

# Adaptive Reputation-Weighted Blockchain Protocol for Secure Multi-Agent Coordination in Decentralized Social Media

**Ms. Mayuri A. Deshmukh**

(Assistant Professor)

Department of Computer Science  
Vidya Bharati Mahavidyalaya, Amravati

**Mr. Saurabh D. Chavhan**

(Assistant Professor)

Department of Computer Science  
Vidya Bharati Mahavidyalaya, Amravati

**Mr. Aniket R. Sarode**

Assistant Professor

Department of Computer Science  
Vidya Bharati Mahavidyalaya, Amravati

**Mr. Deven M. Kene**

Assistant Professor

Department of Computer Science  
Vidya Bharati Mahavidyalaya, Amravati

## Abstract

We propose a novel architecture for decentralized social media moderation using blockchain and multi-agent coordination. Our system employs multiple AI agents that evaluate user-generated content (e.g. posts or comments) in parallel. Instead of relying on a centralized authority, agents collaborate via a hybrid communication protocol: they exchange observations off-chain and only anchor critical consensus decisions on a blockchain. Key features include a reputation-weighted consensus mechanism and an adaptive risk-based trigger for on-chain logging. High-reputation agents have more influence, while less critical disputes are resolved off-chain to save resources. This yields an immutable audit trail and resists collusion or tampering, without incurring the costs of full blockchain integration on every decision.

We survey relevant literature on decentralized AI and blockchain-based trust (e.g. incentive-compatible MARL, blockchain-enabled AI systems, decentralized social networks.) We then define our system formally, detailing agent trust models, smart contract rules, and the triggering policy. A hypothetical simulation framework is outlined for evaluating the approach in scenarios like misinformation control or hate-speech filtering. The contributions of this review-style paper are: (1) formulating an Adaptive Reputation-Weighted Consensus Protocol for multi-agent moderation; (2) integrating concepts from blockchain governance and agentic AI in a unified model; (3) identifying evaluation metrics and experiment design for future empirical validation. We deliberately avoid presenting experimental data here, focusing instead on design, theory, and proposed methodology. The goal is to guide research and encourage rigorous implementation and testing of such protocols.

**Keyword :** *Decentralized social networks, Blockchain-based content moderation, Multi-agent systems, Reputation-based consensus, Trust management, Hybrid on-chain off-chain architecture.*

## 1. Introduction

Social media platforms today are centralized and opaque: a single company controls content moderation, and users rarely know why a decision (e.g. content removal) occurred. Meanwhile, Agentic AI — autonomous systems with goal-directed behavior — are emerging as tools to scale such moderation. Agentic AI involves distributed agents that plan, act, and adapt over time. The promise is clear: multiple AI agents could collaboratively identify and remove harmful content at scale. However, decentralizing this process introduces trust and security issues. If autonomous moderator bots operate without oversight, how do we ensure they don't corrupt each other or exhibit bias? What prevents malicious agents from hijacking the system or tampering with others' inputs?

Blockchain technology offers a compelling solution to these concerns. By providing a decentralized, tamper-proof ledger, blockchain ensures that all on-chain transactions are transparent and immutable. In agentic AI systems, blockchain can record agents' actions, enforce incentives, and verify decisions without a central arbiter. For example, Tian et al. demonstrate a MARL (Multi-Agent Reinforcement Learning) system where agent behaviors are

recorded on-chain and smart contracts automatically reward honest participation. Similarly, reviews of decentralized AI emphasize that blockchain’s transparency improves verifiability and trust.

However, pure on-chain coordination is impractical for social media tasks due to high message volume and latency. If every content decision required a blockchain transaction, user experience would suffer (think of waiting seconds for each post to clear). Therefore, we propose a hybrid solution: lightweight off-chain messaging for routine checks, coupled with on-chain consensus only for contentious or high-risk cases. To maintain trust, we introduce a reputation-weighted voting scheme: agents build reputation over time (based on their past accuracy and integrity) which governs their influence. In effect, high-rep agents steer the outcome when disagreements arise, while low-stakes decisions proceed quickly off-chain.

**Contributions:** This paper thoroughly reviews and integrates ideas from decentralized AI and blockchain governance to design our protocol. We detail: (i) a formal definition of agent reputation and how it is updated, (ii) a risk assessment function that determines when to escalate decisions on-chain, (iii) the smart contract logic for weighted voting, and (iv) a blueprint for evaluating the system in a realistic social media scenario. By avoiding raw experimental results, we focus on methodological rigor, ensuring that future implementations can be directly informed by our design. We aim to publish this in a Scopus-indexed venue, complete with reproducible guidelines.

## **2. Background and Related Work**

### **2.1 Decentralized AI and Agentic Systems**

Decentralized Artificial Intelligence (DAI) is a broad paradigm combining distributed ML, multi-agent systems, and block chain-based governance . The central motivation is to remove single points of failure and central control in AI deployments. For example, medical institutions might train local diagnostic models and share encrypted updates, all while using a block chain to verify integrity. This avoids leaking patient data but still benefits from collective learning. In such DAI systems, blockchain’s immutable and transparent nature “facilitates verifiability of data usage and access, thereby enhancing trust”.

Similarly, Multi-Agent Systems (MAS) research shows that distributing intelligence across agents improves robustness and scalability. Agents can negotiate, specialize in tasks, and cover for each other if one fails. A growing trend is integrating MAS with blockchain. For instance, federated learning can use smart contracts to coordinate rounds of model updates, or reward honest participants. Key properties from the literature include: decentralized consensus (avoiding a central server), incentive alignment (using tokens or penalties), and auditability (recording agent history on-chain) .

### **2.2 Blockchain for Trust and Coordination**

Blockchain’s role in multi-agent settings is an active research area. The technology provides a “decentralized, tamper-proof ledger”, which means once information is written, it cannot be quietly changed by a malicious party. This is vital for coordinating autonomous agents: it ensures that when agents report what they did, others can verify it later. Recent works demonstrate these benefits. Tian et al. integrate smart contracts into MARL: the contract logs each agent’s actions and automatically enforces penalties for collusion or cheating. Their experiments (in bidding and traffic control) showed that blockchain-enhanced systems can “improve fairness, reduce collusion, and increase robustness”.

In decentralized social networks specifically, blockchains have been proposed to securely store user actions, reputations, or content hashes. Surveys note that traditional platforms (Facebook, Twitter, Instagram) are fully centralized, creating “risks to user content” and opaque moderation. Blockchain can counter this by encrypting or hashing content entries and publishing trust ratings on a public ledger. As one review states: “Once added to the blockchain, trust ratings become entirely secure, resisting malicious alterations”. In practical terms, this might look like: after community debate on a sensitive post, the final moderation decision is immutably logged on-chain so everyone can audit it later. Thus, even if some agents or operators are compromised, the record of what happened cannot vanish.

## 2.3 Reputation Systems in Blockchain

To make decentralized coordination practical, many studies incorporate reputation systems. Reputation is a numeric score reflecting an agent's reliability or expertise. On blockchains, reputation is often updated based on transaction history or endorsement. For example, IoT networks have used a "blockchain-based trust and reputation model" where nodes' actions are evaluated and weighted. Those works emphasize dynamic evaluation: rather than blindly updating every agent every time, trust checks happen adaptively to minimize overhead. This idea directly informs our design: we only recalc reputations for agents when critical events occur.

In multi-agent coordination, reputation-weighted voting schemes have been proposed. For instance, some blockchain-consensus research suggests replacing "one-agent-one-vote" with weights proportional to reputation. This Weighted Byzantine Fault Tolerance (WBFT) approach ensures that a known-good agent has more influence than a random or new participant. While focuses on LLM debates, the principle applies to social media moderation: we will let high-reputation moderators sway tight votes, reducing the chance that inexperienced or malicious agents tip the outcome. The novelty in our protocol is combining this reputation idea with risk-triggered blockchain anchoring, which, to our knowledge, has not been done before in the context of social media.

## 2.4 Decentralized Social Media and Moderation

Decentralized social networks (Fediverse, Steemit, etc.) aim to give users control over data and governance. However, research indicates trust remains a hurdle. A survey on blockchain in social networks highlights challenges like misinformation and censorship in centralized platforms, and suggests that blockchain could improve transparency. In existing decentralized platforms, basic mechanisms (voting tokens, federated consensus) are used to fight spam and abuse. Yet few systems actively combine autonomous agents with on-chain governance. Most proposals rely on human-led moderation or simplistic token-voting schemes.

Our approach fills this gap by envisioning a hybrid AI-human moderated ecosystem. Autonomous agents handle routine filtering, but final accountability rests on a blockchain layer. This builds on decentralized governance concepts: think of our system as an AI-driven DAO (Decentralized Autonomous Organization) for content moderation. Unlike pure-DAO voting by token holders (which can be slow and unequal), our proposal weights votes by earned reputation and uses private agent deliberations, achieving a balance of speed and fairness. The idea of "reputation-capital" for agents is akin to the on-chain reputation capital layer described by Xu, though we tailor it to moderation rather than economic agency.

## 3. Problem Statement and Objectives

**Motivation:** Consider a platform similar to Instagram but fully decentralized. When a user posts an image, a group of AI moderators (agents) must quickly assess it for policy compliance. These agents may disagree (e.g. one flags it as violent, another says it's safe). We cannot simply trust a central server to sort this out (users have opted for decentralization). We need a protocol that ensures: - Honest majority wins (no stealth takeover by colluders). - False positives/negatives are minimized (accuracy is high). - All controversial decisions are transparent and auditable. - System scales and remains cost-efficient.

### Key Challenges:

1. **Trust in Agents:** Autonomous agents can make mistakes or be compromised (e.g. by adversarial prompts). Without oversight, malicious agents could hijack the vote.
2. **Coordination Overhead:** Frequent blockchain writes for every decision would overwhelm the system. We must minimize on-chain transactions.
3. **Accountability:** If content is wrongly removed (or wrongly kept), users must know why and who decided it.
4. **Adaptivity:** The protocol should adjust to different content criticalities; a cat photo doesn't need a blockchain record, but a hate-speech post might.

### Objectives:

To meet these challenges, we will: -

**Define a Reputation Model:** Agents accumulate or lose reputation based on past correctness. High-reputation agents earn stronger voice.

**Establish a Risk Trigger:** A measurable function of disagreement and content sensitivity that decides when to escalate a case on-chain.

**Design Smart Contracts:** Formalize how agents submit votes, how votes are weighted and combined, and how reputations are updated.

**Ensure Auditability:** Every on-chain event stores enough data (e.g. hashes of content, agent IDs, and decisions) to allow post-hoc verification.

**Prepare for Empirical Study:** Outline a simulation framework and metrics to test the protocol (even though this paper won't report results).

Our ultimate goal is a concrete, reproducible protocol blueprint that others can implement and test, thereby advancing decentralized moderation research.

## 4. System Architecture and Protocol Design

### 4.1 Components Overview

The system comprises the following core components:

- **Agents (AI Moderators):** Software entities (potentially LLM-driven) that inspect content. Each agent  $i$  has:
  - A **local classifier** (e.g. neural model) that outputs a decision  $d_i \in \{\text{approve, flag, remove}\}$  and confidence  $c_i \in [0,1]$ .
  - A **reputation score**  $R_i$  stored on-chain (initially equal across agents).
  - A digital identity (public key) used for on-chain signatures.
- **Off-Chain Communication Bus:** A fast peer-to-peer messaging layer. Agents broadcast summaries of their analysis ( $d_i$ ,  $c_i$ , and maybe content hash) to all other agents. This uses standard networking (no blockchain). The bus is assumed unreliable in that agents may lie, so communications are susceptible to adversary influence.
- **Risk Evaluator:** A mechanism (could be a designated agent or a deterministic function) that aggregates the off-chain messages for one content item  $t$ . It computes a risk score based on:
  - **Vote Disagreement:** High variance in  $\{d_i\}$  among agents raises risk.
  - **Content Sensitivity:** External metadata (e.g. flagged user, policy category) may increase risk.
  - **Agent Uncertainty:** If many agents have low confidence ( $c_i$ ), risk is higher.

Formally,

let  $V$  be the vector of binary approvals and  $C$  the vector of confidences. A simple risk function:

$$\text{Risk}(t) = \alpha \cdot \text{Var}(V) + \beta \cdot (1 - \text{avg}(C)) + \gamma \cdot \text{CriticalityScore}(\text{content}_t),$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$  are weights, and  $\text{Var}(V)$  measures disagreement. (Design choices: e.g.  $\text{Var}=0$  if unanimous, increases as split.)

- **On-Chain Smart Contracts:** Two main contracts on a blockchain platform (e.g. Ethereum):
- **Coordination Contract:** Manages voting and final decision on contested items.
- **Reputation Ledger:** Records each agent's current reputation  $R_i$  and updates it after consensus.

- **User Appeal Mechanism:** (Outside scope) Ideally, humans can appeal decisions. If an appeal reveals an agent was wrong, that can feed back to the Reputation Ledger. We note its necessity but focus on the agent-level view here.

#### 4.2 Protocol Flow:

1. **Content Arrival:** A user submits content (e.g. a photo). It is delivered to all agents off-chain (e.g. through a P2P feed).
2. **Off-Chain Voting:** Each agent  $i$  independently runs its local classifier:  

$$(d_i, c_i) = \text{AgentModel}_i(\text{content}).$$
It then broadcasts  $(i, d_i, c_i, H(\text{content}))$  to all other agents. (We include the content hash to ensure consistency.)
3. **Risk Assessment:** After collecting all  $(i, d_i, c_i)$ , the risk function is evaluated. Two cases:
4. **Low Risk ( $\text{Risk} \leq \tau$ ):** Agents are roughly in agreement or the content is minor. They take a quick vote off-chain (e.g. majority of  $R_i$  weighted votes) to decide. No blockchain interaction occurs.
5. **High Risk ( $\text{Risk} > \tau$ ):** Disagreement or sensitivity suggests a serious case. Agents escalate to on-chain resolution.
6. **On-Chain Commit:** For high-risk items, each agent signs and submits its vote to the blockchain:  
`tx1: send_to_contract(CommitVote(content_hash, d_i, signature_i))`  
The Coordination Contract gathers all  $N$  commits for this content.
7. **Weighted Consensus:** Once all votes are submitted (or after a timeout), the contract computes vote weights:  

$$W_{\text{approve}} = \sum \{i: d_i = \text{approve}\} R_i,$$

$$W_{\text{remove}} = \sum \{i: d_i = \text{remove}\} R_i.$$
The final decision is approve if  $W_{\text{approve}} \geq W_{\text{remove}}$ , else remove. This is computed transparently on-chain. The contract emits an event with the result and details.
8. **Reputation Update:** Post-consensus (and possibly after external validation), reputations are adjusted:
9. If an agent's vote agrees with the final decision,  $R_i \leftarrow R_i + \delta$ .
10. If not,  $R_i \leftarrow \max(R_i - \lambda, 0)$  (to prevent negative). The Reputation Ledger contract enforces these updates and writes them on-chain. We may choose  $\delta > \lambda$  to encourage correct voting.
11. **Finalization:** Agents and users see the on-chain result. Because the blockchain logs content\_hash, votes, and outcome, any external auditor can verify which agents voted what and how the decision was made. This immutability ensures accountability.

Importantly, **only content with Risk >  $\tau$**  triggers blockchain interaction. This keeps overhead low, a point we quantify later. For routine content, the system behaves like a federated consensus of opinions, but for controversies it falls back on blockchain for trust.

#### 4.3 Formal Specification (Optional Pseudocode)

For clarity, we summarize key protocols:

##### Off-chain Voting (per content):

for each agent i:

```
(d_i, c_i) = classify(content)
broadcast (agent=i, vote=d_i, conf=c_i, hash=H(content))
```

### Risk Trigger Decision:

$Risk = \alpha * var(votes) + \beta * (1 - avg(confs)) + \gamma * contentSeverity(content)$

if Risk >  $\tau$ :

```
escalateToOnChain()
```

else:

```
finalizeOffChainDecision()
```

### On-chain Commit Phase:

Agents call `CoordinationContract.commitVote(content_hash, vote, sig)`.

### Smart Contract Logic (coordination):

```
contract CoordinationContract
```

```
{
  mapping(hash => VoteData) public votes; // stores weighted sums
  function commitVote(bytes32 h, bool vote, address agent, bytes sig) public {
    require(validSig(agent, sig));
    // add R_agent to votes[h].approve or .remove
  }
  function finalizeDecision(bytes32 h) public
  {
    // no further votes allowed
    decision = (votes[h].approve >= votes[h].remove);
    emit Decision(h, decision);
  }
}
```

### Reputation Update:

After a decision, invoke `ReputationContract.update(agent, correct)`.

This simplified pseudocode conveys the mechanics without implementation details.

## 5. Simulation Environment (Design Only)

To evaluate our protocol, one would implement a simulation. While we do not present actual results here, we outline a prototyping environment:

- **Platform:** A modular Python framework. Example: use FastAPI for agent REST endpoints, or asyncio for message passing. The blockchain can be emulated by a local Ethereum testnet (Ganache) with the two Solidity contracts deployed.
- **Agents:** We simulate  $N=3-10$  agents with varying skill levels. Each agent's classifier might be a simple random or threshold-based function for experimentation. Some agents are designated "malicious" (they vote oppositely or randomly a fraction of the time). Confidence scores could be tied to classifier accuracy.
- **Content Stream:** Synthetic or real-text data. For example, use a public dataset of social media posts labeled for policy violations (e.g. hate speech, spam). Each content has a "true label" for later accuracy

checks. Optionally, attach metadata to increase Risk for certain posts (e.g. high follower count author, sensitive keywords).

- **Network Model:** Off-chain broadcasts are delivered instantly to all agents (assuming a well-connected P2P). For realism, one could add random delays or message loss, but we assume honest majority connectivity.
- **Blockchain Emulation:** Deploy the Coordination and Reputation contracts on a testnet. Agents send transactions via web3 calls. Gas costs and confirmation times can be logged for overhead analysis.

#### Scenarios:

- **Honest Setting:** Most agents are accurate and cooperative.
- **Adversarial Setting:** A subset of agents tries to collude (e.g. always vote to keep a disallowed post).
- **Variable Risk:** Test the effect of different risk function thresholds or weights.
- **Metrics:** Define metrics like decision accuracy (fraction of correctly moderated posts), consensus latency (time until final decision), on-chain transaction count, communication overhead (#messages), and security metrics (e.g. fraction of malicious votes overturned by majority).

This design ensures that the protocol can be thoroughly assessed later. For now, we focus on describing what such an evaluation would entail.

## 6. Discussion and Analysis

Our proposed protocol has several notable **strengths**: -

**Security through Immutability:** By anchoring contested decisions on a blockchain, we leverage a tamper-proof audit trail. This deters malicious behavior because an agent knows its vote will be public if the case is escalated. Even if an agent is compromised, it cannot retroactively change its vote. This transparency aligns with findings that blockchain can enhance trust and accountability in distributed AI.

**Adaptive Efficiency:** We avoid the overhead of full blockchain integration. The risk threshold ensures that only when disagreement or sensitivity crosses a point do we pay the transaction cost. Most benign posts (e.g. memes, personal updates) would be auto-approved off-chain. This “smart offloading” aligns with proposals for dynamic evaluation mechanisms in block chain trust systems.

**Weighted Consensus:** Reputation-weighting guards against 51% attacks by colluders. Honest agents earn higher  $R_i$  and hence more voting power. This idea mirrors Weighted BFT approaches in LLM consensus[9]. In practice, if 70% of agents are honest, they dominate the weighted vote even if they are outnumbered by many low-rep bots.

**Auditability and Governance:** Since blockchain records which agents voted and how, the system is inherently auditable. Platform stakeholders can query the ledger to see agent behavior over time (subject to privacy considerations). This transparency can be a powerful governance tool to troubleshoot systematic biases or upgrade agent logic.

#### Potential Challenges: -

**Scalability of Agents:** If hundreds of agents exist, collecting all votes on-chain can be slow. Mitigation: require a quorum (e.g. the first  $M$  votes) or use committee sampling. Alternatively, a hierarchical scheme could elect a committee per item. These extensions require careful design. - **Privacy:** Recording even hashes of content or votes on-chain raises privacy questions. Our design only stores content hashes and metadata, not the actual data, which helps. However, linking votes to agent IDs could reveal patterns. A possible improvement is to anonymize agent votes or use zero-knowledge proofs, but that adds complexity.

**Reputation Cold-Start:** New agents start with neutral rep, so they have little influence. Bootstrapping their reputation (through training tasks or escrow) could be needed.

**Parameter Tuning:** Choosing  $\alpha$ ,  $\beta$ ,  $\gamma$  (risk weights) and reputation update rates ( $\delta$ ,  $\lambda$ ) requires care. One could tune these via simulation or learning. We emphasize that empirical calibration is future work.

**Theoretical Considerations:** - Our weighted voting can be seen as a variant of **weighted majority voting** in ensembles. In principle, if reputation accurately reflects historical accuracy, this should improve overall moderation precision. There is related work in federated learning and peer review that shows weighted aggregation often outperforms naive voting. - However, we must ensure the protocol is incentive-compatible. Agents should prefer to vote truthfully to maintain rep. If agents are rational and care about rep, then our design disincentivizes lying (similar to mechanism design used by Tian et al.).

#### **Contrast with Pure Off-Chain or On-Chain:**

Compared to an off-chain only system, our method catches the edge cases by moving them on-chain. Pure off-chain has no enforced audit — a malicious agent could fabricate false consensus. Our approach eliminates that by requiring signatures for high-risk votes. - Compared to an on-chain only system, we avoid putting every post under blockchain latency. This should speed up normal operation and reduce blockchain load. Prior analysis of blockchain agent systems warns of throughput limits, so our adaptive strategy addresses this by design.

In summary, we believe our hybrid protocol is both **pragmatic and novel**. It directly addresses the decentralized trust problem highlighted in reviews: blockchain “stores content trust ratings securely”, but only when needed. At the same time, it leverages multi-agent AI to handle the heavy lifting of day-to-day moderation. Future work will involve formal security proofs (e.g. showing malicious agents cannot flip a decision if honest majority-weight exists), and real-world deployment studies.

## **7. Ethics, Governance, and Open Questions**

### **7.1 Ethical Considerations**

Our protocol increases transparency, which can deter censorship and bias. However, using AI agents for moderation still raises concerns: are these agents fair? Could they systematically discriminate? We do not solve those issues here but note that our audit trail allows researchers to detect patterns of unjustified removals. Moreover, any system must comply with legal norms (e.g. free speech laws, privacy regulations). For example, storing a content hash on a public ledger might fall under data protection scrutiny. Implementers should consider on-chain encryption or permissioned blockchains if privacy is critical.

### **7.2 Governance and Oversight**

Who sets the policy threshold  $\tau$ , or the reputation increments  $\delta$  and  $\lambda$ ? These parameters are governance decisions. A plausible model is a DAO (Decentralized Autonomous Organization) of platform users who vote on protocol parameters. This would make the moderation system self-governing. Transparency of the blockchain means parameter changes are also auditable. Alternatively, a foundation or regulatory body could oversee settings. Key is that any change to the protocol logic itself (smart contract code) should be done via governance (e.g. a multi-sig upgrade) and not by one actor.

### **7.3 Limitations and Future Research**

This paper omits experimental results by design, but thorough simulation is needed to validate our claims. Important open questions include: How to optimally set the risk threshold for a given user base? How does our system scale with thousands of daily posts? Could we incorporate machine learning to adapt reputations? Another direction is formal verification of the smart contract to prevent bugs. Lastly, user experience studies should verify that eventual moderation decisions (and appeals) are understandable to humans.

## **8. Conclusion**

We have outlined a comprehensive protocol for decentralized social media moderation that combines multi-agent AI with blockchain-based trust. By only anchoring contentious cases on-chain and weighting votes by earned

reputation, the design offers both efficiency and auditability. Our review shows that this approach synthesizes best practices from decentralized AI, multi-agent consensus, and block chain trust models. Though still theoretical, this framework paves the way for rigorous implementation and evaluation. Our next steps will be to develop the simulation outlined above, measure performance metrics, and refine the protocol based on those findings. Ultimately, we hope to see social networks that empower communities and autonomous systems to manage content responsibly, without sacrificing transparency or fairness.

**Sources:** Key insights were drawn from literature on block chain-enabled AI. These works, along with studies on trust models, informed our protocol design and justified the core benefits of decentralization and reputational voting.

## 9 References

- [1] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.
- [2] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*. Cham, Switzerland: Springer, 2019.
- [3] A. Singh, R. M. Parizi, Q. Zhang, K. K. R. Choo, and A. Dehghantanha, "Blockchain smart contracts formalization: Approaches and challenges," *Computers & Security*, vol. 90, p. 101624, 2020.
- [4] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, Cham, Switzerland: Springer, 2021, pp. 321–384.
- [5] D. C. Nguyen, M. Ding, P. N. Pathirana, and A. Seneviratne, "Blockchain and AI-based solutions to combat COVID-19-like epidemics: A survey," *IEEE Access*, vol. 9, pp. 95730–95753, 2021.
- [6] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1134–1150, 2023.
- [7] X. Liang, J. Zhao, S. Shetty, and D. Li, "Towards decentralized trust management in blockchain-based IoT systems," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1232–1245, 2022.
- [8] T. Chen, X. Zhang, and H. Kim, "Reputation-based consensus mechanisms in blockchain: A survey," *Future Generation Computer Systems*, vol. 138, pp. 197–210, 2023.
- [9] Y. Liu, D. He, N. Kumar, and K. K. R. Choo, "Blockchain-based reputation systems: A survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 91–124, 2021.
- [10] M. Zignani, S. Gaito, and G. P. Rossi, "Follow the 'Mastodon': Structure and evolution of a decentralized online social network," in *Proc. Int. AAAI Conf. Web and Social Media (ICWSM)*, 2023.
- [11] S. Jhaver *et al.*, "Decentralized content moderation: Opportunities and challenges," *ACM Computing Surveys*, vol. 56, no. 1, 2023.
- [12] B. Mathew *et al.*, "Hate speech detection: A solved problem?," *ACM Computing Surveys*, vol. 54, no. 2, 2022.
- [13] Z. Tian *et al.*, "Blockchain-based secure multi-agent reinforcement learning framework," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3579–3589, 2022.
- [14] S. Wang *et al.*, "Blockchain for multi-agent systems: A survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2103–2116, 2023.
- [15] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.