Rendering Secure and Trustworthy Edge Intelligence in 5G-Enabled IIoT using Proof of Learning Consensus Protocol

Chao Qiu, *Member, IEEE,* Gagangeet Singh Aujla, *Senior Member, IEEE,* Jing Jiang, Wu Wen, Peiying Zhang

***Abstract*—Industrial Internet of Things (IIoT) and fifth generation (5G) network have fueled the development of Industry 4.0 by providing an unparalleled connectivity and intelligence to ensure timely (or real time) and optimal decision making. Under this umbrella, the edge intelligence is ready to propel another ripple in the industrial growth by ensuring the next generation of connectivity and performance. With the recent proliferation of blockchain, edge intelligence enters a new era, where each edge trains the local learning model, then interconnecting the whole learning models in a distributed blockchain manner, known as blockchain-assisted federated learning. However, it is quiet challenging task to provide secure edge intelligence in 5G-enabled IIoT environment alongside ensuring latency and throughput. In this paper, we propose a Proof-of-Learning (PoL) consensus protocol that considers the reputation opinion for edge blockchain to ensure secure and trustworthy edge intelligence in IIoT. This protocol fetches each edge's reputation opinion by executing a smart contract, and partly adopts the winner's learning model according to its reputation opinion. By quantitative performance analysis and simulation experiments, the proposed scheme demonstrates the superior performance in contrast to the traditional counterparts.**

***Index Terms*—Industrial Internet of Things, blockchain, edge intelligence, proof of learning, reputation opinion.**

I. Introduction

INTERNET OF THINGS, (IoT) has changed the way the devices (or objects) tank to each other and interact with surroundings. IoT encompasses smart devices (like, sensors, tags, and wearable) that gather data, analyze it, and finale a decision (or action). Fifth generation (5G) network has also quicken the way that the devices connect to each other. Recently, the reach of IoT has travelled a significant journey starting from the smart homes applications to myriad smart service ecosystem (including industrial as well as mission critical applications). 5G-enabled Industrial IoT (IIoT) is the most significant supplement of IoT in industrial and manufacturing sector. 5G-enabled IIoT moves across the normal internetworking of smart devices to realize how cyber-physical systems and production or operational processes can transform industrial systems by utilizing big data and analytics.

Meanwhile, integrating artificial intelligence (AI) with IIoT is beginning to receive a tremendous amount of interest, which is treated as one of the most important enabling technologies to achieve smart IIoT [1]. The development of AI has made breakthroughs in a wide spectrum of fields in IIoT, ranging from wireless sensing [2], resource allocation [3], to decision-making [4] and so on.

There is no doubt that the industrial systems need a timely decision making or action so adopting conventional data processing approaches may not fit this revolution. For example, transmitting the data generated by industrial devices to remote cloud servers for processing or analysis may involve high turnaround delay. Due to this reason, the middleware technology, clike edge computing, is very popular to process the workload or analyze the data generated from IoT devices.

Although AI has been regarded as a promising solution to solve the problems in IIoT (supported by edge computing), there are many challenges that hinder its widespread adoption. For example, the training data in IIoT is highly scattered, while AI training approaches are relatively centralized. It is not economical to bring the large volumes of scattered data to a center, as well as low performance, costs and privacy. These challenges have given rise to a new research area in IIoT, i.e., edge intelligence [5]. It pulls the training capacity from the data center to the industrial edges. Such space proximity among data, demands, and intelligence provider, benefits amount of high quality-of-service (QoS) supplement [6].

Nowadays, blockchain has come into existence as a promising technique for widespread applications. It is accessible for every entity, while not controlled by anyone, promising a great benefit to enhance the deployment of edge intelligence [7]. Essentially, edge node trains its own learning model in

This work is partially supported by National Science Foundation of China (Youth) under Grant 62002260, China Postdoctoral Science Foundation under Grant 2020M670654, Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant GML-KF-22-03, Open Project Foundation of Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications under Grant ICNS201901, Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-006, and Shandong Provincial Natural Science Foundation, China, under Grant ZR2020MF006. (*Corresponding authors: Wu Wen and Peiying Zhang*).

Chao Qiu is with the College of Intelligence and Computing, Tianjin University, Tianjin 300354, China, and Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518132, China. Email: chao.qiu@tju.edu.cn.

Gagangeet Singh Aujla is with the School of Computing Science, Durham University, Durham DH1 3LE, United Kingdom. E-mail: gagangeet.s.aujla@durham.ac.uk.

Jing Jiang is with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710061, China. Email: jiangjing@xupt.edu.cn.

Wu Wen is with School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China. Email: wenwu@gzhu.edu.cn.

Peiying Zhang is with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China. E-mail: zhangpeiying@upc.edu.cn.

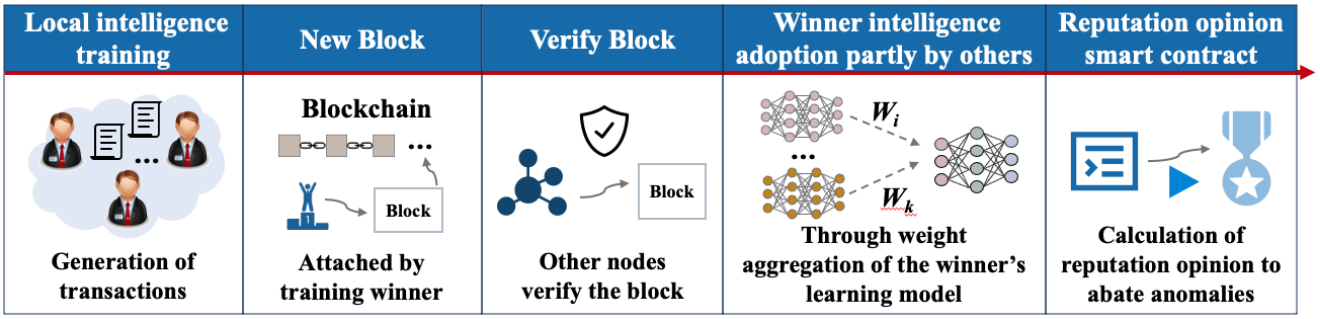


Fig. 1. Research approach

a local manner, then interacting the whole training models in a decentralized blockchain way, finally sharing the better learning model among edges, so as to construct the constantly innovatory edge intelligence, known as blockchain-assisted federated learning [8]. Driving by this trend, there are a lot of works focusing on the Proof of Useful Work consensus protocol to fully release the solution of blockchain assisted edge intelligence, via substituting the hashing operation in Proof of Work (PoW) as the intelligence training [9], such as Proof of Learning (PoL) [10], Proof of Training Quality (PoQ) [11], and Proof of Deep Learning (PoDL) [12]. However, the particularity of IIoT overturns these huge benefits of blockchain-assisted edge intelligence, thus leading to the following research challenges, including security consideration, resource constraints, and performance requirements:

- Security requirements: the security issues in the industrial scene, such as openness, heterogeneity, and non-confidence, have adverse effects on the whole performance of blockchain-assisted edge intelligence. For example, some malicious edges could join in the system and share false training models to impede edge intelligence.
- Resource constraints: as a resource-hungry system, IIoT is associated with problems of poor resources, consequently resulting in lacking enough resources to transmit all the local learning models, also including these false training models from malicious edges.
- Performance requirements: IIoT is also a time latency-sensitive system, which gives rise to the high requirements on the delay and throughput performance. These challenges have opened up demands to construct a security-latency-throughput improved blockchain-assisted edge intelligence system.

A. Research Contributions

To overcome the above discussed research challenges, the research approach adopted in this paper is shown in Fig. 1. Based on this research approach, in this paper, to move towards trustworthy edge intelligence, we propose a PoL consensus protocol with reputation opinion for edge blockchain in IIoT. The main contributions of this article are summarized as follows:

- We propose a reputation opinion based PoL consensus protocol, with high trustworthy supplement.
- We design the smart contract to obtain reputation opinion, so as to abate malicious or accidental anomalies and reliance on trusted intermediaries.
- The proposed trustworthy edge intelligence adopts the winner's intelligence through a weight aggregation of the winner's learning model according to its reputation opinion, instead of completely using the winner's model and discarding all its own local model.
- In addition, we provide the quantitative performance analysis of the proposed approach based on several metrics like, security, latency, and throughput.

B. Organization

The rest of this article is organized as follows. Section III gives an overview of the PoL consensus protocol with reputation opinion for edge blockchain in IIoT. In Section IV, the smart contract to calculate reputation opinion are discussed. Section V presents the quantitative performance analysis, followed by experimental results in Section VI. Finally, conclusions and future works are shown in Section VII.

II. Related Work

A. Security Requirements

Security requirement has become one of the major concerns to uplift the possible way with technological advancement. In addition, the security and the reliability of edge intelligence play a huge role in the effectiveness of IIoT. Just think how would the IIoT perform if the training results and the inference results are malicious. Thus, the exciting new challenges have spawned numerous studies on the security consideration about the edge intelligence-enabled IIoT.

In the light of secure data storage, utilization, search, and deletion in edge intelligence IIoT, the authors in [13] probe into a novel security architecture. The architecture integrates the acquisition, processing, and transition of sensing data in IIoT. Also involved into the secure challenges of edge intelligence IIoT, the authors in [5] discuss the security requirements in the sense of confidentiality and integrity. In this context, compressive sensing is considered to be especially appropriate to solve the above challenges.

From another perspective, edge intelligence has been widely adopted to improve the secure performance of IIoT. For

example, the authors in [14] consider the edge intelligence helped security authentication in IIoT. Meanwhile, with the preponderance of information-centric networking and cyber-physical system, the proposed scheme achieves the balanced working load of IIoT edge devices, under the securing scenario. Taking consideration of the malicious learning servers, the authors in [15] develop a novel fuzzy consensus protocol, integrating delegated proof of state (DPoS) and practical Byzantine fault tolerance (PBFT). The proposed consensus protocol provides the secure aggregation capability in edge intelligence. However, they ignore the quantitative analysis of security performance.

From the above summarization, it can be seen that the security research of edge intelligence IIoT is urgent and intense. There are several gaps in the secure edge intelligence IIoT and the appropriate security quantitative analysis, which explains the purpose and motivation of our article.

B. Learning-based Consensus Protocols

Working-based consensus protocols are taken by the traditional crypto-currency. There is no doubt that these potential costs are not cost-effective, due to the fact that they could not be applied by any others, except for maintaining the blockchain.

Therefore, a series of multiple roles consensus protocols are designed by many researchers. In general, the multiple roles could be classified in two types, namely reaching consensus and achieving AI tasks.

For example, **PoQ** uses federated learning in the permissioned blockchain [11]. Two types of nodes are included, i.e., committee nodes and committee leader. The committee leader collects the transactions, and the committee nodes verifies them. The approval is then distributed to the committee leader. Finally, the transactions are attached to the blockchain with the leader's signature. **PoDL** builds a energy-recycling consensus protocol, where model requester, miners, and full nodes exist [12]. Miners complete deep learning training tasks from mode requester, while full nodes validate the submitted training models based on test datasets. Meanwhile, **Proof of useful work** also changes the mining process as the AI model training [16]. The difference is that the hashed value including the digest including transactions, nonce and the hash of the previous block. And the hashed value determine the hyper-parameters of the training model.

From the above researches, there is a research trending that when taking a consensus in the blockchain, AI tasks are achieved simultaneously. However, the current researches take the seldom consideration about the reputation of participators, as well as the quantitative performance of the consensus protocol.

III. Blockchain Assisted Edge Intelligence Approach in IIoT

In this section, we present the blockchain-assisted edge intelligence approach, i.e., the improved PoL consensus protocol, as well as the adversary model under the consideration.

A. PoL Consensus Protocol

The work in [17] has proposed a blockchain-enabled edge intelligence aspect, which also contributes to relieving the centralized training and resource-limitation problems in IIoT. Assuming that there are N_{MH} edge learning nodes in IIoT, where N_M nodes are malicious to hinder edge intelligence, while N_H nodes are honest to promote edge intelligence. Fig. 2 shows a reputation opinion based PoL consensus protocol in the blockchain assisted edge intelligence, including the following five steps:

*1) **Local intelligence training and the generation of transactions**:* Deep reinforcement learning approach is adopted in each local learning node [18]. Specifically, there are two identical deep networks, namely target networks and evaluated networks, while target networks are kept frozen for a fixed period of time. The evaluated networks are crafted in each episode to minimize loss function $\ell_\Phi(\omega)$ under the local training dataset Φ [18]:

$$\ell_\Phi(\omega) = E[(r + \gamma \max_{a'} Q(s', a', \omega^-) - Q(s, a, \omega))^2], \quad (1)$$

where r is the immediate reward and $\gamma \in (0, 1)$ is a discount factor to balance the immediate reward and the previous ones. $Q(s', a', \omega^-)$ is the expected reward in state s', taking action a', under the weighs and biases set ω^- of neurons in deep networks, and the same in $Q(s, a, \omega)$. The smaller loss function means the more similarity between the evaluated networks and the target ones, further the more accurate evaluation of real Q function and the better learning intelligence of deep networks.

*2) **A new block is attached by the training winner**:* Instead of solving bootless hashing puzzles like in PoW, the 'miners' in PoL compete to obtain the better deep networks. Thus, the winner in PoL is the one with the minimum local loss function. The local loss function in node n can be denoted by [11]:

$$\ell_{\Phi_n}(\omega_I) = \frac{1}{I} \sum_{i=1}^{I} \ell_{\Phi_n}(\omega_i), \quad (2)$$

where I is the total number of local training episodes in node n, $\ell_{\Phi_n}(\omega_i)$ is its i-th training loss function, denoted by eq. (1), and Φ_n is the local training dataset in node n.

The winner then attaches a new un-verified block in the current chain, whose format and the concrete format of each part are presented Fig. 2. It is worth mentioning that *proof* in the block header is the minimum local loss function among transactions in the block.

The block is disseminated over the block system by two types of block propagation protocols, namely the legacy block propagation protocol and the compact block propagation protocol [19]. We mainly consider the former one, i.e., the legacy block propagation protocol, in this article.

*3) **Other nodes verify the newly attached block**:* After receiving the new attached block, other nodes conduct two types of validation, including cryptography, and training results. In the cryptography validation, message authentication codes (MACs) and digital signatures will be verified. Meanwhile, the validity of training results will be verified via re-running the training process, using transaction data. More details are presented in our previous work [17].

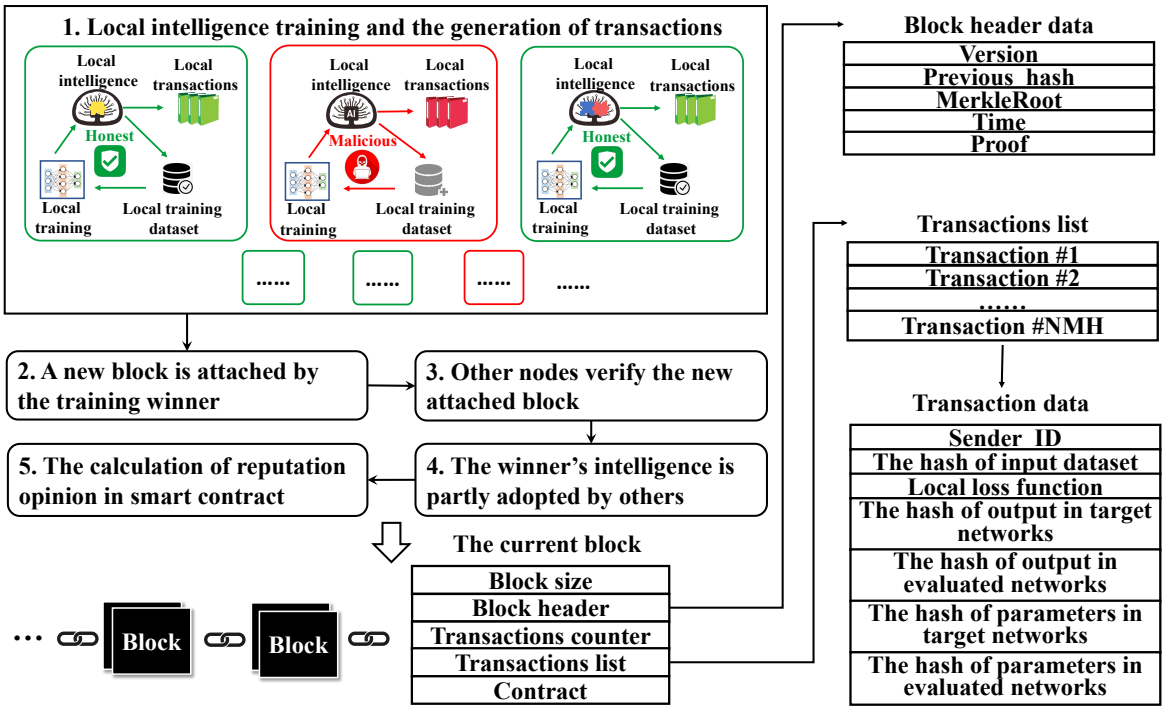


Fig. 2. A reputation opinion based PoL consensus protocol in the blockchain assisted edge intelligence.

*4) **The winner's intelligence is partly adopted by others**:* After verifying the validity of the newly attached block, other node, such as node n, partly adopts the winner's intelligence through a weighted aggregation of the winner's learning model, according to its reputation opinion:

$$\omega_n^{(t_{k+1})} = \omega_n^{(t_k)} + R_{s\to w}^{(t_k)} \omega_w^{(t_k)}, \tag{3}$$

where there are K time slots, denoted by $\{t_0, t_1, ..., t_k, ...t_{K-1}\}$. $\omega_n^{(t_{k+1})}$ and $\omega_n^{(t_k)}$ are the node n's local learning models at time slot t_{k+1} and t_k, respectively. $R_{s\to w}^{(t_k)}$ is the reputation opinion of the winner, calculated by *smart contract* at time slot t_k. Section IV will present how to get the reputation opinion. $\omega_w^{(t_k)}$ is the winner's learning model. Different from our previous work in [17] which discards all the own local model, fully adopting the winner's learning model, the improved solution in this article is with the merits of learning resource conservation and security.

*5) **The calculation of reputation opinion in smart contract**:* After adopting the learning model from the winner, other node, such as node n, needs to score the winner as $score_{n\to w}^{(t_k)}$, in case of malicious nodes sharing the mischievous learning model to hinder edge intelligence:

$$score_{n\to w}^{(t_k)} = \begin{cases} 1, & \text{positive effect} \\ 0, & \text{negative effect} \end{cases}. \tag{4}$$

Smart contract would then be triggered, including the following three parts of functions: charging the fee of adopting the winner's learning model, the calculation of the winner's reputation opinion, the calculation of other nodes' reputation opinions. More details are presented in Section IV.

B. Adversary Model

There are N_{MH} edge nodes in IIoT joining in the PoL consensus protocol, where N_M nodes are malicious and N_H nodes are honest. The malicious nodes are defined as the ones scoring against the original validity of the winner's learning model, i.e., if a winner's learning model is effective, the malicious node will score it as *negative effect* and vice-versa. Besides, the malicious nodes also submit invalid transactions which include mischievous learning models. If happening to be the winner, it will result in the performance inefficiency of the whole edge intelligence.

We set the trustworthiness of node n as $\mathcal{C}_n$, where $\mathcal{C}_n = 1$ when it is malicious and $\mathcal{C}_n = 0$ when it is honest, and $N_M = \sum_{n=1}^{N_{MH}} \mathcal{C}_n$. Due to the fact that there are only two possible trustworthiness outcomes for a discrete random number [20], we define event $\mathbf{A}$ as $\mathcal{C}_n = 1$, and random variable $\mathbf{X}$ as the number of times that event $\mathbf{A}$ happen. Thus, random variable $\mathbf{X}$ follows the Binomial distribution with the parameters of N_{MH} and p_m, i.e., $\mathbf{X} \backsim \mathbf{B}(N_{MH}, p_m)$, where p_m is the probability that $\mathcal{C}_n = 1$.

IV. Designing the Smart Contract

After presenting the PoL consensus protocol with reputation opinion and the adversary model, we will design the smart contract in this section. The overall target of designing the smart

contract is to meet common contractual conditions, including charging fees of using the winner's learning model, and calculating the reputation opinions, so as to abate malicious or accidental anomalies and reliance on trusted intermediaries. As mentioned, there are three parts of functions in the smart contract, as shown in Fig. 3.

Smart contract charges the fees from other nodes, such as node n, according to the score from node n to the winner, i.e., $score_{n\to w}^{(t_k)}$. If $score_{n\to w}^{(t_k)}=1$, the winner charges the fee from node n, otherwise, no charge.

We then present that how to calculate the reputation opinions of all kinds of nodes, including the winner and other nodes.

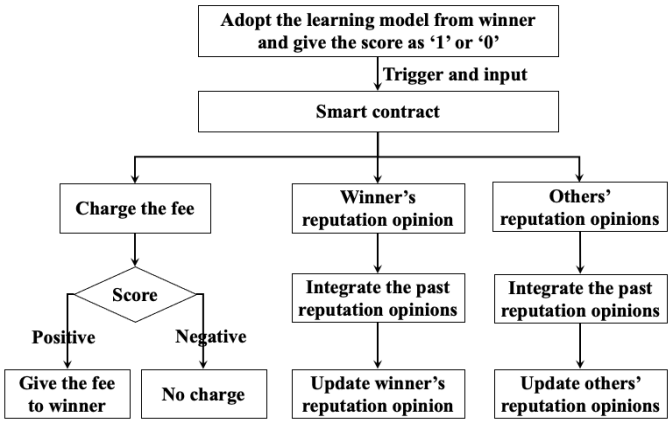


Fig. 3. The operation process in the smart contract.

A. The Calculation of the Winner's Reputation Opinion

We use a subjective logic model to calculate the reputation opinion based on historical interactions, which is widely applied in probabilistic reasoning to estimate the reputation or trustworthiness level [21]. In this article, also-ran nodes partly adopt the winner's learning model according to its reputation opinion, as show in eq. (3), i.e., the learning model from more trustworthy winner will be adopted more, and accordingly get more fees in return. Moreover, the malicious nodes who submit the mischievous learning models will have the lower reputation opinion, thus having less effects on the whole edge intelligence, gradually fading them out the edge intelligence system, thus conducting the trustworthy edge intelligence.

A reputation opinion of winner w at t_k time slot is determined by smart contract via collecting the scores from also-ran nodes to the winner. It is denoted by a opinion vector $v_{s\to w}^{(t_k)}=\{bel_{s\to w}^{(t_k)},dis_{s\to w}^{(t_k)},uncer_{s\to w}^{(t_k)}\}$, where $bel_{s\to w}^{(t_k)},dis_{s\to w}^{(t_k)},uncer_{s\to w}^{(t_k)}$ mean the belief degree, the distrust degree, and the uncertainty degree from smart contract to winner w at t_k time slot, respectively, and $bel_{s\to w}^{(t_k)},dis_{s\to w}^{(t_k)},uncer_{s\to w}^{(t_k)}\in[0,1]$, $bel_{s\to w}^{(t_k)}+dis_{s\to w}^{(t_k)}+uncer_{s\to w}^{(t_k)}=1$. Based on the work in [22], we have:

$$\begin{cases} bel_{s\to w}^{(t_k)}=(1-uncer_{s\to w}^{(t_k)})\dfrac{N_{pos}^{(t_k)}}{N_{pos}^{(t_k)}+N_{neg}^{(t_k)}} \\ dis_{s\to w}^{(t_k)}=(1-uncer_{s\to w}^{(t_k)})\dfrac{N_{neg}^{(t_k)}}{N_{pos}^{(t_k)}+N_{neg}^{(t_k)}} \\ uncer_{s\to w}^{(t_k)}=1-succe_{s\to w}^{(t_k)} \end{cases}, \tag{5}$$

where $N_{pos}^{(t_k)}$, $N_{neg}^{(t_k)}$ are the number of positive interactions, and negative interactions during time slot t_k, respectively. $N_{pos}^{(t_k)}=\sum_{n=1,n\neq w}^{N_{MH}} score_{n\to w}^{(t_k)}$, and $N_{neg}^{(t_k)}=N_{MN}-N_{pos}^{(t_k)}$. $succe_{s\to w}^{(t_k)}$ is the success probability of blockchain communications from the winner to others, and is also the indicator of the trustworthiness of blockchain links from the winner to others. The bigger communication success probability means the better trustworthiness of blockchain links. The reputation opinion of the winner at time slot t_k can be expected as:

$$rep_{s\to w}^{(t_k)}=bel_{s\to w}^{(t_k)}+\alpha_{s\to w}uncer_{s\to w}^{(t_k)}, \tag{6}$$

where $\alpha_{s\to w}\in[0,1]$ means the weight factor considering the influence of the uncertainty degree from the winner to others.

Finally, integrating the past reputation opinions of winner w with the current one, we have the final reputation opinion and update it in the smart contract:

$$\begin{aligned} R_{s\to w}^{(t_k)}=&rep_{s\to w}^{(t_k)}+\gamma_{s\to w}rep_{s\to w}^{(t_{k-1})} \\ &+\gamma_{s\to w}^2rep_{s\to w}^{(t_{k-2})}+...+\gamma_{s\to w}^krep_{s\to w}^{(t_0)}, \end{aligned} \tag{7}$$

where $\gamma_{s\to w}\in(0,1)$ is a freshness fading factor between winner and others, i.e., the more recent reputation opinion has more influence than the past ones.

B. The Calculation of Other Nodes' Reputation Opinions

In order to prevent other also-ran nodes from malignant scoring, on the one hand, so as to hinder edge intelligence, on the other hand, to avoid paying for using the winner's learning model, after calculating the winner's reputation opinion, smart contract also measures other also-ran nodes' reputation opinions. The motivation behind this mechanism, instead of only calculating the winner's reputation opinion, is to enhance the security of PoL.

According to the scores collected from all also-ran nodes, and the principle that the minority is subordinate to the majority, smart contract also scores the winner as $score_{s\to w}^{(t_k)}$:

$$score_{s\to w}^{(t_k)}=\begin{cases}1, & \sum_{n=1,n\neq w}^{N_{MH}} score_{n\to w}^{(t_k)}\geq\frac{1}{2}N_{MH} \\ 0, & \sum_{n=1,n\neq w}^{N_{MH}} score_{n\to w}^{(t_k)}<\frac{1}{2}N_{MH}\end{cases}. \tag{8}$$

According to $score_{s\to w}^{(t_k)}$, smart contract then scores also-ran node n:

$$score_{s\to n}^{(t_k)}=\begin{cases}1, & score_{s\to w}^{(t_k)}=score_{n\to w}^{(t_k)} \\ 0, & score_{s\to w}^{(t_k)}\neq score_{n\to w}^{(t_k)}\end{cases}. \tag{9}$$

Then smart contract calculates node n's reputation opinion, also according to a opinion vector $v_{s\to n}^{(t_k)}=\{bel_{s\to n}^{(t_k)},dis_{s\to n}^{(t_k)},uncer_{s\to n}^{(t_k)}\}$, where $bel_{s\to n}^{(t_k)},dis_{s\to n}^{(t_k)},uncer_{s\to n}^{(t_k)}$ means belief degree, distrust

degree, and uncertainty degree from smart contract to also-ran node n at t_k time slot, respectively.

If $score_{s\to n}^{(t_k)}=1$:

$$\begin{cases} bel_{s\to n}^{(t_k)} = (1-uncer_{s\to n}^{(t_k)})\dfrac{N_{pos}^{(t_{k-1})}+1}{N_{pos}^{(t_{k-1})}+N_{neg}^{(t_{k-1})}+1} \\ dis_{s\to n}^{(t_k)} = (1-uncer_{s\to n}^{(t_k)})\dfrac{N_{neg}^{(t_{k-1})}}{N_{pos}^{(t_{k-1})}+N_{neg}^{(t_{k-1})}+1} \\ uncer_{s\to n}^{(t_k)} = 1-succe_{s\to n}^{(t_k)} \end{cases}, \quad (10)$$

if $score_{s\to n}^{(t_k)}=0$:

$$\begin{cases} bel_{s\to n}^{(t_k)} = (1-uncer_{s\to n}^{(t_k)})\dfrac{N_{pos}^{(t_{k-1})}}{N_{pos}^{(t_{k-1})}+N_{neg}^{(t_{k-1})}+1} \\ dis_{s\to n}^{(t_k)} = (1-uncer_{s\to n}^{(t_k)})\dfrac{N_{neg}^{(t_{k-1})}+1}{N_{pos}^{(t_{k-1})}+N_{neg}^{(t_{k-1})}+1} \\ uncer_{s\to n}^{(t_k)} = 1-succe_{s\to n}^{(t_k)} \end{cases}, \quad (11)$$

where $N_{pos}^{(t_{k-1})}$, $N_{neg}^{(t_{k-1})}$ are the number of positive interactions, and negative interactions during time slot t_{k-1}, respectively. $succe_{s\to n}^{(t_k)}$ is the success probability of blockchain communications among other nodes, and is also the indicator of the trustworthiness of blockchain links among other nodes. The bigger communication success probability means the better trustworthiness of blockchain links. The reputation opinion of node n at time slot t_k can be denoted as:

$$rep_{s\to n}^{(t_k)} = bel_{s\to n}^{(t_k)} + \alpha_{s\to n} uncer_{s\to n}^{(t_k)}, \quad (12)$$

where $\alpha_{s\to n}\in[0,1]$ means the weight factor considering the influence of the uncertainty degree among other nodes.

We then have the final reputation opinion and update it in the smart contract:

$$\begin{aligned} R_{s\to n}^{(t_k)} &= rep_{s\to n}^{(t_k)} + \gamma_{s\to n} rep_{s\to n}^{(t_{k-1})} \\ &\quad + \gamma_{s\to n}^2 rep_{s\to n}^{(t_{k-2})} + ... + \gamma_{s\to n}^k rep_{s\to n}^{(t_0)}, \end{aligned} \quad (13)$$

where $\gamma_{s\to n}\in(0,1)$ is a freshness fading factor among other nodes, i.e., the more recent reputation opinion has more influence than the past ones.

V. Quantitative Performance Analysis

In this section, we present the quantitative performance analysis of PoL consensus protocol with reputation opinion, including security, latency and throughput.

A. Security

We set the probability that the original validation of a block is inverted after PoL consensus protocol as the measure of security performance, denoted by p_{inv}. The bigger p_{inv} means the worse security performance. It includes two parts, i.e, the winner is honest while the verification result indicates it is malicious, and the winner is malicious while the verification result indicates it is honest. Furthermore, p_{inv} is also influenced by each node's reputation opinion, i.e., the smaller reputation opinion may result in the larger inverted probability. As shown in Section III-B, $\mathbf{X}\backsim\mathbf{B}(N_{MH},p_m)$. Thus, p_{inv} is expected as:

$$p_{inv} = (p_{h\to m}+p_{m\to h})\beta^{\sum_{n=1}^{N_{MH}} R_{s\to n}^{(t_k)}}, \quad (14)$$

where $0<\beta<1$ is the converse impact factor of reputation opinions, i.e., the bigger β means the smaller impact of reputation opinion, due to the fact that the bigger p_{inv} means the worse security performance. $p_{h\to m}$ denotes the probability that the winner is honest while the verification result indicates it is malicious. $p_{m\to h}$ means the probability that the winner is malicious while the verification result indicates it is honest. Therefore, $p_{h\to m}$ is denoted as:

$$p_{h\to m} = (1-p_m)\Pr(\mathbf{X}\geq\lceil\frac{1}{2}N_{MH}\rceil), \quad (15)$$

and $p_{m\to h}$ is denoted as:

$$p_{m\to h} = p_m\Pr(\mathbf{X}\geq\lceil\frac{1}{2}N_{MH}\rceil), \quad (16)$$

finally, p_{inv} can be denoted as:

$$p_{inv} = \Pr(\mathbf{X}\geq\lceil\frac{1}{2}N_{MH}\rceil)\beta^{\sum_{n=1}^{N_{MH}} R_{s\to n}^{t_k}}. \quad (17)$$

When $\lceil\frac{1}{2}N_{MH}\rceil$ is even:

$$\begin{aligned} p_{inv} = \beta^{\sum_{n=1}^{N_{MH}} R_{s\to n}^{t_k}}[1-(N_{MH}-k)\binom{N_{MH}}{k} \\ \sum_{i=0}^{k}(-1)^i C_k^i \frac{1}{N_{MH}-k+i} t^{N_{MH}-k+i}\Big|_0^{1-p_m}], \end{aligned} \quad (18)$$

when $\lceil\frac{1}{2}N_{MH}\rceil$ is odd:

$$\begin{aligned} p_{inv} = \beta^{\sum_{n=1}^{N_{MH}} R_{s\to n}^{t_k}}[1+(N_{MH}-k)\binom{N_{MH}}{k} \\ \sum_{i=0}^{k}(-1)^i C_k^i \frac{1}{N_{MH}-i} t^{N_{MH}-i}\Big|_0^{1-p_m}]. \end{aligned} \quad (19)$$

Proof: See Appendix A.1.

B. Latency and Throughput

According to the consensus process of PoL, time latency T_{PoL} includes local training delay T_{tra}, block dissemination delay T_{dis}, and verification delay T_{ver}:

$$T_{PoL} = T_{tra} + T_{dis} + T_{ver}, \quad (20)$$

where T_{tra} is a uniform constant for the whole edge intelligence system.

For block dissemination delay T_{dis}, according to the work in [19], we consider the legacy block propagation protocol in the edge intelligence system. Here, after verifying or generating a block, the node will send an *inventory* message to its neighbors. According to the winner's reputation opinion and whether the block has been received, the receiver will reply to a *getdata* message. When receiving the *getdata* message, the sender node will send the requested block to the receiver. The formats of a *inventory* message and a *getdata* message, containing the winner's reputation opinion, are shown in Table I and Table II. It is worth noting that there is a new

TABLE I
THE FORMAT OF A *inventory* MESSAGE.

Message header				
Start String	Command (inventory)	Reputation opinion $R_{s\to w}^{(t_k)}$	Size	Checksum
The counter of inventories				
Inventory entries				

TABLE II
THE FORMAT OF A *getdata* MESSAGE.

Message header			
Start String	Command (getdata)	Size	Checksum
The number of requested objects			
Requested objects			

string located in the message header, i.e., reputation opinion, compared with the traditional *inventory* message.

Thus, block dissemination delay T_{dis} can be expressed as:

$$\begin{aligned} T_{dis} =& TD_{inv} + PD_{inv} + \\ & R_{s\to w}^{(t_k)}(TD_{get} + PD_{get} + TD_{blo} + PD_{blo}), \end{aligned} \tag{21}$$

where TD_{inv} and PD_{inv} are transmission delay and propagation delay of a *inventory* message. TD_{get} and PD_{get} are transmission delay and propagation delay of a *getdata* message. TD_{blo} and PD_{blo} are transmission delay and propagation delay of a block. Due to the fact transmission delay is related to data size and transmission bandwidth B, and propagation delay is related to propagation distance and rate, $TD_{inv} = s_{inv}/B$, $TD_{get} = s_{get}/B$, $TD_{blo} = s_{blo}/B$, and $PD_{inv} = PD_{get} = PD_{blo}$, where s_{inv}, s_{get} and s_{blo} are data sizes of the *inventory* message, the *getdata* message, and the block. Since whether replying to the *inventory* message is influenced by the currrent winner's reputation opinion, thus the time delays of the *getdata* message and the block are related with the current $R_{s\to w}^{(t_k)}$.

For verification delay T_{ver}, it contains the delay of verifying the winner's training result VT_{win} and the delay of verifying MACs and digital signatures VT_{cry}. According to computation model for deep networks [23], VT_{win} can be denoted as:

$$VT_{win} = \frac{s_{win}c_{win}}{f_n}, \tag{22}$$

where s_{win} is the data size of the winner's training samples to be validated, c_{win} is the number of CPU cycles to validate one training sample, and f_n is the computation capability, i.e., CPU cycles frequency, of verifier n.

In addition, according to the computation mode for cryptographic messages [24], VT_{cry} can be expressed as:

$$VT_{cry} = \frac{(1+N_{MH})(\delta+\theta)}{f_n}, \tag{23}$$

where verifying one MAC and one digital signature need δ CPU cycles and θ CPU cycles, respectively. And there are N_{MH} MACs and N_{MH} digital signature from N_{MH} transactions, i.e., N_{MH} edge learning nodes, as well one MAC and one signature from the un-validated block.

Therefore, the throughput about the number of blocks generated per second can be expressed as:

$$\tau_{PoL} = \frac{1}{T_{tra} + T_{dis} + T_{ver}}. \tag{24}$$

VI. SIMULATION RESULTS AND DISCUSSIONS

In this section, we conduct an extensive experimental analysis to evaluate the performance of the proposed improved PoL with reputation opinion (RO).

In order to assess the performance of the improved PoL, we use the improved PoL edge intelligence approach and the traditional one [10] to solve a joint resource allocation problem in IIoT [17], denoted by *improved PoL with RO*, and *traditional PoL w.o. RO*, respectively in the following figures, with the aim to achieve better joint resource utility.

Fig. 4 presents the training curves tracking resource utility and training episodes, under *improved PoL with RO*, and *traditional PoL w.o. RO*. Meanwhile, Fig. 5 is its fitted result. In the figures, local training happens in the first 1000 episodes, after that the PoL consensus protocol is performed to share better local learning intelligence. Thus, there are mutationally jagged curves when the episode is 1000. However, the improved PoL partly adopts the winner's learning model according to the winner's reputation opinion, which results in the relatively lighter jitter, compared with the traditional one. Meanwhile, as shown in the fitted figure, i.e., Fig. 5, after the sharing learning model in episode 1000, the utility of the improved approach increases more quickly than the traditional one, which indicates that the improved approach also has better learning performance than the traditional one.

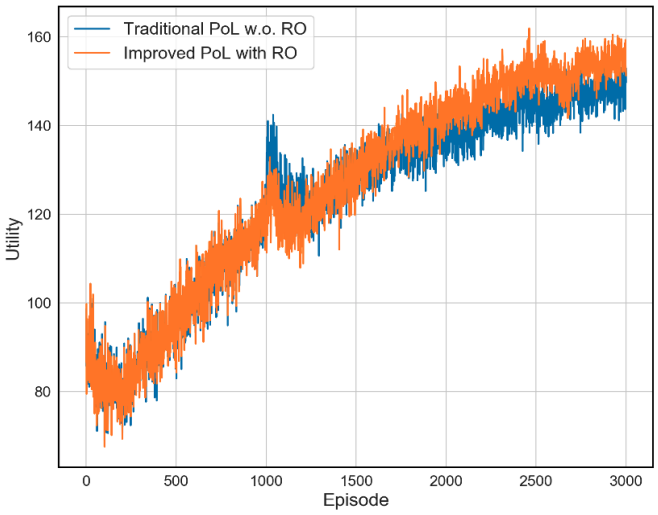


Fig. 4. Training curves tracking resource utility under traditional PoL w.o. RO and improved PoL with RO.

Fig. 6 shows the curves tracking security performance p_{inv} and the total number of edge learning nodes, under two approaches and the different p_m. p_{inv} indicates the probability that the original validation of a block is inverted after PoL consensus protocol, and the bigger one means the worse security performance. As shown in the figure, the security performance of the improved approach is always better than

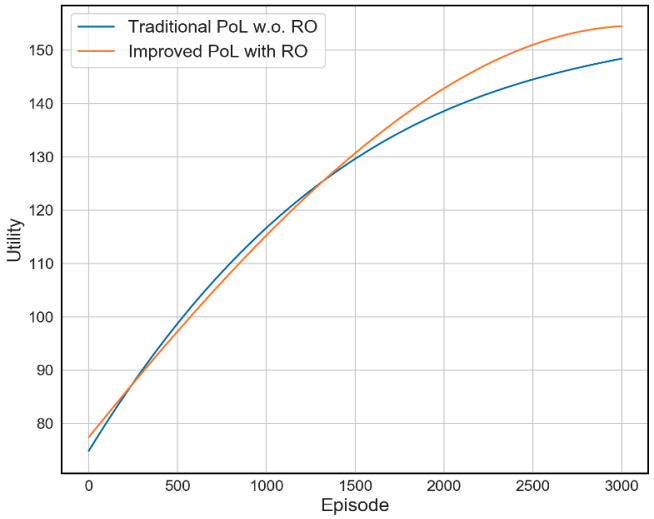


Fig. 5. The fitted training curves of Fig. 4 tracking resource utility under traditional PoL w.o. RO and improved PoL with RO.

the traditional one. The reason is that the also-ran nodes use the reputation opinion to partly adopt the winner's model, rather than totally adopting it, which leads to that although some malicious nodes are selected as the winners by accident, their false learning model also has the lesser effects on others. Meanwhile, when p_m increases, i.e, more nodes may be malicious nodes, the whole edge intelligence system becomes more untrustworthy, which results in the bigger p_{inv}.

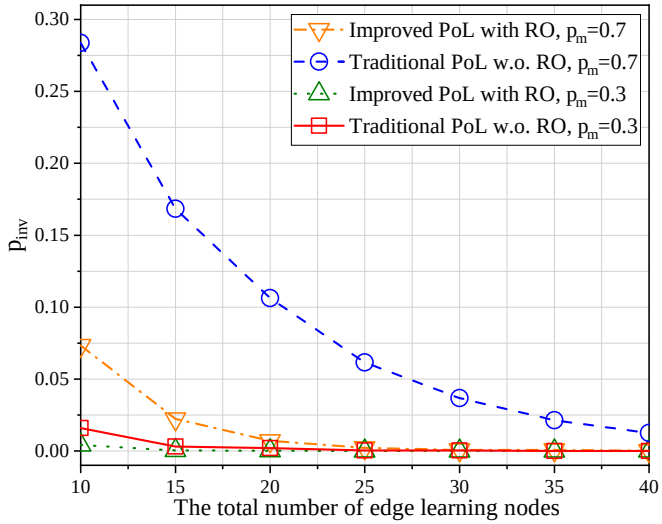


Fig. 6. The curves tracking security performance p_{inv} and the total number of edge learning nodes.

Fig. 7 gives the curves tracking security performance p_{inv} and the converse impact factor of reputation opinion β. As we can see, with the increase of the probability of malicious nodes, the p_{inv} rise, i.e., the whole edge intelligence system becomes more untrustworthy. Meanwhile, due to the factor that the bigger β means the smaller impact of reputation opinion, with the increase of converse impact factor of reputation opinion, the impact of reputation opinion will decrease, further reduce the untrustworthiness of the whole edge intelligence system.

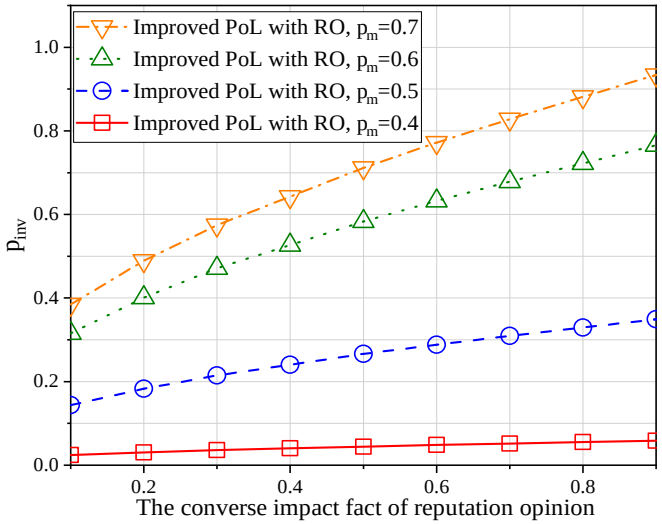


Fig. 7. The curves tracking security performance p_{inv} and the converse impact factor of reputation opinion β.

Fig. 8 and Fig. 9 show the curves tracking the performance of time latency and throughput, under two approaches, respectively. Due to the fact that the receiver of a block decides whether to receive it according to the winner's reputation opinion, rather than totally replying *inventory* message to receive the block, the improved PoL has the better performance in terms of time latency and throughput.

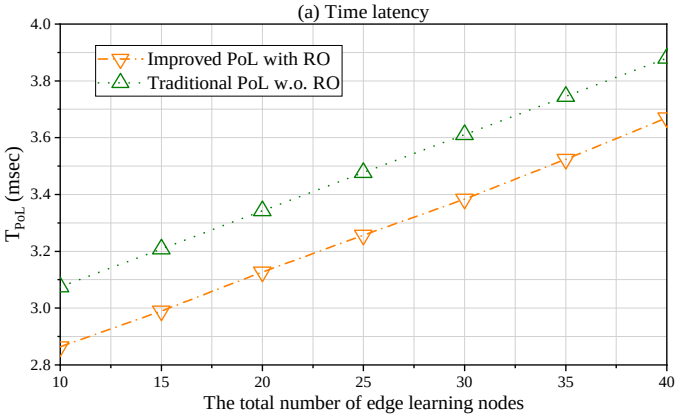


Fig. 8. The curves tracking the performance of time latency.

VII. Conclusion

In this paper, towards trustworthy edge intelligence, we propose a PoL consensus protocol with reputation opinion for edge blockchain in IIoT. By means of smart contract to get each edge's reputation opinion, the learning edges partly adopt the winner's learning model, rather than completely adoption. We also give the quantitative performance analysis of the proposed scheme, including security, latency, and throughput. Finally, simulation results show the effectiveness of our proposed scheme. Meanwhile, some future

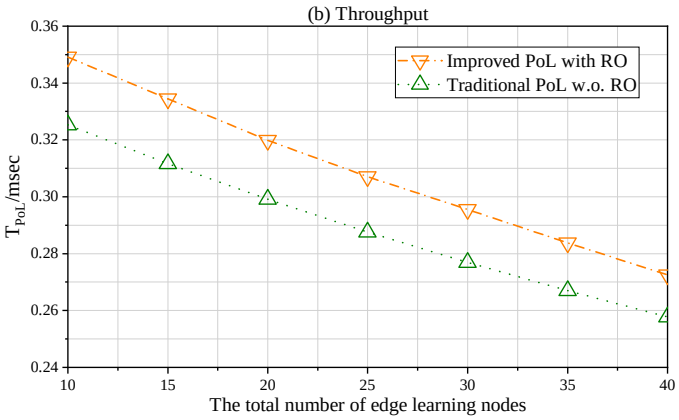


Fig. 9. The curves tracking the performance of throughput.

works are also necessary. For example, incentive mechanism and resource allocation are urgent problems. For one thing, incentive mechanism is the basic problem to attract edges to join in the edge intelligence, in the case of restricted edge capability. For another, flexible resource allocation is one of the characteristics of edge scenarios. How to make sure the excellent edge intelligence contributors obtain more edge resource is yet the question.

APPENDIX A

Appendix A.1:
$\mathbf{X} \backsim \mathbf{B}(N_{MH}, p_m)$, thus we have:

$$\begin{aligned}\Pr(\mathbf{X} \geq \lceil \frac{1}{2} N_{MH} \rceil) &= 1 - \Pr(\mathbf{X} < \lceil \frac{1}{2} N_{MH} \rceil) \\ &= 1 - F(\lceil \frac{1}{2} N_{MH} \rceil; N_{MH}, p_m),\end{aligned} \tag{25}$$

where, $F(\lceil \frac{1}{2} N_{MH} \rceil; N_{MH}, p_m)$ can be denoted by regularized incomplete Beta function. For convenience, let $\lceil \frac{1}{2} N_{MH} \rceil = k$:

$$\begin{aligned}F(k; N_{MH}, p_m) &= I_{1-p_m}(N_{MH} - k, k + 1) \\ &= (N_{MH} - k) \binom{N_{MH}}{k} \int_0^{1-p_m} t^{N_{MH}-k-1}(1-t)^k dt,\end{aligned} \tag{26}$$

when k is even:

$$\begin{aligned}&\int_0^{1-p_m} t^{N_{MH}-k-1}(1-t)^k dt \\ =&\int_0^{1-p_m} t^{N_{MH}-k-1}(C_k^0 + C_k^1(-t) + C_k^2(-t)^2 \\ &+ ... + C_k^k(-t)^k) dt \\ =&C_k^0 \frac{1}{N_{MH}-k} t^{N_{MH}-k}\Big|_0^{1-p_m} - \\ &C_k^1 \frac{1}{N_{MH}-k+1} t^{N_{MH}-k+1}\Big|_0^{1-p_m} + ... + \\ &C_k^k \frac{1}{N_{MH}} t^{N_{MN}}\Big|_0^{1-p_m} \\ =&\sum_{i=0}^{k} (-1)^i C_k^i \frac{1}{N_{MH}-k+i} t^{N_{MH}-k+i}\Big|_0^{1-p_m},\end{aligned} \tag{27}$$

when k is odd:

$$\begin{aligned}&\int_0^{1-p_m} t^{N_{MH}-k-1}(1-t)^k dt \\ =&-\int_0^{1-p_m} t^{N_{MH}-k-1}(t-1)^k dt \\ =&-\int_0^{1-p_m} t^{N_{MH}-k-1}(C_k^0 t^k + C_k^1 t^{k-1}(-1)^1 + ... \\ &+ C_k^{k-1} t(-1)^{k-1} + C_k^k(-1)^k) dt \\ =&-\int_0^{1-p_m} (C_k^0 t^{N_{MH}-1} dt + C_k^1 t^{N_{MH}-2}(-1)^1 dt + ... \\ &+ C_k^{k-1} t^{N_{MH}-k}(-1)^{k-1} dt + C_k^k t^{N_{MH}-k-1}(-1)^k dt) \\ =&-[C_k^0 \frac{1}{N_{MH}} t^{N_{MH}}\Big|_0^{1-p_m} + \\ &(-1)^1 C_k^1 \frac{1}{N_{MH}-1} t^{N_{MH}-1}\Big|_0^{1-p_m} + ... \\ &+ (-1)^{k-1} C_k^{k-1} \frac{1}{N_{MH}-k+1} t^{N_{MH}-k+1}\Big|_0^{1-p_m} + \\ &(-1)^k C_k^k \frac{1}{N_{MH}-k} t^{N_{MH}-k}\Big|_0^{1-p_m}] \\ =&-[\sum_{i=0}^{k} (-1)^i C_k^i \frac{1}{N_{MH}-i} t^{N_{MH}-i}\Big|_0^{1-p_m}].\end{aligned} \tag{28}$$

REFERENCES

[1] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in iot edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[2] R. M. A. Haseeb-Ur-Rehman, M. Liaqat, A. H. M. Aman, S. H. A. Hamid, R. L. Ali, J. Shuja, and M. K. Khan, "Sensor cloud frameworks: State-of-the-art, taxonomy, and research issues," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22 347–22 370, 2021.

[3] I. Alqerm and J. Pan, "Enhanced online Q-learning scheme for resource allocation with maximum utility and fairness in edge-IoT networks," *IEEE Trans. on Net. Sci. and Eng.*, vol. 7, no. 4, pp. 3074–3086, 2020.

[4] G. Rathee, S. Garg, G. Kaddoum, and B. J. Choi, "Decision-making model for securing iot devices in smart industries," *IEEE Trans. on Industrial Infor.*, vol. 17, no. 6, pp. 4270–4278, 2021.

[5] Y. Zhang, H. Huang, L.-X. Yang, Y. Xiang, and M. Li, "Serious challenges and potential solutions for the industrial internet of things with edge intelligence," *IEEE Net.*, vol. 33, no. 5, pp. 41–45, 2019.

[6] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. on Veh. Tech.*, vol. 68, no. 2, pp. 1079–1092, 2019.

[7] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for 5G beyond," *IEEE Net.*, 2020.

[8] F. Ayaz, Z. Sheng, D. Tian, and Y. L. Guan, "A blockchain based federated learning for message dissemination in vehicular networks," *IEEE Trans. on Veh. Tech.*, vol. 71, no. 2, pp. 1927–1940, 2022.

[9] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Comm. Surveys and Tutorials*, vol. 22, no. 2, pp. 1392–1431, 2020.

[10] F. Bravo-Marquez, S. Reeves, and M. Ugarte, "Proof-of-learning: A blockchain consensus mechanism based on machine learning competitions," in *Proc. Conf. on Decentralized App. and Infrastructures (DAPPCON)*, 2019, pp. 119–124.

[11] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. on Industrial Infor.*, vol. 16, no. 6, pp. 4177–4186, 2020.

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

[13] Y. Yu, R. Chen, H. Li, Y. Li, and A. Tian, "Toward data security in edge intelligent IIoT," *IEEE Net.*, vol. 33, no. 5, pp. 20–26, 2019.

[14] Y. Lu, D. Wang, M. S. Obaidat, and P. Vijayakumar, "Edge-assisted intelligent device authentication in cyber-physical systems," *IEEE Internet of Things Journal*, pp. 1–1, 2022.
[15] F. Yang, J. Tian, T. Feng, F. Xu, C. Qiu, and C. Zhao, "Blockchain-enabled parallel learning in industrial edge-cloud network: a fuzzy DPoSt-PBFT approach," in *Proc. Conf. on Globecom Workshops*, 2021, pp. 1–6.
[16] A. Baldominos and Y. Saez, "Coin.AI: A proof-of-useful-work scheme for blockchain-based distributed deep learning,," *Entropy*, vol. 21, no. 8, pp. 723–728, 2019.
[17] C. Qiu, X. Wang, H. Yao, J. Du, F. R. Yu, and S. Guo, "Networking integrated cloud–edge–end in iot: A blockchain-assisted collective q-learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 694–12 704, 2021.
[18] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial internet of things: A dueling deep Q-learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627–4639, 2019.
[19] Y. Shahsavari, K. Zhang, and C. Talhi, "A theoretical model for block propagation analysis in bitcoin network," *IEEE Trans. on Eng. Management*, pp. 1–18, 2020.
[20] S. C. Medin, J. Murray-Bruce, D. Castañón, and V. K. Goyal, "Beyond binomial and negative binomial: Adaptation in bernoulli parameter estimation," *IEEE Trans. on Comp. Imaging*, vol. 5, no. 4, pp. 570–584, 2019.
[21] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.
[22] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1389–1399, 2018.
[23] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. Conf. on Asia Pacific Wireless Commu. IEEE*, 2019, pp. 1–5.
[24] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3298–3304, 2018.

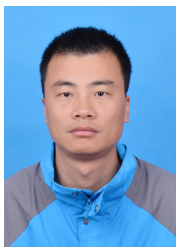
BIOGRAPHIES

Chao Qiu is currently a Lecturer with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, as well as Guangdong Laboratory of Artificial Intelligence and Digital Economy, ShenZhen. She received the B.S. degree from China Agricultural University in 2013 in Communication Engineering and the Ph.D. from Beijing University of Posts and Telecommunications in 2019 in Information and Communication Engineering. From September 2017 to September 2018, she visited Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. Her current research interests include edge computing, edge intelligence, and blockchain.

Gagangeet Singh Aujla is an assistant professor of computer science at Durham University, United Kingdom. Prior to this, he was a postdoctoral research associate with the School of Computing, Newcastle University, United Kingdom. He received his Ph.D. in computer science from Thapar University, India, in 2018. He received the 2018 IEEE TCSC Outstanding Ph.D. Dissertation Award and 2021 IEEE Systems Journal Best Paper Award, which recognized his leading expertise in the application of scalable and sustainable algorithms for cloud data centers, SDN, and smart grid. He is an Area Editor of Ad Hoc Networks (Elsevier).

Jing Jiang received the M.Sc. degree from Xi Dian University, in 2005, and the Ph.D. degree in information and communication engineering from North Western Polytechnic University, China, in 2009. She was the leader of 3GPP LTE MIMO project with ZTE Corporation, China, from 2006 to 2013. Currently, she is a Professor with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications. Her research interests include edge caching, blockchain technology and Artificial Intelligence based wireless communication.

WuWen received the Master of Science degree from the Huazhong University of Science and Technology, in 2009. He is currently an Associate Professor. He is engaged in teaching and scientific research at the School of Computer Science and Cyber Engineering, Guangzhou University. His main research interests include computer application, networks, and service computing.



Peiying Zhang is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He received his Ph.D. in the School of Information and Communication Engineering at University of Beijing University of Posts and Telecommunications in 2019. He has published multiple IEEE/ACM Trans./Journal/Magazine papers since 2016, such as IEEE TVT, IEEE TNSE, IEEE TNSM, IEEE TETC, IEEE Network, IEEE Access, IEEE IoT-J, ACM TALLIP, COMPUT COMMUN, IEEE COMMUN MAG, and etc. He served as the Technical Program Committee of ISCIT 2016, ISCIT 2017, ISCIT 2018, ISCIT 2019, Globecom 2019, COMNETSAT 2020, SoftIoT 2021, IWCMC-Satellite 2019, and IWCMC-Satellite 2020. His research interests include semantic computing, future internet architecture, network virtualization, and artificial intelligence for networking.