Sustainable broadcasting in Blockchain Network with Reinforcement Learning

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Index Terms*—Blockchain, Ethereum, Propagation protocol, Reinforcement Learning, Sustainability**

***Abstract*—Recent estimates put the carbon footprint of Bitcoin and Ethereum at an average of 64 and 26 million tonnes of CO2 per year respectively. To address this growing problem, several possible approaches have been proposed in the literature: creating alternative blockchain consensus mechanisms, applying redundancy reduction techniques, utilizing renewable energy sources and energy efficient devices etc. In this paper, we follow the second avenue and propose an efficient approach based on reinforcement learning that improves the block broadcasting scheme in blockchain networks. Such an improvement concerns both the block propagation time and the number of messages to be broadcasted before the network reaches consistency. The latter determines the amount of traffic and distributed energy consumption across the blockchain network infrastructure.**

In particular, we implemented an reinforcement learning (RL) agent to efficiently re-prioritize the broadcast order in the default blockchain block propagation protocol based on real-time transport layer network information. This approach reduces the average propagation time as well as the number of messages needed to reach network consistency. Since the blockchain networks are highly distributed around the world, this seemingly small improvement could make a big difference in the context of overall energy consumption and network impact on the environment.

To train the agent and validate the approach, we used a blockchain simulator that was modified by adding a broadcast monitoring interface and was integrated into the standard RL environment. We then ran a series of simulations and general statistical analysis to compare performance of the default block propagation scheme and the scheme with RL-agent involved. The analysis and experimental results confirmed that the proposed improvement of the block propagation scheme could cleverly handle network dynamics and achieve better results than the default approach. Additionally, our technical integration of the simulator and developed RL environment can be used as a complete solution for further study of new schemes and protocols that use RL or other ML approaches.

I. Introduction

Motivation. Bitcoin and Ethereum are the most the most carbon intensive blockchain based P2P payment networks, they consume growing amount of energy (Fig. 1) and emit on average 64 and 26 million tons of CO2 per year, respectively [1], [2]. Besides Bitcoin and Etherium, several other large blockchains produce on average more than 750 tons of CO2 eq. each year [2]. Consequently new environmental, social and

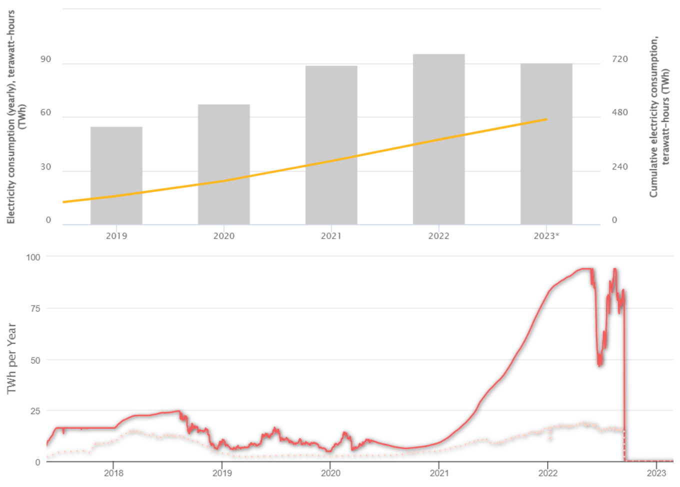

Fig. 1. The Energy Consumption Index of Bitcoin and Ethereum
Note. It shows yearly energy consumption of Bitcoin (upper) and Ethereum (lower) infrastructure measured in terawatt-hours (TWh), data was obtained from https://ccaf.io/cbnsi/cbeci and https://digiconomist.net/ethereum-energy-consumption.

economic challenges confront inventors, researchers and governments [2]. Thus every small improvement on blockchain protocols, schemes and algorithms may lead to significant spatiotemporally distributed effects. Despite that, research in this direction still does not explicitly discuss the environmental consequences of proposed inventions and algorithmic improvements.

State-of-art approaches and our contribution. To address this growing problem, several possible approaches for blockchain networks have been proposed in the literature: creating alternative consensus mechanisms, applying redundancy reduction techniques, utilizing renewable energy sources and energy efficient devices [1]. In this paper, we follow the second avenue and propose an original and efficient approach based on reinforcement learning (RL) that improves the block broadcasting scheme in blockchain-based networks.

The use of the RL-based approach to solve routing problems in different types of networks has been widely discussed in the literature (see for review [3]), but they do not directly address global peer-to-peer networks such as Bitcoin and

Ethereum (see in detail in §II-C). In addition, we want to emphasize the environmental impact of the proposed solution and therefore choose and discuss relevant metrics as a proxy for the environmental impact.

The limitations of existing blockchain routing schemes were recently addressed in [4]. They showed that RL could enhance routing adaptability by selecting more dependable and efficient routes dynamically, but in the context of wireless sensor networks. We extended this concept to Ethereum-related networks, proving that a RL agent can improve broadcasting order effectively.

Experimental methodology and limitations. As we intended to perform cost-effective, controlled, and reproducible experimentation, our approach was tested on the blockchain simulator invented by Faria et al. [5] that was slightly extended by implementing default Ethereum broadcasting mechanism. We validated the proposed scheme improvement on a number of simulations and showed that the approach reduces the average block propagation time while maintaining a good blockchain synchronization rate. But off course simulations in some cases my not cach the real world complexity and this the point for further research.

II. Background and related work

A. Blockchain network and block propagation

It is well known that the blockchain as a concept was introduced with the Bitcoin cryptocurrency system by Satoshi Nakamoto in 2008 [6]. The blockchain could be regarded as an immutable and decentralized database maintaining a continuously growing list of ordered records, called blocks that are secured from tampering and shared among participating members. The blockchain is an important approach for open environments [7], as it is dedicated to memorize data, execute transactions, perform functions and provide trust and secure computations [8].

Bitcoin and Ethereum are the most common blockchain (payment) architectures that are built on an unstructured peer-to-peer (P2P) network model. The P2P architecture of blockchain allows all cryptocurrencies to be transferred worldwide, without the need of any middle-man or intermediaries or central server. With the distributed P2P network, anyone who wishes to participate in the process of verifying and validating blocks can set up a node [9].

New blocks in Bitcoin and Ethereum can only join the chain, when the other nodes validate these blocks, and this is after executing a decentralized consensus procedure [10]. The number of blocks is increasing over time. These blocks are linked together using cryptography to form a chain. Each individual block holds a cryptographic hash of the antecedent one.

When a node initializes, it attempts to discover a set of peers to establish outgoing or incoming transport layer network connections (based on TCP protocol). These TCP connections are used for transaction and block propagation. Each node maintains a list of peers' IP addresses. For instance, according to the default protocol in the Bitcoin core project [11], an average node in the network initiates up to 8 outgoing connections and accepts up to 117 incoming connections. Super nodes may establish more than 8 outgoing connections [12].

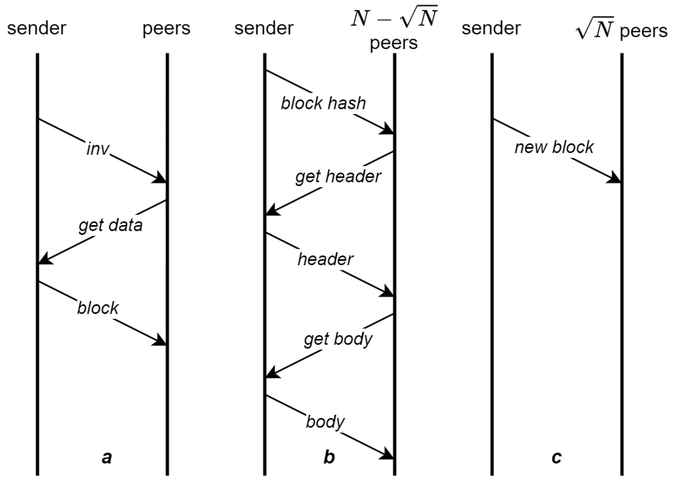

Fig. 2. Standard block-hash propagation scheme (BPP) [14]
Note. a – default propagation scheme in Bitcoin; b, c – default propagation scheme in Ethereum.

The standard block-hash propagation (BPP) scheme (protocol) in Bitcoin/Ethereum blockchain implies that a sending node forwards a new block to its N neighbor nodes [13]. In Bitcoin, when a certain block or transaction arrives and is verified by a node, it notifies the neighbors using an inventory (inv) message to let them know that a new block or transaction is available and ready to send. The inv message consists of the hash of the mentioned block or transaction. When a node receives such a message for a block or transaction that is not already seen by it, it replies to the inv message by a get data message. Upon receiving the $getdata$ message, the node will transfer the block or transaction to the sender of this message (Fig. 2, a). In Ethereum, the sending node randomly selects $\sqrt{N}$ neighbor nodes to forward the full block directly after verifying the block head information. It then announces the block hash to the remaining neighbor nodes after verifying the full block. The neighbor nodes that do not have the block will request the block header and block body successively from this sending node to reconstruct the full block (Fig. 2, b, c).

Message broadcasting and traffic load are the most important components that determine energy consumption in most networks, as well as in blockchain-related networks. However, the great step towards sustainability of blockchain-related networks was done by inventing the proof-of-stake (PoS) consensus algorithm [15]. The idea for PoS began as a way to create a less energy consuming alternative to Bitcoin's proof-of-work algorithm (PoW), which requires miners to solve cryptographic puzzles to verify transactions on the blockchain and involves huge computations that lead to high energy consumption.

B. Reinforcement learning for networks

Reinforcement learning has shown excellent ability in solving some complex problems, as well as in case of blockchain [16]. It uses rewards to enable the algorithm to continuously optimize decision-making in the learning process, thereby learning an optimal mapping from state to action [17]. The rewards for performing actions are delayed, that is to say, the pros and cons of the current action cannot be judged immediately. Only after the action has an impact on the state, the cumulative reward obtained from the execution of the action to a certain moment afterwards is calculated. Then the model accomplishes the optimization of the action by reward [18].

Basically, a task in RL is modeled as a Markov Decision Process (MDP). An MDP M is defined as a tuple $M = (S, A, T, R, \gamma)$, where S and A are the state and action space respectively. $T(s, a, s') : S \times A \times S \to [0, 1]$ is the probability of reaching state s' from state s after executing action a. The reward function $R(s, a, s') : S \times A \times S \to R$ assigns a numerical reward to a state transition from s to s' with respect to the executed action a. A policy $\pi(s, a) : S \times A \to [0, 1]$ defines how the agents should act in the environment through the probability distribution over all actions in every state. The decay factor $\gamma \leq 1$ is used to define the expected discounted return $D_t = \sum_{\infty}^{\infty}{}_{k=0}\gamma^k R_{t+k}$ and the value function $V(s) = E_{A_t \sim \pi(St)}[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s]$. The RL agent learns a policy that maximizes the expected discounted return, where the optimal policy has the maximum expected discounted return [19].

The use of the RL-based approach to solve routing problems in different types of networks has been widely discussed in the literature (see for review [3]). For example, Boyan et al. [20] first combined the Q-learning algorithm with packet routing to dynamically learn the routing situation and find the shortest path in the network. Gao et al. [21] proposed a multi-agent routing algorithm with Q-learning and backpressure, where each routing node needs only local information about the neighbor routing nodes to solve this problem. Mayadunna et al. [22] and Yang et al. [4] proposed a RL-based malicious routing node detection scheme for mobile ad-hoc networks and wireless sensor networks, respectively.

C. Related work and sustainability concern

There is a number of different schemes and protocols proposed to reduce the propagation time and related traffic load in the blockchain-based networks, as well as to ensure network security: compact block propagation [12], hybrid compact block propagation [13], bodiless block propagation [23], etc. They mainly focus on some algorithms and technical improvements of the Bitcoin or Ethereum core protocol itself. This usually requires significant changes in the network design and infrastructure, and in some cases even a fork of the Bitcoin/Ethereum. Therefore, our work focuses on the network transport layer, which is usually invariant of the underlying protocols and requires relatively small changes in the client software. As we try to slightly modify the previously mentioned default propagation procedure (BPP) at the broadcasting step with an RL-based approach, technically, such an approach can be implemented with any propagation scheme that usually does not depend on the transport layer of the network.

As we mentioned above, recent works consider some RL implementations for blockchain based networks to improve their methods and algorithms. For instance, in [7] by combining RL and blockchain for IoT network, the decentralised communication structure for scalable and trustworthy information allocation was developed (for review on RL for IoT see [24], [25]). In [26] deep RL was used to optimize the performance of recently invented Prism proof-of-work blockchain protocol [27] and enhance the number of votes without violating the security and latency performance guarantees [26]. In [28] RL was used to auto-tune network fabric in the permissioned blockchain system and demonstrated an ability to identify optimal network configuration.

There are also a number of papers that are devoted to RL solutions in the context of improving cryptocurrency operations in Bitcoin and Etherium: cryptocurrency exchanges [29], trading [30], [31], portfolio management [32] and, for example, detecting specific agents behaviors [29] etc. Some new work combine blockchain technologies and RL to improve even economic and social systems, e.g. healthcare system [33].

However, these works do not directly address broadcasting problem in global peer-to-peer networks, namely Bitcoin and Ethereum and do not pay much attention to overall environmental impact.

The most closest to ours work [4] has already stated that existing blockchain routing schemes based on randomization or fixed-order broadcasting struggle to cope with network dynamics and in some cases cannot avoid poorly connected or even malicious nodes when routing changes are made in real time. In the case of wireless sensor networks, they showed that RL can improve the self-adaptability of the routing scheme by dynamically selecting more reliable and efficient routing channels [4]. We shifted this frame idea to Bitcoin and Etherium networks and showed that a well-trained RL agent can efficiently re-prioritize initially randomized broadcasting order to achieve network consistency earlier, and reduce possible traffic redundancy.

III. Methods

A. Network topology simulator and RL-agent

Our approach is tested on the blockchain simulator invented by Faria et al. [5]. This is a discrete-event simulator that is flexible enough to evaluate different blockchain implementations, e.g. Bitcoin and Ethereum. The simulator follows a stochastic simulation model, being able to represent random phenomena by sampling from a probability distribution. The network model in the simulator is responsible for knowing the state of each node during the simulation, establishing the connection channels between nodes, and applying a network latency on the messages being exchanged. We slightly extended the network model by implementing BPP broadcasting

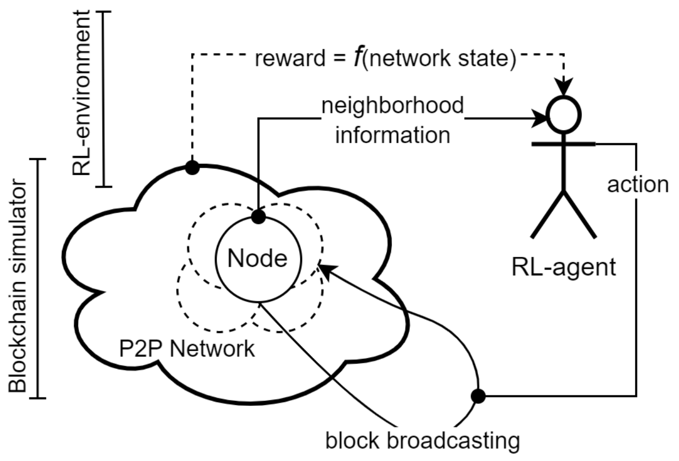

Fig. 3. RL-agent and Blockchain simulator integration

in a recursive way and modified network monitoring to collect appropriate statistics.

The network latency delay in the simulator is applied depending on the geographic location of the destination and origin nodes. Three geographic locations with corresponded latency distribution were provided out of the box (Ohio, Tokyo, and Ireland), and they shows identical results for average propagation time as in real blockchain, but with a slightly different standard deviation [5]. Based on the limited number of TCP connections of the average node ($\sim$125), we built a minimal, but big enough fully connected P2P network architecture of size 150 nodes. Each available geographical location consisted of 50 nodes and half of them were marked as miners with equal hash rate (this is a prerequisite for a proper simulation).

Then we built a standard OpenAI Gym RL-environment [34] with the integrated simulator instance the way it can be independently executed at each step of training after an RL-agent performed its action (Fig. 3). We also modified the node model of the simulator the way it asks the RL-agent to re-order its connections list before starting broadcasting. Thus, the active RL-agent might be involved (or not) independently during simulation.

Based on the tracked information in the simulator, we represented the dynamic network state (s) from a node's perspective as an ordered (corresponded to the node connections list) set of neighbor latencies that change after each communication step. For technical simplicity, we used delay estimates obtained from the transaction propagation process, which is performed before the block propagation phase in simulation. We define an action function as an ordering function over the set of connections between the current node and its neighbors. According to BPP, each node by default randomizes its list of connections and sends full blocks only to $\sqrt{N}$ neighbor nodes, and technically, each action (a) represents the corresponding order for node's connection list.

To train the RL-agent we used the Proximal Policy Optimization (PPO) approach. PPO is a family of policy optimization methods that use multiple epochs of stochastic gradient ascent to perform each policy update. These methods have the stability and reliability of trust-region methods but have better overall performance and are applicable in more general settings [35]. There is strong evidence that PPO based agents are able to solve pathfinding tasks and are better at adjusting such a heuristic [36]. So, we employed the OpenAI Gym environment [34] and the Stable-Baselines3 PPO [37] with default hyperparameters (see Appendix).

B. Key metrics and reward function

Block propagation time is the most important metric for blockchains, since it determines the fork rate [38], [39] and also limits the frequency of payments. For BPP in Ethereum, the block propagation time is dominated by the transmission time of the full block [14] that can be written as:

$$t_{bpp} = l + size \cdot (b^{-1} + t_{proc}),$$

here t_{bpp} indicates block propagation time or total transmission delay; l indicates the network latency; $size$ indicates the block size; t_{proc} indicates the block processing delay; b denotes the bandwidth.

Thus, block propagation time should be considered as the most important metric in case of the block propagation schemes comparison. Because of the continuing block mining and propagation process, we decide to estimate an average block propagation time over the blocks that were successfully propagated to at least 50% of nodes in the network (synchronized blocks) and to consider this metric as a practical approximation of the network consistency – **synchronization time**.

The other important metric is the probability of blockchain forks or uncle blocks. A fork occurs whenever there are two different valid blocks at the same block height competing to form the longest blockchain. This occurs under normal conditions when two miners solve the proof algorithm within a short period of time from each other. In other words, blockchain forks occur as a result of propagation delays in the global network. A faster block mining time would make transactions faster but lead to more frequent blockchain forks, whereas a slower block time would decrease the number of forks but make settlement slower. The faster block propagation time therefore reduces the probability that the older block wins the race.

As the forks probability depends on the block propagation time but also on the frequency of block introduction (mining in Bitcoin or forging in Ethereum), to practically compare block propagation schemes we decide to estimate the number of introduced blocks that were successfully propagated to over 50% of nodes (synchronized) in the network – **synchronized blocks rate**.

As a proxy of the propagation complexity and the environmental impact we decide to count the number of messages that are broadcasted through the network. Thus the third of our metrics is the amount of messages that were emitted to successfully propagate the introduced blocks to over 50% of nodes in the network – **messages per synchronized block**. Based on these metrics we define a simple reward function for our RL-agent:

$$reward = \frac{synchronized\ blocks\ rate}{synchronization\ time}$$

IV. Results

A. *Simulation results*

Since we wanted to test whether a well-trained RL agent can efficiently re-prioritise broadcasting order and improve the default block propagation protocol by doing so, we run a series of fixed-duration Ethereum blockchain simulations and evaluated the key metrics mentioned above. In order to fairly compare BPP with the RL-agent and without it, we performed $k = 1000$ simulations with a 60 sec duration. This duration is minimally sufficient to introduce up to 5 blocks within a single simulation with comparatively less computation time and overall execution time.

At each of the k iterations, one simulation with and one simulation without the RL agent was run using the same random seed and network topology. We then recorded key metrics: synchronization time, synchronized blocks rate, and messages per synchronized block.

Simulation results confirmed that RL-agent based broadcasting is able to achieve significant improvements of network dynamic (Fig. 4). On average over 1000 simulations, synchronization time was reduced by 1.7%, and synchronized blocks rate was increased by 3.4 percent points. The total number of messages per synchronized block was reduced approximately by 5.0%.

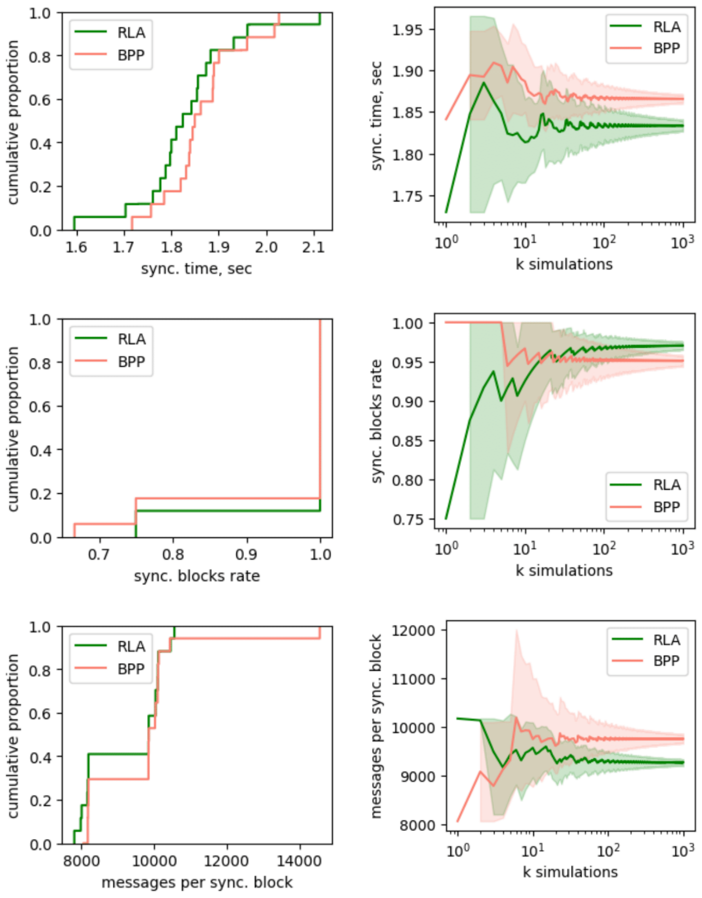

Fig. 4. Main simulation results

Note. The results of 1000 simulations (with different random seed at each one) performed for the default block-hash propagation (BPP) and RL-agent-based block broadcasting (RLA). The figures on the left side represent the cumulative fraction of empirical values (ECDF): the higher and far left the line, the lower the average value. The figures on the right side represent the cumulative average over a certain number of simulations (k), the shaded area is the 95% confidence intervals.

V. Limitations and future work

Using blockchain simulations in experiments is a trade-off between realism and control. Simulations provide a cost-effective, controlled, and safe environment for research and experimentation, but they come with inherent limitations in replicating the complexity of real-world blockchain networks. For instance, simulations can never fully replicate the complexity and dynamics of a real blockchain network; they often rely on simplifications and assumptions, such as uniform network conditions or idealized node behaviors; simulating large-scale blockchain networks with thousands or millions of nodes can be computationally intensive and may not be feasible due to hardware and software constraints etc.

Since we wanted to make not only a controllable and reproducible but also a fair comparison between the default and our improved block propagation protocol, we constructed a simulation-based experimental design. However, in future work we plan to build a small real blockchain testbed, and also consider deeply exploring the proposed approach implementation for other Ethereum-like networks.

VI. Conclusion

In this paper, we have attempted to address the environmental issue associated with blockchain-based P2P payment networks and explicitly discuss the environmental implications of the proposed improvements. We believe that every small improvement in blockchain protocols and its key metrics should also be considered in terms of energy consumption, data flow and network load. Therefore, in this paper, we:

1) developed an improved Ethereum blockchain peer-to-peer network propagation scheme that involves an RL agent to efficiently re-prioritize the broadcast order based on real-time transport layer network information;
2) enhanced an event-driven blockchain simulator by involving an RL agent, and created an simulator-based integrated RL learning environment;
3) validated the proposed scheme improvement on a number of simulations and showed that the approach reduces the average block propagation time while maintaining a good blockchain synchronization rate;
4) emphasized the environmental impact and estimated the required number of broadcast messages before the

blockchain network achieved its consistency.

Code and Data availability

All data and algorithms used can be found in the public repository:...

..

References

[1] V. Kohli, S. Chakravarty, V. Chamola, K. S. Sangwan, and S. Zeadally, "An analysis of energy consumption and carbon footprints of cryptocurrencies and possible solutions," *Digital Communications and Networks*, vol. 9, no. 1, pp. 79–89, 2023.

[2] "Guidelines for improving Blockchain's environmental, social and economic impact," World Economic Forum, 2023, retrieved June 16, 2023 from https://www3.weforum.org/docs/WEF_Guidelines_for_Improving_Blockchain%E2%80%99s_Environmental_Social_and_Economic_Impact_2023.pdf.

[3] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55 916–55 950, 2019.

[4] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, 2019.

[5] C. Faria and M. Correia, "Blocksim: Blockchain simulator," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 439–446.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," May 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[7] T. Alam, "Blockchain-enabled deep reinforcement learning approach for performance optimization on the internet of things," *Wireless Personal Communications*, vol. 126, pp. 995–1011, 2022.

[8] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, no. 3, jul 2019. [Online]. Available: https://doi.org/10.1145/3316481

[9] T. K. Sharma, "Blockchain & role of p2p network," Blockchain Council, 2022, retrieved September 15, 2023 from https://www.blockchain-council.org/blockchain/blockchain-role-of-p2p-network/.

[10] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "Homechain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2020.

[11] "Bitcoin core," Bitcoin Core Project, 2020, retrieved September 15, 2023 from https://bitcoin.org/en/bitcoin-core/.

[12] L. Zhang, T. Wang, and S. C. Liew, "Speeding up block propagation in blockchain network: Uncoded and coded designs," arXiv, 2021.

[13] C. Zhao, T. Wang, S. Zhang, and S. C. Liew, "Hcb: Enabling compact block in ethereum network with secondary pool and transaction prediction," arXiv, 2022.

[14] X. Ma, H. Wu, D. Xu, and K. Wolter, "Cblocksim: A modular high-performance blockchain simulator," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2022, pp. 1–5.

[15] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:42319203

[16] X. Bai, S. Tu, M. Waqas, A. Wu, Y. Zhang, and Y. Yang, "Blockchain enable iot using deep reinforcement learning: A novel architecture to ensure security of data sharing and storage," in *Artificial Intelligence and Security*, X. Sun, X. Zhang, Z. Xia, and E. Bertino, Eds. Cham: Springer International Publishing, 2022, pp. 586–597.

[17] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Computers & Chemical Engineering*, vol. 139, p. 106886, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135420300557

[18] M. Hatem and F. Abdessemed, "Simulation of the navigation of a mobile robot by the qlearning using artificial neuron networks," in *Conférence Internationale sur l'Informatique et ses Applications*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:2909520

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[20] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *NIPS*, 1993. [Online]. Available: https://api.semanticscholar.org/CorpusID:364332

[21] J. Gao, Y. Shen, M. Ito, and N. Shiratori, "Multi-agent q-learning aided backpressure routing algorithm for delay reduction," arXiv, 2017.

[22] H. Mayadunna, S. L. D. Silva, I. Wedage, S. Pabasara, L. Rupasinghe, C. Liyanapathirana, K. K. Kesavan, C. P. Nawarathna, and K. K. Sampath, "Improving trusted routing by identifying malicious nodes in a manet using reinforcement learning," *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 1–8, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:40517242

[23] C. Zhao, S. Zhang, T. Wang, and S. C. Liew, "Bodyless block propagation: Tps fully scalable blockchain with pre-validation," arXiv, 2022.

[24] R. Gasmi, S. Hammoudi, M. Lamri, and S. Harous, "Recent reinforcement learning and blockchain based security solutions for internet of things: Survey," *Wireless Personal Communications*, vol. 132, pp. 1307–1345, 2023.

[25] L. Gao, X. Zhang, T. Liu, H. Yang, B. Liao, and J. Guo, "Energy-aware blockchain resource allocation algorithm with deep reinforcement learning for trusted authentication," in *Smart Grid and Innovative Frontiers in Telecommunications*, M. Cheng, P. Yu, Y. Hong, and H. Jia, Eds. Cham: Springer International Publishing, 2021, pp. 93–103.

[26] D. S. Gadiraju, V. Lalitha, and V. Aggarwal, "An optimization framework based on deep reinforcement learning approaches for prism blockchain," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2451–2461, 2023.

[27] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 585–602. [Online]. Available: https://doi.org/10.1145/3319535.3363213

[28] M. Li, Y. Wang, S. Ma, C. Liu, D. Huo, Y. Wang, and Z. Xu, "Auto-tuning with reinforcement learning for permissioned blockchain systems," *Proc. VLDB Endow.*, vol. 16, no. 5, p. 1000–1012, jan 2023. [Online]. Available: https://doi.org/10.14778/3579075.3579076

[29] M. Schnaubelt, "Deep reinforcement learning for the optimal placement of cryptocurrency limit orders," *European Journal of Operational Research*, vol. 296, no. 3, pp. 993–1006, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721003854

[30] B. J. D. Gort, X.-Y. Liu, X. Sun, J. Gao, S. Chen, and C. D. Wang, "Deep reinforcement learning for cryptocurrency trading: Practical approach to address backtest overfitting," arXiv, 2022.

[31] M. Asgari and S. H. Khasteh, "Profitable strategy design by using deep reinforcement learning for trades on cryptocurrency markets," arXiv, 2022.

[32] Z. Huang and F. Tanaka, "A scalable reinforcement learning-based system using on-chain data for cryptocurrency portfolio management," arXiv, 2023.

[33] A. Lakhan, M. A. Mohammed, J. Nedoma, R. Martinek, P. Tiwari, and N. Kumar, "Drlbts: deep reinforcement learning-aware blockchain-based healthcare system," *Scientific Reports*, vol. 13, p. 4124, 2023.

[34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," arXiv, 2016.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv, 2017.

[36] A. Skrynnik, A. Andreychuk, K. Yakovlev, and A. Panov, "Pathfinding in stochastic environments: learning vs planning," *PeerJ Computer Science*, vol. 8, p. e1056, 2022.

[37] "The proximal policy optimization algorithm," Stable Baselines3 Library, 2021, retrieved September 15, 2023 from https://stable-baselines3.readthedocs.io/en/ master/modules/ppo.html.

[38] 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 *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 3–16.

[39] U. Klarman, S. S. Basu, A. Kuzmanovic, and E. G. Sirer, "bloxroute: A scalable trustless blockchain distribution network whitepaper," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:92979777

Appendix

Experiment details and hyperparameters

We used the same default hyperparameters for the Stable-Baselines3 PPO implementation [**?**]: policy type: actor-critic; timesteps per epoch: 100,000 for general training, 10,000 for experiments and additional training; learning rate: 0.0003; discount factor (γ): 0.99; GAE parameter (λ): 0.95; clipping parameter: 0.2; value function coefficient: 0.5; maximum value for the gradient clipping: 0.5.

Hardware setup for training. CPU: AMD EPYC 7662 64-Core Processor, 256 CPUs; GPU: 2 x A100-PCIE-80GB; 1 TB RAM, 1007.764 GB available; Platform system: Linux-5.10.0-15-amd64-x86_64-with-glibc2.31; Python version: 3.10.6.

Hardware setup for experiments. CPU: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 8 CPUs; GPU: 1 x NVIDIA GeForce MX350; 8 GB RAM, 7.675 GB available; Platform system: Windows-10-10.0.22621-SP0; Python version: 3.9.13.