Integrating Edge Intelligence and Blockchain: What, Why, and How

Xiaofei Wang, Senior Member, IEEE, Xiaoxu Ren, Student Member, IEEE, Chao Qiu, Member, IEEE, Zehui Xiong, Member, IEEE, Haipeng Yao, Senior Member, IEEE, and Victor C. M. Leung, Life Fellow, IEEE

Abstract—Driven by an unprecedented boom in artificial intelligence (AI) and Internet of Things (IoT), edge intelligence (EI) pushes the frontier of AI from cloud to network edge, serving as a remarkable solution that unlocks the full potential of AI services. It is yet facing critical challenges in its decentralized management and security, limiting its capabilities to support services with numerous requirements. In this context, blockchain (BC) has been seen as a promising solution to tackle the above issues, and further support EI. Based on the number of citations or the relevance of emerging methods, this paper presents the results of a literature survey on the integration of EI and BC. Accordingly, we summarize the recent research efforts reported in the existing works on EI and BC. We then paint a comprehensive picture of the limitations of EI and why BC could benefit from EI. From there, we explore how BC benefits EI in terms of computing power management, data administration, and model optimization. In order to narrow the gap between immature BC and EI-amicable BC, we also probe into how to tailor BC to EI from four perspectives, including flexible consensus protocol, effective incentive, intellectuality smart contract, and scalability. Finally, some research challenges and future directions are presented. Different from existing surveys, our work focuses on the integration of EI and BC, develops some general models to help the reader build relevant optimization models in the integrated system, as well as provides detailed tutorials on implementation. We anticipate that this survey will motivate further discussions on the synergy of EI and BC, and offer some guidance in EI, BC, future networks, and other areas.

Index Terms—Edge intelligence (EI), blockchain (BC), edge computing, decentralization, distributed ledger technology (DLT).

I. INTRODUCTION

A LONG with the wave of informatization technology, a booming era of artificial intelligence (AI) has emerged. According to the prediction by Ericsson [1], Internet of Things (IoT) devices will generate 45% of the 40 zettabytes (ZB) Internet data in 2024, while there will be 5.3 billion total Internet users and 29.3 billion networked devices by 2023 [2]. Nevertheless, global devices transferring extremely vast data to cloud datacenters will demand high bandwidth and powerful computational resources [3], thus creating a bottleneck on the restricted network transmission capabilities, computing power of computing infrastructures, and strict delay requirements.

Edge intelligence (EI), as a complementary processing architecture by combining edge computing (EC) [4] and AI, pushes the AI frontier from the cloud to the network edge to open the path for low-latency and critical-computation [5]. Specifically, EI is a burgeoning paradigm integrating network, computing, storage and AI, while providing EI services and satisfying the critical requirements of the Internet era in agile connection, real-time business, data optimization, application intelligence, security and privacy protection, etc. Notably, the celebrated Gartner hype cycle has regarded EI as an emerging technology that will enter a stationary phase in the following 5 to 10 years [6]. In practice, a broad spectrum of fields are moving toward EI acceptance due to its added features, such as IoT [7], smart cities [8] and smart home [9].

However, EI technology may face multiple challenges in its decentralized management and security, further limiting its capabilities to support services with numerous requirements. Specifically, EI leverages cooperation among various
AI services across a multitude of end devices, edge nodes, and clouds. Such cooperations may lead to difficulties in heterogeneous computing-power management, data collaboration, model optimization, etc. Concretely, the following challenges are awaiting us to investigate and address, further accelerating the large-scale implementation of EI applications.

- How to manage the dispersed computing-power resources efficiently and securely while offering a capillary distribution of computing power to the resource-constrained computing platforms needs to be addressed.
- How to connect the geographically distributed edges for handling the intensive distributed data collaboratively has become a significant concern.
- How to perform the distributed training and inference securely based upon private data samples is necessary to consider deeply.

Compared with EI, blockchain (BC) is not a fresh but revolutionary technology, it is an essential concept and core architecture in Bitcoin [10]. BC is expected to play an outsize role due to its unique technology features, such as consensus protocol, cryptography algorithm, distributed ledger technology (DLT), etc. In addition, BC enables a shift in the computing paradigm from centralized control to decentralized control [11], while it records transactions between two parties in a permanent way without the need for third-party authentication. In light of this, BC brings many benefits in terms of reducing costs for trusted third parties, improving the efficiency of resource management, verifying data authenticity, protecting privacy, and ensuring security. Therefore, it can be regarded as a complementary technology to make up for the EI limitations. However, BC also faces many technical challenges when it comes to driving EI, such as storage load, transaction capacity, scalability, and fault tolerance, which prevent many BC systems from being implemented [12].

Therefore, the integration of EI and BC has received extensive attention due to the substantial current interest in EI and BC. This survey aims to investigate how state-of-the-art BC technology is driving EI to open up new horizons for providing ubiquitous intelligent services, as shown in Fig. 1. Accordingly, we explore how BC can benefit EI from computing power management, data administration, and machine learning (ML) model optimization. In order to narrow the gap between immature BC and EI-amicable BC, we also probe into how to tailor BC to EI from four perspectives, including flexible consensus protocol, effective incentive, intellectual-property smart contract, and scalability. This potentially breaks the bottlenecks in BC, further enabling scalable storage and computation on the BC-driven EI.

A. Related Books and Surveys

To our best knowledge, many studies on EC, EI, BC, and related issues have been conducted from a wide range of technical aspects. We list details of the existing surveys in Table I. Some surveys on general EC concepts, opportunities, and challenges are [13]–[15]. Other papers focus on surveying specific applications of EC. In [14], the authors analyze how EC improves the performance of IoT networks with the categorization of EC architectures. The work of [15] presents a hierarchical architecture on EC-enabled IoV. Benefiting from the breakthroughs of deep learning (DL), a set of intelligent applications, the marriage of EC and AI, namely EI, has rapidly gained widespread popularity. In [16], some research efforts on EI, including architectures, frameworks, and key technologies for the DL model are presented. The work of [17] gives a comprehensive survey on EI, and gives some guidance for future EI research. And there are several good survey
papers on BC from specific aspects, such as consensus protocols [18], [19] scalability [20], [21], security [22], and some industrial applications [23]–[25].

With the advantages of these technologies, some integration systems have appeared that incorporate different technologies. Currently, there have been many efforts that attempt to converge IoT, BC, AI, or EC to compensate for each other’s shortcomings. In [26], driven by the breakthroughs in 5G network and BC technology, the synergy of BC with 5G networks and beyond is discussed. Specifically, this study explores and analyzes some of the critical technologies and opportunities that BC empowers diversified 5G services, including cloud computing, EC, network slicing, etc. In [27], the authors propose the integration system of BC and cloud of things. This system aims to establish a decentralized management architecture, improve data privacy and system security, significantly reduce the complexity of system implementation, and thus serve more application scenarios. Moreover, the authors in [28] incorporate BC and EC to securely support massive network servers, data storage, and validity computation in proximity to the end.

With the breakthroughs of AI, some decentralized AI algorithms assisted by BC have attracted considerable interest [29]–[33]. Driven by federated learning (FL), the authors in [29] discuss existing FL architecture, enabling technologies with BC, and highlight their implications to future FL algorithms, whereas [30] performs a comprehensive review of how to utilize BC to facilitate AI applications. Further, Nguyen et al. survey the latest research efforts on the integration of BC and AI for fighting COVID-19 in various applications [31]. In [32], the convergence of BC and ML for communications and networking systems is investigated. This convergence can improve the performance of ML in data and model sharing, decentralized intelligence, etc., while enhancing the functions of BC in scalability, security, privacy, etc. While the authors in [33] follow with the focus on the applications of BC and ML, especially DL and deep reinforcement learning (DRL), in the Industrial Internet of Things (IIoT) and discuss possible security and privacy risks.

Nevertheless, most of the existing surveys suffer from the following limitations: 1) there are few overviews of the convergence of BC, AI, EC, or other technologies; 2) there is no study explicitly formulating the interaction between the above technologies as some general models, and 3) some important implementation tutorials are missing in most of the existing surveys. Unlike existing surveys, our work focuses on the integration of EI and BC, while some general models are developed to help readers build relevant optimization models in integrated systems and provide detailed tutorials for implementation.

In addition, we summarize the number of research publications versus year for the reader’s convenience from IEEE, which is shown in Fig. 2. From Fig. 2, we can find that there is a significant increase in the number of publications every year for BC, EC, and AI. Based on the facts discussed above, BC is expected to provide new opportunities for the combination of AI and EC (i.e., EI) in the foreseeable future. Thus, this survey provides a comprehensive account of EI integrating with BC from several aspects, ranging from overview, motivations, integrated frameworks, and implementation tutorials to research challenges and future directions.

B. Our Tutorial

Even though the emerging BC technological advances have accelerated the process of EI deployment, there are still some significant challenges ahead. Therefore, the main goal of this paper is to take a closer look at vulnerability issues within EI from the deployment perspective, while examining how state-of-the-art BC technology can help with EI deployment. More precisely, we integrate EI and BC, summarized as follows.

- First, we overview the fundamental principles of EI and BC. Specifically, we focus on the 3D feature of EI and the 3C feature of BC, while giving the motivation to support EI by taking advantage of BC’s complementary features.
- Second, we divide the EI deployment into three aspects concerning key elements of EI, i.e., computing power, data, and model, including computing-power management, data administration, and model optimization.
- Third, we tailor the BC to better support EI, including flexible consensus protocol, effective incentive mechanism, intellectuality smart contract, and scalable BC system tailoring.
- Fourth, we show some BC-driven EI applications and give tutorials for implementing BC in these applications.
- Finally, we present several key research challenges with possible solutions.

A taxonomy graph of this paper is presented as Fig. 3. Specifically, we first give the background on EI and BC in Section II. Then, the motivation for integrating EI and BC is explained in Section III. Next, we describe in detail the two integration techniques, i.e., EI-driven by BC (Section IV) and tailoring BC to EI (Section V). In Section VI, we investigate
Fig. 3. The taxonomy graph on the integration of EI and BC.

Some applications and tutorials of BC in EI. Section VII discusses research challenges and future directions. Lastly, the conclusion is drawn in Section VIII.

II. BACKGROUND ON EDGE INTELLIGENCE AND BLOCKCHAIN

In this section, we give an overview of EI from the following aspects: the Definition, Division and Deployment, denoted as 3D feature of EI. Then, we describe the background of BC from the following aspects: the Concept, Construction, and Categories, denoted 3C feature of BC.

A. 3D Feature of EI

With recent breakthroughs in DL, AI applications and services have flourished. Meanwhile, billions of IoT devices are connected to the Internet, generating massive amounts of data at the edges. Driven by AI and IoT, there is an urgent need to unleash the full potential of edge AI. EC is a promising solution to support compute-intensive AI applications on edge devices. The interactive integration of EC and AI, known as EI, empowers each edge node to compute and make decisions, enabling certain complex intelligent applications to be processed locally at the edge. EI is the next stage of the
development of EC and interweaves many concepts and technologies together in a complex manner. Next, we describe the 3D feature of EI in detail.

1) Definition of EI: There is still no standard definition of EI. In 2019, International Electrotechnical Commission defined EI as the capabilities of the data acquisition, storage, analysis, and aggregation with ML algorithms at the edges [34]. Zhang et al. describe the EI in terms of a four-element tuple <Accuracy, Latency, Energy, Memory footprint>. Instead of limiting EI to edge servers or devices, the work in [17] considers EI as a platform to complete DL training and inference offloaded from the cloud. Zhou et al. [5] believe EI should be a paradigm for leveraging available data and resources in the hierarchy of end devices, edge nodes, and cloud centers.

2) Division of EI: EI involves many concepts, technologies, and components intertwined in an enormously sophisticated manner. In this section, we summarize the division of EI into the following three categories based on the correlation between AI and edge environment.

- **Component-Based Division of EI:** Combined with the edge environment, the significant components of AI, i.e., data, model and computation, will turn into edge caching, edge training, edge inference, and edge offloading [35]. As shown in the top left of Fig. 4, based on the edge-cached datasets, edge training learns the optimal model parameters to provide a more precise prediction for edge inference. In addition, edge offloading aims to offload the tasks on devices with limited resources to the edge servers, enabling the computation services to run efficiently, thereby supporting other components.

- **Relationship-based division of EI:** As shown in the bottom left of Fig. 4, there is an interactive relationship between AI and EC. Deng et al. [36] distinguish EI into: AI for edge (solving optimization problems in EC with AI technologies) and AI on edge (how to run AI models on edges). Wang et al. [17] explain that EI is to push DL computation from the cloud to the edges as much as possible while classifying Edge DL into five enabling technologies (i.e., DL on/at/in/for Edge and Edge for DL).

3) Deployment of EI: It is well known that the three key elements of a deployed AI application are data, model, and computation [35]. Specifically, data provides the underlying support for the algorithm, model algorithms can greatly improve the efficiency of the AI model, and the improvement of computing power enables the realization of complex algorithms. Just as a triangle needs three sides to stabilize its shape, EI also needs these three elements to improve itself. Referring to this classification, we divide the deployment methods of EI into three categories, as shown in Fig. 5.

- **Computing-Power Resources Optimization:** It mainly includes software and hardware, as shown in the bottom at Fig. 5. On the one hand, the researches on software optimization focus on optimizing, scheduling, and managing the resources allocation based on AI models to reduce the redundancy latency and improve the system performance [37], [38].

On the other hand, the potential enabling hardware of EI is mainly aimed at accelerating the calculation of NNs. Currently, the most mainstream solution is to use GPUs [39] to parallelize a large number of mathematical operations in NNs. FPGA-based hardware breaks through the power and cost bottlenecks of GPUs at the edge [40], and they perform faster than state-of-the-art GPU implementations. In addition, ASIC-based hardware offers significant advantages in terms of power consumption, reliability, and confidentiality [41].
Data Management: It is to manage massive data at the edges, including data caching and offloading, as shown in the middle of Fig. 5. On the one hand, data caching at the edges can quantify and analyze the users’ requests. After that, the caching deployment can intelligently determine the suitable caching locations, data, AI models and methods [42], [43]. Then, the caching update process replaces the old data with the new in a timely manner [44], [45] to provide real-time, high-quality EI services to end users.

On the other hand, data offloading allows for offloading tasks to other edge devices through AI algorithms. Specifically, it can coordinate computing resources at different levels of the device-edge-cloud, flexibly schedule, decouple and allocate AI tasks to minimize latency and provide computational efficiency for EI applications [46].

Model Establishment: It aims to establish the lightweight model, including model training and inference, as shown in the top of Fig. 5. For the model training at the edges, numerous enabling techniques exist to improve training performance. Specifically, transfer learning (TL) [47] migrates the model suitable for big data to small data, realizing personalized migration. FL [48], [49] and deep neural network (DNN) splitting [50] leverage locally-trained model updates and partially-processed data to improve the security of edge AI models. Gossip training [51] shortens the training latency by exchanging and computing information in an arbitrarily connected network of nodes.

For the model inference at the edge, the model design is committed to reducing the size and latency of NNs for promoting the model effectiveness [52]–[55]. Model compression, such as pruning [56], data quantization [57] and knowledge distillation [58], minimize the model complexity to relieve the pressure of the end devices. Model early-exit [59], [60], model selection [61], model partition [62], [63] and input filtering [64], [65] realize inference acceleration, further speeding up the deployment on edge devices with the limited memory.

B. 3C Feature of BC

BC is not a fresh, stand-alone technology, rather it is an innovative application mode of existing technologies, including distributed data storage, point-to-point transmission, consensus mechanism, encryption algorithm, etc. BC has evolved over the last decade into one of today’s largest breakthrough technologies, with the potential to impact every industry from financial to manufacturing to educational institutions. For instance, some BCs attempt to give control of digital assets to end-users without the need for intermediate nodes. Other BCs are designed to maintain the logical centralization of some information while adopting a decentralized architecture [66]. In order to get a deeper understanding of BC advantages, we then briefly introduce the 3C feature of BC.

1) Concept of BC: In the narrowest sense, BC, as a chained data structure, combines data blocks sequentially in a chronological manner, while it can also be regarded as an encrypted distributed ledger that cannot be tampered with. In the broad sense, BC, as a novel kind of distributed infrastructure and computing paradigm, uses the block-type data structures to validate and store data, while utilizing the distributed consensus algorithm to generate and update the data. Meanwhile, BC leverages cryptography way to ensure the security of data transmission and access, and takes advantage of automated smart contracts to process and operate data [67].

2) Construction of BC: Generally, the BC architecture is decoupled into the following layers [68], [69], including data, network, consensus, incentive, contract and application layer.

- **Data Layer:** This layer encapsulates the chained structure of time-stamped data blocks with asymmetric encryption technology [70].
- **Network Layer:** Once the transaction generates between the parties, it is broadcast to all neighboring nodes by the peer-to-peer network.
- **Consensus Layer:** It makes highly decentralized nodes reach consensus. Currently, the mainstream consensus mechanisms include: proof of work (PoW) [10], proof of stake (PoS) [71], practical byzantine fault tolerance (PBFT) [72], etc.
- **Incentive Layer:** Incentives drive the nodes in BC network to contribute their efforts to validate data by assigning rewards to the corresponding nodes.
- **Contract Layer:** As the self-executing scripts, smart contracts bring programmability into BC, enabling BC unalterable and resistant to malicious attacks [73].
- **Application Layer:** The highest layer in BC is the application layer, consisting of various applications, such as smart city, smart home, smart healthcare, etc.

3) Categories of BC: According to different applications and thresholds, BC is generally classified as public BC, private BC, and consortium BC. Specifically, public BC is also known to be the permissionless system, where anyone can join in the BC and reach consensus freely. A private BC is managed by a single organization. Hence, private BC provides a feasible way for some organizations and applications to handle sensitive data and record-keeping. Consortium BC is operated by a group of pre-selected organizations. Only a small part of nodes will be selected to participate in the BC and reach a consensus. In Fig. 6, we summarize the comparison among public, private, and consortium BCs in the following aspects.
TABLE II
IMPORTANT ABBREVIATIONS LIST IN ALPHABETICAL ORDER

| Abbr. | Definition                          | Abbr. | Definition                          | Abbr. | Definition                          |
|-------|------------------------------------|-------|------------------------------------|-------|------------------------------------|
| AI    | artificial intelligence            | ePoW  | enhanced-proof-of-work             | PoCI  | proof of common interest           |
| BNN   | bayesian neural network            | FPNs  | feed-forward neural networks        | PoET  | proof of elapsed time              |
| BC    | blockchain                         | FL    | federated learning                 | PoS   | proof of stake                     |
| CD    | computing device                   | SG    | fifth-generation                   | PoQ   | proof of training quality          |
| CSP   | computing service provider         | HRL   | hierarchical reinforcement learning | PoW   | proof of work                      |
| CDA   | continuous double auction          | IR    | individual rationality             | PoDL  | proof-of-deep-learning             |
| CNN   | convolutional neural network       | IoT   | industrial internet of things      | PoK   | proof-of-knowledge                 |
| DL    | deep learning                      | IDC   | international data corporation     | PoL   | proof-of-learning                  |
| DLDT  | distributed ledger technology      | IoV   | internet of vehicles               | PoT   | Proof-of-reading                   |
| DNN   | deep neural network                | IoT   | internet of things                 | PoU   | proof-of-utility                   |
| DQL   | deep Q-learning                    | IC    | incentive compatibility            | QoS   | quality of service                 |
| DQN   | deep Q-network                     | LSTM  | long short-term memory             | SFC   | service function chain             |
| DRL   | deep reinforcement learning        | LM    | learning markets                   | SP    | service provider                    |
| DTWN  | digital twin wireless networks     | MDP   | markov decision process            | SAC   | soft actor-critic                  |
| DAG   | directed acyclic graph             | MEC   | mobile edge computing              | TL    | transfer learning                  |
| DML   | distributed machine learning       | MARL  | multi-agent reinforcement learning  | UDEC  | ultra-dense edge computing         |
| DSIC  | dominant-strategy incentive compatability | ML | machine learning              | VANET | vehicle ad hoc network              |
| EC    | edge computing                     | NNs   | neural networks                    | VAE   | variational autoencoder            |
| ECSP  | edge computing service provider    | PPO   | proximal policy optimization       | WAN   | wide area network                  |
| EI    | edge intelligence                  | PBFT  | practical byzantine fault tolerance| ZB    | zettabyte                          |

- **Decentralization**: The major difference among the three types of BC is that public BC is fully decentralized, while consortium and private BCs are partially decentralized or fully centralized, respectively.

- **Immutability**: It is almost impossible to tamper with transactions in a public BC. Differently, the participants in the consortium and private BCs can perform read or write operations.

- **Nonrepudiation, transparency and traceability**: Since a full copy of all transactions ever executed is stored to all the nodes on public BC, it could totally reassure the non-repudiation, transparency, and traceability of transactions of the BC system. As for private and consortium BCs, they cannot or can only partially ensure these properties.

- **Scalability**: In the public BC, it has the weaknesses of cost of low transaction-validation rate, high latency, and extra storage space consumption, limiting the scalability of public BC. As for private and consortium BCs, they have better scalability as the consensus determined by one or multiple groups can be easily reached.

- **Flexibility**: Compared with private and consortium BCs, public BC has less flexibility as configurations of private and consortium BCs are more regulable.

In this paper, we summarize the important definitions and related abbreviations listed in Table II.

C. Opportunities Brought by the Integration of the EI and BC

With the complementary advantages of EI and BC, their integration promises to provide a new set of innovative solutions in security, privacy, decentralization, and transform the network management architectures, as shown in the following.

On the one hand, BC can enhance the security, privacy and synergy of EI services by providing many promising technical features such as decentralization, privacy, immutability, traceability, and transparency [26]. Firstly, BC provides a decentralized fair agreement, improving significantly EI system trust. Secondly, BC can provide better data and resource management services while reducing management costs with the incentive of decentralization. Thirdly, BC reduces edge network complexity, resulting in significant operational cost savings.

On the other hand, EI provides resources and intelligence for BC services [74]. Firstly, the distributed structure of EI enables the deployment of BC service. Specifically, the BC service and EI service share edge resources and intelligence, which in turn saves computing resources and makes intelligent reuse possible. Secondly, the BC applications deployed on the edge scenario can call the capabilities integrated by EI to form a characteristic BC service. Thirdly, BC can take advantage of EI to accommodate compatibility, scalability, and security in improving system performance.

Recently, numerous works have proposed a number of applications that integrate BC and EI. In [75], [76], the authors give a comprehensive survey on BC applications for AI, including supply chain, smart transportation, smart healthcare, and finance. However, these works mainly address some of the problems that BC has for improving AI performance and does not consider the shortcomings of BC itself, as well as the development of EI technologies.

Therefore, this paper focuses on the integration of EI and BC to help build an efficient information value platform, facilitate resource and intelligence sharing at the edge, and improve the performance of BC. In Section III, we elaborate in detail on the motivation for the integration of EI and BC.
III. MOTIVATION OF EDGE INTELLIGENCE AND BLOCKCHAIN

As shown in Fig. 7, the limitations of EI and the complementary advantages of BC are painfully clear. Spontaneously, the appearance of BC-driven EI would be expected to pave the way for the development of emerging intelligent services. In this section, we first discuss the limitations of EI. Then, we elaborate on the benefits of BC in EI. Finally, we give the definition of the integrated system.

A. Limitation of EI

There is no gainsaying that the EI plays the role of the driver for diversified intelligent applications, while it still has many challenges to address. According to the type of EI deployment methods, we correspondingly divide the EI architecture into computing power, data, and model. As shown on the left of Fig. 7, we emphasize the challenges in these three aspects.

**Computing-Power at Network Edge:** The soaring demands for AI services are now posing new challenges to computing-power management at the network edge, as shown below.

For example, i) IoT renders various devices of different capacities concatenate in a network to communicate with each other [24]. How to manage the heterogeneous computing-power resources to adapt to the diversification of customers’ demands raises significant concern. ii) Due to the physical constraints [4], edge nodes could not support the power-hungry or computation-intensive AI services. It is of great interest to focus on how to integrate the dispersed resource to provide a series of computing solutions for the resource-constrained computing platforms. iii) Computing devices with limited resources do not support additional hardware security features, such as TPM, HSM, SGX Enclave, and hardware virtualization, which makes them vulnerable to malicious attacks.

**Data at Network Edge:** The ensuing deluge of data generated at the edges places a heavy burden on data administration and raises other issues [77].

For instance, i) most organizations consider data as a valuable strategic resource. They tend to regard the locally generated data as private property, and are reluctant to share the local data with other organizations [78]. ii) Data collaboration between distributed edges across a wide area network (WAN) may compromise the acceptable time latency. iii) Furthermore, the overwhelming majority of distributed devices are subordinate to a hub-and-spoke topology, or a server-client architecture, which is likely to pose security risks to data administration [79], [80].

**Model at Network Edge:** The intelligent system at the network edge is more vulnerable to some security attacks.

**BC Benefit of BC**

BC is an open, cryptographic, and decentralized system, maintaining immutable ledgers that are accessible for all members but tamper-proof. As shown in the right of Fig. 7, we emphasize the benefits that BC brings to EI.

**Computing-Power Management:** BC will help EI to manage the heterogeneous computing-power, and resource management is valuable, adaptable, and safe.

Specifically, the incentive mechanism is mainly used for various cryptocurrencies, it refers to promoting resource sharing, computing efficiency, and collaborative communication [82]. Moreover, BC can be accessible to all participants without any third-party intervention, which signifies this decentralized system is not controlled by any edge entity and further realizes the distributed sharing of computing power [83]. In addition, BC utilizes short signatures and hash functions to generate blocks for enhancing the security of the computing-power management, while the BC harness the appropriate mechanism to address privacy issues in computing-power trading system.

**Data Administration:** BC will help EI to administrate the diverse data, and make data administration profitable, cooperative, and credible.

In particular, incorporating BC with EI applications can bring many benefits in terms of reducing costs for trusted third parties [84], [85], improving data traceability, verifying data authenticity, protecting privacy [86] and ensuring security [87]. Specifically, BC can remove trusted third parties from data trading [88], addressing the unrealistic and inefficient issues of centralized control and data retention in the EI environments, further maximizing the potential of the data market economy. Meanwhile, BC encapsulates the chained structure of time-stamped data block with asymmetric encryption technology, thus it is desired to solve the data consistency and resist well-known attacks along with maintaining the integrity [89].

**Model Optimization:** BC will help EI optimize the AI models, and make them available, efficient and secure.

Specifically, the incentive mechanism provides a certain economic reward, enabling more members to participate in the model learning process [29]. Due to the transparency, traceability, and reliability characteristics, once the model transaction generates between the parties, it is broadcast to all neighboring nodes. Each edge node will verify the received model information based on an asymmetric cryptography mechanism. Only the valid transactions will be forwarded to other nodes and stored in the BC network [90]. In addition, the smart contract can be considered as a self-executing procedure depending on the script codes, ensuring the unalterability of BC [91]. Therefore, the BC has proven effective in ensuring efficiency and privacy, further it is suitable to solve the trust issue of collaborative inference at edges [92].

C. Integration of EI and BC

As mentioned above, the same decentralization mechanism of both the EI and BC, as well as the complementary
characteristics of BC, are destined for the BC-driven EI. Although BC brings new favorable opportunities for EI, it is still in its infancy, and there remain several critical issues.

- First, consensus algorithms make the system fault-tolerant and achieve reliability in a network composed of unreliable edge nodes. However, no consensus protocol is perfect, and it requires a trade-off between performance, security, scalability, and efficiency.
- Second, incentives in the BC do not motivate the behaviors of edge users in the desired way. Verifier rewards and penalties encourage honest users to invite attacks and destabilize PoS-based BCs, making the incentives ineffective in motivating edge users to maintain the integrated system.
- Third, smart contracts face security issues such as semantic dependencies and pseudonym operation for criminal activities.
- Fourth, scalability is a major barrier to current BC implementations, making it difficult to apply the technology at scale to EI applications.

To better support EI deployment, we correspondingly tailor BC into consensus protocol, incentive, smart contract, and application system, as shown in the right of Fig. 7.

Specifically, the integration of EI and BC focuses on the opportunities that BC brings to EI services and the improvements that BC can make to better support these services. On the one hand, we present an in-depth discussion on the potential of leveraging BC for EI, including computing power management, data administration, and model optimization services. On the other hand, we investigate the improvements of BC for better support of these services, including flexible consensus protocol, effective incentive mechanism, intellectuality smart contract, and scalability tailoring. A detailed description of the integrated system is given in Sections IV and V.

In short, the integration of EI and BC is expected to facilitate the evolution of AI services while realizing common development and prosperity. That is, the technical feasibility of integration consists of the complementary characteristics of BC and EI, while the availability of integration aims at giving play to different advantages of BC and EI, and taking advantage of complementary characteristics to make up for their own limitations.

IV. BLOCKCHAIN DRIVEN EDGE INTELLIGENCE

BC-driven EI focuses on addressing the challenges of EI as described in Section III-A. In this section, we present the EI benefits that can be realized with the assistance of BC, including computing power management, data administration, and model optimization.

A. BC-Driven Computing-Power Management in EI

As one of the most important pillars of EI development, ample computing power fuels the booming of ubiquitous EI applications. However, the tension between computing-hungry EI and computing-constrained edges mainly leads to the following challenges of computing framework and management.

- Weak Incentive: The incentives of computing-power sharing is weak. Accordingly, some EI devices cannot support the increasingly complex computing tasks due to the limited computing resources, while other nodes with abundant computing power may be idle due to a lack of value incentive.
- Complex Environments: The computing-power allocation should take full account of diversified complex factors in EI environments, further perplexing the design of resource allocation strategy.
- Inefficient Performance: In view of a complicated scenario due to the introduction of EI, the service performance that the computation offloading faces is generally inefficient. It is necessary to deeply consider how to reduce the task offloading latency, and promote the quality of service (QoS) of users while ensuring security, privacy, and fairness in the process of computation offloading.
TABLE III
COMPARISON OF UTILITY FUNCTIONS UNDER DIFFERENT MODELS

| Ref. | Utility of leader | Utility of follower | Explanation of variables |
|------|------------------|-------------------|--------------------------|
| [93] | $\sum_{i \in N} (\lambda_i - c) f_i$ | $R \cdot f_i / \sum_{j \in N} f_j - \lambda_i \cdot f_i$ | $f_i$ is the service demand of miner $i$, $c$ is cost, $R$ represents the mining reward, whereas $\lambda_i$ is the unit resource price. |
| [94] | $\sum_{n \in N} \gamma_n \cdot (p_t - T_t) \sum_{m \in N} \alpha_n$ | $\gamma_n \cdot p_t \cdot \alpha_n$ | $\gamma_n$ is discount factor, $\xi$ is cost, $T_t$ denotes the expected block interval time, $p_t$ is resource price at stage $t$, $\gamma_n$ represents the mining reward, whereas $\alpha_n$ is the hash rates of miner $n$. |
| [95] | $\{ \sum_{n \in N} \gamma_n \cdot \alpha_n \}$ | $\sum_{n \in A^n} B_i^n V^n$ | offloading option $\alpha_n \in A^n$, $\gamma_n$ is a given discount rate, $B_i^n$ is the current belief of peer $i$ at stage $t$, $V^n$ is the payoff that BS $m$ receives from its MBC services, $V^n$ is the expected long-term payoff for a given state. |
| [96] | $\frac{\sum_{j=1}^{N} q_j - \sum_{j=1}^{N} q_j - C_j - C_s \cdot \sum_{j=1}^{N} q_j + U_m - \frac{1}{2} \gamma_j}{\gamma_j} \cdot \frac{\sum_{j=1}^{N} q_j}{\gamma_j}$ | $\delta(w_j \cdot q_j - \frac{1}{2} \gamma_j^2) - mpq_j - r$ | $w_j$ means the intention of prosner $j$ to utilize energy, $\delta$ and $\gamma$ are conversion factors, $\delta$ denotes the BC service payment, $p$ is unit energy price, $q_j$ is considered as energy demand, $\gamma$ signifies the transmission loss rate, $U_m$ is the reward from BC, $C_j$, $C_s$, $c_t$ and $\gamma_j$ represent generation, storage, unit transmission and fixed operation cost, respectively. |

Accompanied by the EI deployment on the BC, BC-driven EI offers a viable solution to tackle the above issues and manage the heterogeneous while distributed computing-power resources.

1) Value-Driven Computing-Power Sharing: The value-driven computing cooperations among fogs, edges and end devices are achieving great success in empowering end-users with rich experience by utilizing resource virtualization and sharing [97], [98]. And BC is utilized to incentivize computing nodes to share their computing resources while avoiding trading privacy leakage. In the following, we investigate some related works.

As shown in Fig. 8, prior works mainly focus on designing the incentive mechanisms for the computing device (CD) to rent computing power from the computing service provider (CSP) to run the mobile BC, under the novel computing paradigms and communication protocols. Then the strategies of CSP and CD will be optimized by the game-based methods or auction-based methods for realizing the value-driven computing-power sharing.

Game-Based Methods: There already exist several studies on the game theory and pricing models for computing-power sharing [93]–[95]. Generally, the CSP and CD are self-interested, and they only expect to maximize their utilities. We summarize the utility functions under different game models, as shown in the Table III.

For instance, in [93], the CSP utility is given by the payment from CD minus the energy cost of executing the tasks, while the CD utility is given by the reward of realizing business needs minus the payment for competing resources from the CSP. Then, the game-based approach formulates the interaction between the CSP and CD as the game model.

Specifically, the CSP, as the “leader”, sets the resource price of CD $i$ to $\lambda_i$ and the CD $i \in N$, as the “follower,” sets its resource demand $f_i$. $c$ is the cost, whereas $R$ represents the mining reward. Therefore, the profit optimization problem of CSP is defined as follows:

$$P1 : \max_p \sum_{i \in N} (\lambda_i - c) f_i$$

subject to:

$$C_{u}^1 : p \geq 0,$$

$$C_{u}^2 : \sum_{i \in N} \lambda_i f_i \geq c_f_i$$

(1)

where $C_{u}^1$ considers the non-negativity constraint on price, while constraint $C_{u}^2$ ensures the non-negativity of the leader’s utility. After observing the price strategies of CSP, the CD sets its service demand to earn more profit. The optimization problem of CD can be described by:

$$P2 : \max_{f_i} \sum_{i \in N} (\lambda_i - c) f_i$$

subject to:

$$C_{d}^1 : f_i \geq 0,$$

$$C_{d}^2 : \sum_{i \in N} \lambda_i f_i \geq c_f_i$$

(2)

where $C_{d}^1$ considers the non-negativity constraint on resource demand, while constraint $C_{d}^2$ ensures that the payment of the user $i$ to the CSP is no more than the corresponding reward. After that, the optimal strategies of CSP and CD can be obtained by solving the equilibrium of the game model via some methods, such as heuristic, DL, and RL algorithms.

Specifically, in IIoT, the work in [93] investigates the resource management for the IIoT-based BC network, while modeling the interaction between the computing-power
and prosumers are first initialized with their weights \( \theta \). As shown in Fig. 9, the policy networks of operators, retailers, and prosumers as a three-stage Stackelberg game. For these hierarchical game models with multiple members in the trading market, whether the relation-ship between them can be expressed by a game model? In our previous work [96], we propose a BC-assisted software-defined energy Internet, while formulating the utility optimization problem in the trading process between operators, retailers, and prosumers to call the function.

**Auction-Based Methods:** The auction is an alternative solution for value-driven computing-power sharing while holding essential properties, such as individual rationality (IR), incentive compatibility (IC), etc.

As illustrated in [100], by the continuous double auction mechanism (CDA), a BC-enabled system for computing-power trading enables providers and customers to share their computing resources safely and fairly. Significantly, the transaction data can be recorded in the block. Then they are verified by the consensus mechanism, while the smart contract deployed on the EI acts as a broker in the auction and does not need to reap profits from the spread, to guarantee credibility and fairness of the computing-power sharing.

Further, DRL algorithms are generally leveraged to optimize bidding strategies while maximizing the payoffs of computing-power providers and customers. Unlike the single BC solution described above, a hybrid BC that combines the advantages of public and consortium BCs promotes efficiency and ensures security for all parties in the trading framework [101].

In addition, an optimal auction based on DL for edge resource sharing between the EC service provider (ECSP) and mobile users is introduced in the BC-driven EI network [102]. For the sake of achieving the optimal bidding strategies and maximizing the revenue of the ECSP, the NNs architecture is constructed to precisely fit the optimal auction. In the NNs, the monotone transform functions are leveraged to determine the allocation and payment rules of the NNs architecture, ensuring the IR and dominant-strategy incentive compatibility (DSIC). Different form [102], [103] leverages feed-forward neural networks (FFNs) to derive the optimal auction for resource allocation in the BC-driven EI network.

Generally, the above studies provide the value-incentive for devices to share their computing abilities by the introduction of BC, while the performance of the value-incentive mechanism
aims to maximize the profits of the members in the EI is improved by utilizing the learning algorithms.

2) Performance-Driven Computing-Power Allocation: One of the key challenges that will become the future of computing power allocation is how to improve system performance by taking into account multiple factors in EI [104], [105]. Specifically, a high-performance resource allocation should consider the following key issues: i) adaptive computing allocation, and ii) privacy-preserving computing scheduling.

Adaptive Computing Allocation: Recently, there have been some excellent works on improving the adaptability of resource allocation based on BC [106]–[108]. In [106], Yu et al. leverage BC to guarantee efficient resource cooperation and reliable caching of edge networks. Fu et al. leverage consensus protocol to ensure that the network-wide views can be synchronized and collected in a simplified manner across various EI systems [107].

There is some work that also considers other performance factors in resource allocation. The work in [108] defines the performance of adaptive resource allocation as the latency of the users, throughput of BC, time to finality, decentralization, and security. Additionally, some novel DRL takes the highly dimensional complex data as input to generate optimal actions, thus reducing the complexity of the adaptive resource allocation optimization problem.

Privacy-Preserving Computing Scheduling: In addition, many secure and creditable BC-based computing resource allocation schemes have been proposed. The BC-based trust mechanism not only optimizes the edge resource allocation policy, but also helps MEC address the problem of the selfish edge attacks and the faked service record attacks [109].

By jointly considering the tasks caching, computing, and BC system usage, the work in [110] presents a novel resource allocation framework to reduce unnecessary latency, while promoting caching efficiency and system security in machine-to-machine communications. After data computing and processing, they can be uploaded into the BC, while being authorized by the consensus protocol. To handle the complicated and high-dimension features of the decision process, a dueling DQN algorithm is used to formulate the actions and states offline, while obtaining the network updating online.

Additionally, the types of BC will affect the security performance of resource allocation as well. S. Guo et al. introduce consortium BC and DRL to establish the trusted and auto-adjust service function chain (SFC) orchestration architecture [111]. Based on the consortium BC, the consensus and light block nodes would be selected for resource registration, authentication, and transaction registration with smart contracts to guarantee reliable and autonomous resources allocation.

Unlike consortium BC, private BC is designed to automatically allocate the computing resources for a newly arrived data segment, which eliminates the problem of uneven distribution of resources [112]. Instead of outputting the indirect value function as the suggestion, the asynchronous advantage actor-critic algorithm directly provides the explicit policy, effectively accelerating the allocation efficiency of resources.

3) Service-Driven Computing Offloading: For devices with limited resources, offloading the complex services to EC nodes instead of executing them directly on the device can extend the limited capabilities of terminal devices while improving the quality of service. However, there are still many challenges to computing offloading, including: security and privacy protection, and dynamic computing optimization [113], [119]. Fortunately, benefiting from the advantages of BC, the BC-driven EI systems can manage the above challenges while meeting the services demands of the system.

Secure Offloading Strategy: BC, is known for its security and immutability can provide a promising solution to build trust among the EI system. In [114], an access control mechanism is implemented on the BC to validate and authorize access to mobile devices for the vehicle ad hoc network (VANET). The BC-based access control leverages smart contracts to effectively verify offloading tasks by triggering transactions, further preventing the illegal offloading action of VANET devices.

Meanwhile, the Merkle hash tree, constructed by the offloading data fragments, can accelerate the verification of task offloading [115]. Then the extended DRL can learn the offloading strategy with no need for state transition probability, and the trained DNN can describe the setting properly in the training process, which is more suitable for the high-dimensional VANET scenarios.

Dynamic Computing Optimization: Due to the dynamics in the BC-enabled EI system, computing offloading should consider long-term and comprehensive offloading performance. Recently, many valuable works focus on the above points [116]–[118]. In these studies, the main role of the BC is to build trust among multiple parties of the EI environment, while guaranteeing the data authenticity in the process of computing offloading by the consensus process. The difference between these efforts lies in the formulated optimization problems and the type of solving algorithms.

Li et al. in [116] focus on minimizing the long-term cost of cooperative computing offloading, while league learning is introduced to promote offloading performance by enabling the hierarchical agents to explore the environment collaboratively. Also, the BC-enabled EI framework jointly taking into account the selection of computing-power nodes, offloading decision and block size can further reduce the computation overhead and the energy consumption of systems [117].

In accordance with DNN, the DRL approach can handle the high-dynamic and large-dimensional offloading optimization problem by approximating the action-state value of the agent. However, the DRL-based approaches fail to converge quickly and accurately. By contrast, the work in [118] introduces the DRL combined with a genetic algorithm (DRGO). It generates the candidate solution set of the action to avoid largely unexplored action space, further accelerating the learning process and the convergence performance of the BC-enabled computing offloading.

In addition to system cost, the requirements for different BC services vary greatly, including confirmation latency, throughput, cost, security, etc. As a service-oriented BC system with virtualization and decoupling management, vDLT is presented...
to fulfill different requirements, enabling a paradigm shift from a “BC-oriented DLT system” to a “service-oriented” DLT system [120].

In vDLT, dynamic resource allocation will meet the specific needs of the service while optimizing the use of the network, storage, and computing-power resources. vDLT is decoupled into the execution, control, and application layer, which is shown in Fig. 11. Numerous transactions from the application layer are sent to the control node for prioritization of transactions and resource allocation. Then, the transactions are sent to the execution nodes for executing smart contracts and producing blocks.

When allocating resources, the QoS level of transactions and the state of available execution nodes are carefully considered to maximize the performance of the service-oriented BC system. The work in [121] considers jointly assign networking, computing, and caching resource to achieve the optimal resource utility, which can be formulated as:

$$\max \sum_{b=1}^{B} \sum_{t} a_{u,net}^b(t) \left( p_u r_u^b(t) - \delta_b H_u^b(t) \right)$$

$$+ \sum_{m=1}^{m} a_{u,comp}^m(t) (\eta_u E_u^m(t) - \phi_m q_c e_m)$$

$$+ \sum_{c=1}^{C} a_{u,cache}^c(t) \left( k_u H_u^c r_u^b(t) \varsigma_c^c(t) - \varphi_c \right)$$

s.t. $C_p : a_{u,net}^b(t) \in \{0, 1\}$, (3)

where $a_{u,net}^b(t), a_{u,comp}^m(t)$ and $a_{u,cache}^c(t)$ are the assignment identifier of networking, computing, and caching resource, respectively. $\varsigma_c^c(t)$ is a random variable, $H_u^c(t)$ is the assigned spectrum bandwidth, whereas $E_u^m(t)$ and $r_u^b(t)$ are the maximum computing and networking rate of user $u$, respectively. Likewise, we consider $p_u$ and $\delta_b$ as unit charging fee, and paying fee for accessing the network, respectively. $\eta_u, \phi_m$, and $e_m$ mean unit charging fee, paying energy fee, and energy consumption, respectively. $k_u$ and $\varphi_c$ represent unit charging fee, and paying fee for caching, respectively. Meanwhile, the above problem integrates DRL and BC, bringing intelligence among edges.

4) BC Implementation Tutorial for Computing-Power Management: In order to offer high-quality services at distributed edges while better improving the utilization level of edge nodes, several computing-power and resource trading systems have been designed. The first step for a BC to trade resources is to create a BC that connects resource market members (i.e., seller and buyer) while issuing resource assets. After that, the seller can price and sell its resource assets at will. Alternatively, a buyer can select the appropriate offer to trade. Next, we show the process of trading resources using the Command Line Interface (CLI) provided by MultiChain’s BC platform [122].

- Creating BC: the BC can be created by executing the CLI command [123]. Then, we can get the address of the administrator node that created the BC, which is necessary for the resource market members that want to connect to the generated BC. Once the BC service is started, all members connected to the BC will see the addresses of the wallets assigned to that members via the CLI command “istaddresses” for exchanging assets or making simple transfer transactions.

- Creating assets: by default, the administrator has permission to create and issue assets to support resource transactions.

- Connection of resource market members: similarly, we can connect the resource market members to the generated BC via CLI command. At this point, we need to pass the displayed address (the wallet initially assigned to the members) to the administrator node and inform the node whether it is a buyer or a seller.

- Resource trading: sellers can sell “resource” assets within the number of assets they own, while buyers can view the seller’s details and choose the right offer for the transaction. Further, we can design some trading smart contracts and the payment channel to guarantee the transparency and efficiency of the system [124], [125].

- System implementation: MultiChain provides a JSON-encoded Remote Procedure Calling Protocol (JSON-RPC) application programming interface (API) for applications. In addition, the functions related to BC can be implemented with Savoir, a module that allows you to use the Python3-based API. It also needs to store the information through a database and implement each node as a Web service, which can be done through Django, a python based Web framework.

Summary: With the help of BC, the EI system driven by BC enables the valuable, adaptable and safe computing-power management of sharing, allocation and offloading, and addresses the problems of heterogeneity, dispersibility and security attack in the computing-power management of EI. All in all, we summarize the above-mentioned works of BC-driven EI in computing-power management in Table IV.

B. BC-Driven Data Administration in EI

Nowadays, drastically growing volume and the types of data incur an urgent need to push the frontiers of AI to the edges. However, the inefficient data administration between different multiple parties has seriously set back the development of EI technology for the following reasons.

- Self-Interest: EI devices are often selfish, and they may be reluctant to share data with others. Data sharing should be reasonably monetized to incentivize the paid data sharing by selfish EI devices.
TABLE IV
DETAILS OF BC-DRIVEN EI IN COMPUTING-POWER MANAGEMENT

| Benfits | Ref. | Application | Model | Algorithm | Consensus mechanism | Strategy | Objective function | Property | Security Analysis | Smart Contract |
|---------|------|-------------|-------|-----------|---------------------|----------|--------------------|----------|------------------|----------------|
| Value-Driven Computing-Power Sharing | | | | | | | | | | |
| IoT | [93] | IoT | Stackelberg game | MARL | PoW | Determination of service demand and price | Payment minus cost | Nash equilibrium | × | × |
| MBC | [94] | MBC | Stackelberg game | HRL | PoW | Determination of service price | Discounted payment | Nash equilibrium | × | × |
| IoT | [95] | IoT | Stackelberg game | HRL | - | Determination of service | Payment minus cost | Bayesian equilibrium, fairness | × | × |
| IoT | [100] | IoT | Auction | DRL | PoW/PoS | Determination of bid for computing power | Payment minus valuation | Fairness | ✓ | ✓ |
| Edge computing | [101] | Edge computing | Auction | FL | PoW | Determination of service | Valuation | IR, IC, fairness | ✓ | ✓ |
| IoT | [102] | IoT | Auction | DL | PoW | Determination of bid for computing power | Payment | DSIC, IR | × | × |
| Fog computing | [103] | Fog computing | Auction | DL | PoW | Determination of renting price and service price | Payment | IR, IC, fairness | × | × |
| Performance-Driven Computing-Power Allocation | | | | | | | | | | |
| MBC | [106] | UDEB | Joint optimization | 2Ts-DRL/FL | - | Determination of resource allocation | System cost | Efficient utilization | ✓ | × |
| MBC | [107] | MBC | Multi-objective optimization | DRL | Robust BFT | Determination of view changes, access selection, computing, caching and bandwidth allocation | System rewards | High-dynamic and large-dimensional | ✓ | × |
| MBC | [108] | MBC | Multi-objective optimization | Double-dueling DQN | PBFT/DPoS | Determination of spectrum allocation and block attributes | Long term reward | High-dynamic and large-dimensional | ✓ | ✓ |
| MBC | [109] | MBC | MDP | RL/DRL | PoW/PoS | Determination of resource allocation | Payment minus cost | Credibility | ✓ | × |
| IoT | [110] | IoT | Optimization | DRL | PBFT | Determination of resource allocation | System rewards | High-dynamic and large-dimensional | ✓ | × |
| IoT | [111] | IoT | Combinatorial Optimization | DRL | PBFT | Determination of resource allocation | System cost | Credibility | ✓ | × |
| IoT | [112] | IoT | Continuous-time MDP | DRL | PoW/PoS | Determination of resource allocation | Profit minus cost | Credibility | ✓ | ✓ |
| Service-Driven Computing Offloading | | | | | | | | | | |
| MBC | [113] | MBC | Joint optimization | DRL | Delegated BFT | Determination of offloading decision, power allocation, and block attributes | System reward | cooperativity | ✓ | × |
| IoT | [114] | IoT | Joint optimization | DRL | Redundant BFT | Determination of offloading decision | System cost | High-dimensional | ✓ | ✓ |
| VFC | [115] | VFC | Lagrangian optimization | ML | PoW | Determination of offloading decision | Average task offloading delay | Trustfulness, fairness | ✓ | ✓ |
| MBC | [116] | MBC | Joint optimization | MADRL | - | Determination of offloading decision | System cost | Cooperativity | × | × |
| IoT | [117] | IoT | Joint optimization | DRL | PBFT | Determination of energy allocation | System cost | High-dynamic and large-dimensional | ✓ | × |
| MBC | [118] | MBC | Joint optimization | DRGO | PoW/PoS | Determination of offloading decision | System cost | High-dimensional | × | × |

- **High Latency**: Data sharing in the collaborative end-edge-cloud network will generate unacceptable latency, limiting the effectiveness of some applications, especially for the real-time service with high demands.
- **Security Risk**: Along with the value the data brings, serious issues about data collaboration in EI come, such as privacy leakage, network communication failure, and malicious attackers.

BC is a promising approach to the above problems for efficient data administration. Specifically, BC-driven data administration is embodied in incentive data trading, data caching strategy, and reliable data collaboration, as shown in Fig. 12.
- First, BC can unlock the enormous potential of a data market economy, realizing the exchange and trading of data value.
- Second, the incentives motivate EI devices to cache data while broadcasting them.
- Third, the consensus mechanism and smart contract technologies in BC enable secure data collaboration among EI devices.
Overall, the above three modules cooperate to improve the efficiency of edge data administration from three aspects of incentive, real-time, and security.

1) Incentive Data Trading: Some current researches generally focus on how to share edge data and cooperate through AI algorithms, while ignoring how to incentivize more devices to contribute their data [126]–[128].

Some exceptional properties that BC brings can be of great benefit in realizing the exchange and sharing of data value without traditional intermediaries, as shown in the upper layer of Fig. 12. Furthermore, BC enables data sharing monetized in a distributed peer-to-peer manner, further ensuring the efficiency of the EI system. Additionally, the specific consensus protocols and smart contract can reduce the transaction verification latency and resource consumption cost, promoting trading efficiency and accelerating the funds turnover in the trading process.

In the frequent data-trading scenarios, such as the Internet of Vehicles (IoV), because of the transaction verification delays and the high computing consumption, the devices requesting data generally do not have enough tokens to execute the next transaction immediately.

Proof-of-trading (PoT) protocol [129], jointly considering the market interests of high-power nodes’ and resource consumption, eliminates the above vulnerabilities. Concretely, the monetary aggregates in knowledge trading can be regarded as the stake, and the difficulty of the hash puzzle can be dynamically adjusted on the basis of the stake. In [130], a proof-of-knowledge (PoK) consensus mechanism replaces the complicated cryptography puzzle with the knowledge learning process. It allows the computation consumption in the knowledge trading process to be used for consensus verification in the BC, enabling the trading process to be more lightweight.

Then, the incentive mechanism can release tokens while distributing the corresponding rewards to the miners, motivating more EI devices to contribute data as well as ensuring fairness. Different from the token incentive, a Shapley value solution [131], as a scheme for sharing revenues generated by a coalition, is a novel attempt to incentivize EI devices to perform actively in medical data sharing. Benefit from the programmability and non-repudiation of smart contracts [132], the token trading between transaction participants can be automatically executed as well, incentivizing the devices to act honestly. Meanwhile, the ML algorithm is leveraged to solve disputes about the availability of data trading.

2) Data Caching Strategy: Data caching provides all-in-one solutions to mitigate the mobile traffic on backhaul links effectively. However, the high complexity of the El scenarios and the strong mobility of the EI devices give rise to dynamic edge network topology and time-varying wireless channel conditions, which in turn hinder the design of optimal data caching strategy. In addition, EI devices are generally reluctant to store their data with a caching provider, due to the fact that the caching provider may be less trustworthy and the caching process may leak the private data. Fortunately, BC has the capacity to establish an autonomous, transparent, and invigorative data storage market.

In the following, we investigate some related works that BC can benefit EI in data caching strategy, which is as shown in the middle layer of Fig. 12. With the help of BC, the completed data transactions can be recorded into a block after the transaction verification, addressing the untrusted issues in data caching. After storing the data, the off-chain data storage strategy help users quickly collect and query data [133]. The off-chain data storage strategy generates anonymous datasets quickly, accelerating data administration’s efficiency.

Besides, the new block verifier selection methods [134], i.e., consensus mechanisms, render data caching faster and more efficient. Proof of utility (PoU) consensus [135], integrating BC and data caching, verifies the correctness of data caching, while combining advanced DRL methods, the optimal data caching policy can be derived by learning the edge network topology and channel conditions.

Additionally, the token incentive motivates EI devices to cache data while broadcasting them [136]. In this regard, the EI devices can download the cached contents by receiving broadcast information transmitted by the BC. Meanwhile, the data can be recommended and broadcasted to the relevant EI devices combined with the ML or DL methods, further improving the cache hit rate and robustness of data caching. Furthermore, the smart contract supports the credible data delivery transactions [137], where the caching node selection and caching placement can be formulated as an MDP that can be solved by a DRL approach to obtain the optimal caching strategy by maximizing the accumulated reward.

3) Reliable Data Collaboration: The fast growth in the volume of data in network edges, opens up new possibilities to improve the quality of service for the emerging applications through data collaboration and sharing, but it also poses serious concerns due to the heterogeneity and decentralization of EI devices, such as data tampering, data leakage, etc. These factors may become the bottleneck of EI [144].

BC employs encryption functions to create an add-only, apparently tampered log so that the records in the BC cannot
be tampered with and the activities of malicious customers or manufacturers are traceable. In addition, the consensus mechanism and smart contract technologies enable secure data collaboration among EI devices, as shown at the bottom of Fig. 12. As for secure data collaboration in the EI system, some studies have made attempts from different perspectives.

Due to the DLT characteristic and encryption algorithms of BC, it is desired to solve the data consistency issue during the data collaboration [138]. BC can synchronize and broadcast the root hash value of the transaction among EI nodes, which in turn supports data registration while guaranteeing data consistency and ownership. The introduction of the DRL algorithm can help establish an efficient data collection and sharing scheme in which the fully distributed DRL scheme is leveraged for data collection [139]. At the same time, assisted by the ML, the EI datasets can be processed on-chain [140], and exclude the data-related issues such as repetition, loss of data value, errors, and disruption, providing a lightweight yet effective solution for secure and private EI system.

Based on the advantages of the consensus mechanism in trusted transaction verification, all peers can audit and record the interactions between EI nodes during data sharing, thus supporting multi-party incentives and privacy protection. An enhanced-proof-of-work (ePoW) protocol can authenticate data records and prevent data poisoning attacks caused by tampering with the original data [141]. It can be combined with the variational autoencoder (VAE) and long short-term memory (LSTM) algorithm to protect data from leakage while performing anomaly detection.

Unlike using ML algorithms to secure data, the work in [142] formulates the data sharing as an ML problem while presenting a proof of training quality (PoQ) protocol, so that data privacy can be preserved by sharing data models in security via BC, rather than exposing actual data.

In addition, the BC-driven EI system is based on the smart contract that can support the more flexible implementation of data collaboration strategy, while realizing the trustworthy value exchange between EI nodes [143]. Among the above works, the BC type is permissioned BC, consisting of trusted organizations, which can establish more secure connections among credible EI devices through its encrypted records, further preventing network communication failure, malicious devices attacks, etc.

4) BC Implementation Tutorial for Data Administration: BC brings enhanced security, greater transparency, and traceability to data management. Currently, various BC platforms have been considered to implement this system. Next, we show the process of data management by the Hyperledger Fabric platform, an open-source project from the Linux Foundation, designed by I.B.M.

- Architecture design: In the data administration, considering BC can record the user’s information, while organizations can request detailed information via BC transactions and use it as an access log. Administrators can share a user’s data when the user grants appropriate consent to an organization.

- Initialize BC: Before deploying the BC, we need to have the following prerequisites installed on the data management system: Curl, Nodejs, Git, Python, Go Language, Docker C.E., Docker Compose, etc. After the prerequisites are installed, Fabric samples will be downloaded using Curl. Then, environment variables are updated to ensure Golang works. Finally, the test network is initialized [145].

- Chaincode: The main function of chaincode (smart contract) is to record consent information from the network, query the user’s consent details, and provide historical information. All of the releases of Fabric take an advanced approach to deploying chaincode on the endorsing peer, where the Hyperledger Fabric Node SDK can be used to record data.

- Status register: The admin user and application users need to register [146], obtain credentials and store them in a wallet. Application users will be able to access the chaincode function when the credentials have the appropriate authorization properties.

- Built react applications: React applications for organizations and users are built to interact with the network while invoking the functions of chaincode with the use of API endpoints.

- Data administration: Once the users log in, they can store their data on the database from the react applications while the organizations can gain access to the functions of chaincode to support data sharing and management.

Summary: For the sake of the BC-based storage and computation infrastructures to coexist with the EI infrastructure, the data trading, caching and collaboration need to be integrated with profitability, cooperativity and credibility. EI leverages BC to collect, record, encrypt and broadcast the data transactions, while employing some learning-based methods to address the problems of data administration with lower complexity. To sum up, we summarize the above-mentioned works of BC-driven EI in data administration in Table V.

C. BC-Driven Model Optimization in EI

Specifically, to accommodate the resource-constrained edge scenarios, some problems, such as the size of AI models, computation requirements of modeling, etc., should be optimized accordingly. However, the current model optimization in edge scenes faces many challenges, including i) Inefficient training and ii) Incredible inference. Thanks to the breakthrough of BC technology, many researchers are now offering viable solutions through the BC to tackle the above issues, such as efficiency, and credibility.

1) High-Efficiency Training: Specifically, the high efficiency of learning-based algorithms refers to how to provide some incentives and security assurance for edge servers to improve training performance [147].

3https://curl.se/download.html
4https://nodejs.org/en/
5https://git-scm.com/
6https://www.python.org/
7https://golang.google.cn/
8https://docs.docker.com/
9https://github.com/hyperledger/fabric-sdk-node
On the one hand, these incentive mechanisms for the AI algorithms in edge environments can motivate edge devices to offer high-quality data and participate in model training. Generally, the block reward in BC can motivate the distributed edge devices to exchange and verify the local model updates.

Specifically, based on the nature of BC, the BC-based FL (BlockFL) algorithm can promote the collaboration among edge devices with a larger number of training samples by providing rewards proportional to the training sample sizes [148]. However, the BlockFL does not prevent some malicious behaviors of the clients involved in the training.

Thus, the reliable reputation mechanism with contract theory is coming through [149]. This incentive mechanism stimulates the high-reputation workers that have high-accuracy and reliable local training data to join in the learning process, and improve the accuracy of learning models. Despite the financial incentive, the work in [150] rewards the data contributors of the AI model with points and badges. These non-financial incentives can be recorded on-chain in a smart contract, and identified by the contributor’s wallet address for training a good AI model.

We summarize the generalized architecture of the BC-driven FL model, as shown in Fig. 13. Specifically, the BC-based FL expects to train the global model in each FL iteration to obtain the local model $w_i$. Without loss of generality, the set...
of mobile devices can be denoted as $N = 1, 2, 3, \ldots, N_N$ with $|N| = N_N$, where $N_N$ is the number of the edge nodes. Moreover, the set of training data on the distributed devices is defined as $D_i, B_i$ is the minibatch extracted from $D_i, w_i$ means the model trained by device $i$, then the loss function can be expressed as $f_{M_i}^j(w_i)$. Device $i$ is committed to minimizing $f_{B_i}(w_i)$, as follows:

$$\min F_i(w) = E_{M_i \sim D_i} f_{M_i}(w_i) \quad (4)$$

After several FL iterations, BC-driven FL typically uses the smart contract for aggregating global model $w$ by minimizing $F(w)$ as follows:

$$\min F(w) = \frac{1}{N} \sum_{i \in N} f_i(w_i) \quad (5)$$

Therefore, when designing a BC-driven FL model, we need to clarify how to use the inherent nature of BC to solve the challenges in traditional FL. For example, the smart contracts can be leveraged to aggregate the local models to avoid single points of failure in FL, or to reduce additional resource consumption by combining DAG with BC.

**On the one hand**, the EI model, as the collaborative AI paradigm, brings great benefits for edge networks, while it is still vulnerable to various security attacks of the distributed clients. Currently, there are many works that have studied the security of the EI model training, such as the BC-driven FL models [151]–[154] and other distributed learning-based models [155]–[157]. As shown in Fig. 13, the BC-driven FL models replace the central server with BC to aggregate the local models uploaded from the end devices for obtaining the global model, extending the federation range among the untrustworthy edge devices [29]. And other BC-driven distributed learning algorithms have similar ideas. Nevertheless, BC-assisted distributed learning for intelligent edge devices may increase the risk to edge devices’ sensitive data.

Focusing on the above problems, Kang et al. in [158] introduce reputation, managed by the consortium BC deployed at edge nodes, to defend against unreliable model updates for reliable FL. The consortium BC adopts proof of elapsed time (PoET) consensus and consists of reliable workers. It can record and update the reputation opinions with the workers’ digital signatures. Therefore, for a certain worker candidate, any FL task publishers can obtain the latest reputation opinions by downloading them from BC. By this means, the publishers are in a position to select the high reputation workers to perform the FL tasks securely.

Moreover, a BC-based crowdsourcing FL system leverages differential privacy to prevent adversaries while ensuring that all model updates are held accountable [159]. Despite storing the models on-chain, the locally trained models are stored off-chain by using IPFS. After hash operations, they are sent to BC as the transactions. Then, the consensus nodes verify the authenticity of these transactions, and obtain the global model by aggregating the model parameters received from the distributed devices.

Furthermore, smart contracts can incorporate local differential privacy technology to defend against inference attacks in FL [160]. In [161], the authors integrate BC and FL into IoV, and develop a hybrid BC, i.e., PermiDAG, consisting of the permissioned BC and the local directed acyclic graph (DAG). In order to improve the reliability and security of the proposed scheme, the asynchronous FL architecture is proposed, consisting of three phases: node selection, local training, and global aggregation. Specifically, the DRL algorithm is leveraged to select participating nodes. The selected nodes then perform the asynchronous local training utilizing BC and execute the synchronous global aggregation based on the DAG.

In addition, the encryption scheme [155], error-based aggregation rule [156] and various training contracts [157] can facilitate secure collection and aggregation of the local DL model from multiple edge servers, further enhancing the model training in the edge scenarios.

2) **Credible Inference**: Collaborative inference can help distributed edge nodes to complete large tasks with mutual collaboration in edge-assisted multi-agent systems. Nevertheless, the model inference at the edges is open to security threats. BC can provide a promising solution to address the trust issue of collaborative inference in edge-assisted systems due to its integrity and privacy features.

On the one side, the collaborative inference at the edges can be formulated as the collaborative knowledge graph construction [162]. Specifically, the knowledge graph can greatly facilitate the inference performance of learning models by modeling semantic entities and attributes and pointing out potential relationships between them. Also, PoK consensus selects leaders based on knowledge contributions, which reduces the risk of malicious nodes while reducing traffic load and computational overhead.

On the other side, inductive methods can be utilized to infer certain facts and hypotheses [163], while the knowledge base can be considered as an inference engine for selecting rules accordingly to apply particular symbols in the collaborative inference of the EI framework. This framework focused on optimizing AI training and AI inference in the edge to achieve the interpretability and robustness of AI in the visual EI application system.

3) **BC Implementation Tutorial for Model Optimization**:

Some BC-based FL replaces the central server with BC for model update exchange, while allowing the most honest nodes to enhance each other and continuously improve the model performance. FISCO BCOS 2.0 [99], an open-source BC underlying platform, proposes a set of pre-compiled contracts and frameworks that allow users to write smart contracts using C++. Since it does not enter the EVM for execution, pre-compiled contracts are suitable for the BC-based learning algorithms where the contract logic is simple but frequently invoked, or where the contract logic is fixed and computationally intensive. Next, we give a tutorial on the system implementation of BC for model optimization using FISCO BCOS as an example.

- Pre-compiled contract preparation: Before building the BC, we need to prepare the pre-compiled contract, download the FISCO-BCOS source code,10 and embed the

10https://github.com/FISCO-BCOS/FISCO-BCOS.git
TABLE VI
DETAILS OF BC-DRIVEN EI IN MODEL OPTIMIZATION

| Ref. | Name       | Algorithm | BC Type   | Consensus Algorithm | Contribution                                                                 | The main role of BC                        | Security Analysis | Smart Contract |
|------|------------|-----------|-----------|---------------------|-----------------------------------------------------------------------------|---------------------------------------------|-------------------|----------------|
| [146] | BlockFL    | FL        | Public    | PoW/PoS/ PoS/BFT  | Enable on-device ML with no need for centralized training data              | Promote the collaboration among edge devices | ×                 | ×              |
| [149] | -          | FL        | Consortium| -                  | Stimulate high-reputation devices to participate in model learning          | Ensure the reliable reputation calculation  | ×                 | ×              |
| [150] | -          | ML        | Public    | -                  | Train a model by the dataset built collaboratively on the BC               | Record the incentives                        | ✓                 | ✓              |
| [151] | DTWN       | FL        | Consortium/Private | - | Develop a BC-enabled FL framework for collaborative computing | Share the model information               | ×                 | ×              |
| [152] | iFLBC      | FL        | Public    | PoS/CI             | Bring EI to end-nodes by incorporating FL and BC                          | Store the shared learning model             | ✓                 | ×              |
| [153] | Fleschain  | FL        | Consortium/Public | PoW/BFT/PoW        | Realize the secure collaboration of multiparty data computation             | Store local model parameters for each global iteration | ×                 | ×              |
| [154] | FL-Block   | FL        | Public    | PoW                | Develop a novel BC-enabled FL to solve the subsequent inefficiency of EL    | Verify the global learning model            | ✓                 | ×              |
| [158] | -          | FL        | Consortium | PoET              | Introduce reputation managed by BC for reliable FL                         | Record and update the reputation options    | ✓                 | ×              |
| [159] | -          | FL        | Consortium | Algorand           | Realize a BC-based crowd-sourcing FL system to prevent adversities         | Aggregate the model parameters from the distributed devices | ✓                 | ✓              |
| [160] | -          | FL        | -          | -                  | Integrate BC and differential privacy to prevent poisoning and inference attacks | Utilize smart contracts to prevent malicious or unreliable participants in FL | ×                 | ×              |
| [161] | PermiDAG   | FL        | Consortium/Private | Simplified PoW | Propose a hybrid BC architecture to improve the security of model parameters | Perform the asynchronous local training     | ×                 | ×              |
| [155] | FPPDL      | DL        | Consortium/Public | PoW/BFT/ PoS      | Propose a decentralized Pair and Privacy-Preserving DL (FPPDL) framework to incorporate fairness into FL | Record all trades as immutable transactions | ✓                 | ×              |
| [156] | -          | DL        | Consortium/Private | - | Develop an error-based aggregation rule to prevent attacks | Provide a way to secure interactions among a group of entities | ✓                 | ×              |
| [157] | DeepBlock  | DL        | Private    | -                  | Facilitate the collection and aggregation of the local model from edge nodes | Provide a secure DL operation and removes the control from a centralized authority | ✓                 | ✓              |
| [162] | BCEI       | FL        | Hybrid     | PoK                | Develop a framework for collaborative edge knowledge inference (BCEI)      | Provide a trust solution for multi-party communication and trace transactions | ×                 | ×              |
| [163] | -          | Abductive | Public     | DPoS               | Design an edge AI framework to perform the object detection task          | Enable RSU nodes to store data by transactions | ×                 | ✓              |

pre-compiled contract for the BC-based FL algorithm as well as assign and register the contract address.
- Compile source code: You need to install the compilation dependencies and specify the version cmake,11 then it can be used to compile FISCO BCOS.
- Build BC: First of all, FISCO BCOS development and deployment tool script needs to install OpenSSL, and curl dependency. Secondly, you need to create the operation directory and download the BC installation script. This script can then be used to create a consortium BC, and after starting the BC some on-chain operations can be performed, such as on-chain model aggregation.
- Client Configuration: After configuring the appropriate environment and certificates, you need to define the pre-compiled contract interface and implement the BC-based FL algorithm by calling the defined interface.

Summary: BC-driven EI stimulates the secure collection and aggregation of model information from distributed devices, enhancing the efficiency and credibility of model optimization. Overall, we summarize the above-mentioned works of EI driven by BC in model optimization in Table VI.

V. TAILORING BLOCKCHAIN TO EDGE INTELLIGENCE

Despite the great remarkably benefits brought by BC to EI, BC is still in the initial stage, and also facing critical challenges. In a view of bridging the gap between immature BC and EI amicable BC, it is necessary to acclimatize BC to EI. In this section, we probe into how to tailor BC to EI from four perspectives, including flexible consensus protocol, effective incentive mechanism, intellectuality smart contract, and scalability.

A. Flexible Consensus Protocol Tailoring to EI

Diversified business requirements of EI demand runtime efficiency, flexibility, and security from all operations. This

11http://www.cmake.org/files/v3.18/cmake-3.18.2.tar.gz
factor also holds true for BC-driven ones. However, the current consensus protocols deployed in BC face the following critical challenges.

- **Redundancy**: The existing consensus mechanisms are redundant. For example, the PoW protocol requires brute force to solve the mathematical puzzle, which gives rise to energy waste and further dilutes the value of the BC.
- **Incompatibility**: The current BC lacks compatibility, only enabling one consensus protocol used, which does not adapt to the changing needs of the edge.
- **Security**: Consensus mechanism is facing a wide range of threats, such as majority attack, double spending attack, and so on.

In the interest of addressing the above challenges, there are a number of ongoing researches improving the existing consensus protocols, such as multi-functional design, compatibility, and attack defense.

Popular cryptocurrencies generally adopt work-based consensus protocols to validate the transactions in a distributed ledger. However, these protocols need to consume considerable computing resources to solve the hash puzzle, which is expensive, redundant, and not applied for anything other than validating transactions.

Hence, many researchers are absorbed in designing a lightweight BC system by introducing the multi-functional consensus protocols [164]. Here, we give the general design architecture of the multi-functional protocol, as shown in Fig. 14. In general, these multi-functional protocols replace the laborious and pointless hashed calculation with DL training, enabling BC to share more advanced intelligence among edges. Then the new block will be generated and verified. Afterward, the winner will be selected. That is, when the agreement is reached in BC, AI tasks will be completed simultaneously without any redundant computation.

This improvement has a positive impact on EI. BC recycles computational resources and helps it become a more lightweight system that can be easily adapted to edge scenarios. We summarize the different multi-functional consensus protocols, as shown in Fig. 15.

1. **Multi-Functional Design**: PoDL: Specifically, proof-of-deep-learning (PoDL) mechanism is an energy-recycling protocol [165], as shown in the top half part of Fig. 15. In PoDL-based BC, there exist a model requester, miners, and full nodes. The model requester outsources DL model training to miners, then the miners engage in the DL training tasks, rather than the meaningless hash calculation, and the full nodes validate the training models on test datasets. This protocol only incrementally adds components to block headers, which can be generalized to any PoW-based BC.

PoL: Recently, we propose a proof of learning (PoL) based on AI-Chain [166], which is a distributed and immutable record of learning results, further supporting the intelligence sharing among edges. More specifically, the PoL mining process first requires training NNs locally and then encapsulating each node’s local learning conditions into transactions. Only the winner can issue a block and reach the consensus with other edges. Once a block is generated, other nodes verify the block and eventually update their local intelligence.

Proof-of-Useful-Work: As shown in the left side of Fig. 15, Coin.AI [168] develops a proof-of-useful-work scheme to support its normal running. Unlike the PoDL and PoL, the digest containing the list of transactions, nonce and the hash of the previous block will be hashed in the proof-of-useful-work, while the obtained hash value will then determine the hyperparameters for the trained DL model. Afterward, the miner would win once the miner’s DL performance exceeds a given threshold. Likewise, a proof-of-storage mechanism is introduced to provide the storage of DL models in a distributed manner, which is combined with the proof-of-useful work scheme to support the proliferation of Coin.AI.

PoQ: This protocol [142], integrating FL into the consensus process of permissioned BC, promotes the utilization of computing resources while increasing the performance of the FL algorithm, as shown in the right side of Fig. 15. Specifically, the PoQ consensus process is as follows.

1. The committee node with the highest accuracy would be elected as a committee leader. 2. The leader gathers all
the received transactions, and broadcasts the block to all committee nodes for approval. The model transaction track of a block is verified by the verifying node with the prediction accuracy. An approval will be distributed to the leader once the accuracy is within limits. Further, the transactions will be recorded in the BC and sent to all nodes after being signed with the leader’s signature, if every committee node approves the block containing all transactions.

2) Compatibility Enhancement: Edge scenarios have a large number of flexible intelligence requirements. For example, EI helped vehicular networks need the speedy intelligent decision with the ultra-low latency, while EI helped to cache decision has the less emphasis on latency feature. Meanwhile, different BCs have various performances, such as scalability, decentralization, and security, mainly due to their respective used consensus protocols.

However, the traditional BC does not consider different application conditions, only using a single consensus protocol as the best fit one [172]. Considering the diverse requirements of EI, it is unreasonable to use only one consensus protocol in BC-driven EI. According to the actual application scenarios, the BC should be compatible with multiple consensus mechanisms and has the flexibility to select the most appropriate one to improve the compatibility performance of the BC [120].

The solution is to switch the different protocols flexibly to meet the diversified QoS requirements of edge scenarios while making the BC system more efficient and robust. Since the learning-based algorithms can learn the transition regularities of the system environment, the QoS requirements of the users, and the situation of enabling resources, it would provide a feasible way for BC to select more suitable protocols to perform. For example, the works in [169], [170] design a service-oriented permissioned BC and launch several consensus protocols based on the users’ QoS requirements. Then, the consensus protocol selection, block producer selection, and bandwidth allocation can be formulated as the MDP, where the state, action and reward function are defined as follows.

\[ S(t) = (F, B(t), B(t)) \]  

\[ A = \{A_c(t), A_b(t), A_n(t)\} \]  

\[ Q = (L_1, L_2, L_3) \]  

\[ Q = (L_1, L_2, L_3) \]  

System State: At time slot t, the learning agent senses system state \( S(t) \), including the user’s QoS preference set \( F \), network bandwidth set \( B(t) \), computation capability set \( C(t) \) of all nodes, etc., which can be described as follows:

\[ S(t) = (F, B(t), B(t))^T \]  

System Action: After observing the state of the system, the protocol candidates \( A_c(t) \), block producer \( A_b(t) \), and network bandwidth \( A_n(t) \) would be decided by the learning agent, then system action can be defined as follows:

\[ A = \{A_c(t), A_b(t), A_n(t)\} \]  

Reward Function: Without loss of generality, we assume that there are three protocols in the service-oriented permissioned BC while the goal of this system is to maximize the QoS of the client. The \( L_1 \), \( L_2 \), and \( L_3 \) denote the QoS of the three protocols. Then the reward function can be expressed as:

\[ Q = (L_1, L_2, L_3)^T \]  

Afterward, the above MDP can be addressed by the DRL algorithm to achieve a service-oriented BC. Specifically, a DRL algorithm that learns the process and the architecture of the dynamic BC network, allows the selection strategies of the protocols much faster for the various users’ QoS requirements, while improving the compatibility performance of system operation and making BC more suitable for the edge scenarios.

Unlike the traditional DRL approach, we leverage dueling DRL to learn the relative advantage of action, by evaluating the value function [170]. As shown in Fig. 16, there are two identical deep networks in dueling DRL, one adjusting the weights and biases, and the other periodically updating these parameters. The coordinator uses \( \epsilon \)-greedy policy to determine the selection and allocation strategies for the next state.

Therefore, for the BC compatibility issues, we need to formulate the BC compatibility into a specific optimization problem, such as QoS maximization, and design appropriate learning algorithms to solve it. Experimental results show that the QoS provided by the adaptive BC is always high when jointly considering the selection of consensus protocol, the selection of block producers, and the allocation of bandwidth resources, which illustrates the effectiveness of the pluggable consensus mechanisms.

3) Attack Defense: Recently, several secure and robust consensus protocols in the BC have been researched, which ensure the security requirements in EI. For example, Salimitari et al. [171] propose a two-step consensus protocol for AI-enabled BC. The first step leverages an outlier detection algorithm for the second step consensus (i.e., PBFT) in edge networks. Outlier detection is committed to verifying the compatibility of new data, while discarding the suspicious ones. Applying ML to reach consensus in a BC-enabled edge network can not only reach consensus over new data in milliseconds, but also enhance the overall fault tolerance of the BC.

4) BC Implementation Tutorial for Consensus Protocol Tailoring: Consensus protocols are an essential key element to achieving autonomous property BC. This section lists some relatively lightweight open-source implementations of consensus protocol libraries, including BFT consensus, Raft consensus, Paxos consensus development libraries [173], etc. Specifically, these development libraries can cope with Byzantine failures in distributed systems, including Tendermint Core [174], BFT-MaRt [175], SBFT [176], libbft [177], etc. Furthermore, Hyperledger Fabric uses a Raft consensus-based sorting service, where Raft’s website lists dozens of developed in various languages Raft consensus protocol implementations. Based on cpp-ethereum [178], this section provides guidance for implementing a scalable decentralized trust infrastructure (SBFT) [179] as an example.
Protocol design: In effect, we need to design a specific consensus protocol process, while giving a detailed implementation method to reach consensus. For example, SBFT adds four key design components based on the PBFT protocol: (1) moving from PBFT to linear PBFT; (2) adding the fast path; (3) using encryption to allow single message acknowledgment; (4) adding redundant servers for resiliency and performance improvements.

Cryptography implementation: Some tool libraries are available for cryptographic implementation. For instance, Crypto++ library [180] can be leveraged to implement the cryptographic primitives, while using RELIC cryptographic library [181] to implement boneh–lynn–shacham signatures.

Parameters design: For proper operation of BC, some parameters need to be given, such as the minimum number of client operations per block and the actual number of decision blocks submitted in parallel by the primary node.

Smart contract implementation: Smart contracts need to be deployed so that they can be invoked flexibly, and cpp-ethereumk can provide an implementation of EVM.

Summary: In order to meet the time-sensitivity, elasticity, and safety requirements of EI, tailoring BC to EI helps BC develop the multi-functional consensus protocols to recycle resources at edges, while designing the pluggable consensus protocols to satisfy the various QoS requirements of EI scenarios, as well as some security threats to be alleviated. Overall, we summarize the above-mentioned works of flexible consensus protocol tailoring to EI in Table VII.

B. Effective Incentive Tailoring to EI

BC is able to prosper EI through incentives, which encourage edge learning agents to contribute more intelligence, i.e., training capacity or inference capacity, to form a positive EI ecology. However, the existing incentive mechanisms face various problems:

- The prevalent mining process, such as PoW, is mainly oriented to resource-rich nodes. General devices suffer from limited computing power, and it is hard for them to participate in the mining process. Accordingly, BC is difficult to integrate effectively into EI systems, reducing the motivation for the distributed edge devices to join BC.
- While BC-based cryptographic currency has benefited people’s lives, they have also brought about money laundering, Ponzi schemes, and other cybercriminal activities.

In recent years, several works have been conducted to address the above problems. It further improves the effectiveness of incentive mechanisms, including mining strategy optimization and risk prevention against cryptocurrency.

1) Mining Strategy Optimization: Generally, public BC technology has been widely deployed in mobile environments. Nevertheless, the computation-intensive mining process impedes the practical use of public BC in mobile scenarios, due to the fact that the consumed computing power to perform mining tasks is prohibitively high for edge devices. Some researchers are dedicated to effectively overcoming the limitation mentioned earlier.

For instance, the authors in [182] consider providing the computing power for executing the mining task. As shown in
Fig. 17, utilizing the RL algorithm, the optimal mining strategy in resource management will be obtained by addressing the Stackelberg game between the SP and miners. Concerning incentivizing the participation of the edge devices in the mobile BC mining, the mining task scheduling problem is formulated as an MDP in [183]. Considering the impacts of system performance and privacy levels of all BC users [184], the task offloading, user privacy preservation and mining profit can be formulated as a joint optimization problem. Due to the fact that there is no prior knowledge of the system dynamics, a DRL-based algorithm is designed to efficiently solve the above problems and promote the performance of mining tasks for larger-scale BC scenarios.

2) Risk Prevention Against Cryptocurrency: Since cryptocurrencies have a high level of anonymity, they have been considered the go-to currencies for illegal activities [191]. Many efforts are proposed to prevent cryptocurrency risks.

For example, a model-based ML detection method is proposed to identify fraudulent transactions. However, the traditional supervised algorithm is not suitable for the case of label scarcity [186]. In BC networks, only a small percentage of end-users are reported as scammers, making supervised technique techniques unfeasible [192].

As a result, a list of unsupervised approaches emerges, they regard de-anonymization of cryptocurrency addresses as a clustering problem [187]–[189]. The unsupervised approaches locate anomalous behaviors in the BC by clustering objects and detecting roles. In order to improve the performance of the unsupervised approach, active learning can be leveraged to obtain some labels for achieving performance close to the optimal supervised baseline [190].

3) BC Implementation Tutorial for Incentive Tailoring: The incentive mechanism of tokens in BC encourages miners to participate in validation efforts while receiving tokens and fee rewards in return. In addition, the incentives allow token issuance and trading in an efficient, secure, and fair environment without the involvement of a central regulator. Huobi eco chain (Heco) [193] is a decentralized, energy-efficient public BC and the first product launched by the Firecoin Open Platform [194]. This section provides a detailed tutorial on issuing HRC20 tokens to support BC incentives based on Heco.

• Access to Heco BC: Before issuing Heco Token, you need to connect to MetaMask, which is a plug-in type of Ethereum wallet. Since MetaMask connects to Ethereum BC by default [195], it is necessary to connect MetaMask to the Firecoin ecosystem first.

• Set issuance parameters: The HRC20 token in the Firecoin ecosystem has four main parameters, including the token name, token symbol, number of decimal places, and the total issue amount. After setting the issuance parameters, you are ready to proceed to the next step.

• Connect MetaMask Wallet: Connect to the MetaMask wallet and get the wallet accounts, where the payment account is used to pay the token issue service fee and the token owner account will hold all the tokens issued.

• Submit issuance transaction: After confirming that the issuance parameters are correct, you can submit the issuance transaction through MetaMask for BC operation.

Summary: The researchers conceive the idea of offering a flexible mining strategy for deploying the BC system in edges. In addition, detecting the malicious behaviors of devices provides a secure end-edge environment, further preventing cryptocurrency risk. We summarize the above works of effective incentive mechanism tailoring to EI in Table VIII.
contracts can be rewritten to handle BC’s scalable business. As such, the Eth2Vec performs feature extraction from the technical difficulty of the smart contract. This tool can automatically learn features and separate features from the robustness of the smart contracts. Therefore, many researchers are absorbed in the robustness improvement of smart contracts.

In this subsection, we aim to investigate the performance improvement for smart contracts. Specifically, as shown in Fig. 18, the smart contract typically consists of a set of execution codes and states that can be identified by a specific address. The miner is then responsible for validating and storing transactions in explicit blocks, with the goal of creating a unique contract address for BC users to invoke smart contracts. Thus, it allows trusted transactions and agreements between different, anonymous, fringe parties. Although BC brings more reliable and convenient services, there are still some obstacles to smart contracts, and there are still many issues to be solved when combining EI with BC, mainly including the following.

- For the Turing complete BC system, the current smart contracts are not smart. The legal viability, flexibility, and difficulty of implementation deteriorate the performance of the smart contract, preventing the smart contract from being the basis of real-world agreements [32].
- Smart contracts lack robustness and are vulnerable to malicious parties.

Meanwhile, the extension of learning-based algorithms to security management for BC has been a promising solution. Specifically, as shown in Fig. 18, several learning-based algorithms get the utmost out of the intelligent approaches, such as distributed machine learning (DML) and convolutional neural network (CNN), to improve the performance and detect threats of the smart contract in BC, further benefiting EI.

1) Performance Improvement for Smart Contract: There are potential efficiency problems in the current smart contracts, raising serious concerns about the lightweight BC systems. In this subsection, we aim to investigate the performance improvement of the smart contract.

Robustness: Smart contracts may be subject to errors due to human fallibility. They are generally difficult to modify once they are deployed on the BC. Meanwhile, when the code in a smart contract is rewritten, its performance degrades dramatically. Therefore, many researchers are absorbed in the robustness improvement of smart contracts.

Eth2Vec [196], as an ML-based static analysis tool, owns the robustness against code rewriting of the smart contracts. This tool can automatically learn features and separate feature extraction from the technical difficulty of the smart contracts analysis by ML. As such, the Eth2Vec performs quite well even after the codes are rewritten. Although smart contracts can be rewritten to handle BC’s scalable business behaviors, they require users to write programs in a specified programming language.

Designed to be the novel generation of BC, MATRIX [197] develops an automatically generated smart contract by incorporating the BC and AI technologies. The code generator based on a DNN in MATRIX can automatically transform the script, including the core elements of the users, into an equivalent program. In this way, MATRIX can boost the transaction speed and make the smart contract more effective, further supporting the flexible operations of BC at the edge.

Furthermore, smart contracts are limited by their capabilities at the design stage. Cortex [198] provides the support for combining AI algorithms with smart contracts. Specifically, it can run both existing contracts and inferred contracts with AI. The AI models trained by AI developers are appended to the BC, while smart contracts access the AI models by paying a certain amount of token for creating different types of smart contracts. This integration makes smart contracts more diversified, generalized, and efficient, providing credible assurance for numerous intelligent services.

High Efficiency: Even though the smart contract is being exploited in various fields due to its reliability, there are still problems with its efficiency, including correctness, validity, flexibility, selection, etc.

By incorporating smart contracts, DML, and IoT-based condition monitoring, a smart logistics solution is proposed [199], where the dynamic ML-based method efficiently selects the smart contract for guaranteeing the trust, traceability, and accountability of the IoT system. Other than [199], the ML is applicable for analyzing and decreasing the validity of criminal smart contracts [200]. In this regard, a Q-learning algorithm is introduced to train data feed parameters for the low validity of criminal smart contracts, which can greatly reduce contracts’ validity and further prevent criminals’ illegal behaviors. Further, it helps the AI-enabled smart contract improve efficiency, enabling the AI applications more control and flexibility at the edges.

2) Threat Detection Against Smart Contract: Unlike traditional software programs, smart contract code cannot be changed after its deployment, giving hackers opportunities to exploit potential problems to attack smart contracts, bringing challenges to BC maintenance.

AI is a powerful tool to achieve secure contracts. Specifically, LSTM algorithms can strengthen the vulnerability detection capabilities of smart contracts, enabling BC to analyze and detect defects without requiring predefined or expert knowledge, which will be successfully applied to the edge network [201]. ML can be combined with fuzz testing for smart contract vulnerability assessment, and this incorporation can rapidly adapt to new unknown weaknesses [202].

In addition, learning vector representation (structural code embeddings) for smart contracts with the learning-based approach is useful in response to bugs and exploits created by attackers [203]. This learning-based approach can detect a known set of vulnerabilities of the smart contract, further assisting to enhance the users’ confidence in the reliability of the smart contract.
A representation of smart contract can not only semantically interrelates with each other, but also critically captures essential information [204]. With the DL-based approach, the reentrancy bugs can be precisely detected after vectorizing the representation and inputting it into the models. Different from many efforts that have applied static and dynamic analyses to find smart contract vulnerabilities, the work in [205] presents a color representation method. It focuses on translating the byte-code of the contract develop language into RGB color code, then transforming them into an encoded image. Next, the encoded image is inputted to the CNN to learn and extract the feature automatically, detecting the compiler vulnerabilities of the smart contract.

3) BC Implementation Tutorial for Smart Contract Tailoring: There are already numerous businesses exploring how they can benefit from smart contracts in an active BC solution. Ethereum is currently the most well-known and widely smart contract underlying platform in the world. Smart contracts can be seen as the backend logic of the business, and Ethereum is like an operating system. Most developers only need to focus on writing smart contracts to implement the business logic without modifying the functionality of Ethereum. Therefore, this section will concern how to deploy and invoke smart contracts on Ethereum.

- Set up the development environment: As the official client software provided by the Ethereum Foundation, Geth\textsuperscript{13} can be leveraged to build the development environment. First of all, we need to install Geth and start an Ethereum node with it.

- Write smart contracts: Smart contracts can be written through Solidity\textsuperscript{14} or operated through the local Truffle framework,\textsuperscript{15} where Remix-Solidity IDE\textsuperscript{16} is a browser-based Solidity that provides easy contract development for novices.

- Compile smart contracts: After finishing writing the smart contract, it needs to be compiled with Truffle to get the ABI and BIN files of the smart contract.

- Deploy smart contracts: Before deployment, the Ethereum network needs to be set up using Geth, while the smart contract will be specified on which Ethereum it will be deployed. After adding the deployment script, the smart contract can be deployed to the specified Ethereum private BC via Truffle.

- Invoke smart contracts: After getting the ABI and address information of a smart contract, you can use Geth to create a contract object for invocation.

\textbf{Summary:} Generally, many researchers have studied the questions of improving the robustness and safety of self mechanisms in the BC, making it easier for the BC to be used in edges. The summaries of the above works for intellectuality smart contract tailoring to EI are listed in Table IX.

### Table IX

| Benefits | Ref. | Algorithm | Contributions | The main role of EI | Smart Contract | Storage Analysis |
|----------|------|-----------|---------------|---------------------|----------------|-----------------|
| Intellectuality Improvement for smart contract | [196] | ML | Propose a static analysis tool to automatically learn features | Separating feature extraction from the technical difficulty of the smart contracts analysis | ✓ | × |
| | [197] | ML | Develop an automatically generated smart contracts by incorporating the BC and AI technologies | Automatically transformed the script into an equivalent program | ✓ | × |
| | [198] | ML | Develop a new public chain, Cortex, to provide the support for combining AI algorithms to smart contracts | Provide an access to the smart contract to enable it more diversified, generalized and efficient | ✓ | ✓ |
| | [199] | ML | Propose a smart logistics solution in the supply chain management area by incorporating smart contracts, DML, and IoT-based condition monitoring | Efficiently select the smart contracts for guaranteeing the trust, traceability and accountability of the supply chain | × | × |
| Threat detection against smart contract | [200] | RL | Leverage RL algorithms to decrease the validity of criminal smart contracts for preventing criminals’ illegal behaviors | Train data feed parameters for the low validity of criminal smart contracts | × | × |
| | [201] | LSTM | Strengthen the vulnerability detection capabilities of smart contracts | Enable BC analyze and detect defects without requiring predefined or expert knowledge | ✓ | × |
| | [202] | ML | Develop a ML classification based on the smart contract opcodes feature | Assess and detect smart contracts vulnerabilities | ✓ | × |
| | [203] | LSTM | Present a LSTM-based method for clone detection, bug detection and contracts validation on smart contracts | Learn the structural code embeddings for smart contracts | ✓ | × |
| | [204] | - | Precisely detect reentrancy bug for smart contracts | Capture essential semantic information, while controlling flow dependencies | ✓ | × |
| | [205] | DL | Present a color representation method to detect the compiler vulnerabilities of smart contracts | Learn and extract the feature of the encoded image automatically | ✓ | × |

\textsuperscript{13}https://geth.ethereum.org/ \textsuperscript{14}https://docs soliditylang.org/en/v0.8.11/ \textsuperscript{15}https://trufflesuite.com/ \textsuperscript{16}https://remix.ethereum.org/
which is essentially limited by the bandwidth and communication latency of the Internet, further directly degrading the compatibility and restricting the scalability of the BC system.

Specifically, scalability is the critical barrier that prevents BC from being used as a generic platform in EI. Currently, the existing solutions addressing scalability issues only focus on improving scalability at the expense of other important features, such as decentralization, security, and latency. None of the existing enabling technologies has all the good characteristics of BC. Generally, the scalability can be measured by the following metrics [206].

- **Throughput**: The number of transactions processed per time unit;
- **Networking**: The time for transactions to be verified.
- **Storage**: The size of BC that can be handled by the miners with limited storage capabilities.

1) **Scalability Improvement**: Currently, some researchers stress improving the scalability performance of BC systems, so as to be appropriate for EI. As shown in Fig. 19, the scalable BC system supported by AI algorithms focuses on integrating learning-based algorithms to BC in edge systems, the demands that devices request could be satisfied at a faster rate.

Meanwhile, the consensus protocol of BC is conducive to synchronizing messages of distributed edge systems. Taking the PBFT protocol, for example, the whole consensus process in the EI-based BC system includes six phases: request, admission, pre-prepare, prepare, commit, and reply. Then, the transactions validated by the PBFT can be recorded by BC for supporting the learning algorithms to analyze and optimize the performance of the BC system.

In order to improve the throughput of BC-based software-defined IIoT (SDIIoT), we formulate the trust features of BC nodes and controllers, as well as the computing offloading of the system as a joint optimization problem [207]. However, this work mainly focuses on the throughput among many scalability factors.

Currently, several attempts take into account multiple scalability factors. For example, the work in [208] jointly considers the BC throughput, the processing delay of computation tasks of BC, and operational costs as a multi-objective optimization problem. In [209], the authors consider a four-way tradeoff, including scalability, decentralization, security, and latency.

Generally, the scalability of BC can be optimized for transaction throughput. Then, other attributes, including decentralization, latency, and security, can be considered as the constraints for achieving scalability tradeoffs from multiple multiple influencing factors. Therefore, the scalability problem of BC can be described by:

\[
\begin{align*}
\max & \quad \frac{B_s}{\alpha} \\
\text{s.t.} & \quad C^1_t : G(\rho) \leq \varepsilon_s \\
& \quad C^2_t : T_{F,\delta} \leq w \times T_I, \\
& \quad C^3_t : f \leq F_\delta,
\end{align*}
\]

where \(B_s\) is the block size, \(T_I\) means the block interval, and \(\alpha\) denotes the average size of transactions. And \(G(\rho)\) is the Gini coefficient of the block producers’ stakes, \(w\) represents the block intervals. In addition, the number of malicious validators \(f\) should be limited to the maximum tolerable number of malicious validators \(F_\delta\) with the consensus algorithm \(\delta\).

Due to the high-dimensional actions or states of the scalability problem, the DRL-based algorithms are designed to address the above joint problem. Among them, the offline DNN can be used to approximate the action-value function, while the online dynamic DQN stage is leveraged for action selection and dynamic network updates. To implement the DRL algorithms, the state space, action space, and reward function should be determined. Simulation results demonstrated that the above schemes can improve the scalability of BC.

2) **BC Implementation Tutorial for Scalability Tailoring**

With scalability becoming one of the most prominent barriers to mainstream BC adoption, finding an effective BC scalability solution is inevitable. Currently, you can find different types of solutions being developed to address BC scalability issues.

The MoacChain [210], managed by the MOAC Foundation, is based on a multidimensional BC consisting of a global mother chain and its subchains. MoacChain extends the concept of subchains, which can define their own consensus methods and execution modules [211]. In addition, users and their decentralized applications (DApp) can easily migrate to the MOAC platform without the need to master additional BC technologies. This section gives a detailed tutorial on how to extend subchains based on the MoacChain.

- **Deploy subchain contracts**: DApp deployer deploys a global subchain contract on the V-node, setting up Funccode.
- **Node register**: DAPP deployer invokes RegisterOpen to allow the SCS node to register, while the V-node code is called to push an enroll message to the SCS if it detects that the connected SCS meets the requirements.
- **Participation confirmation**: SCS node initiates a call to a subchain contract to confirm participation in that subchain.
- **Deployment operation**: DAPP deployer calls RegisterClose to close the registration, and the v-node determines if the condition is met during execution.
If it does, the v-node pushes a newSubchain msg to the SCS. Otherwise, the DAPP deployer restarts the process. **Summary:** There are a lot of researches integrating the learning-based algorithms into the BC systems in edge scenario to meet the diversified requirements that devices request at a faster pace. In this way, the fear of scalability bottleneck would be further eliminated.

VI. APPLICATIONS AND TUTORIALS OF BLOCKCHAIN IN EDGE INTELLIGENCE

BC, as a disruptive technology, would be widely available in a variety of applications in EI, while strongly affecting many industries. Table X shows the comparisons of some popular BC platforms that could potentially be adopted for practical applications. Furthermore, there are a number of existing literature highlighting BC-enabled EI applications, which are shown in Fig. 20.

A. INTERNET OF VEHICLE

IoV brings smartness into the vehicular environment while enabling the advent of connected and autonomous vehicles. In fact, the integration of different technologies, such as EC, AI, and NDV, can help improve the standardized vehicle-mounted communication architecture [221]. However, IoV architecture still faces many challenges, including security, privacy, cooperation, etc. BC is considered the best way to effectively break these bottlenecks.

In [222], a number of BC-based solutions are proposed for different layers of EI, including fog layer, edge layer, static multilayer, and dynamic multilayer. In addition, this paper also discusses that the ML algorithm can improve the decision-making process of the execution activities at all layers of the IoV, where BC can further improve the security of the network. However, this survey mainly focuses on the help of the consensus algorithm in BC to the IoV, and does not pay attention to other aspects of BC features. Mollah *et al.* integrate BC and IoV to build a future intelligent transportation system (ITS) and highlight a number of BC-empowered IoV architectures, including EC, vehicular communication systems, AI, etc., [25]. This paper regards that BC can bring benefits to the IoV from seven aspects: decentralization, elimination of intermediaries, security threats, immutability, peer-to-peer trading, trust issues, transparency.
B. Smart Healthcare

Smart healthcare becomes one of the major concerns to be uplifted in every possible way with technological advancement. Especially, with the advent of BC, the complete access, transaction, and storage management taken over by the technology can promote the existing healthcare systems.

For example, a large-scale health data privacy-preserving scheme based on BC technology, named Healthchain, was proposed [223]. Particularly, the health data is encrypted to perform fine-grained access control in this scheme. As such, this introduced Healthchain can prevent both IoT data at the edges and doctor diagnoses to be deleted or tampered with. In [224], an Ethereum-based data access mechanism can resist well-known attacks along with maintaining integrity. Furthermore, there exists other BC platforms have been introduced for the smart healthcare: BlocHIE [225], MedChain [226], MedRec [227], BPDS [228] and BHEEM [229].

C. Smart Manufacturing

BC can be introduced into manufacturing IIoT to support the various operations. Especially, the work of [230] presents a novel iterative bi-level hybrid intelligence model. It is primarily based on the fusion of BC into the digital twin to eliminate unbalance/inconsistency issues and realize a large-scale personalization paradigm in the manufacturing workshop. In [231], a novel trusted platform is established to integrate BC into a cloud manufacturing system. In addition, the platform can not only establish the connection between customers and specific services, but also realize the distributed sharing of data and information. Moreover, an intelligent manufacturing system integrating EI and BC technology is proposed in [232]. The system can balance computational workload and provide a more timely response for terminal devices, while facilitating device-level data transmission and manufacturing service transactions.

D. Smart Grid

With the development of new battery energy storage technologies, a large number of consumers will evolve into prosumers using renewable energy to generate and store electricity. Meanwhile, the smart grid [233] is proposed for providing an efficient, secure, economical, and sustainable power. BC, as a decentralized database platform, enables completely new technological systems and business models for energy management [234], [235].

For example, in [236], a novel BC-based energy framework, entitled DeepCoin, is introduced. This framework incorporates a novel reliable peer-to-peer energy system for high throughput, while making the generation of blocks utilizing short signatures and hash functions for enhancing the security of the smart grid. In [237], it mainly leverages an account-mapping technique to address energy trading users’ privacy in the smart grid, to achieve an objective covering both privacy protection and trading storage. A model permissioned BC edge model for smart grid network, PBEM-SGN, combines BC and EI techniques to bridge up all entities in SGN, while introducing signatures and covert channel authorization techniques to assure users’ validity [238].

E. Next-Generation Communication Network

Driven by the recent breakthroughs in BC technology, next-generation communication networks can provide customized value to meet the increasing demands of user traffic and emerging services. Specifically, BC can realize its full potential in fog or cloud wireless access networks for dynamic access control, integrity and validity of exchanged data, and network resource synchronization [239]. In [32], an integrated system of BC and ML is proposed, providing a promising solution for realizing intelligent, safe, and decentralized data sharing as well as efficient operation of communication and network systems.

As the next important stage of global telecommunication development, the 5G wireless networks promise to bring substantial benefits to global industries [240]. Unlike traditional cellular networks, 5G wireless networks will be decentralized, ubiquitous, and service-oriented, with a particular emphasis on security and privacy requirements from a service perspective. In [26], the authors integrate BC with 5G networks and beyond to provide mobile network services as well as flexibility and security.

D2D communication, as a key technology for 5G, can be established via BC by leveraging the computing power of all participants to run the network, rather than through a third-party intermediary. Specifically, in [241], a BC-based content catching and sharing scheme for D2D networks is proposed, which realizes content sharing between mobile devices through D2D communication, and further improves the robustness and security of D2D networks.

F. Tutorials for Implementing BC in EI Applications

In general, BC can solve many problems such as “one bullet, ten birds”, while it also opens up new opportunities to empower EI services and applications, including financial, smart city, and healthcare services. Specifically, the role of BC in these applications is to provide secure sharing economy services, identify malicious actions from thousands of video frames on the large scale, deal with health data interoperability and security issues, and further improve the security and overall efficiency of EI applications. In the following section, this survey provides a tutorial to successfully deploy BC in EI applications.

1) Identify Use Case: The most important thing in deploying BC in EI applications is to identify the use case in projects. That is, identifying the problems that need to be solved in the use case and whether BC is an irreplaceable solution to those problems is the first step in deploying BC. PwC has developed a list of criteria that can help the organizations determine the role of BC in use case scenario: ① Are there multiple parties updating data? ② Are multiple parties sharing data? ③ Are there validation requirements? ④ Do intermediaries add complexity? ⑤ Do transactions interact with each other? ⑥ Are the interactions time-sensitive? If at least four of the above
criterias are met, then the BC may be a promising solution to the problem.

2) Select Available Platform: There are many BC platforms available, as shown in Table X, each of which considers itself the best in terms of scalability, security, unique features, or capabilities. As such, the appropriate BC platform should be selected depending on the system requirements of the developed EI application. Next, we compare the performance of the different platforms in depth based on the following factors:

- **BC type**: BC types differ depending on whether they are public, private, or quasi-private, i.e., operated by a consortium of stakeholders. The BC type of the use case will be case-dependent. It’s worth noting that Hyperledger Fabric is one of the most permissioned BCs [212], while Multichain is an off-the-shelf platform for creating and deploying private BC within and between organizations.

- **Popularity**: Check out the platform development activity on GitHub or other relevant communities. For example, due to the responsiveness of the Ethereum community and its rich documentation, Ethereum is the most popular platform for creating BC projects [195]. Likewise, Quorum is a customized version of Ethereum developed by financial services company JPMorgan [242]. Likewise, the R3 platform has a strong following in the financial industry [213]. However, the most effective option for businesses just starting with BC is to use BC as a service. That is, we can choose the appropriate platform to create a pre-designed BC based on customized needs.

- **Dependency**: Some BC platforms are still in their infancy and they have introduced many new programming languages, such as Java, Go, Python, Ruby, C++, and others. Before choosing a BC framework, it is important to understand which programming languages are supported by the platform’s SDK. While in comparison with most BC platforms, the Multichain platform [122] does not require learning a new programming language to develop smart contract software solutions and applications, which lets to better performance and apps.

- **Consensus protocol**: In addition to the energy and time-intensive PoW protocols, many BC platforms employ some improved protocols. For instance, EOSIO [215] and Hydrachain [218] platforms carry out a consensus with the utilization of PoS protocol, while FISCO BCOS [99] adopts PBFT protocol. It should be noted that OpenChain works on the partitioned consensus that’s why all the transactions are free of cost [217]. Moreover, Stellar [220] uses the Stellar consensus protocol, which speeds up the time it takes to process and complete transactions on public BC networks.

- **Smart contract**: Not all BC platforms support the deployment of smart contracts which are responsible for enforcing trust between participants. For instance, hyperledger offer diversified BC projects with smart contract support, such as Fabric [212], Iroha [216], Burrow [243], Sawtooth [214], Indy [244], etc, where the most notable project is Hyperledger Fabric. The Neo [219] platform leverages smart contracts to manage digital assets, which also facilitates the P2P exchange of digital assets and currencies.

- **Other features**: The BC network should be able to scale to accommodate the increase in the number of transactions and participants while focusing on the distributed storage and data management features of the platform.

3) Initialize BC: Without loss of generality, we take Ethereum as an example of how to initialize a BC. Specifically, the first block needs to be created with all the characteristics of BC. Then the blocks will be shared for all network nodes. After that, a JSON format file is created to start creating a chain block. Further, several parameters must be specified, such as Nonce (the cryptographic hash for generating random values) and Timestamp (the verification time between two consecutive blocks). After populating the JSON file, the client Geth will create the folder containing the BC and initialize it [245]. In addition, it is necessary to create the cryptocurrency, which will help to transfer data quickly, while obtaining the necessary computing power from the network participants.

4) Create Smart Contract: Similarly, take Ethereum as an example, we introduce how to deploy and run Ethereum smart contracts. Firstly, we need to create smart contracts in the target language based on the functionality we want to implement, such as transfer accounts, and on-chain data store. Then, launch the Geth and Ethereum-Wallet graphical interfaces [246] for Ethereum private BC to deploy the smart contracts. When we deploy a smart contract, the Ethereum-Wallet calls the SOLC to compile the code into EVM bytecode, and then sends the EVM bytecode through Geth’s RPC interface to the Ethereum network, which is verified across the network and written to each Geth-managed BC.

Meanwhile, we can utilize Remix, a Web-based development software, to develop and debug Solidity smart contracts. Once the contracts are tested, they can be published on Ethereum, or any BC platform that supports Solidity smart contracts. Truffle provides a set of tools [246], including Truffle, Ganache, and Mave, to help developers get started with distributed EI applications.

5) Activate BC: After completing the above steps, all that remains is to activate the EI application on the BC network. It’s worth noting that each EI application must be hosted on the main BC. For some hybrid frameworks, it is recommended that you initialize them on a cloud server, for example, applications with on-chain and off-chain entities.

6) Build Ecosystem: As many EI applications begin to be deployed in BC, it will be necessary to build an ecosystem that is a community within the larger BC community. Meanwhile, this ecosystem will help improve understanding of the BC industries and promote trust between them. Furthermore, the BC structure of the EI application needs to be carefully designed to ensure that it can easily resolve any issues with the organization while taking into account BC industry regulations and policies.

---

17https://ethereum.org/en/

18https://wallet.ethereum.org/contracts
**Summary:** In particular, we show some of the applications of merging BC into EI that can bring many benefits, including reducing the cost of trusted third parties, verifying data authenticity, protecting privacy, and ensuring security. We have carried out extensive research and have come to the conclusion that BC-enabled EI could be a game-changer across different industries. Further, we give a tutorial for implementing BC in EI applications.

VII. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Although EI has promising prospects with the help of BC, there still exist ongoing challenges that need to be considered in future research. In this section, we separately discuss open issues, including comprehensive architecture, quantization intelligence, and trading intelligence.

A. Comprehensive Architecture

Currently, emerging technologies, such as AI, 5G, EC, etc., are accelerating the computing power proliferation from cloud servers to distributed network devices. These new opportunities have given rise to the emerging computing frameworks that require ubiquitous connectivity and computing power while taking into account the evolving requirements of networking and computing [247]. As such, the computing-power network rises in response to the proper time and conditions for ubiquitous AI. It aims to connect the resources between clouds, edges, and ends through the network, to provide more flexible high-quality AI services while breaking down the island and monopoly of resources [248], [249].

However, the existing studies focus on resource utilization in an uncompensated manner. BC, as a trusted platform, can support reliable resources management and assure the paid service credibility for autonomous computing-power members [250]. Nevertheless, the complexity of the computing-power network, coupled with the introduction of BC, enables resource allocation to a daunting task. Therefore, how incorporating BC into the computing-power network to provide paid services is still a challenging issue.

B. Quantization Intelligence

The Internet of Everything era opens up new horizons for delivering ubiquitous intelligent services. Naturally, we need to consider whether intelligence is shared and exchanged by a large number of connected edge devices. According to the Big History Project [251], collective learning can store the ideas of individuals and share intelligence efficiently. Meanwhile, as the next networking paradigm, intelligent network [2] makes intelligence as easy to access and share as matter, energy and information.

Due to the outstanding advantages of BC, it can serve as a third-party platform to audit the intelligent decision-making process and provide safe and secure intelligence sharing [252]. The incentives will eliminate some of the self-interested behaviors and allow more edge users to participate in intelligent interactions. However, since each individual has a different level of understanding of intelligence, how quantifying edge intelligence will be critical to the success of intelligence sharing.

In particular, energy can be quantified as the speed of material motion. And in Shannon’s information theory, the Internet of Information uses “entropy” to quantify information, while the Turing Test does not currently have a quantitative measure of intelligence. In summary, the direction of quantization intelligence seems promising but requires a great deal of research time and effort.

C. Trading Intelligence

As the number of edge nodes increases, the collaboration between learning agents in the EI scenario will become a huge challenge. Traditional learning methods are no longer suitable for the large-scale distributed AI collaboration among devices in open and dynamic edge environments. In order to facilitate the collaboration of the learning machines, intelligence trading has been a promising solution. For example, a configurable framework for training the AI model collaboratively on the BC is proposed for models updates efficiently [150].

Nevertheless, many studies do not follow with interest the large-scale distributed collaboration mechanisms. In [253], a scalable market mechanism, i.e., learning markets (LM), is introduced for collaboration and trading. LM leverages the smart contract to encapsulate the scalable collaboration relationships, and construct a trusted database for AI models in the whole collaboration cycle. Moreover, some intelligentizing protocols regard the training process as a working puzzle [166], which can share the local intelligence via the BC system and further facilitate intelligence transactions at edges [254]. Therefore, developing distributed intelligence trading markets based on BC is desirable, and will be helpful for edge nodes to unleash the true potential of AI in the stochastic edge environments.

VIII. CONCLUSION

In this article, we provided a comprehensive examination of the integration of EI and BC, focusing on leveraging the complementary characteristics of BC to make up for the limitations of EI. To this end, we investigated the BC-driven EI and tailoring BC to EI, and further explored how to integrate BC and EI to open up new horizons for providing ubiquitous intelligence services. Specifically, we gave a detailed overview of EI and BC, along with the limitation of EI and why BC could benefit EI. We explored BC-driven EI in terms of computing-power management, data administration, and model optimization, while tailoring BC into four perspectives to narrow the gap between immature BC and EI-amicable BC. After showing some BC-driven EI applications and giving tutorials on their implementation, some of the research challenges and future directions that pertain to the implementation and improvement of BC in EI were presented.

Overall, the integration of EI and BC is still in its infancy, and there are many challenges ahead. While this paper has briefly explored the technologies associated with the BC-driven EI system at a very introductory level and discussed
future research that may benefit from pursuing this vision. We anticipate that this survey will motivate further discussions on the synergy of EI and BC, and offer some guidance in EI, BC, future networks, and other areas.

ACKNOWLEDGMENT

Especially, the authors would like to thank the editors of IEEE COMST and the reviewers for their help and support in making this work possible.

REFERENCES

[1] “IoT Connections Outlook: NB-IoT and CAT-M Technologies Will Account for Close to 45 Percent of Cellular IoT Connections in 2024.” Ericsson. 2019. [Online]. Available: https://www.ericsson.com/en/mobilityreport/reports/june-2019/iot-connections-outlook

[2] F. R. Yu, “From information networking to intelligence networking: Motivations, scenarios, and challenges,” *IEEE Netw.*, vol. 35, no. 6, pp. 209–216, Nov./Dec. 2021.

[3] B. Heintz, A. Chandra, and R. K. Sitaraman, “Optimizing grouped aggregation in geo-distributed streaming analytics,” in *Proc. 24th Int. Symp. High-Perform. Parallel Distribut. Comput.*, 2015, pp. 133–144.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proc. IEEE*, vol. 107, no. 8, pp. 1758–1762, Aug. 2019.

[6] “5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies.” [Online]. Available: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018

[7] Z. Zhao, K. M. Barjiojough, and A. Gerstlauer, “DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2384–2399, Nov. 2018.

[8] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, “Intelligent edge computing for IoT-based energy management in smart cities,” *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, Mar./Apr. 2019.

[9] S. Zhang, W. Li, Y. Wu, P. Watson, and A. Y. Zomaya, “Enabling edge intelligence for activity recognition in smart homes,” in *Proc. 15th IEEE Int. Conf. Mobile Ad Hoc Sens. Syst.*, 2018, pp. 228–236.

[10] S. Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System.” [Online]. Available: https://bitcoin.org/en/bitcoin-paper (Accessed: Dec. 30, 2021).

[11] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, “Applications of blockchains in the Internet of Things: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

[12] M. Vukolic, “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication,” in *Proc. Int. Workshop Open Problems Netw. Security IFIP WG 11.4 (INSec)*, vol. 9591, 2015, pp. 112–125.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[14] W. Yu et al., “A survey on the edge computing for the Internet of Things,” *IEEE Access*, vol. 6, pp. 6900–6919, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2778504

[15] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, “Mobile edge computing-enabled Internet of Vehicles: Toward energy-efficient scheduling,” *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep./Oct. 2019.

[16] E. Li, Z. Zhou, and X. Chen, “Edge intelligence: On-demand deep learning model co-inference with device-edge synergy,” in *Proc. Workshop Mobile Edge Commun.* (MECOM@SIGCOMM), 2018, pp. 31–36.

[17] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of edge computing and deep learning: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[18] S. Kaur, S. Chaturvedi, A. Sharma, and I. Kar, “A research survey on applications of consensus protocols in blockchain,” *IEEE Commun. Netw.*, vol. 2021, pp. 1–22, 2021.

[19] M. Du, X. Ma, Z. Zhang, X. Wang, and Q. Chen, “A review on consensus algorithm of blockchain,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, 2017, pp. 2567–2572.

[20] J. Herrera-Joancomartí and C. Pérez-Solá, “Privacy in Bitcoin transactions: New challenges from blockchain scalability solutions,” in *Proc. 13th Int. Conf. Model. Decis. Artif. Intell. (MDAI)*, vol. 9880, 2016, pp. 26–44.

[21] S. Kim, Y. Kwon, and S. Cho, “A survey of scalability solutions on blockchain,” in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2018, pp. 1204–1207.

[22] M. Conti, S. K. E, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018.

[23] J. Xie et al., “A survey of blockchain technology applied to smart cities: Research issues and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.

[24] H. Dai, Z. Zheng, and Y. Zhang, “Blockchain for Internet of Things: A survey,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[25] M. B. Mollah et al., “Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[26] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Blockchain for 5G and beyond networks: A state of the art survey,” *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102993.

[27] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Integration of blockchain and cloud of things: Architecture, applications and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2521–2549, 4th Quart., 2020.
S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, “When deep reinforcement learning meets federated learning: Intelligent multi-timescale resource management for multi-access edge computing in 5G ultra-dense network,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, Feb. 2021.

X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, “Resource allocation for blockchain-enabled distributed network function virtualization (DNFV) with mobile edge cloud (MEC),” in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops* (INFOCOM WKSHPS), 2019, pp. 1–6.

F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, “Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1689–1703, Mar. 2020.

L. Xiao et al., “A reinforcement learning and blockchain-based trust mechanism for edge networks,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5460–5470, Sep. 2020.

M. Li, F. R. Yu, P. Si, W. Wu, and Y. Zhang, “Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: A deep reinforcement learning approach,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9399–9412, Oct. 2020.

S. Guo, Y. Dai, S. Xu, X. Qiu, and F. Qi, “Trusted cloud-edge network resource management: DRL-driven service function chain orchestration for IoT,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6010–6022, Jul. 2020.

Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, and Z. Ming, “Blockchain-based edge computing resource allocation in IoT: A deep reinforcement learning approach,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2226–2237, Feb. 2021.

J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, “Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6214–6228, Jul. 2020.

X. Zheng, M. Li, Y. Chen, J. Guo, M. Alam, and W. Hu, “Blockchain-based secure computation offloading in vehicular networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4073–4087, Jul. 2021.

H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, “Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, Jul. 2021.

Z. Li, M. Xu, J. Nie, J. Kang, W. Chen, and S. Xie, “NOMA-enabled cooperative computation offloading for blockchain-empowered Internet of Things: A learning approach,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2364–2378, Feb. 2021.

L. Yang, M. Li, P. Si, R. Yang, E. Sun, and Y. Zhang, “Energy-efficient resource allocation for blockchain-enabled Industrial Internet of Things with deep reinforcement learning,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2318–2329, Feb. 2021.

X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, “Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

F. Li, H. Yao, J. Du, C. Jiang, Z. Han, and Y. Liu, “Auction design for edge computing offloading in SDN-based ultra dense networks,” *IEEE Trans. Mob. Comput.*, vol. 21, no. 5, pp. 1580–1595, May 2022.

F. R. Yu, J. Liu, Y. He, P. Si, and Y. Zhang, “Virtualization for distributed ledger technology (vDLT),” *IEEE Access*, vol. 6, pp. 25019–25028, 2018.

C. Qiu, X. Wang, H. Yao, Z. Xiong, F. R. Yu, and V. C. M. Leung, “Bring intelligence among edges: A blockchain-assisted edge intelligence approach,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

“Enterprise Blockchain.” [Online]. Available: https://www.multichain.com/ (Accessed: Jan. 30, 2022).

S. C. Oh, M. S. Kim, Y. Park, G. T. Roh, and C. W. Lee, “Implementation of blockchain-based energy trading system,” *Asia Internet Innov. Entrepreneurship*, vol. 11, no. 3, pp. 322–334, 2017.

M. G. Xevgenis, D. G. Kogias, P. Karkazis, H. C. Leligou, and C. Z. Patrikakis, “Application of blockchain technology in dynamic resource management of next generation networks,” *Information*, vol. 11, no. 12, p. 570, 2020.

K. M. Venkateswarlu, S. Avizheh, and R. Safavi-Naini, “A blockchain-based approach to resource sharing in smart Neighbourhoods,” in *Proc. Financial Cryptography Data Security Int. Workshops, AsiaUSEC, CoDeFi, VOTING, WTSCT, Kota Kinabalu, Malaysia, Revised Select. Papers*, vol. 12063, 2020, pp. 550–567.

X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, “Blockchain-based deep reinforcement learning for Internet of Things with decentralized cooperative edge caching,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9441–9455, Oct. 2020.

Z. Ning et al., “Deep learning in edge of vehicles: Exploring transactionality for data transmission,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 10, pp. 5737–5746, Oct. 2019.

F. Wang et al., “Dynamic distributed multi-path aided load balancing for the global data center networks,” *IEEE Trans. Netw. Serv. Manage.*, vol. 19, no. 2, pp. 991–1005, Jun. 2022.

X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, “Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 12, pp. 6367–6378, Dec. 2019.

H. Chai, S. Leng, Y. Chen, and K. Zhang, “A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3975–3986, Jul. 2021.
[226] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, “MedChain: A design of blockchain-based system for medical records access and permissions management,” IEEE Access, vol. 7, pp. 164595–164613, 2019.

[227] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “MedRec: Using blockchain for medical data access and permission management,” in Proc. 2nd Int. Conf. Big Data (BigDataCon), 2016, pp. 25–30.

[228] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, “BPDs: A blockchain based privacy-preserving data sharing for electronic medical records,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), 2018, pp. 1–6.

[229] J. Vora et al., “BHEEM: A blockchain-based framework for securing electronic health records,” in Proc. IEEE Globecom Workshops, 2018, pp. 1–6.

[230] J. Leng et al., “ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 50, no. 1, pp. 182–192, Jan. 2020.

[231] Z. Li, A. V. Barenji, and G. Q. Huang, “Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform,” Robot. Comput. Integ. Manuf., vol. 54, pp. 133–144, Dec. 2018.

[232] H. Cui, Z. Chen, N. Liu, and B. Xia, “Blockchain-driven contents records,” in Proc. 2nd Int. Conf. Open Big Data (OBD), 2016, pp. 25–30.

[233] J. Kang, R. Yu, X. Huang, Y. Zhang, and E. Hossain, “Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains,” IEEE Trans. Ind. Informat., vol. 13, no. 6, pp. 3154–3164, Dec. 2017.

[234] M. A. Ferrag and L. A. Maglaras, “DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids,” IEEE Trans. Eng. Manag., vol. 67, no. 4, pp. 1285–1297, Nov. 2020.

[235] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, “Privacy-preserving energy trading using consortium blockchain in smart grid,” IEEE Trans. Ind. Informat., vol. 15, no. 6, pp. 3548–3558, Jun. 2019.

[236] Y. Wang, J. Wang, X. Shen, and M. Qiu, “Permitted blockchain and edge computing empowered privacy-preserving smart grid networks,” IEEE Internet Things J., vol. 6, no. 5, pp. 7992–8004, Oct. 2019.

[237] P.-H. Kuo, A. Mourad, and J. Ahn, “Potential applicability of distributed ledger to wireless networking technologies,” IEEE Wireless Commun., vol. 25, no. 4, pp. 4–6, Aug. 2018.

[238] N. Parwar, S. Sharma, and A. K. Singh, “A survey on 5G: The next generation of mobile communication,” Phys. Commun., vol. 18, pp. 64–84, Mar. 2016.

[239] H. Cui, Z. Chen, N. Liu, and B. Xia, “Blockchain-driven contents sharing strategy for wireless cache-enabled D2D networks,” in Proc. 17th IEEE Int. Conf. Commun. Workshops (ICC), 2019, pp. 1–5.

[240] “Build on Quorum, The Complete Open Source Blockchain Platform for Business.” [Online]. Available: https://cnsys.com/ (Accessed: Dec. 30, 2021).

[241] “Burrow—The Boring Blockchain.” 2019. [Online]. Available: https://www.hyperledger.org/blog/2019/10/08/burrow-the-boring-blockchain [Hyperledger Indy]. [Online]. Available: https://www.hyperledger.org/use/hyperledger-indy (Accessed: Jan. 30, 2022).

[242] P. V. Klaine, L. Zhang, and M. A. Imran, “An implementation of a blockchain-based data marketplace using Geth,” in Proc. 3rd Conf. Blockchain Res. Appl. Innov. Netw. Serv. (BRAINS), 2021, pp. 15–16.

[243] R. M. A. Latif, M. Farhan, O. Rizwan, M. Hussain, S. Jabbar, and S. Khalid, “Retail level blockchain transformation for product supply chain using truffle development platform,” Clust. Comput., vol. 24, no. 1, pp. 1–16, 2021.

[244] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, “Compute first networking: Distributed computing meets ICN,” in Proc. 6th ACM Int. Conf. Cloud Computing (CLOUD), 2019, pp. 67–77.

[245] A. U. G. Dennis, N. Naresh, and S. Schildt, “Towards in-network computing infrastructures for connected vehicles,” in Proc. 2 KaV/S Fachgespräch Netw. Softw., 2019, pp. 67–77.

[246] S. Huang et al., “Intelligent Eco networking (IEN) III: A shared in-network computing infrastructure towards future Internet,” in Proc. 3rd Int. Conf. Hot Inf. Centric Netw. (HotICN), 2020, pp. 47–52.

[247] X. Wang, X. Ren, C. Qiu, Y. Cao, T. Tuleb, and V. C. M. Leung, “Net-in-AI: A computing-power networking framework with adaptability, flexibility, and profitability for ubiquitous AI,” IEEE Netw., vol. 35, no. 1, pp. 280–288, Jan./Feb. 2021.

[248] “The Big History Project.” [Online]. Available: https://www.bighistoryproject.com (Accessed: Jan. 30, 2022).

[249] F. R. Yu, “From the Internet of information to the Internet of intelligence,” 2019, arXiv:909.08068.

[250] L. Ouyang, Y. Yuan, and F.-Y. Wang, “Learning markets: An AI collaboration framework based on blockchain and smart contracts,” IEEE Internet Things J., early access, Oct. 21, 2020, doi: 10.1109/JIOT.2020.3032706.

[251] H. Xu and B. Jiang, “Study on a security intelligence trading platform based on blockchain and IPFS,” J. Comput. Virok. Hacking Techn., vol. 17, no. 2, pp. 131–137, 2021.

Xiaofoi Wang (Senior Member, IEEE) received the master’s and Doctoral degrees from Seoul National University from 2006 to 2013. He is currently a Professor with the College of Intelligence and Computing, Tianjin University, China. He is also with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, China. He was a Postdoctoral Fellow with The University of British Columbia from 2014 to 2016. Focusing on the research of social-aware cloud computing, cooperative cell caching, and mobile traffic offloading, he has authored over 140 technical papers in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS MAGAZINE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE INFOCOM, and the IEEE SECON. In 2017, he received the “Fred W. Ellersick Prize” from the IEEE Communication Society.

Xiaoxu Ren (Student Member, IEEE) received the B.S. degree from the College of Science, Inner Mongolia University of Technology, China, in 2016. She is currently pursuing the Ph.D. degree with the College of Intelligence and Digital Economy (SZ), Shenzhen University, Tianjin, China. Her current research interests include machine learning, computing power networking, edge intelligence, and blockchain.

Chao Qiu (Member, IEEE) received the B.S. degree in communication engineering from China Agricultural University in 2013, and the Ph.D. degree in information and communication engineering from Beijing University of Posts and Telecommunications in 2019. She is currently a Lecturer with the College of Intelligence and Computing, Tianjin University, China. She is also with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University. 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.
Zehui Xiong (Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore. He is currently an Assistant Professor with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design. He is also an Honorary Adjunct Senior Research Scientist with Alibaba-NTU Singapore Joint Research Institute. He was the Visiting Scholar with Princeton University and University of Waterloo. He has published more than 150 research papers in leading journals and flagship conferences and many of them are ESI Highly Cited Papers. His research interests include wireless communications, Internet of Things, blockchain, edge intelligence, and Metaverse. He has won over 10 Best Paper Awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He is the recipient of IEEE Early Career Researcher Award for Excellence in Scalable Computing, the IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, the IEEE Internet Technical Committee Early Achievement Award, the IEEE Best Land Transportation Paper Award, the IEEE CSIM Technical Committee Best Journal Paper Award, the IEEE SPCC Technical Committee Best Paper Award, the IEEE VTS Singapore Best Paper Award, the Chinese Government Award for Outstanding Students Abroad, and the NTU SCSE Best Ph.D. Thesis Runner-Up Award. He is currently serving as the Editor or a Guest Editor for many leading journals including IEEE Journal on Selected Areas in Communications, IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, IEEE Transactions on Cognitive Communications and Networking, IEEE Transactions on Network Science and Engineering, IEEE Systems Journal, and IEEE/CAA Journal of Automatica Sinica. He is serving as the Associate Director of Future Communications R&D Programme, Singapore.

Haipeng Yao (Senior Member, IEEE) received the Ph.D. degree from the Department of Telecommunication Engineering, University of Beijing University of Posts and Telecommunications in 2011, where he is a Professor. He has published more than 100 papers in prestigious peer-reviewed journals and conferences. His research interests include future network architecture, network artificial intelligence, networking, space-terrestrial integrated network, network resource allocation and dedicated networks. He has served as an Editor of IEEE Network, IEEE Access, and a Guest Editor of IEEE Open Journal of the Computer Society and Journal of Network and Systems Management (Springer). He has also served as a member of the technical program committee as well as the Symposium Chair for a number of international conferences, including the IWCMC 2019 Symposium Chair and the ACM TUR-C SIGSAC2020 Publication Chair.

Victor C. M. Leung (Life Fellow, IEEE) is a Distinguished Professor of Computer Science and Software Engineering with Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, University of British Columbia (UBC), Canada. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. He received the 1977 APEBC Gold Medal, the 1977–1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, the 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and the 2018 ACM MSWiM Reginald Fessenden Award. He coauthored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE SYSTEMS JOURNAL Best Paper Award, the 2018 IEEE CSIM Best Journal Paper Award, and the 2019 IEEE TCGCC Best Journal Paper Award. He is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Transactions on Computational Social Systems, IEEE Access, IEEE Network, and several other journals. He is a Fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of “Highly Cited Researchers.”