AIArena: A Blockchain-Based Decentralized AI Training Platform

Zhipeng Wang
Imperial College London
London, United Kingdom
zhipeng.wang20@imperial.ac.uk

Rui Sun
Newcastle University
Newcastle upon Tyne, United
Kingdom
rui.sun@newcastle.ac.uk

Elizabeth Lui
FLock.io
London, United Kingdom
elizabeth@flock.io

Tuo Zhou
The University of Hong Kong
Hong Kong, China
zhoutuo@connect.hku.hk

Yizhe Wen
FLock.io
London, United Kingdom
yizhe@flock.io

Jiahao Sun
FLock.io
London, United Kingdom
sun@flock.io

Abstract

The rapid advancement of AI has underscored critical challenges in its development and implementation, largely due to centralized control by a few major corporations. This concentration of power intensifies biases within AI models, resulting from inadequate governance and oversight mechanisms. Additionally, it limits public involvement and heightens concerns about the integrity of model generation. Such monopolistic control over data and AI outputs threatens both innovation and fair data usage, as users inadvertently contribute data that primarily benefits these corporations.

In this work, we propose AIArena, a blockchain-based decentralized AI training platform designed to democratize AI development and alignment through on-chain incentive mechanisms. AIArena fosters an open and collaborative environment where participants can contribute models and computing resources. Its on-chain consensus mechanism ensures fair rewards for participants based on their contributions. We instantiate and implement AIArena on the public Base blockchain Sepolia testnet, and the evaluation results demonstrate the feasibility of AIArena in real-world applications. Our project page is available at https://train.flock.io/explore.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Theory of computation** → **Design and analysis of algorithms**.

Keywords

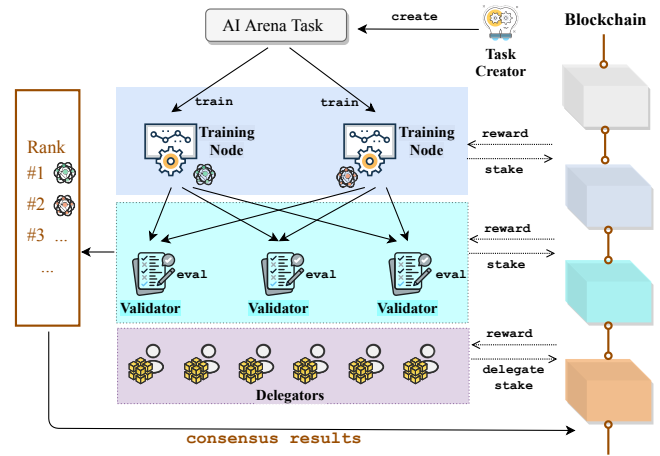Blockchain; Machine Learning; Decentralized Artificial Intelligence

Figure 1: Overview of AIArena, a blockchain-based decentralized AI training system.

1 Introduction

The growth of artificial intelligence (AI) has led to groundbreaking applications in many fields. However, it also brings challenges, especially because a few large companies control most of its development and use. This concentration of power increases bias in AI systems [1] and reduces public involvement in important decisions. This lack of transparency can exacerbate issues such as misuse of AI technologies or unethical practices, as there is little external oversight in the AI centralized systems. Such centralized control also slows down innovation in AI [4, 7] and leads to the unfair use of user data, which mainly benefits these companies. These compounded challenges underline the urgent need for a shift to decentralized AI (DeAI) [9], where control and access are distributed across a broader and more inclusive range of participants.

Blockchain's decentralized nature can help address the above challenges by allowing multiple participants to collaborate on AI development without relying on a central authority [2, 3]. Smart contracts [11] can automate processes such as distributing rewards or verifying contributions, fostering fairness and efficiency. Blockchain also enhances data integrity, as participants can verify the source and ownership of data, preventing unauthorized use and ensuring compliance with ethical standards [8].

In this work, we propose AIArena, a blockchain-based platform designed to decentralize AI training in an open and collaborative environment, where participants can contribute models and computing power. Specifically, AIArena leverages on-chain consensus mechanisms to ensure that only valid contributions are rewarded, while incentivization models encourage active and meaningful participation to prevent issues such as free-riding or fake identities.

Contributions. Our contributions are summarized as follows:

- We propose AIArena, a blockchain-based decentralized platform for AI training. AIArena enables the AI model to be optimized directly on users' devices using their own data or publicly available datasets, and finally derives the optimal model based on blockchain consensus. By leveraging blockchain, AIArena ensures sustained machine learning (ML) contributor (i.e., training node and validator) engagement and provides fair rewards based on their contributions in enhancing models.

- We introduce a delegation stake design for AIArena, enabling users with limited computational resources to participate in model generation by delegating their stake to preferred ML participants. This design encourages broader involvement in the ML model generation process, further promoting decentralization.

- We instantiate and implement AIArena on the public Base blockchain Sepolia testnet[1]. Over an approximate 7-month period, AIArena has engaged 603 training nodes, 1,051 validators, and 63,265 delegators, collaboratively generating 18,656 models across 16 training tasks. These results highlight the practical feasibility of AIArena in real-world applications.

- We evaluate three popular tasks performed by AIArena contributors. Their models not only outperform baseline models but also exceed the performance of larger state-of-the-art (SOTA) models. Notably, in the code co-creation task, our contributors collaboratively curated the largest existing Move code dataset, along with their corresponding code instructions and comments.

2 AIArena System Design

2.1 Participants

Fig. 1 shows the overview of AIArena system. There are various categories of participants in the AIArena system:

- **Task Creators:** Task creators are responsible for defining training tasks and their specific requirements, including selecting and designing appropriate learning algorithms for model training and validation. To promote decentralization, the created tasks can be reviewed and verified by the stakeholders within AIArena.

- **Training Nodes:** Given a created task, training nodes are responsible for training the model on their service using public data. To participate, training nodes must stake blockchain assets or tokens to establish their eligibility. Training nodes will receive rewards based on their stake amount and performance.

- **Validators:** Validators are responsible for assessing the work completed by training nodes and submitting validation scores that influence the distribution of rewards. To participate, validators must stake tokens, which not only allows them to validate assigned tasks but also ensures fair task allocation.

- **Delegators:** Delegators contribute to the AIArena system by supporting the staking process of other participants without directly engaging in task training or validation. They stake tokens on behalf of other participants, boosting the delegatees' capacity to earn greater rewards. In return, delegators share in the rewards earned by their delegatees, according to predefined algorithms that reflect their staked contributions.

2.2 Training and Validation

2.2.1 Training. We consider the dataset held by the training node, $\mathcal{D}_{\text{local}}$, which contains locally sourced data samples, comprising feature set X and label set Y, with each sample $x_i \in X$ corresponding to a label $y_i \in Y$. We define a predictive model f, aiming to learn patterns within $\mathcal{D}$ such that $f(x_i) \approx y_i$.

To quantify the prediction metric, e.g., accuracy, the task trainer will introduce a loss function $L(f(x_i), y_i)$, assessing the discrepancy between predictions $f(x_i)$ and actual labels y_i. A generic expression for this function is: $L = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i)$, where N denotes the total sample count, and l signifies a problem-specific loss function, e.g., mean squared error or cross-entropy loss.

The optimization goal is to adjust the model parameters θ to minimize L, typically through algorithms such as gradient descent: $\theta_{\text{new}} = \theta_{\text{old}} - \eta \nabla_\theta L$, where η represents the learning rate, and $\nabla_\theta L$ the gradient of L with respect to θ. Utilizing the aggregated dataset $\mathcal{D}$, parameter θ is iteratively updated to reduce L, consequently improving the model's predictive accuracy. This optimization process is conducted over a predefined number of epochs E, each epoch consisting of a complete pass through the entire dataset $\mathcal{D}$.

2.2.2 Validation. Consider a selected group of validators, denoted as $V_j \in V$, each equipped with the evaluation dataset $\mathcal{D}_{\text{eval}}$ from the task creator. This dataset consists of pairs (x_i, y_i), where x_i represents the features of the i-th sample, and y_i is the corresponding true label. The model, trained by designated training nodes, is denoted as θ_p^{task}. The primary objective of θ_p^{task} is to predict the label $\hat{y}_i$ for each feature vector x_i contained within $\mathcal{D}_{\text{eval}}$. To assess the performance of θ_p^{task} on $\mathcal{D}_{\text{eval}}$, we use a general evaluation metric denoted by *eval*. We exemplify with accuracy as follows:

$$eval(\theta_p^{task}, \mathcal{D}_{\text{eval}}) = \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{eval}}} \mathbf{1}(\hat{y}_i = y_i) \qquad (1)$$

Here, $\mathbf{1}$ represents the indicator function that returns 1 if the predicted label $\hat{y}_i$ matches the true label y_i, and 0 otherwise. The function $|\mathcal{D}_{\text{eval}}|$ denotes the total number of samples within the evaluation dataset. Each predicted label $\hat{y}_i$ from the model θ_p^{task} is compared against its corresponding true label y_i within $\mathcal{D}_{\text{eval}}$. The calculated metric result (e.g., accuracy) serves as a quantifiable measure of θ_p^{task}'s effectiveness across the evaluation dataset.

2.3 Consensus and Reward Distribution

We assume there are n submissions $(O_1, \ldots, O_n)$ from n training nodes with stakes $(t_1, \ldots, t_n)$, and m validators $(V_1, \ldots, V_m)$ with stakes $(s_1, \ldots, s_m)$. Each validator $V_j (1 \leq j \leq m)$ evaluates the n models submitted by the training nodes, producing a score vector $\vec{r}_j = (r_{j1}, \ldots, r_{jn})$. These scores reflect the performance of each model according to predefined criteria as shown in Eq. 1.

---

[1]https://sepolia.basescan.org/address/0x7b6bde1d173eb288f390ff36e21801f42c4d8d91

2.3.1 Reward Distribution within One Task. Within a single task, the reward distribution between training nodes and validators is determined based on their relative stake amounts. Let the total daily reward allocated to a task be denoted as R_0. The total rewards for training nodes are: $R_0 \cdot \left(\gamma + (1 - 2\gamma) \cdot \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} t_i + \sum_{j=1}^{m} s_j} \right)$. Similarly, the total rewards for validators are: $R_0 \cdot \left(\gamma + (1 - 2\gamma) \cdot \frac{\sum_{j=1}^{m} s_j}{\sum_{i=1}^{n} t_i + \sum_{j=1}^{m} s_j} \right)$. The parameter γ controls the split rewards, defining the balance between fixed and stake-dependent reward components.

2.3.2 Reward for Training Nodes. The final scores of the submitted model are determined through a weighted aggregation: $\vec{r} = \left(\frac{\sum_j r_{j1} \cdot s_j}{\sum_j s_j}, \ldots, \frac{\sum_j r_{jn} \cdot s_j}{\sum_j s_j} \right)$. This means that the evaluations of validators with higher stakes have a larger impact on the final outcome. We then compute the following geometry series: $g_k = \frac{1-q}{1-q^m} \cdot q^{k-1}$, in which k denotes a given training node's rank amongst its peers in the same task, whereas q is the common ratio of the geometric series and m is the number of training nodes in a given task.

We finally compute the total rewards allocated for the training nodes as well as their delegators, which is based on the quality of their submission and their total amount of stake: $f_i(g_i, t_i) = \frac{g_i \cdot t_i^{\alpha_t}}{\sum_{k=1}^{n} g_k \cdot t_k^{\alpha_t}}$, where t_i the total stake amount from the training node i as well as its respective delegators. α_t is a system parameter that determines the influence of the stake on the reward distribution.

If a training node i's stake in the task is t_n and stakes delegated to training node i is t_d, i.e., $t_i = t_n + t_d$, then the actual reward for training node i is $f_i \cdot \left(\sigma + (1 - \sigma) \cdot \frac{t_n}{t_n + t_d} \right)$. σ is the reward ratio set by the training node, which determines the ratio of rewards shared between the training node i and its respective delegators.

2.3.3 Reward for Validators. For each validator V_j, we compute the distances between their score and the final aggregated score: $\vec{\Delta}_j = (\Delta_{j1}, \ldots, \Delta_{jn}) = \left(\left| \frac{\sum_j r_{j1} \cdot s_j}{\sum_j s_j} - r_{j1} \right|, \ldots, \left| \frac{\sum_j r_{jn} \cdot s_j}{\sum_j s_j} - r_{jn} \right| \right)$.

We define a distribution function f_i, which satisfies: *(1)* $f_i(\Delta_{1i}, s_1) + \ldots + f_i(\Delta_{mi}, s_m) = 1$; *(2)* f_i decreases over the distance Δ_{ji}, and *(3)* f_i increases over the stake amount s_j. To fulfill the three criteria, we can employ a modified version of the Softmax Function: $f_i(\Delta_{ji}, s_j) = \frac{e^{-\lambda_v \Delta_{ji}} \cdot s_j^{\alpha_v}}{\sum_{k=1}^{m} e^{-\lambda_v \Delta_{ki}} \cdot s_k^{\alpha_v}}$. The parameters λ and α play crucial roles. λ_v controls the sensitivity of the function to the distance Δ_{ji}. A higher λ_v increases the function's sensitivity to score accuracy, emphasizing the importance of precise evaluations. α_v determines the influence of the stake amount s_j on the reward distribution.

If the validator finishes multiple (i.e., N) validation tasks, then its reward ratio is $\sum_{i=1}^{n} f_i(\Delta_{ji}, s_j)$. If a validator's stake in the task is s_v, and s_j is its accumulative stake by considering the total delegation amount s_d on this validator, i.e., $s_j = s_v + s_d$, the actual reward ratio for this validator is $\left(\sum_i f_i(\Delta_{ji}, s_j) \right) \cdot \left(\sigma + (1 - \sigma) \cdot \frac{s_v}{s_v + s_d} \right)$, where σ is a parameter set by the delegatee.

2.4 Delegate Staking

Delegators may entrust their tokens to participants of their choosing to receive a passive investment income stream. The receivers can thus amplify their stake and rewards. These rewards are shared with the delegators, to attract users who have tokens but lack the technical expertise to perform AI model training or validation.

Specifically, the delegator rewards depend on: *(1)* The quality of the training nodes or validators selected for delegation, and *(2)* The amount of stake delegator has delegated.

Formally, the the reward for a delegator who delegates to a training node can be calculated as: $f_i \cdot (1 - \sigma) \cdot \frac{t_d}{t_n + t_d}$, whereas f_i refers to the total reward distributed to the training node i and delegator based on the quality of the training node's submission, t_d is the stake amount from this given delegator, t_n is the stake amount from the training node i. Similarly, the reward for a delegator who delegates to a validator can be calculated as: $\left(\sum_i f_i(\Delta_{ji}, s_j) \right) \cdot (1 - \sigma) \cdot \frac{s_d}{s_v + s_d}$ in which s_d refers to the stake amount from a given delegator and s_v is the stake amount of the validator the delegator delegated to.

2.5 Various Validation Phases

To address potential attacks, such as lookup attacks and model-stealing attacks by malicious training nodes, AIArena validators employ diverse validation datasets across different phases. For a task spanning x days, the process is divided into three phases:

(1) **Submission Phase**: This initial phase lasts x_0 days and provides daily rewards to training nodes based on the validator consensus results for each day. The continuous reward mechanism encourages sustained participation by offering consistent feedback and reinforcing contributors' efforts.
(2) **Final Validation Phase**: Lasting x_1 days, this phase utilizes a validation dataset distinct from the one used in the submission phase. This diversification increases the difficulty for malicious actors to exploit predictable validation scenarios, enhancing the system's robustness.
(3) **Challenging Phase**: Spanning $x - x_0 - x_1$ days, this phase allows any validator to issue a challenge if they suspect a model has been stolen. Training nodes must provide proof of legitimate training, such as proof of learning [5, 6] or on-chain address-based watermarking [10]. If they fail to do so, the successful challenger receives the rewards allocated to the malicious training node, incentivizing honest participation.

This phased approach ensures a dynamic and secure validation process, mitigating risks associated with malicious training activities.

3 Implementation and Evaluation

3.1 On-Chain Implementation

We implement AIArena on the public Base blockchain Sepolia testnet. We leverage Solidity to implement the on-chain reward calculation and distribution smart contracts. We run the system from April 30, 2024, to December 9, 2024. For each day, 1074 tokens will be minted and rewarded to the participants, and $\gamma = 0.7$. Table 1 reports the summary statistics of all trained tasks on AIArena.
Number of Participants. In total, we observe that 603 training nodes, 1,051 validators, and 63,265 delegators have participated in the system to train and validate the 16 various tasks, accumulating 18,926 training and 2,225,254 validation submissions (see Fig. 2). Across most tasks, the number of validators consistently exceeds the number of training nodes. For example, in task 10, overall 391 validators participated, compared to only 128 training nodes. Even

Task ID	Task Name	Duration (days)	Training Nodes				Validators				Delegators
			Number	Rewards			Number	Rewards			
				Mean±Std	Min	Max		Mean±Std	Min	Max	
1	D&D MBTI Chatbot	10	5	1879.50±1614.41	11.00	3895.53	6	708.84±63.79	666.05	821.97	5
2	CharacterX - Yae Miko	14	11	1033.72±402.23	537.17	1745.63	9	483.30±306.93	33.45	930.03	3
3	AI Friend - Loki	21	39	635.06±621.60	59.99	2177.59	49	188.55±504.51	0.00	2850.25	21
4	AI Friend - Hutao	21	38	733.90±588.06	124.82	1948.08	46	220.17±322.86	0.00	1533.65	11
5	Farcaster GPT	18	35	334.59±361.29	2.30	1240.02	34	334.13±326.71	0.04	877.44	46
6	Multilingual Query Analyzer	15	27	423.95±352.11	9.10	1263.14	48	204.57±311.66	0.00	1501.77	37
7	Professor Grump	6	13	268.50±151.78	91.08	543.35	38	117.47±133.31	1.27	510.54	17
8	FLock GPT	21	31	443.31±674.22	8.90	2505.02	102	146.40±207.77	0.02	1089.89	36
9	Text2SQL Agent Model	28	153	113.82±296.88	0.00	1600.63	172	95.13±130.40	0.00	659.68	5
10	Text-to-SQL	28	128	21.46±134.26	0.00	1284.67	391	40.20±87.74	0.00	850.28	0
11	Real Life Simulator	28	146	100.69±305.20	0.00	1803.46	365	44.88±90.68	0.01	707.10	0
12	Berachain GPT	28	110	64.50±241.34	0.00	1739.92	456	54.72±85.79	0.01	758.29	0
13	AI Research Agent	32	67	13.98±46.38	0.00	302.88	413	30.03±57.75	0.01	527.84	61345
14	Move Code Co-creation	14	11	5.58±9.34	0.17	30.63	107	13.06±39.90	0.00	269.48	14899
15	Role-play LLM in Korean	27	42	42.40±79.44	0.00	318.89	352	62.39±125.80	0.00	854.83	60830
16	Investment Report Generator	15	24	6.04±10.14	0.05	40.03	98	18.36±40.23	0.03	249.95	18770
	Total	224	603	165.11±403.45	0.00	3895.53	1051	69.80±158.77	0.00	2850.25	63265

Table 1: A total of 16 tasks have been trained and validated over a period of seven months. Starting from Task 13, participants are able to create delegation pools to create delegators, resulting in a significant increase in the number of delegators.

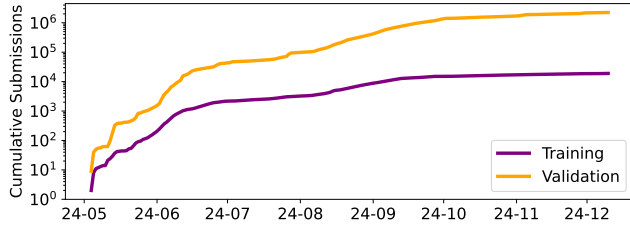for smaller-scale tasks such as task 14, the number of validators outnumbered training nodes.

Figure 2: Training and validation submissions over time.

Reward Distribution. The reward distribution among training nodes and validators reveals interesting patterns. Training nodes generally receive higher rewards per participant compared to validators before in tasks 1-12, due to the smaller number of training nodes. However, the variability in rewards among training nodes is much higher, as evidenced by the standard deviation, which indicates a broader range of effort and contribution levels. This suggests that while training tasks may offer higher potential rewards, they also come with greater uncertainty, whereas validation tasks provide more consistent but relatively lower rewards. Such dynamics could further explain the observed user preference for validation tasks, which offer steadier returns with less variance in earnings.

3.2 Off-Chain Implementation

To demonstrate the feasibility of AIARENA in realistic scenarios, we evaluated three diverse and popular tasks using the methodology proposed by the platform[2]. The results are summarized in Table 2. Meanwhile, we employ the Human Preferences Evaluation method, with the voting results by GPT-4o presented in Fig. 3. The results show that all Rank-1 training nodes' models outperform baseline

[2]https://github.com/FLock-io/llm-loss-validator

LLM	Text-to-SQL (Task 10)	Life Simulator (Task 11)	Code Co-creation (Task 14)		
	$\mathcal{L}$	$\mathcal{L}$	$\mathcal{L}_{\text{CMC}}$	$\mathcal{L}_{\text{MBLC}}$	$\mathcal{L}_{\text{N}}$
Rank-1	0.080	0.648	-	-	0.257
Rank-2	0.084	0.648	-	-	0.272
Rank-3	0.085	0.688	-	-	0.284
Phi3-4B*	1.030	1.073	0.450	0.533	0.504
Qwen2.5-7B*	1.055	1.397	0.379	0.530	0.575
Qwen2.5-14B**	1.090	1.345	0.370	0.512	0.556
Yi1.5-9B*	0.322	1.288	0.631	0.726	0.631

Table 2: Model performance (loss $\mathcal{L}$) across three tasks.
CMC and MBLC denote models finetuned exclusively on the respective datasets: CMC (Co-created Move Code from AIARENA) and MBLC (Move-Bytecode-LLVM-Compiler project from CMC). N indicates the native output of the model without any finetuning. *: Models based on foundational LLMs; **: SOTA, larger-scale LLMs.
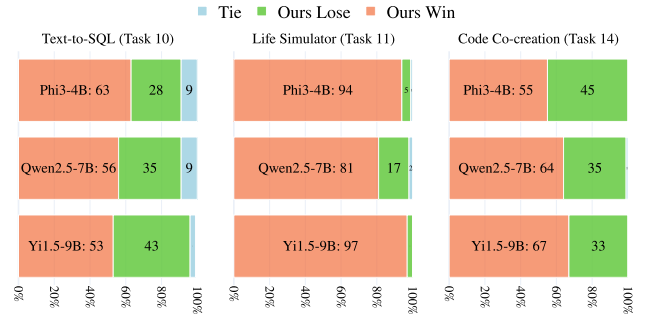

Figure 3: Comparison of the Rank-1 Model in AIARENA with Baselines, evaluated using the Human Preferences Evaluation method judged by GPT-4o.

models, larger SOTA models, and baseline LLMs finetuned on the provided or subset CMC and MBLC datasets.

Text-to-SQL for Onchain Data (Task 10). Text-to-SQL, which translates natural language into SQL queries, has advanced in various domains but remains largely unexplored for complex blockchain

data, a gap that limits accessible, data-driven insights into on-chain transactions, token transfers, and smart contract states. To address this problem, we launched a text-to-SQL task on the AIArena platform, starting from a seed dataset of structured blockchain transaction data, and encouraged participants to enrich it with diverse examples and refined schemas. All participants then fine-tuned LLMs on these enhanced samples, thereby improving their text-to-SQL capabilities specifically for blockchain queries.

Real Life Simulator (Task 11). Life Simulator is a dynamic, LLM-powered game where players guide a character's growth through career, relationships, and personal choices, experiencing unique, adaptive outcomes in a realistic virtual world. However, most current LLMs tend to generate overly optimistic life stories, which deviate significantly from realism and undermine the core purpose of a life simulator. To address this, the AIArena platform has been utilized to gather contributions from the community, enhancing LLMs' ability to generate more realistic and diverse life scenarios.

Code Co-creation (Task 14). Accurate code generation and understanding are critical when leveraging LLMs for advanced automation, particularly in low-resource blockchain languages. Building on this need, we focus our efforts on the Move language, aiming to develop a model capable of generating accurate instructions and comments. Through our AIArena platform, a diverse community of contributors collectively curated what we believe, to the best of our knowledge, to be the largest Move instruction dataset to date (see https://huggingface.co/datasets/flock-io/move-code-comment). Drawing from 514 distinct Move projects, they enriched the source code with detailed instructions and comments.

References

[1] Robert Challen, Joshua Denny, Martin Pitt, Luke Gompels, Tom Edwards, and Krasimira Tsaneva-Atanasova. 2019. Artificial intelligence, bias and clinical safety. *BMJ quality & safety* 28, 3 (2019), 231–237.

[2] Xuhui Chen, Jinlong Ji, Changqing Luo, Weixian Liao, and Pan Li. [n. d.]. When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. In *2018 IEEE Big Data*.

[3] Nanqing Dong, Zhipeng Wang, Jiahao Sun, Michael Kampffmeyer, William Knottenbelt, and Eric Xing. 2024. Defending against poisoning attacks in federated learning with blockchain. *IEEE Transactions on Artificial Intelligence* (2024).

[4] Oren Etzioni. 2016. Democratizing artificial intelligence through open AI models. *AI Magazine* 37, 1 (2016), 50–52.

[5] Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. [n. d.]. Proof-of-learning: Definitions and practice. In *2021 IEEE S&P*.

[6] Yixiao Lan, Yuan Liu, Boyang Li, and Chunyan Miao. 2021. Proof of learning (PoLe): Empowering machine learning with consensus building on blockchains. In *AAAI*.

[7] Gabriel Axel Montes and Ben Goertzel. 2019. Distributed, decentralized, and democratized artificial intelligence. *Technological Forecasting and Social Change* (2019).

[8] Khaled Salah, M Habib Ur Rehman, Nishara Nizamuddin, and Ala Al-Fuqaha. 2019. Blockchain for AI: Review and open research challenges. *IEEE access* (2019).

[9] Zhipeng Wang, Rui Sun, Elizabeth Lui, Vatsal Shah, Xihan Xiong, Jiahao Sun, Davide Crapis, and William Knottenbelt. 2024. SoK: Decentralized AI (DeAI). *arXiv preprint arXiv:2411.17461* (2024).

[10] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. 2020. Model watermarking for image processing networks. In *AAAI*.

[11] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. 2019. Smart contract development: Challenges and opportunities. *IEEE TSE* (2019).