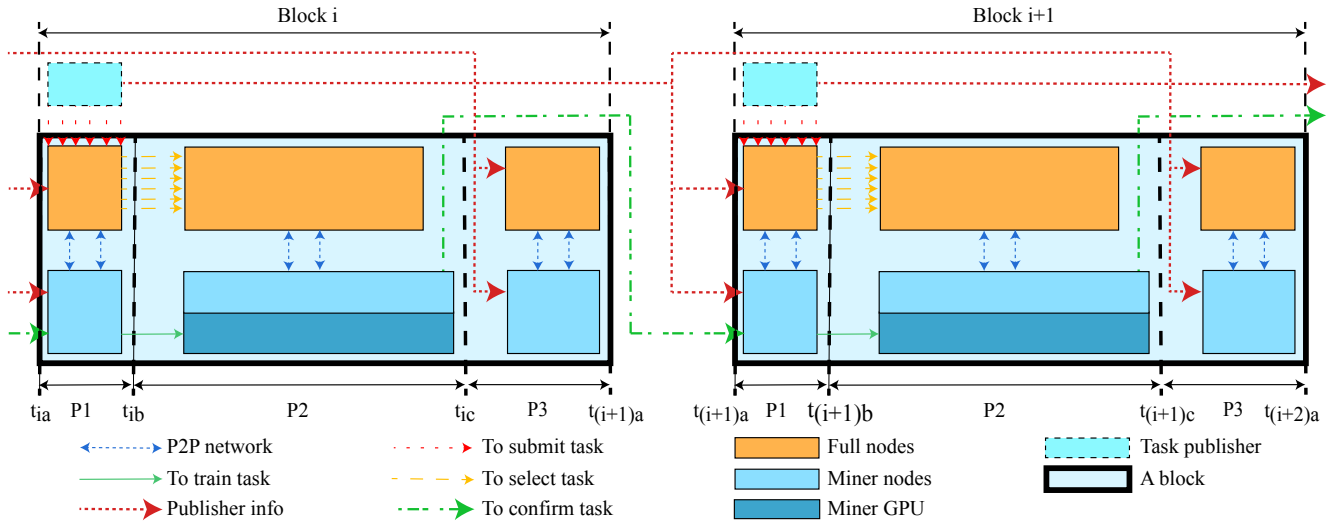
Fig. 5. Description of the block mining in detail.

Fig. 6. Our augmented memory pool.
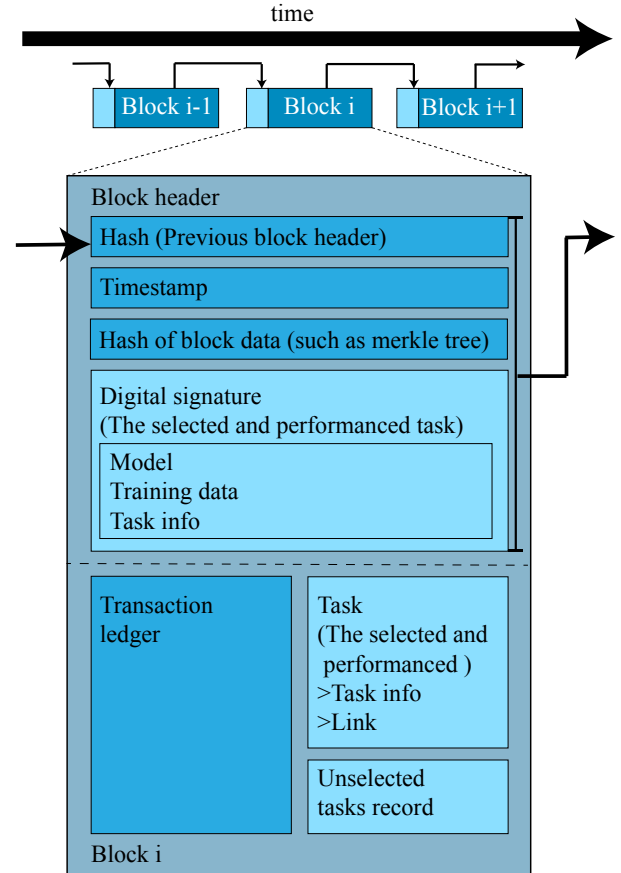

Fig. 7. An overview of the block i.

relative short. We introduce the equations which will schedule DL training tasks based on ranking score.

In Equation 1, q_i is the task queue index which will effect the value of ranking score and block reward distribution. The value of L and l_i, must be positive integers, represent the estimated maximum length and current length of the task queue, respectively. Here k is the scaling factor. When k is bigger than one, the threshold (L/k) of the task queue length will be shorter than K. When k is between zero and one, the threshold of the task queue will be bigger than K.

$$q_i = \frac{\ln (k \times l_i)}{\ln L}, \quad k > 0 \qquad (1)$$

In Equation 2, 3, the value of d_{ni} and d_{ci} represent the hardness of task in terms of transmission over network and Floating-Point Operations (FLOPs) of model, respectively.

$$d_{ni} = \frac{\text{model size} + \text{data size}}{\text{median network bandwidth}} \qquad (2)$$

$$d_{ci} = \frac{\text{FLOPs}}{\text{median computation power}} \qquad (3)$$

As shown in Equation 4, the ranking score of a task is based on the task reward, difficulty of the task (d_{ni} and d_{ci}) and the task queue index (q_i). The Equations 1,2,3,4 will suggest a ranking list of tasks based on the scores. In the current design, all miner nodes will train the task with the smallest ranking score. Thus, the system will prefer a task

with high task reward, small model size and data size. When the length of the task queue is shorter than the threshold (L/k), the system will prefer a complicated task. When the length of the task queue is longer than the threshold, the system will prefer a task with less FLOPs.

$$\text{ranking score} = \frac{d_{ni} + (d_{ci})^{q_i}}{\text{task reward}} \qquad (4)$$

For the block reward, the Equations 5,7 suggest the ratio of block reward distribution that will encourage publishers to submit DL training tasks by sharing block rewards if the length of task queue is shorter than the threshold. This design will try to ensure that the task list will not be empty.

$$\text{block reward}_{\text{publisher}} = max(0, 1 - (\frac{k \times l_i}{L})^2) \qquad (5)$$

$$\text{block reward}_{\text{miner}} = 1 - \text{block reward}_{\text{publisher}} \qquad (6)$$

B. Overview of block mining

In a traditional blockchain, the attributes of the block header, as shown in Fig 7, include the block number, the hash value of the previous block header, the hash of the block data, a timestamp, the size of the block and the nonce value; the block data part contains a ledger that records transactions [57]. We introduce three new attributes to the block header in our blockchain: (1) digital signature of segmentation model, training data, and segmentation task information, (2) the task that is selected and performed by all miners, (3) the list of all unfinished tasks that need to be performed in the future. Each attribute will be explained in the subsequent sections.

In Fig. 5, it shows the scheduling of tasks for block i and block $i + 1$. For each block, the interval is split into three partitions which named as Phase 1 (P1), Phase 2 (P2) and Phase 3 (P3), respectively. For block i, it starts at time t_{ia} and ends by $t_{(i+1)a}$. For each phase in block i, P1 starts at time t_{ia} and ends by t_{ib}; P2 starts at time t_{ib} and ends by t_{ic}; P3 starts at time t_{ib} and ends by $t_{(i+1)a}$. The periods of P1, P2 and P3 are fixed where the length of time for P2 is much longer than those of P1 and P3.

As shown in Fig. 5, the procedure will be introduced in details. In general, the length of Phase 2 is much longer than the length of Phase 3, because the training time will be significantly longer than the testing time. The Phase 1 of block i starts at time t_{ia}. Task publishers can submit DL training tasks during this phase. The publisher will need to submit its own ID, reward, and links (model address and data address). In real world scenarios that memory pool may not hold the same view of task ranking list due to network delay, it may need an additional ledger to confirm all submitted tasks and a target task will be selected from the confirmed list. However, as it is claimed in Assumption 3, full nodes will hold the same view on memory pool, thus we will not consider this case and will address this issue in the future. In the current work, all submitted tasks will be recorded in the unselected list, as described in Section IV-D. At the same time, the miners are training the segmentation task which was ranked at the first place by the end of the last block. After a task appeared in the unselected task list, it will join the ranking which is based on the task reward. Only the task with the highest task reward will win the chance to be trained.

Once the target training task is confirmed, miner nodes start fetching data and model, and the publisher of the selected task will release the key to the encrypted model. After the model and data are ready, each miner will evaluate the complexity of the task by training the model for one epoch. Then miners will start training with GPUs for a certain number of epochs. The number of epochs was evaluated by each individual miner and it can be different among miners. This number is measured to ensure that the miner can stop training before Phase 3, yet finish the last entire epoch. The behavior of other participators is described in Sections IV-D,IV-E.

As shown in Fig 5, the primary job of Phase 3 is to test and validate the biomedical image segmentation model which was trained during Phase 1 of the current block. When the time t_{ib} (shown in Fig. 5) arrives, the publisher will provide API for miner nodes to test their own model and for full nodes to validate the winner model.

Miner nodes generate a digital signature which is shown as Fig 7 digital signature frame. At the same time, miners will check the accuracy of their own models. All miners will submit their accuracy values and model links. In addition, the model link is required to include all checkpoint models for verification purpose. This policy is to make sure that the final model is truly a trained model. The full nodes will sort all submitted accuracy values and verify the model with the highest accuracy. The miner, who submitted the best model, will generate the block. Meanwhile, as described in Section IV-D, the task ledger will be confirmed which also means all unconfirmed tasks are moved into unselected task list. The target task for the next block is selected from the updated unselected task list, and traditional transaction confirmation is finished.

The essential property of blockchain is that any full node should be able to verify the history data. In this case, it is necessary to check that the accepted biomedical image segmentation models are trained from training dataset only. In addition, the testing results must be the same as they were claimed. As described above, full nodes will be able to fetch all the checkpoint models as the reference to verify the training through the model link in the task ledger in block data.

C. DL model ownership protection with DNN-watermark

Once a miner submits a model, it is very important to protect the ownership. Otherwise, attackers will be able to steal others submissions to generate seemingly valid blocks without performing the actual work. For example, an attacker may attempt to steal a published model and perform short-term extra training to generate a distinct model. Such an attack will lead to a valid block since the block and the model are not bound to each other unlike the hash value and the block in PoW-based consensus. To prevent such attacks, we adopted DNN-watermark to protect DL model ownership [1].

1) Embedding watermarks into deep neural networks

In order to claim and protect the ownership of the deep learning model, the miner will generate a watermark and embed the watermark into the deep neural network.

For each individual miner, they will firstly generate a unique watermark from the current block data which is derived from the hash of the block body and the block header. We will apply the direct watermark method as described in the DNN-watermark [1]. A watermark in this method is a binary matrix of n columns and m rows. The method requires that each column of the matrix should contain only one 1 and the rest are 0s. We will use the first two rows to encode the hash of the block body and the block header into the matrix, which is the watermark to be embedded into DL models. The encoding rule is to check the hash value bit by bit. If the i-th bit of the hash is 0, the first row of the i-th column is 1 in the watermark matrix; if the i-th bit of the hash is 1, the second row of the i-th column is 1 in the watermark matrix. Since the first transaction (i.e., CoinBase transaction) is unique, the hash value of the block data and the corresponding watermark matrix is unique unless there is a hash collision, which can be neglected since a cryptographic hash (e.g., SHA-256) is adopted in practice.

The generated watermark is embedded into DL models via regular DL training methods with an extra term in the loss function that is optimized during the training (Eq. 7). $E(w)$ is the total loss function, where $E_0(w)$ is the original loss function related to the classification errors and $E_R(w)$ is the extra regularizer term for embedding the watermark. Because a watermark-related term is introduced, the value of weights in the model will be updated to follow the requested distribution such that the watermark matrix will be embedded into the weights. This distribution of the weight values can be easily detected and this will be demonstrated in the experiment section. As described in DNN-watermark [1], the normal regularizer will address the over-fitting issue and this additional regularizer will train the weights of model to follow certain statistical bias.

$$E(w) = E_0(w) + \lambda \times E_R(w) \qquad (7)$$

2) Determination of existence of watermark

Once a miner submit a DL model and corresponding watermark which is unique as discussed before, full nodes will determine whether a watermark matches or not. As introduced in [1], the watermark extraction is done by projecting DL model weights using the unique watermark from miner and it can be considered as a binary classification problem with one layer. Therefore, we will find one array of confidence level. In our experiment, we picked 0.9999 as the threshold. Only if the confidence level of all watermark elements is higher than 0.9999, the miner will be detected as honest miner and model will be confirmed as matched. Only a DL model submitted by honest miner could be considered as winner candidacy.

3) Ownership protection

Note that the embedded watermark is generated from the block which is publicly known, whereas the robustness of the watermark in the original method [1] relies on the confidentiality of the watermark. Therefore, it is possible to remove or overwrite the embedded watermark in our scenario. For example, attackers may attempt to remove the watermark by subtracting the same extra term in the loss function instead of adding it and perform the training. By doing so, it is possible for the attackers to remove the watermark, but it takes many times more epochs of training to remove it as we show in the experiments (Section V). It will only need less than 5 iterations of training to embed the watermark in the model, but attacker miner will need more than 30 iterations of training to remove the honest miner's watermark. Therefore, training on top of others' models will not help a malicious miner to win the block reward. In other words, by spending the same amount of computation, the attackers could achieve better models with higher accuracy if they train their own models from the scratch without needing to remove existing watermarks.

In the third phase of each block, all full nodes will validate submissions which start from the best performance model. A full node will need to check whether the miner's watermark is matching the embedded watermark in the submitted DL model. In our experiment, only if each elements of the confidence level is higher than 0.9999, the unique miner's watermark is confirmed as matching DL model. If it matches, the model will be considered as genuine submission. If miner's watermark cannot be detected from the submitted model, the miner will be considered as malicious miner and this submission will be pruned.

D. Handling multiple tasks

Unlike in [32] where at most one task can be accepted by the blockchain, our novel blockchain is capable of accepting multiple tasks and handle them with the aforementioned ranking mechanism (Section IV-A). We achieve this by augmenting the full nodes' mempool to keep all the unselected tasks (Fig 6). Namely, multiple tasks submitted by publishers will reside in the mempool until they are selected and performed by the miners.

We allow task publishers to submit their tasks to full nodes only during Phase 2. To submit a task, the publisher will need to broadcast the followings to full nodes: publisher ID, task reward value, a link for downloading training dataset as well as the model (i.e., its hyperparameter). At the same time, the publisher will write and launch the smart contract that will send the task reward to the winning miner later when the task is performed and corresponding model is announced in the blockchain. Once the publisher submits a task, it will go to the full nodes' mempool and become an unfinished task. The unfinished tasks will stay in the mempool until the miners select and perform them.

With multiple tasks, it becomes important to let miners agree on the same task to be performed. Otherwise, it is hard to choose the winner by choosing the highest-accuracy model, since comparison of accuracy among different tasks is meaningless. Furthermore, as we will describe in Section IV-F, attackers may attempt to double spend by creating forks, and it is necessary to provide a task selection for the miners to agree on one task for each block.

Our blockchain defines that all miners must choose the task with the highest reward from the unfinished tasks in full nodes' mempool (ties are broken in a pre-defined manner). Due to the assumption that full nodes' views on the mempool are consistent, all full nodes have the same set of tasks in their mempool, and it is the blockchain policy to choose the task with the highest reward. Therefore all the miners must select and perform the same task for a specific block.

E. Handling large models and training data

In order to reduce the network traffic, a task publisher will only need to submit the model link and dataset link instead of submitting the model and the dataset directly during the task submission process. Also, to save the block storage, the link will be stored in blocks instead of actual models or datasets.

Miners still need to retrieve training data from the publisher, which may lead to network delay of tens of seconds or even more. To reduce this time loss, we let task publishers release the training dataset earlier in Phase 3 of the previous block's mining. After Phase 2 for block i, the task to be performed for block $i + 1$ has been determined already, therefore the miners can start to download. Note that miners are not able to continue training in Phase 3. Otherwise, the model will be different from the one committed in Phase 1. One issue of such training data release is that the miners with high network bandwidth are advantaged because they can start mining earlier than others. To avoid this and make mining fair, we let task publishers encrypt the training data with any efficient symmetric-key encryption (*e.g.,* AES [58]) and release the encrypted training data instead. Then, the publisher releases the key at the end of Phase 3. By doing so, the network delay caused by a key is negligible (*e.g.,* ≤ 256 bits for AES), and the miners who have finished downloading the encrypted training data can decrypt it and perform the training task immediately. The decryption causes extra delay as well, but the decryption itself can be considered as the work that miners need to prove. Note that, with the Assumption 1 in Section III, the task publisher will not release the key to any specific miners in advance.

Symmetric-key encryption such as AES does not expand data, but it is possible that encrypted training data cannot be fully downloaded within Phase 3 because of the large volume. Motivated miners will monitor the tasks being submitted to full nodes and start fetching the training data even before Phase 3, but our mechanism may have to limit the training data to an acceptable size.

F. Handling forks

Instead of considering the longest chain as the correct one, we let full nodes in our blockchain consider the chain who has the most highest-accuracy models as the correct one. The intuition behind this form of fork resolution is similar to that in existing cryptocurrencies based on PoW mechanisms. Namely, generating a correct block with a small-enough hash value is challenging in PoW-based cryptocurrencies, and a chain will be considered correct if it has the most correct blocks with small-enough hash values (*i.e.,* being the longest chain). In our blockchain, generating a valid block with the highest-accuracy model is challenging, therefore we treat the chain with the most highest-accuracy models as the correct one.

G. Validating past blocks

Newly-joined full nodes need to verify the entire blockchain. When checking block i, full nodes will need to check the unselected tasks record from block $i - 1$ and the selected task for block i to see whether the task selected in block i has the smallest ranking score. Here, the ranking score is given by Equation 4. The full nodes will have to verify whether the model accuracy is the same as the one claimed by the winner miner. Then, the full node will verify the digital signature we introduced in the block header to verify the integrity of data. Finally, existing validation (*e.g.,* correctness of hash calculation, transaction validity) will be performed.

This work has some limitations at the current version. We assume full nodes have consistent view as well as synchronized time clock. Achieving a design with the same robustness against various attacks without relying on these assumptions is our immediate future work. Besides, we store all unconfirmed tasks in the block, however the block size is limited to several megabytes. This limits the total number of tasks that can be handled by our blockchain. Breaking this limit is another future work.

H. Properties of our blockchain

Synchronized tasks: The augmented mempool stores all unselected tasks, and these mempools will be stored in every block. Miners will have to select the task with the highest reward from this list (that is available in the previous block), therefore all miners are able to agree on the task to perform. Therefore, full nodes are guaranteed to deal with the same task during the block validation. We highlight that this synchronization is achieved without relying on third-party entities, and therefore it does not harm the decentralization of blockchain.

Redefining confirmation: Because of the way full nodes choose the next block in our blockchain, whether blocks can be reversed does not depend too much on the number of confirmations (*i.e.,* the number of blocks after them on the blockchain). Rather, the accuracy of the models on the blockchain determines it. Namely, if the block contains a model with a high accuracy, it is challenging to generate another block with another model with a higher accuracy. Then, reversing the previous blocks ahead of the block with a high-accuracy model requires the amount of work needed for training a higher-accuracy model. Therefore, the blocks become hardly reversible after there being multiple high-accuracy models along the chain. Then, we may define the confirmations of a block as the number of high-accuracy models appearing after it rather than the number of blocks after it.

Hardness of double spending: Full nodes will accept the blocks in Phase 3 if and only if their headers are received in Phase 1. Therefore, as long as full nodes are honest, even if adversaries delay the submission of their blocks in order to afford more time in training, they are not allowed to submit

blocks with the *better* models (who were trained with more time) because the block headers did not appear in Phase 1.

Even if the majority of the full nodes collude with miners, double spending without 51% computing resources is still a low-probability event. During the training process, the optimization algorithms seek local optima with certain randomness because no known algorithms can strategically find the global optimum. Therefore, if only the highest-accuracy models are accepted, it is challenging to further improve the accuracy beyond it, as shown in Fig. 10,12. If adversaries wish to double spend in our blockchain by controlling majority of the full nodes, they must present another chain with more highest-accuracy models. Because the accuracy of the model depends on the hyperparameters and initial weights, choice of which is random, we conjecture that it is extremely difficult to generate another chain with more highest-accuracy models unless the adversary possesses more than 51% of the computing resources for the image segmentation training.

Dataset and model provision: Training dataset may have large volumes, however it is necessary for performing the published tasks. Therefore, we assume the task publishers will host training dataset for the miners' access.

Besides, full nodes who need to verify the whole chain (*e.g.,* newly-joined full nodes) need to access the historical models provided by the winning miners. We also let task publishers store the image segmentation models they collected from the miners, and provide the models to full nodes for their verification. There may be model privacy concerns, however addressing privacy concerns is an orthogonal problem, and we do not address that in this paper.

In case the storage of models (100KB-10GB per model) becomes a burden to the task publisher, we can save the storage by freeing up some earlier models with lower accuracy because the tamper-proofness is guaranteed by high-accuracy models only. Accordingly, we can also let full nodes verify the high-accuracy models only. By doing so, the blocks with high-accuracy models will still prevent the double spending, and the publishers need to store one model (the ultimate one that has the highest accuracy) per task only.

Network delay: Unlike the existing work [32], blocks submitted to full nodes do not include the trained models any more. Instead, the block contains the links providing access to the models. Therefore blocks do not need to be very large. Our blockchain does require some extra attributes in the block header as well as various information of tasks in the block. However, the storage burden of those extra data in the block is negligible.

However, miners' access to training data does involve non-negligible network delay which owing to the characteristics of the tasks performed by the miners. If tasks do not need to take large data as input, miners will experience less extra network delay.

Honesty of task publishers: Task publishers are assumed to be honest in this paper, however this assumption can be relaxed if we adopt smart contract capable of calling external APIs. For example, if we have a function `API_query(string URL,string link)` that sends the link of a model `link` to a web-based API `URL` and returns its accuracy against the test

```solidity
pragma solidity 0.4.0;
contract TaskContract {
    uint256 private reward;
    uint256 private accuracy;
    string apiForTesting;
    function TaskContract() public{
        taskReward = 1 ether;
        requiredAccuracy = 9500; // 95%
        apiForTesting = xx.yy.com/test
    }
    function testAndPay(string linkOfModel) public{
        require(API_query(apiForTesting, linkOfModel
            ) >= accuracy);
        msg.sender.transfer(taskReward);
    }
}
```

Fig. 8. A toy example of smart contract that guarantees task reward payment.

TABLE I
OVERHEAD BENCHMARK BASED ON 1000 TIMES TESTING

Model	Digital signature (s)		Network (s)	
	AVG.	STD.	AVG.	STD.
U-net (270MB)	1.25	0.051	3.77	0.322
FCN (212MB)	0.98	0.042	3.01	0.291
MNIST (2MB)	0.001	0.000	0.37	0.012
ARS (3MB)	0.002	0.000	0.41	0.001

dataset behind the API `URL`, we can let task publishers submit and deploy a smart contract transaction that looks like Fig. 8. It sends the task reward (1 ether) to the message sender if s/he provides a link of well-trained model that yields a high-enough accuracy ($\geq 95\%$) after the API call to the publisher's API for testing (*e.g.,* xx.yy.com/test). Then, we can let task publishers announce their tasks by deploying smart contract transactions at the blockchain instead of announcing them to full nodes. By doing so, task publishers are unable to reject the task reward payment. Oraclize [59] can be used to implement such external API call in Ethereum-based smart contract transactions, which supports access to any API on the Internet. However, further study needs to be done to understand the security as well as burden to the full nodes, the miners, and even the task publishers.

V. EXPERIMENT

A. Experiment setup

The experiments were conducted in small scale local network on the machines with Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, 32Gb RAM, GTX 1080 Ti.

To exploit computation power of blookchain for the image segmentation tasks, we adopt two widely used networks: fully conventional networks (FCN) and U-net for for 2D and 3D biomedical image segmentation respectively.

For FCN, we adopt the same network as that in the work [60], a 34-layer FCN, which applied bottleneck design and modified the decoding part to improve the accuracy. We use the MICCAI 2015 Gland Challenge dataset which has 85 training 2D images and 80 test 2D images. The loss function, learning rate, regulation parameters, and training epoch are also the same as that in the work [60]. For U-net, we adopt a general configurations: (a) four resolution steps, and each resolution step contains two layers of $3 \times 3 \times 3$ convolutions,
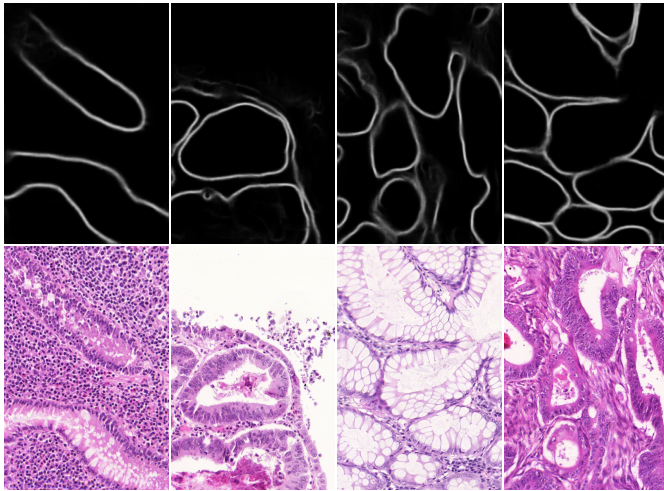
Fig. 9. FCN image segmentation result. (The upper/lower row demonstrates the segmentation results and original gland histology images, respectively.)
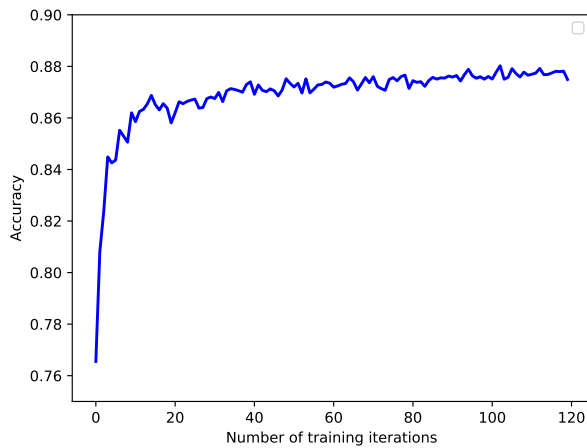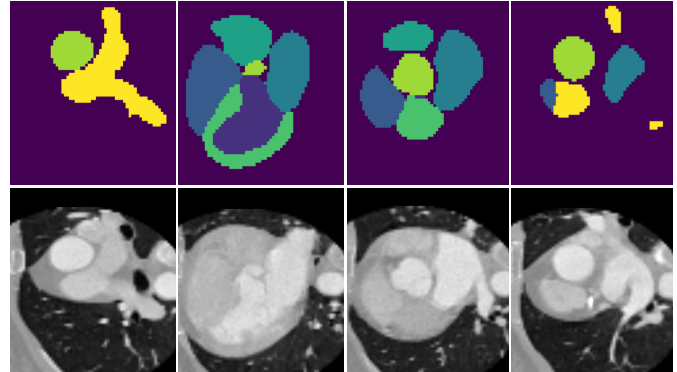

Fig. 11. U-net image segmentation result. (The upper/lower row demonstrates the segmentation results and original cardiac CT images, respectively.)
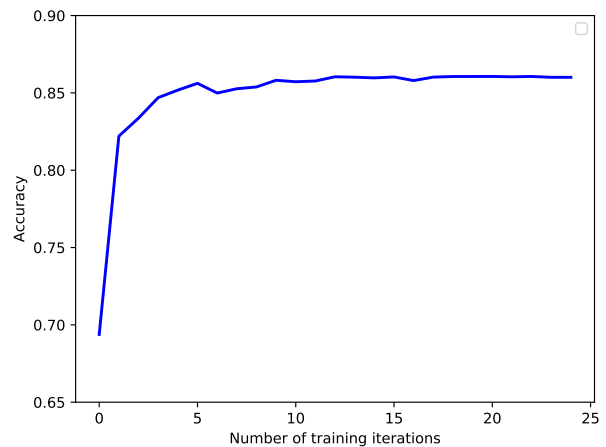

Fig. 10. FCN Image segmentation accuracy results.

Fig. 12. U-net Image segmentation accuracy results.

rectified linear unit (ReLu), and $2 \times 2 \times 2$ max pooling/up-sampling; (b) the number of filters in higher resolution step doubles that in its lower resolution step, and the initial (lowest) resolution step. We use the CT images in MMWHS 2017 heart segmentation challenge which has 20 training 3D images and 40 test 3D images. The loss function, learning rate, regulation parameters, and training epoch are the same as that in the work [61]. For both the two networks, we use Dice metric for evaluation.

B. Benchmark tests

Instead of the brute-force algorithm, the miner nodes performed image segmentation tasks as described in Section V-A. Fig. 9,11 shows the segmentation results with FCN method and U-net method, respectively. The accuracy evaluation results of FCN and U-net are demonstrated in Fig. 10,12. It can be seen that additional training based on a well performed model can hardly improve the performance of the model, thus it will be prevented from double spending as the discussion in Section IV-H.

Table I shows the extra overhead of digital signature and network. The digital signature was achieved by SHA-256

algorithm and the extra network overhead was evaluated by transmitting the winner model through a local network in accuracy validation step. Both overheads are much smaller than the image segmentation training time. Therefore, our mechanism utilized most of power on useful tasks and it potentially could be a contribution to both computer vision and blockchain society. Since we assumed the dataset is public accessible, the data loading time is not evaluated in the experiment. Extra storage overhead incurred by augmented mempool and novel task ledger is negligible which was discussed in Section IV-H.

C. Ownership protection evaluation

Full nodes will only accept a model if the watermark of the submitted model matches the watermark of the co-responding miner. Therefore, because of the embedded watermark in the DL model, attacker will need to pay additional penalty in term of removing the embedded watermark after the attacker miner steals the winner DL models from honest miners.

In Fig 14, it shows the watermark confidence level for honest miner and attacker miner. The honest miner achieved very high watermark confidence level in less than 5 iterations. But the attacker miner will have to remove the original watermark first and it will cost around 30 iterations in our case.
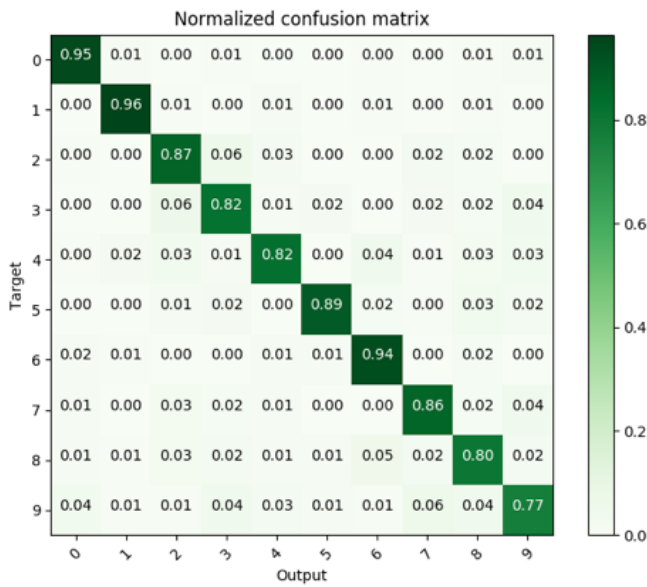
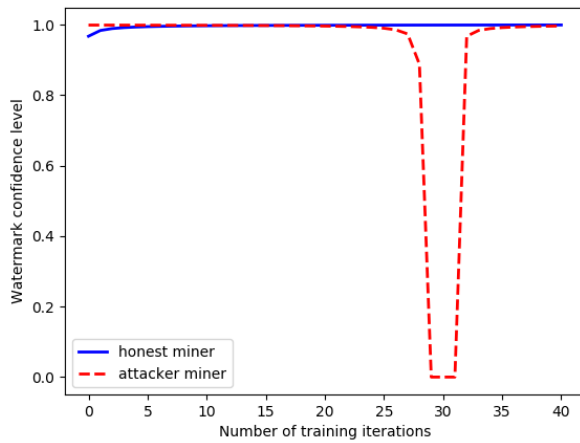Fig. 13. Confusion matrix for MNIST image classifier.

Fig. 14. Honest miner training versus attack miner training.

VI. CONCLUSION

In this paper, we presented a blockchain design that lets miners to perform biomedical image segmentation model training instead of hash calculation for block mining. Our blockchain design addresses the limitations of existing PoUW consensus mechanisms. The useful work involved in our design is practical because various disease diagnosis required customized models trained on specific dataset. Our blockchain is able to handle multiple tasks submitted by different task publishers, and it also provides a solution to handle DNN models as well as training datasets with large size. We performed quantitative experiments with real-world data to show that the extra overhead introduced by our design is acceptable.

REFERENCES

[1] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 269–277. 1, 2, 6, 7

[2] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008. 1

[3] Digiconomist, "Bitcoin energy consumption index @ONLINE," https://digiconomist.net/bitcoin-energy-consumption, March 2019, (accessed: 03.06.2019). 1

[4] A. De Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801–805, 2018. 1

[5] C. Mora, R. L. Rollins, K. Taladay, M. B. Kantar, M. K. Chock, M. Shimada, and E. C. Franklin, "Bitcoin emissions alone could push global warming above 2 c," *Nature Climate Change*, vol. 8, no. 11, p. 931, 2018. 1

[6] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: A comprehensive introduction.* Princeton University Press, 2016. 1

[7] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012. 1

[8] Burstcoin, "Burstcoin @ONLINE," https://www.burst-coin.org, March 2019, (accessed: 03.06.2019). 1

[9] D. Romano and G. Schmid, "Beyond bitcoin: A critical look at blockchain-based systems," *Cryptography*, vol. 1, no. 2, p. 15, 2017. 1

[10] T. Wang, J. Xiong, X. Xu, and Y. Shi, "Scnn: A general distribution based statistical convolutional neural networkwith application to video object detection," in *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI19)*, 2019. 1

[11] X. Xu, Q. Lu, L. Yang, S. Hu, D. Chen, Y. Hu, and Y. Shi, "Quantization of fully convolutional networks for accurate biomedical image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8300–8308. 1

[12] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, p. 216, 2018. 1

[13] X. Xu, Q. Lu, T. Wang, J. Liu, C. Zhuo, X. S. Hu, and Y. Shi, "Edge segmentation: Empowering mobile telemedicine with compressed cellular neural networks," in *Proceedings of the 36th International Conference on Computer-Aided Design*. IEEE Press, 2017, pp. 880–887. 1, 2

[14] X. Xu, Q. Lu, T. Wang, Y. Hu, C. Zhuo, J. Liu, and Y. Shi, "Efficient hardware implementation of cellular neural networks with incremental quantization and early exit," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 4, p. 48, 2018. 1

[15] Z. Liu, X. Xu, T. Liu, Q. Liu, Y. Wang, Y. Shi, W. Wen, M. Huang, H. Yuan, and J. Zhuang, "Machine vision guided 3d medical image compression for efficient transmission and accurate segmentation in the clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8300–8308. 1

[16] Y. Gao, Y.-F. Li, S. Chandra, L. Khan, and B. Thuraisingham, "Towards self-adaptive metric learning on the fly," in *Proceedings of the 2019 World Wide Web Conference (WWW '19), San Francisco, CA, USA, May 13–17, 2019*, 2019. [Online]. Available: https://doi.org/10.1145/3308558.3313503 1

[17] X. Xu, F. Lin, W. Xu, X. Yao, Y. Shi, D. Zeng, and Y. Hu, "Mda: A reconfigurable memristor-based distance accelerator for time series mining on data centers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018. 1

[18] X. Xu, F. Lin, A. Wang, X. Yao, Q. Lu, W. Xu, Y. Shi, and Y. Hu, "Accelerating dynamic time warping with memristor-based customized fabrics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 4, pp. 729–741, 2018. 1

[19] X. Xu, T. Wang, Q. Lu, and Y. Shi, "Resource constrained cellular neural networks for real-time obstacle detection using fpgas," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2018, pp. 437–440. 1

[20] X. Xu, D. Zeng, W. Xu, Y. Shi, and Y. Hu, "An efficient memristor-based distance accelerator for time series data mining on data centers," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2017, pp. 1–6. 1

[21] X. Xu, Q. Lu, T. Wang, Y. Hu, C. Zhuo, J. Liu, and Y. Shi, "Efficient hardware implementation of cellular neural networks with incremental quantization and early exit," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 4, p. 48, 2018. 1

[22] Y. Gong, B. Li, C. Poellabauer, and Y. Shi, "Real-time adversarial attacks," *arXiv preprint arXiv:1905.13399*, 2019. 1
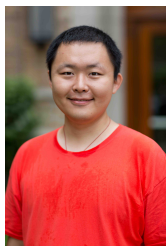
[23] Z. Liu, S. Luo, X. Xu, Y. Shi, and C. Zhuo, "A multi-level-optimization framework for fpga-based cellular neural network implementation," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 4, p. 47, 2018. 1

[24] B. Shen, C. W. Forstall, A. D. R. Rocha, and W. J. Scheirer, "Practical text phylogeny for real-world settings," *IEEE Access*, vol. 6, pp. 41 002–41 012, 2018. 1

[25] Y.-F. Li, Y. Gao, G. Ayoade, H. Tao, L. Khan, and B. Thuraisingham, "Multistream classification for cyber threat data with heterogeneous feature space," in *Proceedings of the 2019 World Wide Web Conference (WWW '19), San Francisco, CA, USA, May 13–17, 2019*, 2019. [Online]. Available: https://doi.org/10.1145/3308558.3313572 1

[26] Y. Ding, J. Liu, J. Xiong, and Y. Shi, "On the universal approximability and complexity bounds of quantized relu neural networks," *arXiv preprint arXiv:1802.03646*, 2018. 1

[27] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," *July 7th*, 2013. 1, 4

[28] A. Shoker, "Brief announcement: Sustainable blockchains through proof of exercise," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 2018, pp. 269–271. 1, 4

[29] H. Turesson, A. Roatis, H. Kim, and M. Laskowski, "Deep learning models as proof-of-useful work: A smarter, utilitarian scheme for achieving consensus on a blockchain," 2018. 1, 3

[30] A. Baldominos and Y. Saez, "Coin. ai: A proof-of-useful-work scheme for blockchain-based distributed deep learning," *arXiv preprint arXiv:1903.09800*, 2019. 1, 3

[31] F. Bravo-Marquez, S. Reeves, and M. Ugarte, "Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions," in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. IEEE, 2019, pp. 119–124. 1, 3

[32] C. Chenli, B. Li, Y. Shi, and T. Jung, "Energy-recycling blockchain with proof-of-deep-learning," in *IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, 2019. 1, 3, 7, 9

[33] S. Grieggs, B. Shen, P. Li, C. Short, J. Ma, M. McKenny, M. Wauke, B. Price, and W. Scheirer, "Measuring human perception to improve handwritten document transcription," *arXiv preprint arXiv:1904.03734*, 2019. 2

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 2

[35] P. Ballester and R. M. Araujo, "On the performance of googlenet and alexnet applied to sketches," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[37] H. Chen, X. Qi, e.-Z. Cheng, P.-A. Heng *et al.*, "Deep contextual networks for neuronal structure segmentation." in *AAAI*, 2016, pp. 1167–1173. 2

[38] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 424–432. 2

[39] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571. 2

[40] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, "Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images," *NeuroImage*, 2017. 2

[41] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3d deeply supervised network for automatic liver segmentation from ct volumes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 149–157. 2

[42] T. Wang, J. Xiong, X. Xu, M. Jiang, H. Yuan, M. Huang, J. Zhuang, and Y. Shi, "Msu-net: Multiscale statistical u-net for real-time 3d cardiac mri video segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 614–622. 2

[43] X. Xu, T. Wang, Y. Shi, H. Yuan, Q. Jia, M. Huang, and J. Zhuang, "Whole heart and great vessel segmentation in congenital heart disease using deep neural networks and graph matching," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 477–485. 2

[44] Z. Liu, X. Xu, T. Liu, Q. Liu, Y. Wang, Y. Shi, W. Wen, M. Huang, H. Yuan, and J. Zhuang, "Machine vision guided 3d medical image compression for efficient transmission and accurate segmentation in the clouds," *arXiv preprint arXiv:1904.08487*, 2019. 2

[45] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 2

[46] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv preprint arXiv:1703.05390*, 2017. 3

[47] A. Poelstra *et al.*, "Distributed consensus from proof of stake is impossible," *URL: https://download. wpsoftware. net/bitcoin/old-pos. pdf*, 2014. 3

[48] T. Ogawa, H. Kima, and N. Miyaho, "Proposal of proof-of-lucky-id (pol) to solve the problems of pow and pos," in *Blockchain*. IEEE, 2018. 3

[49] Y. L. Sanket Kanjalkar, Joseph Kuo and A. Miller, "I cant believe its not stake! resource exhaustion attacks on pos," in *Financial Cryptography*, 2019. 3

[50] A. Back *et al.*, "Hashcash-a denial of service counter-measure," 2002. 3

[51] B. Li, C. Chenli, X. Xu, T. Jung, and Y. Shi, "Exploiting computation power of blockchain for biomedical image segmentation," *arXiv preprint arXiv:1904.07349*, 2019. 3

[52] R. Pass and E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model," in *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 3

[53] K. Chatterjee, A. K. Goharshady, and A. Pourdamghani, "Hybrid mining: exploiting blockchain's computational power for distributed problem solving," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. ACM, 2019, pp. 374–381. 4

[54] A. F. Loe and E. A. Quaglia, "Conquering generals: an np-hard proof of useful work," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*. ACM, 2018, pp. 54–59. 4

[55] B. Li, J. Wu, and Y. Shi, "Privacy-aware cost-effective scheduling considering non-schedulable appliances in smart home," in *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*. IEEE, 2019, pp. 1–8. 4

[56] N. Shibata, "Proof-of-search: Combining blockchain consensus formation with solving optimization problems," *IEEE Access*, 2019. 4

[57] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," National Institute of Standards and Technology, Tech. Rep., 2018. 6

[58] P. Mahajan and A. Sachdeva, "A study of encryption algorithms aes, des and rsa for security," *Global Journal of Computer Science and Technology*, 2013. 8

[59] T. Bertani, "Understanding oracles," 2016. 9

[60] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2017, pp. 399–407. 9

[61] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018. 10

Boyang Li Boyang Li is currently pursuing a Ph.D. degree with the Department of Computer Science and Engineering at the University of Notre Dame, Notre Dame, IN, USA. He received his B.S. degree in Electrical Science and Technology from Xian University of Post and Telecommunication, Xian, Shaanxi, China. He received his Masters Degree in electronics and computer engineering from the University of Southampton, Southampton, Hampshire, UK. His research interests include machine learning, optimization, and novel blockchain mechanism. His paper has won the best student paper award (IEEE BIOMETRICS COUNCIL, 2019)
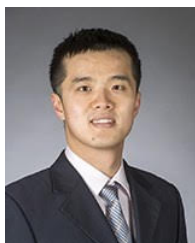
Changhao Chenli Changhao Chenli is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering at the University of Notre Dame, Notre Dame, IN, USA. He received his B.S. degree and M.S. degree in Renmin University of China of information security and software engineering, Beijing, China, in 2016 and 2018 respectively. His research interest includes blockchain technology, smart contract and data provenance. His paper has won a best student paper award (IEEE BIOMETRICS COUNCIL, 2019).
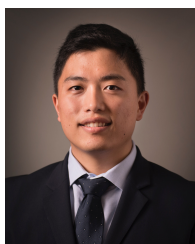
Xiaowei Xu(S'14-M'17) received the B.S. and Ph.D. degrees in electronic science and technology from Huazhong University of Science and Technology, Wuhan, China, in 2011 and 2016 respectively. He worked as a post-doc researcher at University of Notre Dame, IN, USA from 2016 to 2019. He is now a AI researcher at Guangdong Provincial People's Hospital. His research interests include deep learning, and medical image segmentation. He was a recipient of DAC system design contest special service recognition reward in 2018 and outstanding contribution in reviewing, Integration, the VLSI journal in 2017. He has served as TPC members in ICCD, ICCAD, ISVLSI and ISQED.

Yiyu Shi(S'06-M'10-SM'15) is currently an associate professor in the Department of Computer Science and Engineering at the University of Notre Dame, the site director of NSF I/UCRC Alternative and Sustainable Intelligent Computing, and the director of the Sustainable Computing Lab (SCL). He received his B.S. in Electronic Engineering from Tsinghua University, Beijing, China in 2005, the M.S and Ph.D. degree in Electrical Engineering from the University of California, Los Angeles in 2007 and 2009 respectively. His current research interests focus on hardware intelligence and biomedical applications. In recognition of his research, many of his papers have been nominated for the Best Paper Awards in top conferences. He was also the recipient of IBM Invention Achievement Award, Japan Society for the Promotion of Science (JSPS) Faculty Invitation Fellowship, Humboldt Research Fellowship, IEEE St. Louis Section Outstanding Educator Award, Academy of Science (St. Louis) Innovation Award, Missouri S&T Faculty Excellence Award, NSF CAREER Award, IEEE Region 5 Outstanding Individual Achievement Award, and the Air Force Summer Faculty Fellowship. He has served on the technical program committee of many international conferences including DAC, ICCAD, DATE, ISPD, ASPDAC and ICCD. He is on the executive committee of ACM SIGDA, a member of IEEE CEDA Publicity Committee, deputy editor-in-chief of IEEE VLSI CAS Newsletter, and an associate editor of IEEE TCAD, IEEE Access, ACM JETC, VLSI Integration, and IEEE TCCCPS Newsletter.

Taeho Jung is an assistant professor of Computer Science and Engineering at the University of Notre Dame. He received the Ph.D. from Illinois Institute of Technology in 2017 and B.E. from Tsinghua University in 2011. His research area includes data security, user privacy, and applied cryptography. His paper has won a best paper award (IEEE IPCCC 2014), and two of his papers were selected as best paper candidate (ACM MobiHoc 2014) and best paper award runner up (BigCom 2015).