The Case for AI Based Web3 Reputation Systems

Navin V. Keizer*, Fan Yang*, Ioannis Psaras*†, George Pavlou*
*University College London,† Protocol Labs

***Abstract*—Initiatives such as blockchains and decentralized storage networks are pushing for a decentralized Web3 to replace the current architecture. At the core of Web3 are network resource sharing services, which allow anyone to sell spare network capacity in return for rewards. These services require a way to establish trust, as parties are potentially malicious. This can be achieved by reputation systems.**

In this paper we make the case for using deep reinforcement learning in Web3 reputation calculation. More specifically, we propose a model which allows for decentralized calculation of scores with high personalization for the user.

***Index Terms*—Reputation System, Deep Reinforcement Learning, Blockchain, Web3, Resource Sharing Services**

I. Introduction

Over the past decade, there have been increasing calls for a decentralized Web3 which aims to address the disadvantages of the current centralized infrastructure, including a single point-of-failure, censorship, and data privacy.

An important aspect of a decentralized Web3 is the ability to outsource tasks to spare resources, creating *network resource sharing* (NRS) services. This is essential as central servers (e.g. the Cloud) should not be blindly trusted. NRS services can broadly be classified as storage, computation, or bandwidth sharing services. A service may also target all of these, as is the case for decentralized content delivery networks.

Sharing network resources in a decentralized network isn't a new concept, but what makes Web3 initiatives unique is their integration with blockchains to create an incentive layer. Classical peer-to-peer (P2P) systems suffered from a number of problems, rendering them useless in the long term, including, free-riding, instability due to churn, and security vulnerabilities [1]. By providing a fair exchange for performed work in the form of cryptocurrency rewards, blockchain-based NRS services add incentives, security, and robustness.

One prominent example of a NRS service is Filecoin [2], a decentralized storage market. A blockchain is used as an incentive layer, allowing clients and sellers to create storage deals on a public ledger and reward the storage node accordingly. As storage on blockchain is highly inefficient, data is stored locally at storage nodes.

While the blockchain can be used to establish trust for transactions on-chain, the actual NRS service is provided off-chain and occurs directly between two parties. This means that we cannot rely solely on an honest majority of the network for security. A simple illustration is a provider node which promises a service, but is not able to complete the service. While it does not gain extra rewards, the client may experience additional negative consequences. As any node in the network is potentially malicious there is a risk with every deal.

To discern between honest and malicious parties a *reputation system* is needed. Generally, a reputation system is a mechanism which produces a score for nodes in a network, indicating the trust in a likely positive experience with them. The reputation system aggregates a number of metrics and follows a scoring mechanism to produce scores. A common use-case of reputation systems is in e-commerce, where trust is established between buyers and sellers using transaction feedback. However, these types of reputation systems rely on a centralized infrastructure, and are therefore unsuitable for Web3 applications.

P2P research presented a number of distributed trust and reputation systems [3], but these ultimately did not reach mass adoption due to their complexity and security vulnerabilities. While these systems used a range of metrics, both public and private, to the best of our knowledge none have incorporated blockchain data.

II. Web3 Reputation System Requirements

Currently, NRS services either do not use a reputation system, or rely on a centralized service. Not only are these a single point-of-failure, but they also lack transparency and control which metrics and scoring functions are used. For example, Storage.Codefi[1], a 3rd party Filecoin Dashboard, puts more emphasis on storage faults than ask price when calculating reputation scores, which may not represent all user preferences and their risk aversion.

Instead, a decentralized reputation system is needed for Web3 services. Due to the close integration of decentralized services and blockchain, a Web3 reputation system should *use on-chain data* to derive metrics for computation of reputation scores. Furthermore, the system should be *accurate, quickly converging, secure*, and *lightweight*.

Finally, reputation scores are highly subjective metrics. One node's view of a sellers reputation might be different from others, based on personal preference and experience. In fact, a single seller may be seen reputable as a storage node, but not so much in other services. Therefore, a highly *personalized* reputation service is needed, which self-adapts based on a personal preference profile.

III. Artificial Intelligence Based Reputation

In this work, we propose to use artificial intelligence (AI) - specifically deep reinforcement learning (DRL) - to assist in calculating personalized reputation scores. This meets the requirements as described in section II, creating a decentralized reputation solution for Web3 services. Using AI allows us to take into account a wide range of metrics to produce an optimized scoring function, as the algorithm continually

[1]https://storage.codefi.network

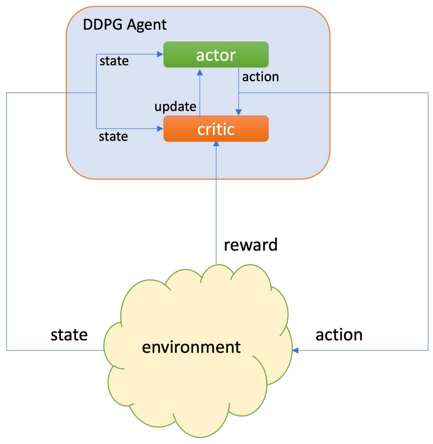

Fig. 1. Overview of our DDPG algorithm.

finds the ideal weights, and may take into account otherwise overlooked metrics.

A reputation score calculation is comprised of two parts: gathering information and computing scores. Our proposed solution uses blockchain data directly as input, as most NRS service deals are stored on-chain, and therefore much can be inferred from past transactions. On top of this, the user of the reputation system submits a personal preference profile, which may include preference for *e.g.* cost savings or security.

The second part, *i.e.* the computation of scores, is taken care of by our DRL algorithm, which is learning from past experiences to create a scoring function. The specific DRL algorithm we propose is Deep Deterministic Policy Gradients (DDPG) [4]. The main advantage of DDPG compared to others is its ability to output approximate continuous actions. Since our application should have a wide range of possible outputs, other DRL algorithms will not be eligible. As shown in Figure 1, the algorithm takes environment information as input state, after which the actor inside the agent outputs an action. This action will be random in the beginning, but becomes smarter over time with training, as feedback is received by the critic based on the reward of an output action. This way, the critic continually updates the actor.

Using the personal preference profile of a user, our model generates a personalized reputation scoring function, which the user then uses with blockchain data to calculate scores. While training the model can be computationally intensive, the actual calculation of scores when the model weights are obtained is lightweight. Training the model is initially done using historical blockchain data, after which it is continually updated with usage data.

IV. Filecoin Example

We now illustrate how our model could work on decentralized storage markets such as Filecoin. The input state to our agent consist of a node's information (who we want to score), inferred from blockchain transactions, as well as a virtual balance of the user. Examples of blockchain inferred data are average deal time, storage size available, failed deals, and time since joining the network. Meanwhile, a preference profile of the user is required in order to evaluate the action. Initially the model will need to be trained based on historic data pulled from the chain in order to become accurate.

Every input cycle, the DDGP algorithm will output an action (a reputation score from 0 to 100) based on the input states. In our example, the consequence of a bad output could be a loss of profit due to a wrong recommendation, as well as failing to make a deal with an honest counterparty. Evaluating actions is essential to give feedback to our model for it to learn.

After an output score is generated, it is mapped to a *sinc* function which indicates the probability P of making a deal with the node. P is 1 if the score is over 95, and P is 0 if it is below 30. The accuracy of an action (and therefore the reward) is decided by the distance between P and the user decision D. In our initial training, D is obtained by inferring the preference profile from the node's information, whereas this preference can be obtained from user feedback. When P and D are both below or above 0.5, the reward will be positive, and this reward is increased the closer they are together. The reward is added to the virtual balance which is going to be forwarded to the next state. After enough training the model will learn user preference and optimize this balance.

V. Discussion

We have described how the DRL model can be used to create a personalized reputation scoring system. While calculating scores is fairly lightweight when the function has been trained by the model, the actual training phase can be computationally intensive. Therefore, an open question remains how this training would happen in practise. A completely distributed scenario would have every node run its own DRL algorithm, but this might not be feasible for all nodes, especially mobile users, as their devices may be relatively resource poor and have battery constraints. Furthermore, this would create lots of redundant work, as many users will have similar preference profiles and therefore similar functions.

A more feasible architecture would allow nodes in the network to outsource the training of the algorithm, while calculating the scores locally. They would submit their preference profile, fetch the matching function, and calculate the score locally after querying the blockchain for input data. We plan to explore these possibilities, as well as the performance of the proposed model further in our future work.

References

[1] P. G. López, A. Montresor, and A. Datta, “Please, do not decentralize the internet with (permissionless) blockchains!” 2019.
[2] “Filecoin: A decentralized storage network,” Protocol Labs, Tech. Rep., 2017.
[3] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” 2007.
[4] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *CoRR*, 09 2015.
[5] N. V. Keizer, O. Ascigil, I. Psaras, and G. Pavlou, “Rewarding relays for decentralised nat traversal using smart contracts,” 2020.
[6] R. Dennis and G. Owenson, “Rep on the roll: a peer to peer reputation system based on a rolling blockchain,” 2016.
[7] R. Zhou and K. Hwang, “Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing,” 2007.