Independent verification of validator performance in the *XRP* ledger

Francis Behnen[1] , **Mitchell Olsthoorn**[1] , **Annibale Panichella**[1]
[1]EEMCS, Delft University of Technology, the Netherlands
F.J.Behnen@student.tudelft.nl, {M.J.G.Olsthoorn, A.Panichella}@tudelft.nl

Abstract

The *XRP* network (*Ripple* network) is a global transaction network that settles transactions in seconds. It is a technology that provides new opportunities for traditional financial institutions and startups. In the *XRP* network, participants depend on each other's functioning. However, at present *Ripple Labs, Inc.* is the only actor monitoring some form of performance. This paper presents a way for independent individuals to verify the score of validators. Furthermore, to improve the insight into the performance of validators, a new metric is introduced. Together these improvements contribute towards a healthy decentralised *XRP* network.

Keywords: Blockchain, Performance metric, Performance monitoring, Decentralised trust management, *XRP*

1 Introduction

Decentralised transaction databases (blockchains) have become a very active field of research and many companies operate in this domain as well [1]. Most public blockchains use an algorithm to get participants that do not trust each other to cooperate. With the result of a calculation, a participant can proof to the rest of the blockchain network that a certain set of transactions should be applied to the database (ledger) next. Every other participant can verify this calculation result and decide whether it is indeed correct to apply the proposed set of transactions next. They do not have to know, and hence not trust, the participant that provided the result. These Proof of Work (PoW) algorithms require a lot of energy and are slow [2].[1]

To mitigate the problem of energy consumption and create a faster system, *Ripple Labs, Inc* (hereafter *Ripple*) proposed the *XRP* Ledger network. Unfortunately, the system they designed also introduces the need for some form of trust between the network participants.

A large body of literature exists on decentralised reputation and trust algorithms. They all have their specific application, but can also be compared [3]. Researchers proposed to store trust, or reputation, on a blockchain [4], and researched trust in other domains such as Internet of Things (IoT) [5] and social and adhoc networks [6]. However, because most blockchains do not need trust management to keep the network itself running, there is no literature about trust between nodes of a blockchain network itself.

To evaluate the trustworthiness of *XRP* nodes, specifically the *XRP* validator nodes, *Ripple* introduced a central list: https://xrpcharts.ripple.com/#/validators. This list shows the validators with their score as measured by nodes monitored by *Ripple*. These scores are in turn used to select the highest performing validators, in order to create a separate list of trustworthy validators the rest of the network should follow, according to *Ripple*. Such a central list is against the core principle of blockchains to not have a central authority. After all, this situation makes *Ripple* the de facto authority on which validators are trustworthy. Furthermore, the underlying technology *Ripple* employs is not scalable. This calls for a scalable solution without central authority for validators to assess their peers.

Two contributions towards this goal are made in this paper. First, we develop a program to locally monitor validators and compute their scores. After that, a new metric, built on top of those scores, is introduced. This metric increases the understanding

[1]This is simplified beyond the point that it is technically correct, but should bring home why it is worthwhile to investigate cryptocurrencies not based on PoW. For an excellent, technically correct, explanation, see the video *But how does bitcoin actually work?* by Grant Sanderson.

of the actual performance of validators.

The next section discusses required background that should make clear what the design limitations are. Section 3 examines related work on trust algorithms. In section 4 the solution is introduced and in section 5 it is evaluated. Section 6 discusses the ethical aspects surrounding this research specifically and blockchains in general. Section 7 provides a conclusion and recommendations for future work.

2 Background

This section provides background on the mechanics of the system to illustrate the problem to be tackled. The first subsection explains the *XRP* Ledger Consensus Protocol (*XRP LCP*). This is the protocol through which new ledgers are established. The paragraphs after that introduce the *XRP* peer-to-peer network. This is the network that, among other things, facilitates the traffic required for the protocol to work.

2.1 The *XRP* Ledger Consensus Protocol

The *XRP* Ledger (*XRPL*) network works with special nodes called 'validators' to determine the next ledger. All validators receive transactions that users want to execute. Together the validators decide what transactions to apply next. This is done in two stages: deliberation and validation. In the deliberation stage validators iteratively propose a set of transactions to apply [7, p. 5]. If a validator receives the same set of transactions from a set quorum of other validators (e.g. >80%) it moves to the validation stage and actually applies the transactions.

Validators do not listen to all other validators in the network. Since it is an open decentralised network anyone can join at any moment. To prevent their validator from following malicious validators, every validator operator configures a Unique Node List (UNL). This UNL represents a list of validators they trust to not defraud the network [7, p. 3] [8, p. 3]. To aid in keeping a current list it does not have to be configured manually, but can be automatically downloaded and updated from a UNL provider.

Currently, *Ripple*[2] and Coil[3] are the only UNL providers and their process in determining this list is opaque. *Ripple* does publish a continually updated list of scores and says that that is what they use in determining their provided UNL, referred to as *dUNL* for *default UNL*. There was however no way to verify if these scores were correct. This paper will present a way to solve this, but to understand the difficulties in solving this problem, another key concept has to be understood first.

[2]https://vl.ripple.com

[3]https://vl.coil.com

2.2 The operation of the peer-to-peer network and problem for new validators

Validators do not connect directly but communicate over a peer-to-peer network instead. This network consists of all validators and all tracking-nodes. Tracking-nodes do not validate, but merely provide an entry point into the network for users. All tracking-nodes also have a UNL. Every node, validator or tracking-node, finds its peers via a peer discovery protocol when booting. The result is a, essentially random, set of up to 21 direct peers. There are two processes in this network that are important to understand, to get at the problem it creates for new validators.

Nodes only relay messages to its peers that are coming from trusted validators (validators on their UNL). This rule has two consequences. First of all, it implies that messages from untrusted validators are only seen by its direct peers. The second consequence is that most nodes should have mostly the same UNL. Otherwise some nodes may not receive messages from all its trusted nodes. This would happen for example if a node trusts a validator that is not trusted by any of its peers.

Ripple records performance only through nodes that expose their public API to them. Since these nodes only have a limited number of direct peers, *Ripple* can only observe parts of the network. For an untrusted validator to become trusted it has to connect to a *Ripple* node and then show a high performance for an extended period of time. Unfortunately, as explained earlier, these peer connections are mostly random.

This creates a problem for new validators that want to become trusted validators. They have to 'get lucky' and connect to a *Ripple* node. Next to providing insight into the performance of a validator this paper also aims to aid in improving this process of becoming a trusted validator.

3 Related work

Various types of trust algorithms exist, each with their strengths. To replace *Ripple*'s *dUNL* list, some form of a trust algorithm will be needed to counter the influence of malicious actors on the perceived scores. This section discusses three types of algorithms for this, and mentions some potential drawbacks. For a more in-depth comparison of these algorithms, readers are referred to an excellent review by Fan, Liu, Zhang, et al. [3]. In this section *score* and *trust* are used interchangeably, because most literature is on trust algorithms, but the application in this context would be to obtain scores.

The first option is a globally consistent score. This is interesting, because that could be a one-on-one substitute for *Ripple*'s list. EigenTrust [9] is one of the most well-known algorithms of this type. However, this type of trust algorithm is computationally heavy [3] and may not be needed. For better protection against certain types of attacks EigenTrust is vulnerable to, a descendant, like EigenTrust++ [10], can be considered.

On the other hand it may be worthwhile looking into AdaptiveTrust [5] for its high resistance against malicious attacks [3]. The drawback of this algorithm is that it does not provide a global score, instead every participant has its own set of scores about all other participants.

A third option that may be well suited for this application is to use distributed hash tables per the idea in the original EigenTrust paper [9]. In this setup, the owner of the server running the instance of the program would no longer be the same as the owner of the validator being monitored. Instead, all instances decide via a distributed hash table which validators should be monitored by which instance. All instances judging the same validator can then decide the correct score by majority vote. This may remove the need for a trust algorithm completely, but still allow for a global score and not be computationally intensive.

4 Approach

We developed a system that measures the score of individual validators using the same technique as *Ripple* does.[4] A new metric is built on top of this score to improve the understanding of a validator's performance.

4.1 Personal monitorer

The system is built as a standalone program for a validator operator to monitor their validator. It connects through the *WebSocket* protocol to a tracking node directly connected to the validator under monitoring. To guarantee a connection between the node and the validator it is preferred that they are in a cluster together. On boot, the program connects to the configured node and subscribes to the callback for received ledger submissions. More on this API is found in its documentation: https://xrpl.org/subscribe.html.

The program is split in a data aggregation and a data processing function that are run separately (`data_aggregation` and `data_analysis` folders in the repository). The data aggregation function subscribes to the node as described before. It writes every hour the submissions from that hour. The data processing function can analyse this data over arbitrary time-frames. This is useful for reproducing this research; in production, the processing should update automatically and compute over the largest available time-frame.

The processing function should display the current performance of the validator under monitoring, possibly with the raw score from the last hour. In the next subsection, the score directly computed from the aggregated submissions is explained. After that, the new metric for validator performance is introduced.

4.2 Score aggregation

The (existing) score used is the ratio between the number of correct submitted ledgers and the total number of ledgers that should have been submitted during a given time-frame:

$$\frac{correct\ submitted}{total} = score. \tag{1}$$

A *correct submitted* ledger is a ledger submitted by the validator that ends up being the canonical ledger, and *total* number of ledgers is the number of canonical ledgers. The canonical ledger is the ledger that all validators (eventually) use to base new ledgers on. In Bitcoin, it would be a block of the longest chain.

Inspecting the scores resulting from the ledger collection can already give an idea of the validator's performance. However, this requires manual inspection. To overcome this manual inspection of validators' scores by operators a metric is introduced built from these scores.

4.3 Performance Trend Metric

The goal of the Performance Trend Metric is to improve the understanding of the performance of the validator. In isolation, the raw score (eq. 1) does not tell much about the performance of a validator. If a validator has a score of 1.0000 (100% agreement) it is clear that it cannot do better, but it is not clear when it stops being a 'good' score.

To add meaning to the score, the score of the validator of interest is compared to the scores of all *dUNL* validators. First, a statistical significance test is performed to determine if an observed difference between these samples would be significant. Then, if the difference between the scores of the validator of interest and the scores of the *dUNL* validators is indeed significant, the effect size is computed. The effect size is used as the basis for a colour based reporting scheme, according to the thresholds in table 1. If there is no significant difference (p-value $\geq 5\%$) the metric will always report Green. This should give insight into the performance of a validator, without the need to search for and monitor other validators' scores.

[4] Available at:
https://gitlab.ewi.tudelft.nl/ripple/brp-2020/validator-score/ripple-validator-reputation-scoring

Effect size	Effect significance	Metric label	User explanation
> 0.4265	negligible	Green	Your validator's performance is indistinguishable from a *dUNL* validator
≤ 0.4265	small	Green	
≤ 0.335	medium	Amber	Your performance is drifting away from the *dUNL* validators
≤ 0.263	large	Red	Your performance is significantly lower than expected from a *dUNL* validator

Table 1: Relative Validator Performance metric thresholds. These are the thresholds used for the Vargha-Delaney effect size to determine the metric label. The effect size is only used if the validator's performance is significantly different; p-value $<$ 5%, see table 2

As significance test the non-parametric two-sample Kolmogorov-Smirnov test is used [11]. The two-sample Kolmogorov-Smirnov test tests the hypothesis that both samples are drawn from the same distribution, by quantifying a distance between the empirical distribution functions. It is useful in this context as it works well for non-normal distributions and is sensitive against all types of differences. The test statistic is defined as:

$$D_{n,m} = \sup|F_{1,n}(x) - F_{2,m}(x)| \tag{2}$$

With $F_{1,n}(x)$ and $F_{2,m}(x)$ the empirical distribution functions, `sup` the supremum function and m and n the sizes of the samples. The supremum function returns the least element that is greater than or equal to all elements in a set, in other words, the *least upper bound*. For a given confidence level α the null hypothesis is rejected if:

$$D_{n,m} > \frac{1}{\sqrt{n}} \cdot \sqrt{-\ln(\frac{\alpha}{2}) \cdot \frac{1 + \frac{n}{m}}{2}} \tag{3}$$

For the effect size Cliff's delta is computed and transformed to the A_{12} statistic of Vargha and Delaney [12] [as cited in 13]. The significance thresholds in table 1 are taken from Romano, Kromrey, Coraggio, et al. [14, p. 14] and transformed for the A_{12} as well. Readers familiar with this statistic will know that the significance is mirrored over 0.5. This is not important for our purpose, and hence everything above the negligible threshold in the table is treated as negligible. As presented in [12, eq. (14)] the statistic is computed with the following formula:

$$A_{12} = (R_1/m - (m+1)/2)/n, \tag{4}$$

where R_1 is the rank sum of the first sample [13] and m and n defined as before.

Table 2 shows how the metric would behave over the course of 78 days under different circumstances. Day 23 shows that, even though the validator score sample is significantly different from the *dUNL* sample (p-value $<$ 5%, column 3), the effect size is small (column 4 and 5) and therefore the metric turns green. Furthermore, it shows that the metric can give confidence in the validator on days that it has a low score, see day 56 and 58. On the other hand, it also signals that a validator is really falling behind even though the scores every day are near perfect (day 71 and on).

To treat a new validator with caution and mitigate any 'newcomer advantage' the score-list (column 2) of the validator under inspection is prepended with 30 0% scores. This ensures that a validator does not turn green shortly after coming online, but has to exhibit high performance for multiple weeks first. The metric is computed on a rolling window of 30 days.

5 Evaluation

To evaluate the introduced system it is first of all interesting to know whether the locally aggregated scores are more accurate than the (global) scores aggregated by *Ripple*. This would legitimise the use of a local monitorer instead of the data provided by *Ripple*. After those results, the new metric is evaluated. To demonstrate the usefulness of the introduced metric, historical data is analysed to show how the new metric may have helped the validator operator at that moment. The code for this historical analysis is available under `historical_analysis` in the repository.

5.1 Design

To compare the accuracy between local and global scores the program ran for multiple contiguous hours on different days. Next to the hourly saving of the received ledger submissions as described in the approach section, *Ripple*'s observations of that hour are also saved to storage. Ripple's data is pulled from the *Data API v2*: https://xrpl.org/data-api.html. From this data, the validators are identified where the difference with what *Ripple* observed was at least 5. If *Ripple* consistently misses submissions for a validator, the local program may be more accurate for that validator and vice versa. The program was set up to

Day	Score (%)	p-value (%)	Eff. size	Eff. significance	Metric label {G, A, R}	std (%)	Mean *dUNL* (%)	Mean validator (%)	iqr (%)	Median *dUNL* (%)	Median validator (%)
1	100.0	0.00	0.0190	L	Red	16.28	97.18	3.33	0.00	100.00	0.00
2	99.6	0.00	0.0221	L	Red	12.67	97.92	6.65	0.00	100.00	0.00
3	100.0	0.00	0.0414	L	Red	10.39	98.59	9.99	0.00	100.00	0.00
	...										
10	100.0	0.00	0.1606	L	Red	5.86	99.43	33.32	0.00	100.00	0.00
11	99.3	0.00	0.1624	L	Red	5.59	99.46	36.63	0.00	100.00	0.00
12	100.0	0.00	0.1818	L	Red	5.36	99.49	39.96	0.00	100.00	0.00
13	99.0	0.00	0.1830	L	Red	5.66	99.41	43.26	0.00	100.00	0.00
14	100.0	0.00	0.2019	L	Red	5.46	99.45	46.59	0.00	100.00	0.00
15	99.9	0.00	0.2054	L	Red	5.29	99.46	49.92	0.00	100.00	49.48
16	100.0	0.00	0.2246	L	Red	5.15	99.47	53.26	0.00	100.00	99.12
17	100.0	0.00	0.2427	L	Red	5.00	99.50	56.59	0.00	100.00	99.43
18	100.0	0.00	0.2614	L	Red	4.87	99.51	59.92	0.00	100.00	99.73
19	100.0	0.00	0.280	M	Amber	4.75	99.53	63.26	0.00	100.00	99.94
20	100.0	0.02	0.2987	M	Amber	4.63	99.55	66.59	0.00	100.00	99.98
21	100.0	0.06	0.3046	M	Amber	4.54	99.55	69.92	0.00	100.00	100.00
22	100.0	0.24	0.3237	M	Amber	4.46	99.55	73.26	0.00	100.00	100.00
23	100.0	0.75	0.342	S	Green	4.38	99.56	76.59	0.00	100.00	100.00
	...										
29	100.0	60.02	0.4545	N	Green	3.91	99.64	96.59	0.00	100.00	100.00
30	99.9	63.20	0.4600	N	Green	3.85	99.64	99.92	0.00	100.00	100.00
31	100.0	66.12	0.4624	N	Green	3.79	99.64	99.92	0.00	100.00	100.00
32	99.8	66.40	0.4632	N	Green	3.73	99.65	99.93	0.00	100.00	100.00
33	100.0	45.19	0.4489	N	Green	3.67	99.66	99.93	0.00	100.00	100.00
34	98.5	44.95	0.4451	N	Green	3.62	99.66	99.88	0.00	100.00	100.00
35	96.2	26.86	0.4267	N	Green	3.57	99.67	99.75	0.00	100.00	100.00
36	100.0	14.08	0.4114	S	Green	3.52	99.68	99.75	0.00	100.00	100.00
	...										
55	100.0	13.00	0.4177	S	Green	3.74	99.67	99.81	0.00	100.00	100.00
56	19.8	5.78	0.3985	S	Green	4.13	99.63	97.13	0.00	100.00	100.00
57	100.0	5.37	0.3970	S	Green	4.09	99.64	97.14	0.00	100.00	100.00
58	95.2	2.08	0.3785	S	Green	4.06	99.64	96.97	0.00	100.00	100.00
59	99.9	0.71	0.3624	S	Green	4.02	99.65	96.97	0.00	100.00	100.00
60	100.0	2.19	0.3795	S	Green	3.99	99.65	96.97	0.00	100.00	100.00
	...										
64	100.0	2.29	0.3855	S	Green	3.86	99.67	97.03	0.00	100.00	100.00
65	99.9	2.41	0.3891	S	Green	3.83	99.67	97.15	0.00	100.00	100.00
66	100.0	2.39	0.3885	S	Green	3.81	99.68	97.15	0.00	100.00	100.00
67	100.0	0.81	0.3731	S	Green	3.78	99.68	97.15	0.00	100.00	100.00
68	100.0	0.24	0.3579	S	Green	3.75	99.68	97.15	0.00	100.00	100.00
69	99.9	0.06	0.3414	S	Green	3.73	99.68	97.15	0.00	100.00	100.00
70	94.3	0.06	0.3376	S	Green	3.70	99.68	96.96	0.00	100.00	100.00
71	100.0	0.01	0.323	M	Amber	3.68	99.69	96.96	0.00	100.00	99.99
72	99.9	0.00	0.3071	M	Amber	3.65	99.69	96.96	0.00	100.00	99.99
73	100.0	0.00	0.3056	M	Amber	3.63	99.69	96.95	0.00	100.00	99.99
74	100.0	0.00	0.2896	M	Amber	3.61	99.70	96.95	0.00	100.00	99.98
75	99.9	0.00	0.2720	M	Amber	3.58	99.70	96.95	0.00	100.00	99.98
76	100.0	0.00	0.257	L	Red	3.56	99.70	96.95	0.00	100.00	99.98
77	100.0	0.00	0.2419	L	Red	3.53	99.71	96.95	0.00	100.00	99.98
78	99.9	0.00	0.2250	L	Red	3.51	99.71	96.95	0.00	100.00	99.98

Table 2: The introduced Performance Trend Metric with additional statistical information on the samples. The marked cells are the days that the metric changed label. Note that the introduced metric is designed to convey the performance over a period of time, instead of on a day-to-day basis. This becomes clear for example at day 56, where a substantially lower score of the validator has no immediate impact on the metric. The std and iqr columns are the standard deviation and interquartile range of the *dUNL* scores respectively.

monitor validator `ripple1.ewi.tudelft.nl`.[5]

The new metric is empirically evaluated by analysing daily scores over a period where it may not have been clear how a validator performed by just looking at the score. The situation is analysed from the perspective of an operator of the validator. The scores of this validator are compared to the scores, in the same interval, of all *dUNL* validators.

5.2 Results

This section discusses the results of the described evaluations. First, the accuracy of the score aggregation is discussed and compared to *Ripple*'s aggregation. After that, the introduced metric is empirically evaluated using real-world data.

Score accuracy

The data aggregation was done from May 25th to May 28th. In table 4 in Appendix A all scores are found that did not match per hour, filtered as described before.[6] The address column provides the last six characters of the address.

Our local scores are not always in agreement with *Ripple*'s global scores. It is however close. Assuming that every hour 900 ledgers were closed and 40 validators connected, it turns out that for ~0.03% of the recorded submissions there was a major disagreement. Crucially, there was never major disagreement on the monitored validator `ripple1.ewi.tudelft.nl`. This shows that local scores are equally good as the basis for the new metric as *Ripple*'s scores.

Metric evaluation

A historical interval of scores is investigated. During the recording for the score accuracy described above no interesting situations occurred, therefore historical data is used that better illustrates the usefulness of the metric. The results are presented in table 3, this interval starts at the 4th of January. The validator under investigation is `validator.xrptipbot.com`.[7]

Although the validator does not start perfect, with this metric it becomes clear that it is actually not doing worse than *dUNL* validators as the metric turns green on day 26. The new metric is intended to convey the performance *trend* of a validator, not be affected by day-to-day disturbances. This property is seen at work on day 29 and 31 where the daily score drops significantly below the average, but the metric is not immediately affected. Only after an extended period of time without perfect scores, starting day 36, the

[5]Address: nHDDkPeX4CzMrXAQXNQqVSxkPmRbrxReC5NUHWPCmDezfADKKwDQ

[6]The raw data is found in the repository with the source code, in the folder `data/consecutive_only/`.

[7]Address: nHUXeusfwk61c4xJPneb9Lgy7Ga6DVaVLEyB29ftUdt9k2KxD6Hw

Day	Score	label
1	100.00	R
2	100.00	R
3	99.91	R
4	100.00	R
5	99.81	R
6	99.98	R
7	98.45	R
8	96.23	R
9	99.96	R
10	100.00	R
11	100.00	R
12	100.00	R
13	99.98	R
14	100.00	R
15	100.00	R
16	99.99	R
17	100.00	R
	...	
21	100.00	A
22	99.99	A
23	100.00	A
24	99.99	A
25	100.00	A
26	100.00	G
27	100.00	G
28	100.00	G
29	19.76	G
30	100.00	G
31	95.18	G
32	99.93	G
33	100.00	G
34	99.98	G
35	100.00	G
36	99.95	G
37	99.99	G
38	99.94	G
39	99.95	G
40	99.97	G
41	99.96	G
42	99.92	G
43	94.28	G
44	99.98	A
45	99.92	A
46	99.96	A
47	99.96	A
48	99.88	A
49	99.98	R
50	99.99	R
		...

Table 3: Score and new metric compared. The marked cells are the days that the metric changed colour. Day 1 is Jan. 4th 2020 from validator `validator.xrptipbot.com` (Address: nHUXeusfwk61c4xJPneb9Lgy7Ga6DVaVLEyB29ftUdt9k2KxD6Hw)

metric drops to the amber warning zone and quickly after that to red.

6 Ethical aspects of this work

This work has been conducted with the ethics of cryptocurrencies and responsible research in mind. Considerations around this will be discussed in the following two subsections. First it is discussed that, although there are serious concerns around privacy and energy usage surrounding cryptocurrencies in general, it turns out that this is less applicable to *XRP* . After that, the efforts of making this work reproducible are set out.

6.1 Privacy, speed and energy

This section first discusses that *XRP* possesses the same privacy characteristics as Bitcoin. After that, a possible consequence of the fast settlement speed is described. It concludes with a note on energy usage.

Privacy
Bitcoin has been perceived as a useful tool for criminal activity in the past due to its anonymity. Despite its design considerations around privacy at inception [15, p. 6], it turns out that Bitcoin is only pseudo-anonymous. Companies actively try to link public addresses with real people[8] and academic research has shown that it can be even easier for law enforcement that can subpoena information [16]. Although privacy was not a design consideration for *XRP* like it was for Bitcoin [8], it does possess similar characteristics. *XRP* also runs on a public ledger where transactions are linked to public addresses. From a technical perspective, this makes it equally hard to link real people to transactions. Furthermore, *XRP* is not mined and hence has to be acquired by receiving it from someone else. This must mean that it is always traceable to an institution that swapped it for fiat currency, which should be able to identify the other party per Customer Due Diligence (CDD) and Know Your Customer (KYC) regulations. These considerations combined convince the authors that this work contributes to a technology that is not likely to be used for anonymous criminal activity.

Speed
Within the context of criminal activity, it should also be considered that *XRP* is sent in a matter of seconds, and when a ledger is validated transactions are irreversible. This makes it a useful tool for moving money out of a country very fast. Although the above paragraph suggests the identity behind addresses can probably be unearthed, this may come too late. Where law enforcement is probably able to halt international fiat transactions if they operate within twenty-four hours, this may not be fast enough for *XRP*. Such activity is very much against the interest of *Ripple* and any other connected company. Therefore the authors trust that this risk is kept to a minimum.

Energy
The introduction already alluded to the fact that proof of work algorithms like the algorithm Bitcoin employs have enormous energy costs [2][17]. As discussed in this paper, *XRP* uses a completely different system. Therefore this work does not contribute to or support high energy usages.

6.2 Reproducibility

An effort was made to make this work reproducible as recommended by and per the guidelines in the Netherlands Code of Conduct for Research Integrity [18]. The presented validator scores were computed online and the data was not saved in permanent storage in earlier iterations of the code. Splitting aggregation and computation made it possible to share the dataset so the presented scores can be verified. Furthermore, a replication study is made easier by providing the code itself.

7 Conclusions & Future work

The presented program, alongside the presented new metric, gives validator operators better insight into the performance of their validator. This is beneficial to validator owners and hence to the overall stability of the *XRP* network.

This work can be extended by building a score trust algorithm into the tool. When adopted widely, that would render *Ripple*'s score list obsolete and remove *Ripple* from its position as an authority. If instances of the program determine the score of all validators of which it is receiving messages independently, enough instances of the program are running, and they decentrally determine a global score, the *Ripple* score list is not needed anymore. All these instances together could, in that case, provide a score per validator independent of *Ripple*'s list.

Next to extending the program with this new functionality, it can also be improved. In particular, more research can be done into the best method to test for statistical significant difference in the score samples. It is not clear what type of differences are most relevant for this data. Depending on the answer to this, the Cucconi test may turn out to be better suited to test for significance as the Kolmogorov-Smirnov test is sensitive against all possible types of differences between distribution functions. The Cucconi test, on the other hand, was only proposed to test location and scale.

[8] Bitcoinist.com: "Yes, your Bitcoin transactions can be tracked - and here are the companies that are doing it"

Another interesting direction is to test for significance directly from the effect size statistic. Vargha and Delaney describe a technique for this in their paper proposing the A_{12} statistic [12].

Furthermore, other thresholds for the effect size could be researched. The thresholds used in this research are common, but not specialised for this type of data. It may be possible to find heuristics to base the thresholds on for this specific application.

Next to that, the program could be extended to group the performance on the domain name, instead of by address. Validators with the same domain tend to have the same operator, so such a grouping could potentially provide new insights for and on these operators.

A last possibly interesting extension is to provide a second *year window* next to the thirty-day window implemented now. Such a window may better convey the performance under maintenance. It is likely that in a year, maintenance is performed. Within a month this is less likely. A year window is therefore potentially better suited to convey information on maintenance performance.

References

[1] Konstantinos Christidis and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things". In: *IEEE Access* 4 (2016), pp. 2292–2303.

[2] Alex de Vries. "Bitcoin's growing energy problem". In: *Joule* 2.5 (2018), pp. 801–805.

[3] Xinxin Fan, Ling Liu, Rui Zhang, et al. "Decentralized Trust Management: Risk Analysis and Trust Aggregation". In: *ACM Computing Surveys (CSUR)* 53.1 (2020), pp. 1–33.

[4] YongJoo Lee, Keon Myung Lee, and Sang Ho Lee. "Blockchain-based reputation management for custom manufacturing service in the peer-to-peer networking environment". In: *Peer-to-Peer Networking and Applications* 13.2 (2020), pp. 671–683.

[5] Ray Chen, Fenye Bao, and Jia Guo. "Trust-based service management for social internet of things systems". In: *IEEE transactions on dependable and secure computing* 13.6 (2015), pp. 684–696.

[6] Jochen Mundinger and Jean-Yves Le Boudec. "Analysis of a robust reputation system for self-organised networks". In: *European transactions on telecommunications* 16.5 (2005), pp. 375–384.

[7] Brad Chase and Ethan MacBrough. "Analysis of the XRP ledger consensus protocol". In: *arXiv preprint arXiv:1802.07242* (2018).

[8] David Schwartz, Noah Youngs, Arthur Britto, et al. "The ripple protocol consensus algorithm". In: *Ripple Labs Inc White Paper* 5.8 (2014).

[9] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. "The eigentrust algorithm for reputation management in p2p networks". In: *Proceedings of the 12th international conference on World Wide Web*. 2003, pp. 640–651.

[10] Xinxin Fan, Ling Liu, Mingchu Li, et al. "EigenTrustp++: Attack resilient trust management". In: *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. IEEE. 2012, pp. 416–425.

[11] JL Hodges. "The significance probability of the Smirnov two-sample test". In: *Arkiv för Matematik* 3.5 (1958), pp. 469–486.

[12] András Vargha and Harold D Delaney. "A critique and improvement of the CL common language effect size statistics of McGraw and Wong". In: *Journal of Educational and Behavioral Statistics* 25.2 (2000), pp. 101–132.

[13] Andrea Arcuri and Lionel Briand. "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering". In: *Software Testing, Verification and Reliability* 24.3 (2014), pp. 219–250.

[14] Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, et al. "Exploring methods for evaluating group differences on the NSSE and other surveys: Are the t-test and Cohen'sd indices the most appropriate choices". In: *annual meeting of the Southern Association for Institutional Research*. Citeseer. 2006, pp. 1–51.

[15] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Bitcoin.org* (2008). URL: https://bitcoin.org/bitcoin.pdf.

[16] Steven Goldfeder, Harry Kalodner, Dillon Reisman, et al. "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies". In: *Proceedings on Privacy Enhancing Technologies* 2018.4 (2018), pp. 179–199.

[17] M. R. Comans, O. N. de Haas, R. Jongerius, et al. "Stop Boiling the Oceans: A Review on Energy Efficient Proof of Work Alternatives". In: *Bachelor Seminar TI3706, Delft University of Technology* (2019).

[18] KNAW, NWO, TO2-federatie, et al. "Nederlandse gedragscode wetenschappelijke integriteit". In: *DANS* (2018). URL: https://doi.org/10.17026/dans-2cj-nvwu.

Appendices

A Ledger submissions

Hour	Address	Missed		Score	
		local	*Ripple*	local	*Ripple*
	May 25th 2020, 11:02:59 CET, 9 hours				
1	NePRTw	0	0	1.0000	0.0000
2	yMRWJW	5	0	0.9947	1.0000
	ZDfnYe	5	0	0.9947	1.0000
	NePRTw	0	0	1.0000	0.0000
3	NePRTw	0	0	1.0000	0.0000
4	NePRTw	0	0	1.0000	0.0000
5	ZDfnYe	1	11	0.9989	0.9884
	NePRTw	0	0	1.0000	0.0000
6	GgPBjp	7	16	0.9926	0.9831
	NePRTw	0	0	1.0000	0.0000
7	NePRTw	0	0	1.0000	0.0000
8	ZDfnYe	5	0	0.9947	1.0000
	NePRTw	0	0	1.0000	0.0000
9	KxD6Hw	0	5	1.0000	0.9947
	ZDfnYe	0	8	1.0000	0.9916
	NePRTw	0	0	1.0000	0.0000
	May 26th 2020, 10:19:08 CET, 14 hours				
2	ZDfnYe	8	0	0.9915	1.0000
	yMRWJW	10	0	0.9894	1.0000
	DHn7GK	10	0	0.9894	1.0000
4	ZDfnYe	11	0	0.9883	1.0000
	yMRWJW	9	0	0.9904	1.0000
	DHn7GK	9	0	0.9904	1.0000
5	ZDfnYe	0	13	1.0000	0.9863
6	ZDfnYe	6	0	0.9936	1.0000
	yMRWJW	6	0	0.9936	1.0000
	DHn7GK	8	0	0.9915	1.0000
7	QVajam	0	26	1.0000	0.9723
8	ZDfnYe	9	0	0.9904	1.0000
	yMRWJW	11	0	0.9883	1.0000
	DHn7GK	11	0	0.9883	1.0000
9	ZDfnYe	0	6	1.0000	0.9936
10	ZDfnYe	5	0	0.9947	1.0000
11	KxD6Hw	0	6	1.0000	0.9937
12	ZDfnYe	8	0	0.9915	1.0000
	yMRWJW	13	0	0.9863	1.0000
	DHn7GK	13	0	0.9863	1.0000
	May 27th 2020, 12:49:19 CET, 7 hours				
1	ZDfnYe	8	14	0.9916	0.9853
	KxD6Hw	0	7	1.0000	0.9926
	QVajam	34	0	0.9642	1.0000
	yMRWJW	9	0	0.9905	1.0000
	DHn7GK	11	0	0.9884	1.0000
2	QVajam	4	9	0.9958	0.9905
3	ZDfnYe	6	0	0.9937	1.0000
	yMRWJW	8	0	0.9916	1.0000
	DHn7GK	7	0	0.9926	1.0000
7	yMRWJW	5	0	0.9945	1.0000
	DHn7GK	7	0	0.9923	1.0000
	May 28th 2020, 12:54:15 CET, 7 hours				
5	yMRWJW	9	0	0.9902	1.0000
7	yMRWJW	7	0	0.9924	1.0000
	ZDfnYe	7	0	0.9924	1.0000
	KxD6Hw	0	6	1.0000	0.9935

Table 4: Validator scores per hour for the validators that our local program and the Ripple score did not agree on. It turns out that in less than 0.1% of the submissions there is disagreement. If *missed* and *score* report 0 simultaneously *Ripple* had no data on this validator. Validator WvcyRj is left out completely because it was unreliable.