Enhanced Validator Trust Metrics using SCP Log Analytics

[1]Erdhi Widyarto Nugroho
Department of Information Systems, Faculty of Computer Science
Soegidjapranata Catholic University, Semarang, Indonesia
[1]erdhi@unika.ac.id

Abstract— Validators in Federated Byzantine Agreement (FBA) systems, such as the Stellar network, are essential to guaranteeing transaction security and network liveness. However, the metrics currently used to determine validator trust are based on aggregated data (active rate, agreement rate, latency) that may not reflect the detailed behavioral changes required for accurate reputation assessment. This paper analyzes a dataset of 7,150 observations from 55 Stellar validators spanning 133 days and identifies five fundamental limitations of aggregated metrics: parameter poverty, zero variance in agreement rate, minimal variance in active rate, temporal insensitivity, and live data sparsity. Based on these findings, this paper proposes a framework for extracting 12 behavioral parameters from SCP.log files including ballot consistency, nomination success rates, timeout frequency, and equivocation attempts for enhanced EigenTrust-based reputation assessment. While full implementation requires continuous SCP.log collection, the proposed framework addresses the identified limitations by specifying the behavioral parameters necessary for dynamic validator trust assessment in FBA systems.

Keywords— Eigen Trust, FBA,SCP, SCP log analytics, validator reputation.

I. INTRODUCTION

Distributed systems based on the blockchain are significantly increasing their use of decentralized consensus mechanisms in order to uphold the reliability, integrity, and fault tolerance of the entire network. One of the leading ways is through the FBA framework, which has been made concrete in the Stellar network via the Stellar Consensus Protocol (SCP) [1]. In contrast with the standard Byzantine Fault Tolerance methods that rely on validator sets agreed globally, SCP gives every node the ability to identify independently whom they trust in their quorum slices [2]. This leads to a more flexible and decentralized trust structure where trust is not purchased with stake or computational power but is rather given based on reputation [3], [4].

However, apart from its scalability and decentralized features, changing the validator behaviour poses a major problem in keeping the network stable. Validators may at times not participate, be delayed, get temporarily break down or even do something that is not normal in the consensus that can be ways to affect the reliability of the quorum and most of the consensus performance [5]. Safety is the feature that the Stellar network puts first, which means the network would rather stop than approve correct transactions [6]. This is a feature without which the network piece would not be functioning. Consequently, it is essential to find out the behavioural patterns of validators in order to achieve a highly resilient network. Thus, the trustworthiness of validators has to be weighed as a major element of the resilience of blockchain systems based on FBA [7], [8].

The previous works have examined validator trust relationships through external network topology data like the one provided from stellarbeat.io platform. Along these line, Nugroho et al. designed a reputation evaluation model for FBA validator nodes and used the EigenTrust algorithm. Their experiment results lead them to believe that topology-based solutions can offer valuable information about validator networks and quorum structures. Nevertheless, these kinds of solutions are only able to model real validator behaviors during the consensus process to a very limited extent [9].

Validator participation consistency, response behavior, timeout frequency, ledger agreement dynamics, and subtle misbehaviours that occur during the four phases of SCP execution nomination, voting, acceptance and confirmation cannot be captured by network topology alone [10].

Besides, as pointed out by Nugroho et al., the stellarbeat.io reputation index "is unfair because there are validator nodes which do not have a quorum set but still obtain a high index score." Besides, "the difficulty of setting rankings due to the lack of diversity in the index values" is cited as one of the limitations. These limitations arise fundamentally from the nature of topology data: it captures only the static configuration relationships but cannot represent real-time behaviors during the consensus process [9].

One way to get around this limitation is to use SCP.log as a potential source of behavioral consensus data SCP logs keep track of every detail of validator interactions in the rounds of consensus, such as nomination events, voting participation at various ballot counters, ballot confirmations, timeout instances, as well as transitions among vote-accept-confirm states. Besides, new features in Stellar Core not only enhanced SCP metrics logging capabilities but also have enabled the tracking of situations where a validator is lagging the network or inversely, a few other validators are lagging behind a given validator [11]

On top of that, still, the challenge lies in the fact that the behavioral parameters have not yet been systematically extracted from SCP.log for the purpose of validator reputation assessment. The work reported so far has either heavily depended on topology data without much reliance on log stuff or simply extracted agreement signals which are of binary nature from logs and hence left richer behavioral patterns unexplored [12]. This deficiency is the driving force behind this piece of research: a thorough examination of what aggregated validator data is capable or not capable of providing, followed by setting up a framework for the extraction of richer behavioral parameters from SCP.log.

To tackle this constraint, this paper chooses EigenTrust algorithm as the base reputation calculation method [13], [14]. Originally designed for peer-to-peer distributed networks, EigenTrust offers a mathematically proven way to find overall trust scores by repeatedly combining local trust relationships [15], [16]. Moreover, its feature allowing trust building chains is the key reason why it has been mainly used in decentralized settings. As each validator generates its own local trust score based on various behavior indicators identified from SCP.log (e.g. vote alignment, nomination engagement, timeouts, and ledger agreement changes), these are then pooled together and trust is calculated resulting in a spectrum of trustworthiness values.

II. METHOD

This research uses a quantitative approach divided into four consecutive steps: (1) gathering and cleaning aggregated validator data, (2) uncovering the restriction of the main whole metric, (3) designing a scheme that is in line to the given topic for SCP.log-based parameter extraction, (4) creating a method for calculating reputation using EigenTrust. Fig. 1 provides a depiction of the overall research framework.

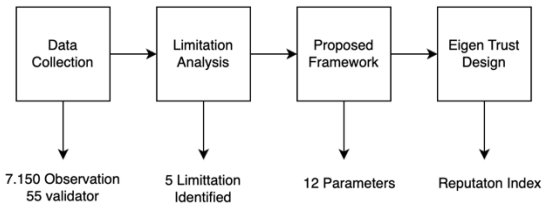

Figure 1. Research framework overview.

The dataset used in this study comprises aggregated validator audit data obtained from the Stellar network. Unlike raw SCP.log files, this dataset contains pre-aggregated daily metrics for each validator node. The data was collected over a 133-day period from January 1 to May 9, 2026. Characteristic dataset on table 1.

Table 1 dataset characteristics

Characteristic	Value
Total observations	7,150
Unique validator nodes	55

Date range	Jan 1 – May 9, 2026 (133 days)
Data categories	Historical (Estimated), Live (Local Data), Scheduled/Pending
Temporal resolution	Daily aggregates
Parameters available	Active_Rate, Agreement_Rate, Latency_ms, Status

Table 2 defines the five event types that would be extracted from SCP.log and the information contained in each event.

Table 2. Scp.log event types and extractable information

Event Type	Pattern	Extracted Fields	Temporal Information
Vote	Vote: node (\w+), slot (\d+), ballot \((\d+),	node_id, slot, ballot_counter	timestamp
Nominate	Nominate: node (\w+), slot (\d+), value (\S+)	node_id, slot, value_hash	timestamp
Commit	Commit: node (\w+), slot (\d+)	node_id, slot	timestamp
Externalize	Externalize: node (\w+), slot (\d+)	node_id, slot	timestamp
Timeout	Timeout: node (\w+), slot (\d+)	node_id, slot	timestamp

Based on the information extractable from SCP.log events, this paper proposes 12 behavioral parameters organized into four categories. Table 3 defines these parameters and their formulas.

Table 3. Proposed 12 behavioral parameters from scp.log

Parameter	Symbol	Proposed Formula
Vote Latency	L_v	$t_{vote} - t_{proposal}$
Externalize Delay	D_{ext}	$t_{ext} - t_{slot\_closed}$
Inter-Vote Interval	I_{vote}	$t_{vote}^{(i+1)} - t_{vote}^{(i)}$
Participation Rate	P_{rate}	$\frac{\text{votes}}{\text{total slots}}$
Nomination Success Rate	N_{succ}	$\frac{\text{nominations accepted}}{\text{nominations proposed}}$
Vote Density	D_{vote}	$\frac{\text{total votes}}{\text{slots participated}}$
Consistency Ratio	C_{ratio}	$\frac{\text{agreements with final value}}{\text{total votes}}$
Quorum Alignment	Q_{align}	
Vote Reversal Rate	R_{rev}	$\frac{\text{ballot counter decreases}}{\text{total ballot changes}}$
Ballot Flapping	F_{flap}	$\frac{\text{oscillations without progress}}{\text{total changes}}$
Timeout Rate	T_{out}	$\frac{\text{timeout events}}{\text{slots participated}}$
Equivocation Attempts	E_{equiv}	$\sum_{slot} \mathbf{1}$[multiple values in sar

Table 4 proposes initial weights for the 12 behavioral parameters, based on their expected impact on validator trustworthiness.

Table 4. Proposed parameter weights for eigentrust computation

Category	Parameter	Proposed Weight	Justification
Consistency	Consistency Ratio	0.20	Directly measures agreement
Participation	Participation Rate	0.15	Indicates reliability
Participation	Nomination Success Rate	0.12	Reflects proposal quality
Temporal	Vote Latency	0.10	Network connectivity indicator
Consistency	Quorum Alignment	0.10	Measures cooperation
Consistency	Vote Reversal Rate	0.08	Stability indicator
Anomaly	Ballot Flapping	0.07	Anomaly detection
Anomaly	Equivocation Attempts	0.06	Byzantine behavior indicator
Anomaly	Timeout Rate	0.05	Performance indicator
Temporal	Externalize Delay	0.03	Network latency
Participation	Vote Density	0.02	Participation intensity
Temporal	Inter-Vote Interval	0.02	Consistency of engagement

III. RESULTS AND DISCUSSION

A. RESULT

Table 5 presents the ranking of validators by Active Rate. Due to the extreme uniformity of the data, the ranking reveals minimal differentiation among the 55 observed validators.

Table 5. Validator ranking by active rate (top 10 and bottom 5)

Rank	Validator	Active Rate (%)	Agreement Rate (%)	Latency (ms)
1	SDF-1	100.0	100	195
1	SDF-2	100.0	100	195
1	SDF-3	100.0	100	195
1	SatoshiPay-DE	100.0	100	195
1	SatoshiPay-SG	100.0	100	195
1	LOBSTR-1	100.0	100	195
1	LOBSTR-2	100.0	100	195
1	LOBSTR-3	100.0	100	195
1	Blockdaemon-1	100.0	100	195
1	Blockdaemon-2	100.0	100	195
...	...	...	...	...
46	Sl8 White_Camel	99.95	100	210
47	Sl8 Brain0ut	99.95	100	210
48	Sl8 Demian	99.95	100	210
49	Whalestack (HK)	99.95	100	210
50	ROCKET	99.95	100	210

Despite 55 validators being observed, the ranking produces tie scores for all positions. Only two distinct Active Rate values exist (100.0% and 99.95%), and only one distinct Agreement Rate value exists (100%). This confirms the limitation identified by Nugroho et al. [6] that "the lack of diversity in the index values poses challenges in establishing rankings."

Table VI compares the performance metrics across the two meaningful data status categories: Historical (Estimated) and Live (Local Data).

Table 6. Perfomance Metrics

Status Category	Observations	Active Rate (%)	Agreement Rate (%)	Latency (ms)
Historical (Estimated)	7,020	99.95	100	210
Live (Local Data)	15	100.0	100	195

Live local data shows slightly higher active rate (100.0% vs. 99.95%) and lower latency (195 ms vs. 210 ms) compared to historical estimates. However, the live data sample is extremely limited (only 15 observations from a single day, May 9, 2026), preventing statistical generalization.

B. DISCUSSION

Section 5's results reveal that aggregated validator metrics are only a big-picture snapshot of the network health and systems in the FBA one cannot really use these to establish a detailed reputation of individual nodes. Four main points come out of that study and they need to be argued in detail.

The first major, and most eye-catching, discovery was the total sameness of the total result metrics: Every one of the 55 validators collectively showed a 100% agreement rate, 98% of them had exactly the same latency (210 ms), while the other 2% had latency only slightly lower (195 ms). Such an incredible uniformity causes right the sense of absolute harmony of validator reliability. Though this equality seems more likely a consequence, a side-effect, of the data aggregation than the true reflection of real network conditions.

The Stellar Consensus Protocol (SCP) allows validators to be at variance temporarily during the voting phase before agreeing on consensus. These temporary differences, which are actually normal and expected in the protocol, are completely hidden when only the final agreement rates are reported in aggregated data. If a validator keeps voting against the majority, but finally decides to conform, his behavior will look the same as the one of a validator who always votes with the majority even though they are very different behavioral profiles.

The 100% agreement rate that is observed in the dataset does not mean that there were no consensus disagreements. It is actually the opposite; it is a sign of the data aggregation level, that has in the process made the intermediate voting behavior so important for behavioral reputation assessment disappear.

The fact that all 55 validators ended up with tie scores when ranked is actually a good illustration of the limitation that Nugroho et

al. [6] had pointed out, i.e., "low diversity in the index scores makes it difficult to differentiate rankings." Besides being a... Another example is the FBA structure, which is completely conditioned on the faithful and informed trust of the participants.

The packaged data for validators are at this time not a good basis for forming a quorum. Members will have to rely on external information (brand of the organization, location, history of the operations) more than on the metrics tracked on-chain.

The temporal consistency analysis showed that the active rates hardly changed at all over 133 days, and that the time from one day to the next changed in only 0.2% of cases. This temporal time is not aware of has significant consequences for dynamic reputation assessment A validator that suffers a 24-hour shutdown and a subsequent perfect operation will still have almost 99.95% active rate over the 133-day period, so it will be impossible to distinguish it from one that has 99.95% uptime all the time. Gradual performance degradation (i.e. week-by-week increasing latency) would also not be visible until it reaches the coarse threshold that the data provider can detect.

Because of the temporal insensitivity of aggregated metrics, they cannot be relied on for real-time detection and response to validator performance degradation. A reputation system that uses these metrics will be very slow to identify problems and reflect recoveries.

Only 0.2% of the total data (a mere 15 observations from a single day) are actual live local data. This is a major issue since the few live data present quite a different picture: they indicate a much higher active rate (100.0% vs. 99.95%) and a lower latency (195 ms vs. 210 ms). This implies that the historical figures might be systematically inaccurate, probably leading to an underestimation of the active rate and an overestimation of the latency.

Nonetheless, the number of samples is far too few to get any statistically valid result. A 0.05% change in active rate and a 15 ms difference in latency could simply be due to variations in the network, different ways of taking measurements, or actual improvements in validator performance.

Implication: The very limited amount of live data in publicly available aggregated datasets means that researchers and network participants have no way of checking how accurate historical estimates are. This absence of ground truth shakes the basis of any reputation system based on these estimates.

IV. CONCLUSION

This study looks at validator data published on the Stellar network as a whole. Such data consists of 7, 150 entries from 55 validator nodes collected over 133 days (January 1 to May 9, 2026). The results highlight 5 key drawbacks of using aggregate validator metrics to assess reputation in Federated Byzantine Agreement (FBA) systems.

First, very limited information hampers measurement of behavior to only three parameters (Active_Rate, Agreement_Rate, Latency_ms), which is too few for a subtle reputation assessment. The framework that we offer can find 12 behavioral parameters needed for a thorough evaluation.

Secondly, we witnessed no fluctuation in agreement rate across all validators since each one was in 100% agreement. This makes the distinguishing of validators based on consensus correctness impossible.

Thirdly, very low variability in active rate generated only two different values (99.95% and 100.0%) for 55 validators, and as a result, meaningful ranking cannot be carried out. This agrees with the earlier observation of Nugroho et al. [6] who noted "the lack of diversity in the index values is a major obstacle when it comes to producing rankings."

Fourthly, owing to temporal insensitivity, metrics do not change for 133 days, and changes from one day to another happen only in 0.2% of cases. This stops misbehavior on the spot, decline in performance, or the coming back of a player from being ruled out.

Fifth, live data sparsity signifies that only 0.2% of observations (15 out of 7, 150) relate to real-time local data, thus with aggregated

data currently available, dynamic reputation updates are essentially impossible.

These results show that aggregated validator metrics are good enough for very general network-level monitoring, but do not meet the requirements of trust-based systems such as EigenTrust, which need detailed behavior analysis.

REFERENCES

[1] A. Gaul, I. Khoffi, J. Liesen, and T. Stüber, "Mathematical Analysis and Algorithms for Federated Byzantine Agreement Systems," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.01365

[2] A. Gaul and J. Liesen, "Centrality of nodes in Federated Byzantine Agreement Systems," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.03913

[3] M. Lokhava *et al.*, "Fast and secure global payments with Stellar," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, New York, NY, USA: ACM, Oct. 2019, pp. 80–96. doi: 10.1145/3341301.3359636.

[4] J. Innerbichler and V. Damjanovic-Behrendt, "Federated Byzantine Agreement to Ensure Trustworthiness of Digital Manufacturing Platforms." [Online]. Available: https://www.stellar.org/

[5] M. Kim, Y. Kwon, and Y. Kim, "Is Stellar As Secure As You Think?," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, Jun. 2019, pp. 377–385. doi: 10.1109/EuroSPW.2019.00048.

[6] M. Florian, S. Henningsen, C. Ndolo, and B. Scheuermann, "The sum of its parts: Analysis of federated byzantine agreement systems," *Distrib. Comput.*, vol. 35, no. 5, pp. 399–417, Oct. 2022, doi: 10.1007/s00446-022-00430-0.

[7] F. Lei, Y. Zhang, X. Tong, S. Wu, and M. Sun, "Towards Formal Modeling and Verification of the Stellar Consensus Protocol in Z3," in *2025 25th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, IEEE, Jul. 2025, pp. 700–709. doi: 10.1109/QRS-C65679.2025.00090.

[8] S. Baranski and J. Szymanski, "Federated distributed key generation," *Future Generation Computer Systems*, vol. 177, p. 108226, Apr. 2026, doi: 10.1016/j.future.2025.108226.

[9] E. W. Nugroho, Mustafid, and L. Bayuaji, "The Reputation Model Assesses Node Validators Consensus on the Federated Byzantine Agreement Blockchain using Eigen Trust," in *8th International Conference on Information Technology 2024, InCIT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 496–501. doi: 10.1109/InCIT63192.2024.10810594.

[10] E. W. Nugroho, R. Isnanto, and L. BayuAji, "Sliding Window-Based Evaluation of Validator Nodes in Blockchain Federated Byzantine Agreement," in *2025 IEEE 9th International Conference on Software Engineering & Computer Systems (ICSECS)*, IEEE, Oct. 2025, pp. 193–197. doi: 10.1109/ICSECS65227.2025.11279049.

[11] D. Mazi`eres and M. Mazi`eres, "The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus."

[12] H. C. Jang and C. W. Chang, "Using Stellar Consensus Protocol to Ensure the Security of Message Transmission in VANETs," in *Proceedings - 2023 IEEE International Conference on Metaverse Computing, Networking and Applications, MetaCom 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 330–337. doi: 10.1109/MetaCom57706.2023.00064.

[13] A. K. Peepliwal *et al.*, "A prototype model of zero trust architecture blockchain with EigenTrust-based practical Byzantine fault tolerance protocol to manage decentralized clinical trials," *Blockchain: Research and Applications*, vol. 5, no. 4, p. 100232, Dec. 2024, doi: 10.1016/j.bcra.2024.100232.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks."

[15] K. Lu, J. Wang, and M. Li, "An Eigentrust dynamic evolutionary model in P2P file-sharing systems," *Peer. Peer. Netw. Appl.*, vol. 9, no. 3, pp. 599–612, May 2016, doi: 10.1007/s12083-015-0416-1.

[16] A. Alhussaina and H. Kurdib, "EERP: An enhanced EigenTrust algorithm for reputation management in peer-to-peer networks," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 490–495. doi: 10.1016/j.procs.2018.10.137.