IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 May 2024, accepted 12 May 2024, date of publication 14 May 2024, date of current version 21 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401076

RESEARCH ARTICLE

Blockchain-Empowered Metaverse: Decentralized Crowdsourcing and Marketplace for Trading Machine Learning Data and Models

HUNG DUY LE, VU TUAN TRUONG, AND LONG BAO LE, (Fellow, IEEE)
Institut national de la recherche scientifique (INRS), University of Quebec, Montreal, QC H5A 1K6, Canada

Corresponding author: Long Bao Le (long.le@inrs.ca)

ABSTRACT The Metaverse relies on advanced machine learning (ML) techniques to facilitate the seamless mapping between the virtual and physical realms. ML-based technologies also enable metaverse service providers (MSPs) to offer a diverse range of intelligent virtual services to metaverse users (MUs). However, it can be challenging for MSPs to collect sufficient metaverse data to train ML models by themselves. As a result, MSPs can be interested in seeking contributions from MUs in both ML data and models. To address these challenges, we propose MetaAICM, a blockchain-based framework that empowers the metaverse through two key components. Firstly, it incorporates a distributed crowdsourcing system that allows MSPs to gather metaverse data and ML models from MUs. Secondly, it features a decentralized marketplace, enabling MUs to proactively collect data and train ML models for sale using their metaverse devices and computing resources. MetaAICM leverages blockchain and smart contracts to achieve decentralization, ensuring security and privacy without relying on a trusted third-party authority or additional trust assumptions between MUs and MSPs. Numerical studies show that MetaAICM offers high processing performance and cost efficiency, while the framework is implemented on top of a consortium blockchain to show its feasibility.

INDEX TERMS Metaverse, blockchain, crowdsourcing, machine learning, marketplace, decentralized application.

I. INTRODUCTION

The metaverse, envisioned as the next-generation Internet, is an immersive digital representation of the physical world where users can participate via augmented/virtual reality (AR/VR) devices [1], [2], [3] and wearable haptic devices [4]. It enables individuals to engage in a wide range of virtual activities such as learning, gaming, working, and socializing [5]. Among various advanced technologies, artificial intelligence (AI) plays a pivotal role in enabling this virtual realm [6], [7]. For example, computer vision (CV) algorithms leverage real-world data captured by Internet-of-Things (IoT) devices and unmanned aerial vehicles (UAVs) to construct the metaverse's virtual environment. Additionally, machine learning (ML) models deployed on wearable devices analyze user data, such as appearance, gestures, and facial expressions, to reflect user behaviors into avatar actions [8]. While metaverse publishers typically handle these tasks, metaverse service providers (MSPs) also require ML models to deliver virtual services to metaverse users (MUs). For example, MSPs can utilize generative models to create AI-generated content like metaverse virtual items, clothing, and decorations for the virtual environment [9]. Another example is related to virtual events where event organizers can employ detection models to identify unauthorized MUs accessing the virtual space by analyzing their profile such as reputation, location, and operational history.

MSPs could be interested in seeking and leveraging the expertise in AI/ML and computational resources from the metaverse community to construct and train their desired ML models. On the other hand, ML professionals capable of designing and training ML models may face challenges

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Vecchio.

in collecting the required data due to privacy concerns and the large-scale and heterogeneous nature of the metaverse. As a result, it is mandatory to provide MUs and MSPs with a decentralized platform that allows them to exchange both metaverse data and ML models in a trustless distributed environment with low trading fees, high performance, and privacy preservation.

Traditional trading and crowdsourcing systems offered by third-party authorities are often vulnerable to a single point of failure (SPoF) and trust issues, free-riding and false-reporting attacks among participants, in addition to the lack of transparency, privacy, and incentive mechanisms [10]. Although these limitations might be acceptable for certain conventional centralized platforms, they should be eliminated to fit with the decentralized metaverse where scams and frauds are exacerbated by various novel social engineering attacks [11]. To this end, blockchain is a potential solution for trust management in such metaverse trading systems with immutable, transparent, and auditable properties [12]. In particular, blockchain provides a reliable storage environment to record trading transactions, while smart contracts and blockchain's committee can replace third-party authorities in decision-making, policy enforcement, reputation, and incentive management [13].

Motivated by the aforementioned urgent needs, this paper proposes MetaAICM, a blockchain-empowered framework that enables AI crowdsourcing and provides the marketplace to facilitate intelligent virtual services in the metaverse. In MetaAICM, MUs can leverage their devices' capability and available resources to collect data or train ML models for specific purposes, then list them on MetaAICM's marketplace for sale. On the other hand, if MSPs cannot find the expected data/models on the marketplace, they can initialize a crowdsourcing task on MetaAICM's decentralized applications to obtain the desired products from other MUs. MetaAICM ensures that payment is only finalized when the traded product is valid, while both parties do not need to trust each other. To address scalability, transaction speed, and sustainability, MetaAICM incorporates a reputation-based Raft consensus protocol.

A. RELATED WORK AND RESEARCH GAPS

In any crowdsourcing and trading systems, it is important to evaluate the contribution of each worker, thereby distributing the award fairly to honest workers and punishing malicious workers who submitted low-quality results. Most existing frameworks assume that there exists an evaluation function that can, somehow, assess the quality of the submitted data/models automatically. Unfortunately, assessing the quality of data is not a simple task. Therefore, the novel contribution of MetaAICM compared to existing works is that it enables a practical solution for evaluating the submitted data/models. With a novel decentralized design, MetaAICM does not require the assumption of such an evaluation function, while also resolves various security risks such as privacy leakage, false-reporting, and free-riding attacks.

1) BLOCKCHAIN-BASED DATA MARKETPLACE

Prior to our work, several studies explored the applications of blockchain for data trading and data marketplace. For instance, the authors in [14] designed a blockchain-based market for IoT data trading with three different trading modes, namely general trading, selling on demand, and buying on demand. All of these functions are regulated by smart contracts, thereby eliminating SPoF and the intervention of the third-party middleman. However, this framework does not take into account the situation in which the sellers act dishonestly by committing incorrect data. In such cases, the buyers will lose their funds while only receiving low-quality data. Furthermore, without proper encryption techniques, the traded data can be leaked widely due to the transparent property of blockchain. To address the privacy issue, the authors in [15] take a novel approach to data trading, in which the sellers are not required to send the original data to the buyers. Instead, a buyer only purchases the processed analysis results of the seller's dataset. The analysis process is carried out by certain trusted nodes that use the trusted hardware called Software Guard Extensions (SGX) to execute smart contract functions. However, the smart contract is only suitable for relatively simple tasks due to its resource limitation and complexity, thus limiting the framework's capacity.

Regarding the marketplace, the authors in [16] implemented a decentralized IoT data marketplace, which is based on blockchain to improve fairness and trust management. This framework supplements a security manager layer consisting of multiple storage operators who take responsibility for data storage. In addition, the traded data are encrypted before being sent to the storage operators to ensure privacy. However, if the buyers claim that they obtained a low-quality dataset after the decryption, it is almost infeasible to determine which party (i.e., the buyer, seller, or storage operator) was malicious. The authors in [17], proposed a solution to data-trading center, which has the ability to retain data by taking advantage of blockchain and ML. In addition, it employs a similarity-learning technique to verify data availability, while introducing a new role called ''arbitration institution'' to resolve any controversy between the buyers and sellers. However, relying on such the trusted third-party may cause the single-point-of-failure (SPoF) if it acts maliciously or being under attacks.

2) BLOCKCHAIN FOR CROWDSOURCING

There have been several existing works that employ blockchain technology to engineer different crowdsourcing frameworks. In particular, the authors in [18] proposed a blockchain-based crowdsourcing design named CrowdBC. In CrowdBC, different smart contracts are deployed to regulate various tasks, thereby limiting human interactions and trust issues. The framework can resist malicious requesters, workers, and miners. However, CrowdBC requires an assumption that the requesters must be able to provide an evaluation function for the requested task on smart contract

TABLE 1. Comparison between MetaAICM and existing frameworks.

Feature	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	MetaAICM
Practical contribution evaluation									✓
SPoF & manipulation resistance	✓	✓	✓		✓			✓	✓
False-reporting attack		✓		✓	✓	✓		✓	✓
Free-riding attack		✓	✓	✓	✓	✓	✓		✓
Sybil attack	✓	✓	✓		✓	✓	✓	✓	✓
Privacy leakage		✓	✓			✓	✓	✓	✓
Incentive mechanism		✓	✓	✓	✓	✓		✓	✓
Reputation system			✓		✓			✓	✓

codes to assess the contribution of workers. This assumption would be difficult to realize in practice since the requesters are often just unprofessional clients who cannot write smart contracts by themselves, while sophisticated tasks such as AI/ML are even more challenging to be implemented on smart contracts due to their complexity and resource demand. The authors in [20] proposed zkCrowd, a blockchain-enabled crowdsourcing platform emphasizing user privacy with the integration of zero-knowledge proof [22]. In zkCrowd, both public and private blockchains are leveraged to enable flexible adjustment of the privacy degree corresponding to the user's demand. However, zkCrowd also requires a reward distribution function that is capable of evaluating the contribution of workers and correspondingly distribute rewards to the workers, which would be difficult to realize in most practical use cases. On the other hand, several studies such as ZebraLancer [19] and BPCM [21] focus on resolving privacy issues in crowdsourcing systems. Although the frameworks can enable privacy preservation by hiding the true identities of participants, the problem regarding contribution evaluation remains unsolved.

3) BLOCKCHAIN-BASED METAVERSE APPLICATIONS

Blockchain technology has been leveraged to develop various metaverse applications [12], [23], [24]. The authors in [25] proposed using smart contracts to efficiently manage and automate the interaction between MSPs and MUs. This blockchain-based design allows MSPs to optimize resource allocation when offering virtual services and applications to MUs and encourages MUs to contribute their resource to support the operation of the metaverse thanks to an incentive mechanism based on Stackelberg game theory. The authors in [26] proposed a blockchain-based federated learning (FL) framework to enable the industrial metaverse. By leveraging blockchain, the system can offer two privacy options for FL in the physical and virtual spaces using various sub-blockchains, which communicate with each other by using cross-chain communication techniques. On the other hand, the authors in [27] constructed a campus-oriented prototype of the metaverse, in which blockchain is leveraged to enable the platform's economic system. For instance, the blockchain-based non-fungible token is used to represent virtual items and user-generated content. Nonetheless, the use of blockchain to enable metaverse AI/ML based services such as ML crowdsourcing or decentralized marketplace for ML models and metaverse data has not been investigated in these studies.

Comparison of our proposed design and existing frameworks is given in Table. 1. It can be seen that the proposed MetaAICM framework addresses the contribution evaluation issue and it tackles many important security risks and attacks, which have not been handled adequately by other related works.

The preliminary results of this work are presented in a conference paper where we described a decentralized ML crowdsourcing system for the metaverse, which solved the problem of contribution verification and reward distribution by using committee-based model validation and a blockchain oracle network. This paper is an extension of the mentioned work, in which a decentralized marketplace is integrated to enable both selling and buying on demand. Furthermore, intensive experiments are supplemented to illustrate the full potential of MetaAICM.

B. KEY CONTRIBUTIONS AND PAPER STRUCTURE

To fill the research gaps presented above, this paper proposes MetaAICM, a blockchain-empowered framework that enables ML crowdsourcing and marketplace to facilitate intelligent services in the metaverse. The novel contributions of MetaAICM can be summarized as follows:

- Our MetaAICM framework enables a blockchain-based decentralized marketplace that encourages MUs to proactively collect metaverse data or train ML models for sale. If the desired data and models are not available on the marketplace, the crowdsourcing mode allows MSPs to crowdsource ML models from machine learning workers (MLWs), while MLWs can also request metaverse data from data workers (DWs).
- Unlike existing blockchain-enabled frameworks, MetaAICM does not require task requesters to provide an evaluation function that can automatically assess the contribution of workers. Thanks to our decentralized design, other trust assumptions are also eliminated and the involved participants do not have to trust each other. A concrete incentive mechanism is designed that motivates MUs to contribute computational resources

to empower metaverse ML services, while a reputation system is integrated to filter out malicious entities who act dishonestly.

- We show that MetaAICM can resist various security threats such as denial of service (DoS), SPoF, data leakage, and Sybil attacks. In addition, we present extensive numerical results to confirm that MetaAICM offers high performance and automation with low operation costs.

The rest of this paper is structured as follows. Section III presents the system model and design overview of MetaAICM. Section IV proposes the detailed architecture and operation of MetaAICM with security analysis. Section V describe the numerical results for MetaAICM and Section VI concludes the paper.

II. PRELIMINARIES

A. BLOCKCHAIN

Blockchain is a decentralized digital ledger that records transactions across many computers in such a manner that the registered transactions cannot be altered retroactively. This technology ensures the integrity and transparency of data without the need for a central authority or intermediary. At its core, a blockchain consists of multiple blocks linked together as a chain, where each block contains a certain number of transactions. These transactions are verified by participants in the network, known as nodes, through a consensus mechanism, ensuring security and mutual agreement on the state of the ledger at any given time. The immutable nature of blockchain technology not only enhances security but also builds trust among users, making it a foundational technology for various applications in the metaverse [24], such as security [23], digital asset management [28], [29], networking [30], and distributed learning [31].

1) BLOCKCHAIN TYPES

In terms of accessibility, there are two main types of blockchain, including permissionless and permissioned blockchains [32]. Permissionless blockchain is sometimes referred to as public blockchain, which is open to the public and anyone can participate, read, write, or audit the blockchain without requiring explicit permission. Bitcoin [33] and Ethereum [34] are prominent examples of public blockchains, characterized by their transparency, security, and immutability. On the other hand, if a blockchain belongs to the permissioned category, its participants have to obtain explicit permission to join the network, which can be granted by a central authority or by existing participants. Permissioned blockchain includes private and consortium blockchains. In a private blockchain, there is a single authority controlling the network, while a consortium blockchain is often regulated by multiple authorities.

Permissionless blockchains provide the highest extent of transparency and openness, as their data are available on the Internet and anyone can download and audit the information. On the other hand, permissioned blockchains prefer privacy and controllability over transparency. Depending on specific use cases and system requirements, a suitable type of blockchain could be chosen accordingly.

2) CONSENSUS MECHANISM

Blockchain consensus refers to the mechanism by which all participants reach an agreement on the state of the network without the need for a central authority. Various consensus algorithms have been developed and used in different blockchain networks, each with its own set of advantages and trade-offs. Some of the most common consensus mechanisms include Proof of Work (PoW) [35], [36], Proof of Stake (PoS) [37], Raft [38], practical Byzantine Fault Tolerance (pBFT) [39], [40], and Proof of Authority (PoA) [41].

Consensus algorithms play a key role in deciding the characteristics of a blockchain network, including scalability, decentralization, and security. Furthermore, each consensus mechanism takes a different approach to offer fault tolerance capability. For example, pBFT tackles the byzantine problem via a three-stage voting mechanism, while PoA filters out malicious nodes by only selecting reputable nodes to conduct the consensus process. On the other hand, certain algorithms like Raft and Paxos offer crash fault tolerance instead of byzantine fault tolerance [38]. It is also possible to improve the security of Raft and Paxos by integrating additional techniques for selecting consensus nodes (e.g., techniques that are based on the amount of stake or reputation scores). Our design uses the reputation score for eliminating malicious nodes from the consensus, while the consensus process is crash fault-tolerant, which is the key characteristic of the Raft-based protocol.

3) DECENTRALIZATION AND SCALABILITY

In any blockchain network, there is an inevitable trade-off between decentralization and scalability. A highly decentralized network like Bitcoin often suffers from low processing speed because it takes more time to reach consensus among numerous nodes due to networking delay and synchronization problems. In contrast, semi-decentralized blockchain systems usually offers higher scalability as they allow to reach consensus more quickly, but they are usually prone to manipulation and single point of failure (SPoF).

Several existing blockchains attempt to achieve a balance between these two aspects by taking a semi-decentralized approach. Specifically, they first select a limited number of consensus nodes to form a committee. Then, a consensus protocol (e.g., pBFT) is carried out among the committee members instead of the entire network like Bitcoin's PoW, resulting in a significantly higher throughput compared to the fully decentralized approach. In terms of electing nodes for the committee, we can assess the reliability of each node based on different factors such as their amount of stake (i.e., PoS) or reputation score. Our work is also a

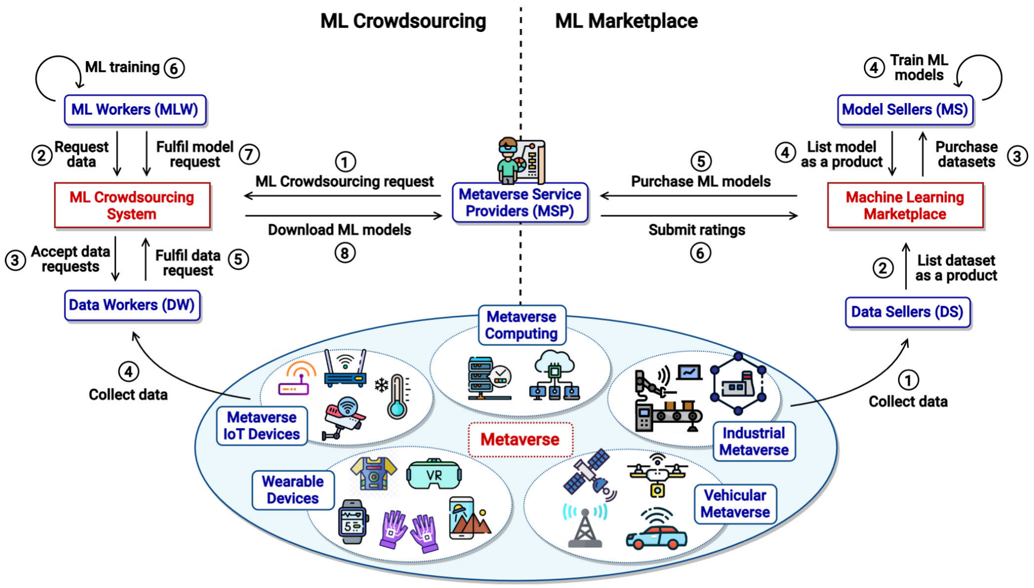


FIGURE 1. Overview of the metaverse and the architecture of MetaAICM with two operation modes, namely crowdsourcing system and decentralized marketplace.

semi-decentralized design, which offers a balance between decentralization and scalability.

We would like to note that various research studies have investigated different methods to improve scalability without sacrificing decentralization [42]. Some potential techniques are sharding [43] and various variants of layer-2 solutions such as rollups, sidechains, and state channels. However, each of the methods has its own limitations and design challenges. For example, sharding might decrease the decentralization and security of each shard, while the bridges from layer-2 blockchains to their main chains are prone to various attacks.

B. SMART CONTRACTS

Smart contracts have significantly expanded the utility of blockchain technology, extending its use beyond mere transactional functions. Essentially, a smart contract functions like a computer program embedded with predefined rules and logic. Its code is executed across multiple nodes within the blockchain network through a consensus mechanism. This execution process ensures that outcomes are determined by the majority consensus, safeguarding against manipulation as long as most nodes operate honestly. The transparency of smart contracts, with both their code and outcomes verifiable on-chain, bolsters trust and verification. However, this same transparency poses challenges, particularly for processing sensitive information like private keys or credentials, which become visible on the blockchain and accessible to all nodes. Several blockchain platforms now facilitate smart contract deployment, notably Ethereum [34] and Hyperledger [44]. These platforms are engineered to execute smart contracts reliably, without the risks of fraud, downtime, or interference from third parties.

C. IPFS DATA STORAGE

The InterPlanetary File System (IPFS) [45] is a peer-to-peer distributed file system. By dispersing file storage across a network of nodes, IPFS enhances data redundancy and resilience, making it robust against loss. It uniquely identifies files and their constituent blocks through cryptographic hashes, which not only secures data integrity but also streamlines data retrieval by fetching content from the nearest node rather than a central server. This decentralized approach significantly accelerates access speeds and decreases bandwidth demands. IPFS finds critical use in applications requiring data integrity and availability, notably in blockchain-based decentralized applications (DApps). Here, while the blockchain stores only the references to data in the form of IPFS URLs or hashes, the actual data resides on IPFS. This methodology drastically cuts storage costs and amplifies data accessibility.

III. SYSTEM MODEL AND DESIGN OVERVIEW

A. SYSTEM MODEL AND DESIGN GOALS

As illustrated in Fig. 1, the metaverse is envisioned to leverage numerous devices (e.g., IoT sensors, smart vehicles, industrial devices, wearable AR/VR and haptic devices) and computational infrastructure (e.g., edge/cloud servers and data centers) for its operation. Specifically, these devices continuously collect and generate a massive amount of data, which is especially valuable for AI/ML-enabled tools and services [46]. However, these devices and the collected data are often owned by numerous distributed MUs instead of a centralized publisher like traditional media platforms. Therefore, it poses significant challenges in data collection including privacy issues [47], trust management, and incentives for the data collectors. For instance, MUs may not want to disclose their data widely to the public (i.e., privacy issue), while malicious participants might try to obtain the data without paying tokens (i.e., trust issue).

Our design of the crowdsourcing system and marketplace aims to achieve the following goals:

- To engineer decentralized frameworks allowing metaverse data and ML models to be traded or crowdsourced for intelligent virtual services in the digital world.
- To effectively tackle the prevalent issues of security, privacy, and trust management that are inherent in a trustless environment and remain unaddressed thoroughly.
- To make sure that the proposed designs can achieve desired levels of scalability, decentralization, and security while being sustainable (i.e., it does not require a significant amount of energy for its operations as in the Bitcoin blockchain).

To this end, our proposed designs rely on a consortium blockchain using a Raft-based consensus mechanism as we will explain in more detail in the next section. In the following, the term ‘‘blockchain committee'' is used to refer to a group of blockchain consensus nodes.

B. DESIGN OVERVIEW

In this paper, we propose both crowdsourcing and marketplace designs for different application scenarios as we will highlight in the following.

1) OVERVIEW OF CROWDSOURCING DESIGN

In crowdsourcing systems, data collectors are called data workers (DWs), while ML professionals who offer ML training services are ML workers (MLWs). There are three main smart contracts regulating the crowdsourcing process in our proposed framework as follows.

- Machine Learning contract (MLC): It allows MSPs to crowdsource ML models from MLWs.
- Data contract (DC): This smart contract enables MLWs to crowdsource metaverse data from DWs.
- Reputation contract (RC): It manages the reputation profiles of participants and can be considered as a reference indicating the reliability of individuals.

The crowdsourcing mode's operation is illustrated in the left side of Fig. 1 with a total of 8 steps. Specifically, to crowdsource a ML model from MUs, a MSP$_i$ first submits a ML crowdsourcing request to the smart contract MLC (step 1). The MSP$_i$ also deposit to MLC a certain number of metaverse tokens as a reward for the workers. Then, interested MWs can accept the request from MLC and start training the requested model. During the training process, the MWs can further request more metaverse data from DWs if needed. To do so, a MW$_j$ must commit a data request to the smart contract DC (step 2), which triggers a new data crowdsourcing task. After some DWs collected data and fulfilled the data request (steps 4 and 5), the MW$_j$ can use the obtained data to train its model (step 6). Finally, once finishing the ML crowdsourcing task (step 7), the MSP$_i$ will download the final ML model from MLC, and the deposited tokens are distributed automatically to honest workers.

2) OVERVIEW OF MARKETPLACE

Instead of waiting for specific crowdsourcing requests, MUs can also proactively collect metaverse data to build their own datasets and list them on the marketplace for sale. These MUs are called data sellers (DSs). Similarly, certain model sellers (MSs) can design/train ML models which meet the metaverse's demand, then sell them on the marketplace with a predefined price and descriptions. The marketplace's backend is constructed on top of a smart contract called marketplace contract (MPC). The general operation of the MetaAICM's marketplace is described in the right side of Fig. 1.

For privacy preservation, the ML models/data are stored on InterPlanetary File System (IPFS) [45] instead of using the on-chain storage. Moreover, the IPFS URL is encrypted by the buyer's public key. In case a buyer is not satisfied with the received product (e.g., she claims that the seller sent incorrect data, or the data has been modified), a two-stage comparison mechanism is activated to validate the reports. Consequently, malicious buyers will lose their tokens due to the dishonest disputation reports, while the transaction is canceled automatically if the report is verified to be honest. When the payment is finalized, the buyer can submit a rating for the product with a score from 0 to 10, which reflects the product's quality and also impacts the seller's reputation.

C. THREAT MODEL

The threat model is described in the following. In MetaAICM, it is assumed that any entities, including DWs, MLWs, DSs, and MSs, can be malicious. These involved participants do not trust each other and always want to maximize their benefit by committing malicious activities or colluding together to conduct attacks. Furthermore, we assume that there is no trusted authority in our framework, and the operation of any entity can be corrupted suddenly. Under these assumptions, there can be a wide range of potential attacks that MetaAICM aims to cope with as described in the following:

1) FALSE-REPORTING ATTACK

In the crowdsourcing mode, after receiving a high-quality dataset $\mathcal{D}_i$ from the worker DW$_i$, the data requester MLW$_j$ may intentionally misreport that it has not received $\mathcal{D}_i$ from DW$_i$, or the received $\mathcal{D}_i$ is of low quality. The adversary goal is to avoid paying the fee $\mathcal{F}_i$ associated with $\mathcal{D}_i$. Similarly, a model requester MSP$_j$ might also deny a high-quality model $\mathcal{M}_i$ committed by the worker MLW$_i$ to repudiate the payment $\mathcal{F}_i$. As a result, honest workers would not be rewarded for their contribution and effort, while the malicious requesters could obtain high-quality products (i.e., $\mathcal{D}_i$ and $\mathcal{M}_i$) without paying the associated fee $\mathcal{F}_i$.

In the ML marketplace, a buyer MSP_i/MS_i could also conduct false-reporting attack by claiming that the purchased model/dataset is low-quality or even inaccessible, although that product is valid and of high quality. Consequently, it impacts directly on the benefits of the data/model sellers and allows illegal financial gains from the buyer side.

2) FREE-RIDING ATTACK

Regarding the crowdsourcing system, a worker MLW_i could accept a model request from a MSP_j, then only commit an arbitrary model $\mathcal{M}_i$ instead of making real effort to train the model. The purpose of this malicious MLW_i might be either attacking the requester MSP_j with a harmful model, or just for gaining metaverse tokens without real contribution. Similarly, a malicious DW_i could also submit an arbitrarily low-quality dataset to poison the MLWs and obtain the free-riding rewards. In contrast to false-reporting attack, this free-riding attack poses a threat to the benefits of the requesters instead of the workers.

In the marketplace of MetaAICM, a similar form of free-riding attack can manifest when a seller sells a product that is substantially lower in quality than described, or not as advertised. Such a deceitful act leads buyers to pay for products of negligible value, while enabling the seller to earn tokens with minimal or no genuine effort in data collection or model training. This form of attack directly impacts the marketplace's reliability and can erode the trust of buyers in the system.

3) SYBIL AND DOS ATTACK

Sybil attack [48] refers to the circumstance in which an attacker generates a large number of fake identities to attain superior impact, thus manipulating the system. For instance, in the data crowdsourcing system, a Sybil worker DW_i may use its k fake identities to submit k invalid datasets for a crowdsourcing task $\mathcal{T}_j$, increasing the possibility that one of its datasets is chosen and rewarded by the requester MLW_j. In terms of the marketplace, the Sybil identities can dominate the rating/reputation system by submitting multiple dishonest ratings to the seller.

Using a similar approach, the attacker can also conduct a DoS attack by committing numerous buying/selling requests on the marketplace, surpassing the processing capacity of the blockchain. As a result, this might corrupt the entire system instead of each individual.

4) PRIVACY LEAKAGE

In the crowdsourcing system, when a data worker DW_i submit its dataset $\mathcal{D}_i$ to fulfil a data request from the requester MW_j, it is possible that $\mathcal{D}_i$ is also accessible by other entities in the system as on-chain information is transparent. As a result, some malicious entities may obtain sensitive information from $\mathcal{D}_i$, or even use such this dataset to fulfil the corresponding data request $\mathcal{T}_i$ and compete with the original worker DW_i. This not only raises privacy issues, but also threatens the workers' benefit.

On the other hand, hackers can attack the marketplace's storage environment to access the datasets and ML models listed on the marketplace. Consequently, the attackers could steal sensitive information from the data/models without paying the products' fees.

IV. PROPOSED FRAMEWORK

We describe the proposed metaverse crowdsourcing system and marketplace design in this section.

A. METAVERSE ML CROWDSOURCING SYSTEM

The proposed crowdsourcing system is depicted in Fig. 2 with two main components, which are ML model crowdsourcing and data crowdsourcing as presented in the following.

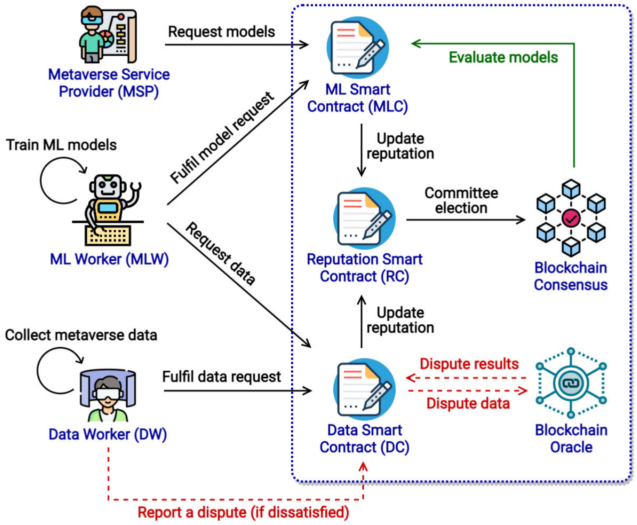


FIGURE 2. The crowdsourcing system of MetaAICM.

1) ML MODEL CROWDSOURCING

The proposed crowdsourcing design for ML models is summarized in Algorithm 1, in which a MSP_i crowdsources ML models from n different MLWs. Firstly, MSP_i commits a "model crowdsourcing" transaction to activate the MLC smart contract, where the transaction structure is as follows:

$$T_{mc} = \{T_{\text{token}} = \Theta | T_{\text{desc}} | t_{\text{deadline}} | R_{\min} | Sig_{sk}(T_{mc})\}, \quad (1)$$

where Θ is the task reward (resulting in certain metaverse tokens), T_{desc} is the task description (i.e., requirements for the crowdsourced model), t_{deadline} is the task deadline, $R_{\min}$ is the minimum reputation required to join the task, and $Sig_{sk}(T_{mc})$ is the digital signature signed by the MSP_i's secret key.

MLWs who are interested in the crowdsourcing task can train a ML model that satisfies the requirements in the task description. Once finishing training, the participated MLWs upload their solutions (i.e., the trained ML models) to IPFS, then submit the IPFS URLs to the smart contract MLC. When the task deadline is reached, the task requester MSP_i must publish the validation data D_{val} to the blockchain committee

Algorithm 1 ML Model Crowdsourcing

```
Input: $MSP_i$, $T_{desc}$, $t_{deadline}$, $R_{min}$, $D_{val}$, $\Theta$, $\mathcal{W} = \{MLW_i\}_1^n$.
Output: The highest-performance model $Model_{best}$.
MSP_i initializes a model crowdsourcing request via a transaction: $T_{mc} = \{T_{token} = \Theta | T_{desc} | t_{deadline} | R_{min} | Sig_{sk}(Tx)\}$;
for $MLW_j \in \mathcal{W}$ do
    if reputation of $MLW_j < R_{min}$ then
        $MLW_j$ is refused;
    else
        $MLW_j$ trains $Model_j$ according to task description $T_{desc}$;
        $MLW_j$ stores $Model_j$ on IPFS to obtain the link $URL_j$;
        $MLW_j$ submits $URL_j$ to the smart contract MLC;
    end if
end for
if $t_{current} == t_{deadline}$ then
    $MSP_i$ publishes the validation data $D_{val}$ to the committee;
end if
The committee downloads all submitted models from IPFS;
The committee evaluates every model on the validation data $D_{val}$ to obtain $Model_{best}$ with the highest performance;
return $Model_{best}$;
```

via IPFS. Next, consensus nodes of the committee download all crowdsourced ML models and the validation data from IPFS, then use this dataset to evaluate the models. Based on the evaluation results, the highest-performance model ($Model_{best}$) is selected as the final solution for the task. The IPFS URL of the selected model is emitted to the requester MSP_i by the smart contract MLC, thus finishing the model crowdsourcing task.

The MLW whose model was chosen as the final solution is rewarded 50% of the deposited tokens Θ, while the remaining tokens are distributed to other MLWs whose model's performance is higher than a predefined threshold (e.g., the accuracy being greater than 85%). However, if the requester MSP_i fails to publish the validation data when the deadline is reached, the deposited tokens are distributed equally to all MLWs who submitted a model, regardless of their model's performance. In that case, although the $Model_{best}$ is not determined, the requester still obtains all the submitted models.

2) DATA CROWDSOURCING

If MLWs do not possess the necessary data to train their models, they can initialize a data crowdsourcing task to collect the desired metaverse data from other MUs. In this case, the MLWs are also called data requesters. In comparison to the model crowdsourcing process presented above, the data crowdsourcing procedure poses other fundamental challenges:

a: PRIVACY ISSUE

Unlike ML models, the leakage of crowdsourcing data may threaten the privacy of DWs. Therefore, the crowdsourcing data must be encrypted before being submitted to ensure privacy preservation.

b: DATA EVALUATION

While the quality of ML models can be evaluated based on a common metric such as accuracy or loss, there is no similar standard metric for data evaluation. Therefore, MetaAICM allows data requesters to evaluate the crowdsourced datasets and decide the reward distribution by themselves. False-reporting attacks are discouraged by requiring data requesters to deposit crowdsourcing fees into the smart contract DC in advance, while the decentralized blockchain oracle can resolve any disputation between data requesters and DWs.

The proposed data crowdsourcing process is summarized in Algorithm 2, in which a data requester MLW_i must commit a "data crowdsourcing" transaction to initialize the task where the transaction content is:

$$T_{dc} = \{T_{token} = \Theta + \alpha | f_{data} | pk | t_{deadline} | R_{min} | Sig_{sk}(T_{dc})\}, \tag{2}$$

where α is a compensation stake that will be recompensed to the DWs whose dataset is honest but not recognized by the requester, f_{data} is a list of desired features for the crowdsourcing dataset, pk is MLW_i's public key, and the remaining elements are similar to those presented in (1).

Interested DWs can collect data to fulfill the crowdsourcing request. The collected datasets must comprise all data features listed in f_{data}. Upon reaching the designated deadline, DWs upload their datasets to IPFS, receiving a unique URL for each submission. They then encrypt these URLs using the MLW_i's public key pk_i, and each resulting encrypted string is submitted to the smart contract DC via a transaction Sub_j. Although accessible publicly, these encrypted strings can only be decrypted by MLW_i using their private key, ensuring the privacy of the submitted data.

Consequently, MLW_i can retrieve all the crowdsourced datasets from the smart contract DC. MLW_i must evaluate these datasets and submit the evaluation results to the DC for reward distribution. The evaluation results indicate which datasets are honest (e.g., clean and high-quality datasets) or dishonest based on MLW_i's decision. As a result, Θ deposited tokens are distributed to DWs whose dataset is stated to be honest, while the rest of the DWs will not receive any reward.

In cases where DWs contest the evaluation results, they are entitled to initiate a dispute. This is done by triggering the DC's disputation function and submitting the URL of their datasets hosted on the IPFS. The smart contract DC then verifies the legitimacy of each disputation request by ensuring it aligns with the corresponding transaction Sub_j,

Algorithm 2 Data Crowdsourcing

```
Input: $\text{MLW}_i, f_{\text{data}}, pk, t_{\text{deadline}}, R_{\min}, \Theta, \alpha, \mathcal{W} = \{\text{DW}_i\}_1^n$.
Output: qualified datasets $D$.
$T_{dc} = \{T_{\text{token}} = \Theta + \alpha | f_{\text{data}} | pk | t_{\text{deadline}} | R_{\min} | Sig_{sk}(T_{dc})\}$;
for $\text{DW}_j \in \mathcal{W}$ do
    if reputation of $\text{DW}_j < R_{\min}$ then
        $\text{DW}_j$ is refused;
    else
        $\text{DW}_j$ collect $\text{Dataset}_j$ according to $f_{\text{data}}$;
        $\text{DW}_j$ stores $\text{Dataset}_j$ on IPFS, obtaining $\text{URL}_j$;
        $\text{DW}_j$ encrypts $\text{URL}_j$, obtaining an encrypted string $\text{eURL}_j$;
        $\text{DW}_j$ submits $\text{eURL}_j$ to DC via transaction $Sub_j$;
    end if
end for
if $t_{\text{current}} == t_{\text{deadline}}$ then
    $\text{MSP}_i$ publishes evaluation results to smart contract DC;
    DC distributes rewards to qualified datasets;
end if
if $\text{DW}_j$ send dispute request and $\text{URL}_j$ to DC then
    DC verifies the dispute request aligns with $Sub_j$;
    DC triggers the Oracle network to re-evaluate $\text{Dataset}_j$;
    if $\text{Dataset}_j$ is qualified then
        DC sends compensation to $\text{DW}_j$;
    end if
end if
return $D$;
```

where a DW_j previously submitted an encrypted URL. Upon confirmation of this alignment, the dispute is escalated to the decentralized blockchain oracle network. This network comprises multiple professional nodes that independently assess whether the disputed datasets meet the required quality standards. Each node casts a vote to determine if a dataset is qualified, and the final decision is based on the compounded result of the majority votes. This decision is then relayed back to DC, which enforces the outcome. If the DWs succeed in their dispute, they are collectively awarded the compensation stake α, shared equally among them. In contrast, if there is no disputation request or no DWs won their dispute, the compensation stake α is given back to MLW_i after certain block times t_{lock}.

B. METAVERSE ML MARKETPLACE

According to the market's demand, MUs can proactively gather data or train ML models for sale on MetaAICM's marketplace, bypassing the need to wait for specific crowdsourcing requests. To simplify the discussion, both datasets and ML models are referred to as "products" in this context, with the entities involved being broadly categorized as buyers and sellers.

To list a new product on the marketplace, a seller first uploads the product's data to IPFS and obtains a corresponding IPFS URL. The URL can be considered a representation of the product as everyone can download the product if they have its URL. The seller then initiates the marketplace listing process by submitting a transaction to MPC, structured as follows:

$$T_{\text{list}} = \{P_{\text{price}} | P_{\text{desc}} | \text{H}_{\text{s}} | Sig_{sk}(T_{\text{list}})\}, \tag{3}$$

where P_{price} is the product's price, P_{desc} is the product's description (e.g., features list of a dataset, or ML model's type such as regression/classification), H_{s} is the hash of the IPFS URL, and $Sig_{sk}(T_{\text{list}})$ is the transaction's digital signature.

A buyer interested in purchasing a product initiates the process by submitting a buying request and depositing the requisite tokens, as per the product's price, into the MPC. Upon receiving this request, the MPC automatically emits an event to notify the seller. Subsequently, the seller encrypts the product's IPFS URL using the buyer's public key, resulting in an encrypted URL denoted as URL_{s}, which is then sent to the MPC. The buyer retrieves URL_{s} from the MPC and decrypts it using their secret key to access the original URL and download the product's source data. While the encrypted URL_{s} is publicly accessible as on-chain information, decryption is exclusive to the buyer due to the need for their secret key.

In cases where the buyer is not satisfied with the product, suspecting issues such as data modification or seller misconduct, they have a set period t_{chal} to initiate a dispute. This is done by invoking the "disputation" function of the MPC, accompanied by the decrypted URL. The MPC re-encrypts this URL with the buyer's public key to generate URL_{b} and then compares it with the previously stored URL_{s}. Should there be a discrepancy between URL_{s} and URL_{b}, it indicates the buyer has intentionally provided an incorrect URL, leading to the transfer of the buyer's deposited tokens to the seller, thereby finalizing the purchase. Conversely, if the URLs match, the MPC hashes URL_{b} to get H_{b}, and compares it with H_{s} from the listing transaction (3). A mismatch at this stage implies a modification of the IPFS URL during the transaction, leading to a refund of the deposited tokens to the buyer. If no discrepancies are found, the buyer might be regarded as a potential DoS attacker, and their deposit is transferred to the seller.

If no disputes arise within the timeframe t_{chal}, the deposited tokens are automatically transferred to the seller, thus concluding the payment process. Additionally, the buyer has the option to rate the purchased product, on a scale from 0 to 10, using the "rating" function of the MPC. This rating contributes to the product's average score, which is displayed on the marketplace for reference. The workflow of the MetaAICM marketplace is presented in Fig. 3.

C. INCENTIVE AND REPUTATION SYSTEM

MetaAICM's incentive framework includes token rewards and a reputation mechanism. Token rewards are allocated

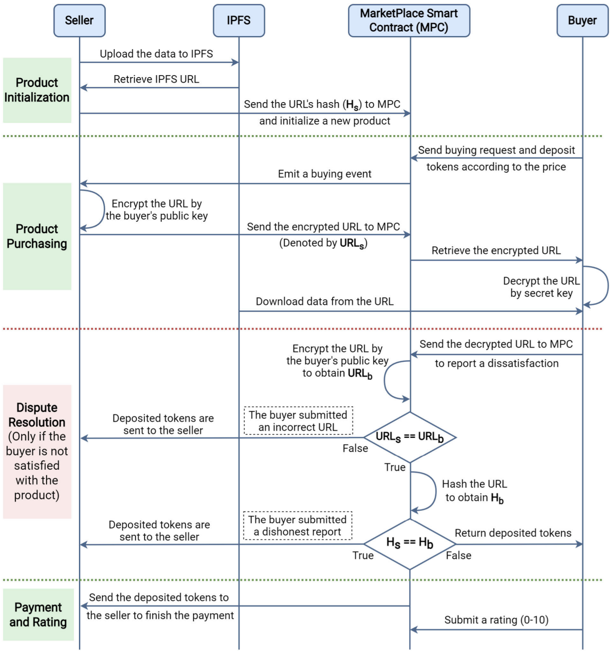

Seller
IPFS
MarketPlace Smart Contract (MPC)
Buyer
Product Initialization
Upload the data to IPFS
Retrieve IPFS URL
Send the URL's hash ($\mathbf{H_s}$) to MPC and initialize a new product
Send buying request and deposit tokens according to the price
Emit a buying event
Product Purchasing
Encrypt the URL by the buyer's public key
Send the encrypted URL to MPC (Denoted by $\mathbf{URL_s}$)
Retrieve the encrypted URL
Decrypt the URL by secret key
Download data from the URL
Send the decrypted URL to MPC to report a dissatisfaction
Encrypt the URL by the buyer's public key to obtain $\mathbf{URL_b}$
Dispute Resolution (Only if the buyer is not satisfied with the product)
Deposited tokens are sent to the seller
The buyer submitted an incorrect URL
$\mathbf{URL_s == URL_b}$
False
True
Hash the URL to obtain $\mathbf{H_b}$
Deposited tokens are sent to the seller
The buyer submitted a dishonest report
$\mathbf{H_s == H_b}$
True
False
Return deposited tokens
Payment and Rating
Send the deposited tokens to the seller to finish the payment
Submit a rating (0-10)

FIGURE 3. The workflow of MetaAICM's marketplace.

to the entities guaranteeing the operation of the system: blockchain oracle nodes and consensus nodes. Oracle nodes whose decision is the same as the majority of the oracle network are rewarded metaverse tokens, while those whose decision is opposite will face deductions. Consensus nodes in the committee also receive tokens for each new block added onto the blockchain after the consensus process.

Beyond token incentives, MetaAICM integrates reputation scores to incentivize contributors for their honesty and valuable contributions. The reputation scores play a crucial role in determining participants' eligibility for selection as consensus nodes in the blockchain committee for subsequent rounds. The reputation management is autonomously handled by the smart contract RC. This smart contract tracks and updates each avatar's reputation score based on contributions to the system. Importantly, all changes to reputation scores result from transactions, which are executed and verified by the blockchain committee, guaranteeing that no individual can unilaterally modify their reputation score; attempts to do so through unauthorized transactions will be rejected.

For each participant i, the reputation score r_i is updated in response to their behaviors in defined contexts:

- Model Crowdsourcing: MLWs whose model's performance is higher than the required threshold will receive a reward. The remaining MLWs incur a reputation penalty due to their low-quality models.
- Data Crowdsourcing: DWs whose datasets are accepted by the data requester, or win disputes, will receive reputation score rewards. The remaining DWs have their reputation scores slashed. In case of successful disputes, the data requesters also face reductions in their reputation scores.
- Marketplace: Sellers gain reputation reward for receiving high ratings (e.g., ratings ≥ 8) and lose reputation for low ratings (e.g., ratings ≤ 3). Moreover, each negative disputation decision results in punishment to the reputation score of the seller/buyer.

Let α refer to the change in reputation score following an event, with $\alpha = R$ standing for the rewarding case and $\alpha = P$ standing for the penalty case. Thus, the reputation r_i can be computed as follows:

$$r_i = \begin{cases} r_i + \dfrac{A_i}{P_i}, & \text{if } \alpha = R \\ r_i - 1, & \text{if } \alpha = P \end{cases}$$

where A_i represents the number of unique avatars interacted with by user i through crowdsourcing and marketplace activities, P_i represents the total participation instances of user i, including all submissions of crowdsourcing tasks and buying/selling activities in the marketplace. This incentive mechanism ensures that rewards are directly proportional to the diversity and frequency of positive contributions, thereby enhancing the system's resilience to collusion attacks. In the Metaverse, the number of unique avatars is significantly fewer than the number of activities, which influences the rate of reputation increase. Consequently, users with numerous activities experience a slower reputation increase compared to new users. This design is effective in preventing the domination of the system by longstanding users and encourages new users to actively contribute to the Metaverse.

D. REPUTATION-BASED RAFT CONSENSUS

The developed smart contracts are deployed in a consortium blockchain that must be appropriately designed to meet our design goals. Note also that the consensus mechanism deployed to reach agreements on newly added blocks over time strongly impacts the resulting blockchain performance. In this paper, our selection of a consensus mechanism for MetaAICM aims at managing the blockchain network effectively while maintaining transaction integrity, security, and efficiency. Considering the demands of the metaverse environment, particularly the need for high transaction throughput and a decentralized, trust-based system, we identify the following requirements for our consensus mechanism:

- High Transaction Throughput: Our consensus protocol must efficiently handle large-scale metaverse data, supporting the network's high transaction rates.
- Alignment with Reputation Mechanism: The consensus algorithm must be well-suited for the integration of a reputation system for node selection, enhancing network trust and reliability.
- Fault-Tolerance Capacity: In a distributed system with multiple consensus nodes, some of the nodes might crash or behave maliciously during its operation, thus hindering the consensus process. Therefore, an important requirement of MetaAICM's consensus algorithm is to ensure its seamless operation even if a certain proportion of the consensus nodes are inoperative.

- Sustainability: Unlike energy-intensive algorithms like PoW [49], our consensus protocol must be efficient in both energy consumption and data storage, thereby contributing to the overall sustainability of the entire blockchain network.

In the MetaAICM, the consortium blockchain plays the key role in managing identity and ensuring the trustworthiness of the system. Common consensus mechanisms include PoW, PoS, PBFT, BFT-Smart [50], and Raft. PoW and PoS are predominant in public blockchain environments, where participants are anonymous. These consensuses, while central to common cryptocurrencies (e.g., Bitcoin, Ethereum) have limitations in scalability (i.e., limited transaction throughput), making them less ideal for the Metaverse. Conversely, PBFT, BFT-Smart, and Raft are frequently utilized in consortium blockchains. However, PBFT and BFT-Smart still have the scalability limitation due to the heavy communication overhead of their voting mechanisms, especially as the network size expands. On the other hand, Raft offers better performance and scalability even when the number of nodes is large. It is also known for its simplicity and understandability, simplifying the implementation and maintenance process [38]. It is designed primarily for crash-fault tolerance, allowing the system to sustain up to 50% node failures [51], yet lacks safeguards against malicious behaviors. Therefore, to address this limitation, we construct the MetaAICM's consensus protocol partly based on Raft, with the integration of our reputation system for the election of consensus nodes. This integration aims to mitigate the vulnerability to malicious activities, hence, enhancing the security and reliability of consensus in a consortium blockchain setting.

In particular, MetaAICM features a blockchain committee comprising multiple consensus nodes. These nodes are responsible for transaction verification, block proposal, and ensuring a consistent ledger across the network. The committee's structure includes an *authorized committee partition*, which reserves u% of slots for metaverse organizations responsible for maintaining the infrastructure and operations of the virtual world. The rest of the slots are allocated to normal nodes through a reputation-based election mechanism, forming what we term the *dynamic committee partition*. This partition is subject to re-election every k rounds, as illustrated in Fig. 4, ensuring continual adaptability and responsiveness to the evolving needs of MetaAICM, while the authorized partition provides stability and ongoing governance.

1) CONSENSUS PROCESS

The consensus mechanism in MetaAICM, based on the Raft model [38], is integral to maintaining a consistent and reliable blockchain ledger. This process is depicted in Fig. 4 and involves three distinct states for a consensus node: follower, candidate, and leader. The primary goal of this mechanism is to achieve a robust, democratic, and failure-resistant system for block proposal and ledger consistency.

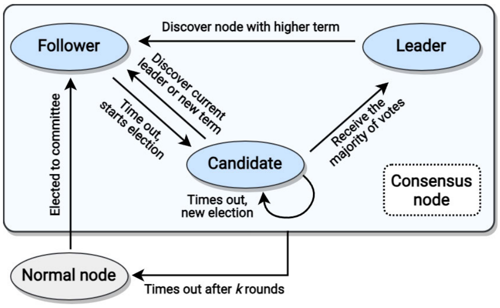

Follower
Discover node with higher term
Leader
Discover current leader or new term
Time out, starts election
Receive the majority of votes
Elected to committee
Candidate
Times out, new election
Consensus node
Normal node
Times out after k rounds

FIGURE 4. State transition of a consensus node.

a: CONSENSUS OPERATION

The Raft model achieves consensus through a streamlined leader-based approach. In this system, the designated leader is responsible for log management and data replication to the follower nodes. The leader node proposes new log entries (blocks) and ensures that followers replicate these entries consistently. Followers, in turn, accept and apply these entries to their local state. This process minimizes the risk of conflicting entries and ensures a single, agreed-upon sequence of logs across the network. The efficiency of the Raft model lies in its simplicity - with a single leader coordinating log replication, the process becomes more predictable and manageable, reducing the overheads typically associated with more complex consensus models.

b: LEADER ELECTION

The model's leader election is a democratic process, transitioning nodes through the roles of follower, candidate, and leader. All nodes initially act as followers. If a follower fails to receive a heartbeat message from the current leader within a specified time frame, it perceives this as a signal of leader failure and shifts to a candidate role. The candidate then seeks votes from other nodes to become the new leader. Achieving a majority vote is crucial for the candidate to assume the leader role, preventing any single node from dominating the process and ensuring wide network support for the elected leader. The leader, once elected, resumes sending heartbeat messages to maintain its authority and keep the network's state synchronized.

This Raft-based election process has been shown to offer 50% fault-tolerance capacity [38]. In other words, even if up to 50% of the nodes are compromised, MetaAICM still operates correctly and seamlessly. Furthermore, as there is only one leader verifying transactions at a time, it reduces significantly energy consumption while improving the transaction throughput. As a result, MetaAICM provides a higher extent of sustainability compared to other energy-intensive consensus protocols like PoW.

2) REPUTATION-BASED COMMITTEE SELECTION

In MetaAICM, the composition of the dynamic committee partition is periodically refreshed every k rounds. This process is orchestrated by the current leader, who executes

Algorithm 3 for selecting new nodes for the upcoming period, where M denotes the number of nodes to be elected. The leader begins the election process by generating a random number and its corresponding proof, $\langle \phi_1, \pi \rangle$, using a Verifiable Random Function (VRF) algorithm [52]. The seed for the VRF is derived from the hash of the previous block, thereby ensuring resistance to manipulation. The leader then iteratively hashes ϕ_1 for $M - 1$ times to produce a set of M random numbers, denoted as $\Phi = \{\phi_1, \ldots, \phi_M\}$. Each ϕ_i in Φ is instrumental in the node selection. The probability of a node being elected is proportional to its reputation score. The reputation scores of all nodes are represented as K indexed reputation units, $R_{\text{spread}} = \{r_1, \ldots, r_K\}$. For each ϕ_i, the leader selects a reputation unit, r_{Idx}, from R_{spread}. The node owning r_{Idx} is then elected to the committee, ensuring that nodes with higher reputation are more likely to be chosen.

Algorithm 3 Reputation-Based Committee Election

```
Input: Indexed reputation units $R_{\text{spread}} = \{r_1, \ldots, r_K\}$, number of nodes $M$, previous block $\mathcal{B}$.
Output: Nodes list for the next dynamic partition $\mathcal{N} = \{N_1, \ldots, N_M\}$, proof of randomness $\pi$.
Generate seed from block: seed $\leftarrow$ hash($\mathcal{B}$);
Compute VRF random number and proof: $\langle \phi_1, \pi \rangle \leftarrow \text{VRF}_{sk}(\text{seed})$;
Generate $M - 1$ additional random numbers by hashing $\phi_1$ $M - 1$ times, obtaining $\Phi = \{\phi_1, \ldots, \phi_M\}$;
for each $\phi_i \in \Phi$ do
    Compute index: $\text{Idx} = K \cdot \frac{\phi_i}{2^{||\phi_i||} - 1}$, where $||\phi_i||$ is the length of $\phi_i$ in bits;
    Append the node $N_i$ with reputation unit at Idx to $\mathcal{N}$;
end for
return $\mathcal{N}, \pi$
```

By integrating this reputation-based committee election, we ensure that highly reputable nodes are selected to regulate the consensus process, while those with low reputation are not able to dominate the system. This further improves the reliability of the framework. Moreover, the election process can be verified by any participant via validating the proof π, thereby preventing any single entity from manipulating the system.

3) BOOTSTRAPPING CONSENSUS PROCESS

At the bootstrapping stage of the system, all nodes are initialized with an equal reputation score of zero. To mitigate the risk of malicious entities infiltrating the blockchain committee during this initial phase (i.e., the cold-start period), reputation scores are not utilized for committee selection for the first W rounds. Instead, the MetaAICM platform publisher appoints a set of Trusted Seed Nodes (TSNs) to form the blockchain committee. After the cold-start period, once W rounds have elapsed and nodes have been assigned reputation scores reflecting their behavior, the committee election mechanism (refer to Algorithm 3) activates, electing consensus nodes based on these scores.

Metaverse avatars can apply to become TSNs during MetaAICM's initialization phase, with selections made based on the avatars' profiles. Operating within a consortium blockchain framework, MetaAICM ensures that the identities of these avatars are authenticated before they can assume roles within the system.

E. SECURITY ANALYSIS

1) FALSE-REPORTING ATTACK

In MetaAICM's crowdsourcing mode, false-reporting attacks are prevented by requiring the task requesters to deposit the task reward Θ when submitting the crowdsourcing transaction T_{mc}/T_{dc}. Once the crowdsourced solutions are revealed, the crowdsourcing smart contracts automatically distribute Θ to every MLW_i and DW_j whose solutions are not considered low-quality or malicious. Therefore, requesters cannot repudiate the payment by false reporting.

Regarding false-reporting attacks in the marketplace, MetaAICM deployed the smart contract MPC to efficiently eliminate false-reporters. If a buyer tries to report a dispute indicating that the received data is not the same as the one listed on the marketplace, this request will be validated by MPC via a two-step verification process. Firstly, if the buyer intentionally committed an incorrect URL (i.e., $\text{URL}_\text{s} \neq \text{URL}_\text{b}$), the report is considered a false-reporting attack and the tokens Θ are sent to the seller. Then, if the URL's hashes are identical between the seller and buyer (i.e., $\text{H}_\text{s} = \text{H}_\text{b}$), this means that the seller submitted a correct product and the buyer is trying to conduct a false-reporting attack.

2) FREE-RIDING ATTACK

In terms of crowdsourcing, every Model_i from $\text{MLW}_i \in \mathcal{W}$ is validated with the validation data D_{val} of the task publisher. This helps filter out the free-riders who submitted low-quality results. Furthermore, these malicious workers are subject to certain punishments, resulting in the reduction of tokens and reputation scores. On the other hand, the crowdsourced datasets can be validated by the decentralized blockchain oracle via the disputation mechanism. This further prevents the existence of free-riding data workers.

In the MetaAICM's marketplace, free-riding attacks are prevented based on the rating system. If a Model_i or a Dataset_j is of low quality, it would receive low ratings from the buyers. These ratings not only deter the low-quality products, but also impact the overall reputation scores of the free-riding sellers. As a result, the buyers are aware of each product's quality based on its ratings, while the free-riders are exposed due to their low-reputation profile.

3) PRIVACY LEAKAGE

In MetaAICM, data privacy is ensured based on different encryption techniques. Specifically, each URL_i associated with the crowdsourced Dataset_i is encrypted by the MLW's public key pk_j, ensuring that it is not accessible by any parties except the task requester MLW_j. Similarly, the datasets and models on the marketplace are also protected by the buyer's public key. Only the buyer with its private key can carry out decryption and obtain the products. Other blockchain entities, although being able to read any transaction information, cannot access the IPFS storage to download the data.

4) SYBIL AND DOS ATTACKS

MetaAICM employs a comprehensive reputation system designed to thwart Sybil attacks effectively. Within this framework, each avatar operates as an independent entity. Thus, the system prevents the aggregation of reputation scores from multiple avatars into a single, artificially inflated score. This system ensures that participants with multiple avatars, particularly those with low reputation scores, are restricted in their influence and capabilities within the network. By limiting the participation of such low-reputation entities, the integrity of the consensus process and other critical operations of the system is maintained. Therefore, our design significantly diminishes the potential impact of Sybil attackers, who typically rely on creating numerous fake identities to manipulate or disrupt network activities.

In the case of DoS attacks, MetaAICM's design inherently discourages such attempts. Once a group of attackers collude to conduct DoS, their balance would be drained quickly due to the punishment mechanism. Therefore, DoS attackers will gain nothing other than the loss of their tokens.

5) TRUST ISSUE AND SPOF

MetaAICM effectively addresses trust issues and eliminates SPoF by adopting a decentralized architecture across all its operations. The platform's approach involves a crowdsourcing system and marketplace that operate without any centralized authority, relying instead on a consortium blockchain to distribute trust across the network. This ensures that no single entity controls the system, thereby enhancing security and reliability for all participants. In MetaAICM, both data and ML models are stored on the IPFS, a peer-to-peer storage solution. Moreover, the role of MSPs is designed to contribute to the metaverse's development without centralizing control or influence over the blockchain or the system. This decentralization ensures that MetaAICM's operations are transparent and verifiable, fostering a secure and trustworthy environment. By maintaining the platform through a robust network of decentralized nodes, we not only bolster its resilience against failures and disruptions but also guarantee uninterrupted service availability.

V. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

MetaAICM is deployed as a permissioned blockchain based on Hyperledger Fabric [44], an open-source blockchain development platform. The implementation source code is published on GitHub[1] with detailed instructions. To take part in the system, each blockchain node in MetaAICM maintains a Docker container to execute its operation (e.g., training ML models, performing consensus mechanisms, executing smart contracts, and submitting transactions). Consensus nodes use PyTorch to perform model evaluation in model crowdsourcing. A blockchain benchmarking tool named Hyperledger Caplier is used to simulate transaction workloads and monitor the network's performance. IPFS is also re-implemented locally to prevent Internet latency, with 100 peer-to-peer nodes.

B. PERFORMANCE EVALUATION

1) BLOCKCHAIN PERFORMANCE UNDER DIFFERENT WORKLOADS

In the experiment illustrated in Fig. 5, 100 clients are set up to simultaneously submit transactions invoking smart contract functions of three main systems, namely data crowdsourcing, ML model crowdsourcing, and decentralized marketplace. The transaction workload increases from 100 transactions per second (TPS) to 2000 TPS. According to Fig. 5, when the workload is less than 1000 TPS, the system can afford most of the submitted transactions with a transaction processing rate of more than 92%. However, when the workload increases beyond 1200 TPS, the transaction processing rate decreases sharply. At 2000 TPS, only 50% of transactions are processed in each round. This indicates that the system's processing capacity is limited to around 1000–1200 TPS. Similarly, the average latency of the network (i.e., the average time it costs for a transaction to be processed) is negligible until reaching the mentioned saturation point of 1000 TPS.

2) BLOCKCHAIN PERFORMANCE WITH VARYING BLOCK SIZE

Fig. 6 presents the results of another experiment in which the blockchain performance is monitored with different block sizes (from 100 to 1000 transactions per block) and under different workloads (from 700 to 2000 TPS). It is shown that the system with the smallest block size of 100 transactions can efficiently handle the workload of 1200 TPS. With a higher transaction workload, its performance starts decreasing rapidly. Intuitively, if the block size is small, it costs more blocks to process/store the same number of transactions. However, Fig. 6 also shows that an excessively large block size of 1000 transactions per block achieves lower performance than another test case with 500 transactions per block. This is because the consensus time is often higher with larger block sizes. When the block size becomes larger than the actual processing demand, it might cause redundancy and lead to lower performance.

3) CONSENSUS PERFORMANCE

In terms of blockchain consensus, the performance of MetaAICM's consensus is compared to another baseline

[1] https://github.com/duyhung2201/MetaAICM

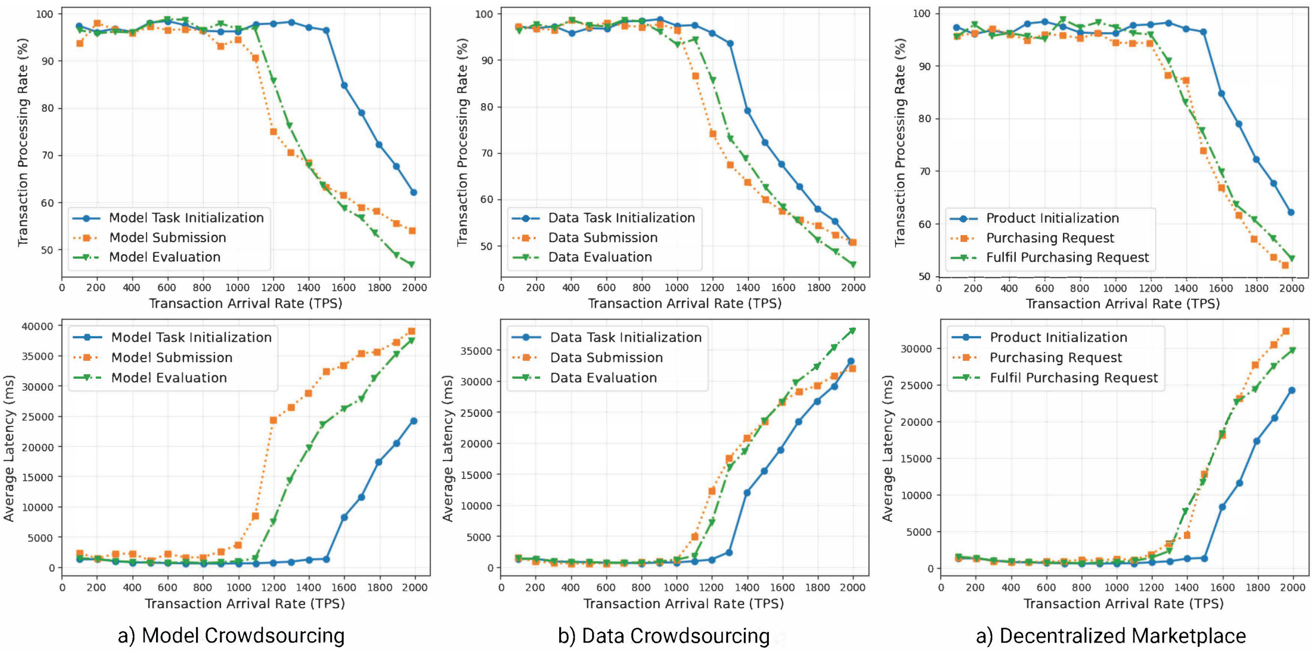


FIGURE 5. Performance of smart contract's functions under different workloads, ranging from 100 to 2000 TPS.

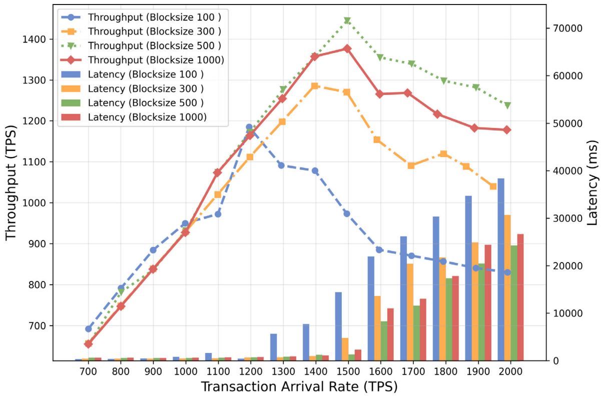


FIGURE 6. Blockchain performance as the block size varies from 100 to 1000 transactions per block.

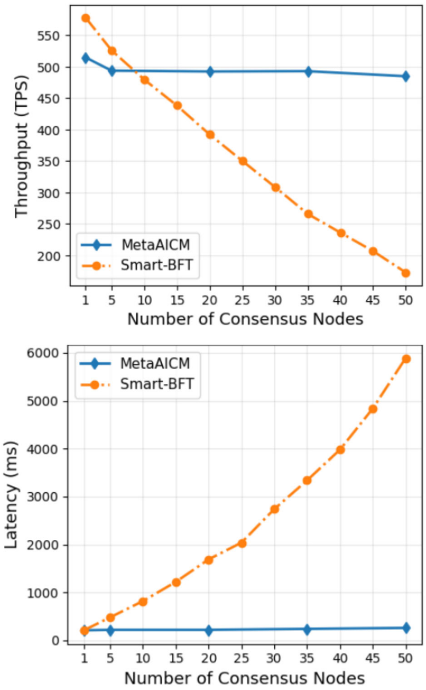


FIGURE 7. Performance of MetaAICM's consensus compared to BFT-Smart when the committee size varies from 1 to 50.

framework using BFT-Smart consensus algorithm [50]. The transaction arrival rate is set to 600 TPS in this experiment, and the committee size ranges from 1 to 50 consensus nodes. As shown in Fig. 7, MetaAICM maintains a high throughput and low latency regardless of the committee size, while the baseline system's performance drops rapidly when the committee size increases. As a result, MetaAICM allows more consensus nodes to participate in the system, making it more decentralized without sacrificing throughput and latency.

4) STORAGE ENVIRONMENT EVALUATION

Table 2 shows both the upload and download time of the system according to several benchmarking ML datasets and models. In general, a larger dataset/model often leads to a higher upload and download time. On the other hand, it is observed that the upload time is significantly higher than the download time. The main reason for this additional delay is that a file must be divided into smaller chunks when being uploaded to IPFS, while each chunk also costs a certain amount of time to generate a corresponding hash value.

TABLE 2. Upload and download time of the system according to different benchmarking datasets and ML models.

Dataset	Size (KB)	Upload (ms)	Download (ms)
Reuteurs	3931	79.08	7.80
Ratings1m	24,017	223.84	42.74
Flowers-102	24,567	234.55	43.98
IMDB	27,712	256.51	45.88
MNIST	53,594	374.25	66.41
LFW	176,334	1312.62	214.46
CIFAR-10	180,000	1335.29	264.69
SVHN	297,965	2341.91	368.98
Model	**Size (KB)**	**Upload (ms)**	**Download (ms)**
MobileNet	16,871	169.88	24.39
NASNetMobile	22,884	192.57	41.88
DenseNet21	32,635	265.78	44.36
Xception	89,889	694.78	141.49
InceptionV3	94,016	607.33	131.78
ResNet50	100,651	726.15	138.78
ResNet152V2	237,353	1698.98	357.59
ConvNeXtBase	346,879	2446.19	506.08
VGG16	540,532	3788.18	612.51

In contrast, the download time is relatively smaller since the IPFS is deployed locally, making it easier to search for the desired data chunks based on the provided hashes.

5) REPUTATION SCORE EXPERIMENT

This experimental analysis examines the behavioral patterns of five distinct user groups over 100 crowdsourcing tasks to evaluate the performance of the reputation mechanism, as depicted in Figure 8. The x-axis represents the cumulative number of crowdsourcing tasks, while the y-axis measures the average reputation score for each group. Specifically, Group 1 consistently submits honest work, whereas Group 4 only engages in malicious activities. Group 2 starts with honest contributions for the first 40 tasks before switching to malicious behavior for the remainder of the experiment. Group 3 exhibits a combination of honest and malicious submissions throughout.

Group 1 demonstrates a consistent increase in reputation, though with a diminishing rate, attributable to a decrease in the diversity of interactions (i.e., unique avatars encountered) as the number of tasks increases. Group 2 experiences an initial reputation boost due to positive contributions, followed by a significant decline post-task 40, coinciding with their shift to malicious activities. This reputational decline is initially steep, stabilizing as their score lowers, which is a consequence of reduced task eligibility due to previously malicious submissions. It can be noted that past behaviors are transparent in the blockchain, which affects the participating ability of bad entities in the system. Group 3's reputation fluctuates, mirroring their mixed behavioral pattern, with periods of increase reflecting honest contributions and decreases corresponding to malicious actions. Conversely, Group 4 undergoes a sharp decline in reputation due to persistent malicious behavior, with the rate of decline stabilizing towards the experiment's end. This stabilization occurs as their reputation diminishes, hence progressively restricting their access to tasks. Eventually, group 4 users are excluded from further participation in the system.

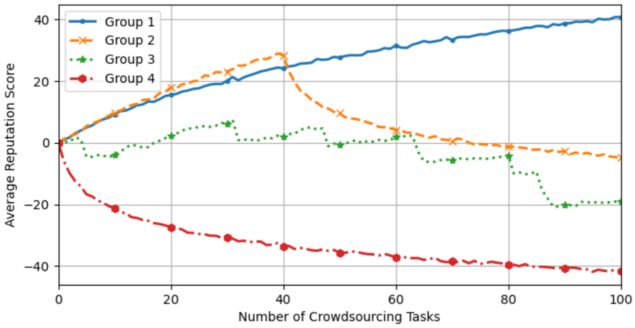


FIGURE 8. Average reputation scores change.

VI. CONCLUSION

This paper presented MetaAICM, our proposed blockchain-based framework aiming to maximize the potential of ML and intelligent services for the metaverse. In MetaAICM, we designed a distributed ML crowdsourcing system and a decentralized ML marketplace that enables both selling and buying on demand, serving both metaverse data and ML models. Instead of relying on a trusted authority, our design is completely decentralized and requires no trust assumption among participants, thereby being resistant to SPoF and trust issues. MetaAICM is robust against free-riding and false-reporting attacks thanks to the disputation resolution mechanism, while user privacy is guaranteed with the use of encryption techniques. Moreover, the concrete incentive and reputation mechanisms can efficiently eliminate malicious actors, while encouraging MUs to contribute their available recourse to the system. Experimental results showed that MetaAICM offers high performance with low processing latency and high throughput.

VII. OPEN CHALLENGES AND RESEARCH DIRECTION

A. PRIVACY THREATS AND COUNTERMEASURES

Although MetaAICM uses asymmetric encryption techniques to protect the trading data/models, other potential techniques could be useful to address privacy threats in more sophisticated metaverse application contexts such as homomorphic encryption, Zero-Knowledge Proofs (ZKP), Differential Privacy (DP), and Secure Multi-party Computation (SMC). However, it should be noted that each method comes with its own strength and disadvantages. For instance, homomorphic encryption and ZKP are restricted to a limited set of mathematical operations, while DP can reduce the quality of the data. Detailed studies of these techniques for different metaverse applications are outside the scope of our current work. Thus, we would like to leave them for our future work.

B. SCALABILITY ISSUE

Based on the proposed Raft-based consensus algorithm, MetaAICM achieves the throughput around 1,500 TPS. As MetaAICM is only used for the crowdsourcing/ marketplace purposes instead of other applications like gaming, finance, healthcare, and education, this performance would be sufficient. However, a real-world metaverse environment might require even higher throughput as the number of metaverse users can be enormous. Therefore, development of efficient scalability techniques like sharding and layer-2 solutions is still necessary and worth further research. In our future work, we plan to apply scalability methods to scale up the throughput for the large-scale metaverse environment.

C. DATA EVALUATION METHODS

While evaluating ML models is simply computing their performance on a given test data, evaluating the training data is much more challenging. In MetaAICM, we let requesters evaluate the data, and then allow workers to submit a dispute request in case of dissatisfaction, which will be verified by a blockchain oracle. However, the oracle is often less decentralized than the blockchain due to its small size. Therefore, design of an efficient method for automatically evaluating ML data is an open research direction.

REFERENCES

[1] Y. X. G. Xingren, "Virtual sound modeling and real-time rendering in aircraft simulator," *Acta Aeronauticaet Astronautica Sinica*, vol. 30, no. 7, p. 1305, 2009. [Online]. Available: https://hkxb.buaa.edu.cn/EN/abstract/article_9935.shtml

[2] X. Yang, G. Gong, and X. Wang, "Real-time visual system of night-flying across the sea," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 34, no. 1, p. 35, 2008.

[3] Y. Xinying, G. Guanghong, Z. Bo, and H. Zhanpeng, "Virtual modeling and rendering technologies of high-seas environment," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 35, no. 4, pp. 493–496, 2009.

[4] H. Wang, H. Ning, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14671–14688, Aug. 2023.

[5] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 656–700, 1st Quart., 2023.

[6] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zheng, "Fusing blockchain and AI with metaverse: A survey," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 122–136, 2022.

[7] V. T. Truong and L. B. Le, "Security for the metaverse: Blockchain and machine learning techniques for intrusion detection," *IEEE Netw.*, early access, Jan. 9, 2024, doi: 10.1109/MNET.2024.3351882.

[8] Y. Zhou, H. Huang, S. Yuan, H. Zou, L. Xie, and J. Yang, "MetaFi++: WiFi-enabled transformer-based human pose estimation for metaverse avatar simulation," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14128–14136, Aug. 2023.

[9] Y. Lin, H. Du, D. Niyato, J. Nie, J. Zhang, Y. Cheng, and Z. Yang, "Blockchain-aided secure semantic communication for AI-generated content in metaverse," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 72–83, 2023.

[10] M. Wang, T. Zhu, X. Zuo, M. Yang, S. Yu, and W. Zhou, "Differentially private crowdsourcing with the public and private blockchain," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8918–8930, May 2023.

[11] J. Wu, K. Lin, D. Lin, Z. Zheng, H. Huang, and Z. Zheng, "Financial crimes in Web3-empowered metaverse: Taxonomy, countermeasures, and opportunities," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 37–49, 2023.

[12] V. T. Truong, L. Le, and D. Niyato, "Blockchain meets metaverse and digital asset management: A comprehensive survey," *IEEE Access*, vol. 11, pp. 26258–26288, 2023.

[13] H.-N. 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.

[14] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in IoT networks," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6487–6497, Apr. 2021.

[15] W. Dai, C. Dai, K. R. Choo, C. Cui, D. Zou, and H. Jin, "SDTE: A secure blockchain-based data trading ecosystem," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 725–737, 2020.

[16] A. Dixit, A. Singh, Y. Rahulamathavan, and M. Rajarajan, "FAST DATA: A fair, secure, and trusted decentralized IIoT data marketplace enabled by blockchain," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2934–2944, Feb. 2023.

[17] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102331–102344, 2019.

[18] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1251–1266, Jun. 2019.

[19] Y. Lu, Q. Tang, and G. Wang, "ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 853–865.

[20] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "ZkCrowd: A hybrid blockchain-based crowdsourcing platform," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4196–4205, Jun. 2020.

[21] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1407–1419, Dec. 2019.

[22] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Netw.*, vol. 35, no. 4, pp. 198–205, Jul. 2021.

[23] V. T. Truong and L. B. Le, "MetaCIDS: Privacy-preserving collaborative intrusion detection for metaverse based on blockchain and online federated learning," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 253–266, 2023.

[24] Y. Wang, Z. Su, and M. Yan, "Social metaverse: Challenges and solutions," *IEEE Internet Things Mag.*, vol. 6, no. 3, pp. 144–150, Sep. 2023.

[25] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "MetaChain: A novel blockchain-based framework for metaverse applications," in *Proc. IEEE 95th Veh. Technol. Conference: (VTC-Spring)*, Jun. 2022, pp. 1–5.

[26] J. Kang, D. Ye, J. Nie, J. Xiao, X. Deng, S. Wang, Z. Xiong, R. Yu, and D. Niyato, "Blockchain-based federated learning for industrial metaverses: Incentive scheme with optimal AoI," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Aug. 2022, pp. 71–78.

[27] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 153–161.

[28] V. T. Truong, H. D. Le, and L. B. Le, "Trust-free blockchain framework for AI-generated content trading and management in metaverse," *IEEE Access*, vol. 12, pp. 41815–41828, 2024.

[29] V. T. Truong and L. Bao Le, "A blockchain-based framework for secure digital asset management," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 1911–1916.

[30] Y. Fu, C. Li, F. R. Yu, T. H. Luan, P. Zhao, and S. Liu, "A survey of blockchain and intelligent networking for the metaverse," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3587–3610, Feb. 2023.

[31] V. T. Truong, D. N. M. Hoang, and L. B. Le, "BFLMeta: Blockchain-empowered metaverse with Byzantine-robust federated learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 5537–5542.

[32] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[33] A. M. Antonopoulos and D. A. Harding, *Mastering Bitcoin*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.

[34] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum, Zug, Switzerland, Yellow Paper 151, pp. 1–32, 2014.

[35] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 3–16.

[36] Y. Wang, H. Peng, Z. Su, T. H. Luan, A. Benslimane, and Y. Wu, "A platform-free proof of federated learning consensus mechanism for sustainable blockchains," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3305–3324, Dec. 2022.

[37] F. Saleh, "Blockchain without waste: Proof-of-stake," *Rev. Financial Stud.*, vol. 34, no. 3, pp. 1156–1190, 2021.

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

[39] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99, Mar. 1999, pp. 173–186.

[40] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4601–4613, Jun. 2019.

[41] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. Italian Conf. Cybersecur.*, vol. 2058, 2018, pp. 1–11.

[42] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[43] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. Int. Conf. Manage. Data*, Jun. 2019, pp. 123–140.

[44] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. ACM EuroSys Conf.*, 2018, pp. 1–15.

[45] E. Daniel and F. Tschorsch, "IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 31–52, 1st Quart., 2022.

[46] S. Zeng, Z. Li, H. Yu, Z. Zhang, L. Luo, B. Li, and D. Niyato, "HFedMS: Heterogeneous federated learning with memorable data semantics in industrial metaverse," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 3055–3069, Jul./Sep. 2023.

[47] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 319–352, 1st Quart., 2023.

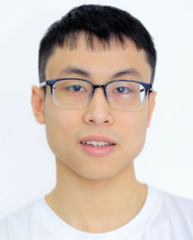
[48] J. R. Douceur, "The Sybil attack," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Germany: Springer, 2002, pp. 251–260.

[49] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 104–121.

[50] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State machine replication for the masses with BFT-SMART," in *Proc. 44th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2014, pp. 355–362.

[51] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 172–181, Jan. 2020.

[52] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *Proc. IEEE Annu. Symp. Found. Comput. Sci.*, Oct. 1999, pp. 120–130.



HUNG DUY LE received the B.Eng. degree in information systems from Hanoi University of Science and Technology (HUST), Vietnam, in 2022. He is currently pursuing the M.Sc. degree with the Institut National de la recherche scientifique (INRS), University of Quebec, Montreal, QC, Canada. His research interests include blockchain and enabling technologies for metaverse and future internet.

VU TUAN TRUONG received the B.Eng. degree in electrical and computer engineering from Hanoi University of Science and Technology (HUST), Vietnam, in 2021. He is currently pursuing the Ph.D. degree with the Institut National de la recherche scientifique (INRS), University of Quebec, Montreal, QC, Canada. His research interests include blockchain, machine learning, and enabling technologies for metaverse, wireless networks, and future internet.

LONG BAO LE (Fellow, IEEE) received the B.Eng. degree in electrical engineering from Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree in telecommunications from Asian Institute of Technology, Thailand, in 2002, and the Ph.D. degree in electrical engineering from the University of Manitoba, Canada, in 2007. He was a Postdoctoral Researcher with the University of Waterloo, from 2007 to 2008, and Massachusetts Institute of Technology, from 2008 to 2010. Since 2010, he has been with the Institut National de la recherche scientifique (INRS), University of Quebec, Montreal, QC, Canada, where he is currently a Full Professor. He is a coauthor of the books *Radio Resource Management in Multi-Tier Cellular Wireless Networks* (Wiley, 2013) and *Radio Resource Management in Wireless Networks: An Engineering Approach* (Cambridge University Press, 2017). His current research interests include smartgrids, radio resource management, network control and optimization, and emerging enabling technologies for 5G-and-beyond wireless systems and the metaverse. He was a member of the Editorial Board of IEEE Transactions on Wireless Communications and IEEE Communications Surveys and Tutorials. He is also an Editor of IEEE Transactions on Communications and IEEE Transactions on Cognitive Communications and Networking.