SOCIALTRUSTR: Blockchain Solution for the Traceability and Validity of Online Content Dissemination

SOCIALTRUSTR: Blockchain Solution for the Traceability and Validity of Online Content Dissemination

Manuel Arto*, Luca Sciullo*†, Lorenzo Gigli*†, Federico Montori*†

*Department of Computer Science and Engineering, University of Bologna, Italy
†Advanced Research Center on Electronic Systems, University of Bologna, Italy

***Abstract*—Disinformation is one of the most insidious challenges of our times, and there is an increasing need for sophisticated systems that can deliver truthful content to users. Traditional centralized approach require the trust on a single entity, which often is subject to bias and cannot keep up the pace to the rate at which fake information is produced. Recent proposals aim to leverage the online communities in the process of fact-checking using blockchain-based systems, which can provide the traceability, immutability and transparency of interactions. In this paper, we propose SocialTrustr, a blockchain-based system aimed at ensuring the traceability and validity of content shared within social environments. SocialTrustr is designed to encourage online honesty by rewarding users who publish truthful content and perform honest validations, through a public consensus mechanism based on reputation. We implement our system, release it open source and evaluate it in a real deployment by simulating online user behavior.**

***Index Terms*—Blockchain, fact-checking, reputation**

I. Introduction

In the recent digital landscape, characterized by the rapid evolution of technologies and the pervasiveness of digital media, disinformation emerges as an increasingly complex and widespread phenomenon. The overwhelming amount of data exchanged has made disinformation one of the most difficult and insidious challenges of our time. This phenomenon fuels the polarization of opinions but also undermines the public trust in the media and institutions [1].

The need for effective tools to track and validate information has become an important matter, driving the adoption of new, innovative approaches. Nowadays it has become clear that conventional solutions can not keep up the pace with the proliferation of disinformation. They in fact often rely on the intervention of experts or fact-checking agencies [2]. However, these approaches reveal clear limitations, primarily due to their centralized nature and susceptibility to bias. Additionally, their slowness and lack of scalability in relation to the growing volume of data present further challenges.

In this context, the blockchain technology emerges as a promising solution to redesign the dynamics around the online trust. Its inherent features such as decentralization and data immutability present a revolutionary potential in the context of online information verification. Integrating such principles in the battle against disinformation could not only improve the efficacy of fact-checking, but also actively involve the online community in the effort.

In this paper, we propose **SocialTrustr**, a blockchain-based system designed to contrast disinformation using an approach based on decentralized reputation systems. Leveraging the "wisdom of the crowds" [3], SocialTrustr provides a platform where each user actively contributes to content validation, being incentivized to behave honestly by a consensus mechanism based on reputation, that was already widely explored in the context of IoT [4]. We believe that our decentralized approach and the validation system based on the voting entropy are a valid alternative to the traditional fact-checking approaches, transforming each user in an active member of the validation community. Our work spans from its theoretical fundamentals to the practical implementation and validation of the system on a real blockchain.

In detail, we make the following contributions:

- We propose a blockchain-based architecture through which users can share content and forward it to others on-chain, using IPFS and The Graph as supporting systems to make content storage and traceability efficient.
- We propose a system through which users can request the validation of a piece of information by relying on a set of special users called verifiers, who reach a consensus mediated by Proof-of-Stake policies driven by user reputation.
- We implement the system on a set of EVM-compatible smart contracts, release it open-source, and evaluate its fairness and resilience by simulating the behavior of a crowd of users over time.

The paper is structured as follows: Section II introduces the background and related works, Section III describes the architecture of the system and how involved actors interact, Section IV explain the dynamics associated with the consensus process and how the system token is managed, Section V explains how reputation values are drawn after the consensus process, Section VI presents our evaluation of the system, finally VII concludes the paper.

II. Related Works

Several solutions have been proposed to tackle the problem of fake news. Some of them focus on automatically verifying news through AI algorithms, while others directly involve the user community in the validation process. However, these solutions present significant issues due to the centralization of the verification authorities. The blockchain has established itself as a common ground to contrast this issue, however, many of the blockchain-based proposed solutions still depend on a certain level of centralization in the selection of validators or in the management of information.

The usage of Distributed Ledger Technologies (DLTs) and blockchain to contrast disinformation was endorsed in [5]. Authors claim that these technologies guarantee the source and the traceability of data, providing an immutable, verifiable and transparent transaction registry, creating a peer-to-peer secure platform for archiving and excanging information according to policies specified in smart contracts.

Following, many approaches were proposed such as [6], which uses an on-chain crowdsourcing mechainsm together with a behavior classification system to distinguish among true and fake news. However, a system based on machine learning outputs could not be completely immune from manipulation and could introduce randomness in the final decisions, which detaches from the deterministic outputs that blockchain systems are built upon.

TRUSTD [7] is a conceptual software framework which adopts blockchain and digital signature for the verification of online contents, including human intervention in the verification process. The framework calculates the trust score of online content on top of a list of validators and their respective trust score. However, the latter is generated by setting the trust between the authors and the validators; this lack objectivity as the same group of verifies may have different outputs depending on the trust scores assigned by different authors.

Setting TRUSTD as a baseline, a system relying on a unique trust value for verifiers is proposed [8]. This solution relies on an incentive mechanism based on the entropy of a given pool of votes by trusted users. The experiments are mostly oriented to the performance side rather than the effectiveness of the system. Moreover, it allows anybody to become a trusted user by buying digital tokens, and the influence and the trust of single users is actually based on the amount of digital tokens owned over the total, centralizing the power in the hands of the "wealthiest".

ProBlock [9] tackles the problem by proposing a dynamic model with a system of safe vote based on blockchain. ProBlock uses a probabilistic model to predict the truthfulness of news based on the feedback of reviewers. The solution presents good results, however it relies on a private blockchain, which hiders the open-access concept that must be part of a disinformation detection system.

The solutions cited above only deal with the news validation, completely disregarding the process of tracking their diffusion and their source. This is taken into account in [10], where a blockchain-based system for tracking fake news from unknown sources is proposed. The solution uses a collaborative on-chain and off-chain storage model and an algorithm to guarantee the consistency between on-chain and off-chain data.

The analysis of the current research landscape underlines how the usage of the blockchain technology in contrasting disinformation is timely, as well as how many of the currently proposed solutions only focus on one aspect of the process, disregarding the others. SocialTrustr aims to provide a holistic approach to traceability and verification in response to this gap.

III. Architecture & Implementation

In this Section, we illustrate the software architecture and the implementation details of the whole system together with the interactions among the software components and the actors involved, as shown in Figure 1. We distinguish three types of users, i.e., (i) the *publisher*, (ii) the *reader*, and (iii) the *verifier*. The *publisher* is meant to publish some content on the platform after becoming a *trusted user*. Following the approach described in [8], this is achievable by purchasing a badge (step #1), whose cost is defined by a predefined amount of tokens required by the platform. This amount is deducted from the system balance, and if the balance is insufficient, the system mints the necessary tokens. The minting of new tokens is reserved exclusively for users who have not yet reached Trusted status. Thereafter, the user saves the media content on IPFS (step #2), obtaining a code identifier (CID) that can be then saved on the chain (step #3); this mechanism preserves all the benefits of on-chain storage but significantly reduces the amount of data that is stored. Contents saved on chain are automatically indexed to ease their future retrieval (step #4) by a user. A *reader* can ask to verify a specific published content (step #5): this triggers a validation phase in which a set of *verifiers* retrieve this content and vote its validity (step #6). Both the steps #3, #5, and #7 are ruled by a mechanism of staking-compensation to avoid Sybil attacks and useless requests, as better detailed in Section V. All the operations involved in the blockchain are performed by custom Smart Contracts written in Solidity; in particular, Smart contracts constitute the core component of the SOCIALTRUSTR system, as they manage the logic for sharing, evaluating, and handling TRS tokens, i.e., the token used in the system for managing consensus, better described in Section IV. The development process for the smart contracts was streamlined and accelerated through the use of Foundry [1], a suite of tools for blockchain development that simplifies tasks such as managing a local blockchain via Anvil, writing tests in Solidity, and creating scripts to interact with the smart contracts deployed.

We decided to divide the system into three smart contracts, each representing a specific functionality within the system: ContentSharing for the operations of content storage, ContentEvaluation for managing the content evaluation provided by the verifiers, and TrustToken for all the operations required

[1] https://github.com/foundry-rs/foundry

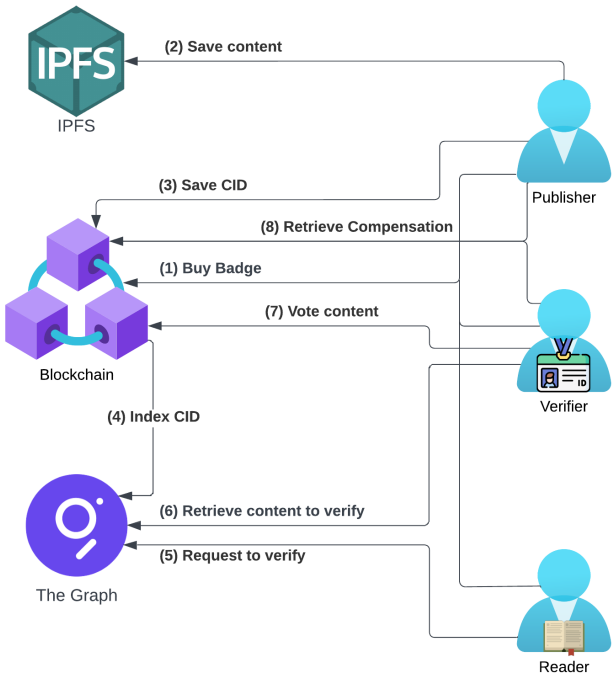


Fig. 1: Software architecture and users'interactions.

by the TRS token. Additionally, we use *The Graph*[2] to automatically index the contents and to create a subgraph that can be queried with GraphQL. All the code written is open source and available online [3].

IV. Consensus & TRS Token

One key challenge of SOCIALTRUSTR architecture is guaranteeing the system's resiliency against malicious attacks. Sybil attacks [11], in particular, can compromise the entire system's integrity since users can create multiple identities to manipulate the final outcome. Following the work of [12], SOCIALTRUSTR proposes a Sybil-resistance mechanism based on the concepts of Badge and Trusted users.

The system uses a consensus similar to Proof-of-Stake, where users must buy a predefined quantity of TRS tokens to become Trusted. These tokens, representing economic value, function as Badges and attest to a user's credibility. This requirement increases the cost of creating multiple identities, thereby discouraging Sybil attacks.

In order to perform actions such as content sharing and validation, users must stake their TRS tokens. This staking mechanism encourages users to act honestly since a misbehavior can result in significant economic losses. On the other hand, the system guarantees token rewards to those users who make valid contributions to the network.

TRS is an ERC-20 token pegged to the value of a fiat currency like USD or EUR and can be exchanged between users of the platform. As previously said, new participants can buy TRS to acquire Badges and become Trusted users. If the amount of tokens within the system is not enough, the SOCIALTRUSTR smart contract protocol will mint the missing amount. The minting process is tightly restricted to new users to prevent inflation and ensure economic stability. Additionally, the protocol allows Trusted Users to sell their TRS back to the system at a fixed exchange rate, providing a flexible exit mechanism while maintaining the token's value consistency. These measures ensure that the token supply dynamically adjusts to user needs without compromising the platform's financial integrity.

V. Content evaluation and Users' Trust

In SOCIALTRUSTR only the Trusted users are capable of sharing contents within the system. However, the content validation is not an immediate process. After a new content submission, there is a fixed period of time during which the other Trusted users can review the information, assess the reliability of the contributor, and assign a Confidence Score.

It is important to decouple the economic stake based on TRS with the user's reliability score, quantified as a numerical value ranging from 0 to 100 (default 50) and denoted as Affidability Factor (AF). This separation is crucial to prevent to over concentrate decision-making power in the hands of few people with huge token holdings. Instead, the reliability metric ensures that trust and consistent contributions over time are the foundation of influence within the system, promoting democratization in content validation.

In Figure 2 we introduce the steps of the algorithm that governs content evaluation, redistribution of tokens, and dynamic reliability adjustment.

1) Trusted shares a content: A Trusted user stakes TRS to share a content, formally starting the evaluation phase.
2) Voting round 24h: Trusted users stake their TRS to evaluate the new content. Each user assign a confidence score to its vote, specifying if the content is valid or not.
3) Calculate Trust Scores: After the end of the evaluation period the system calculate both the Score of True (SoT), representing the possibility that a content is valid, and the Score of False (SoF), representing the possibility a content is invalid. These scores are calculated as:

$$S = \sum_{i=1}^{n} AF_i \times Conf_i \tag{1}$$

4) Determine content validity: After calculating SoT and SoF, the validity of the content is determined by comparing the scores: the content is deemed valid or invalid based on which score is higher.
5) Calculate entropy of Trust Scores: Entropy quantifies the uncertainty or disorder within the evaluations provided by Trusted Users. We used the Shannon's entropy formula [13] to model the level of consensus among users:

$$Entropy(S) = -\sum_{i=1}^{k} p_i \log_k p_i \tag{2}$$

[2]https://thegraph.com/
[3]https://github.com/ManuelArto/SocialTrustr

S is the set of evaluation outcomes, k is equals to 3 since we have 3 possible evaluation classes: True, False, and Neutral. Hence, we define:

$$\begin{aligned} p_1 &= \frac{\sum_{i \in T_T} Conf_i}{n} \\ p_2 &= \frac{\sum_{i \in T_F} Conf_i}{n} \\ p_3 &= 1 - p_1 - p_2 \end{aligned} \tag{3}$$

These proportions correspond to the 3 classes: p_1 to True; p_2 to False; p_3 expresses for the evaluators expressing their opinion as Neutral.

6) Calculate Rewards and Punishments for TRS: We describe the case for $SoT > SoF$, since we follow the same calculations for the opposite scenario. Our algorithm redistributes the staked TRS tokens based on the performance of the different users. When users provide incorrect evaluations, their staked tokens are redistributed to those who evaluated correctly, proportionate to the confidence scores expressed. The amount of tokens removed from the stake of T_F users is calculate following:

$$Punishment = \sum_{i \in T_F} -(Stake * Conf_i) \tag{4}$$

while we redistribute the removed tokens to T_T users according to this formula:

$$Reward_i = \frac{Conf_i}{Tot_{Conf}} * Punishment \tag{5}$$

7) Adjust trust levels: The user trust level is updated following the calculated entropy that directly affects the penalties and rewards. High entropy means that there is a lot of uncertainty between the evaluators, and for this reason the system decrease the severity of the penalties since it's difficult to identify a clear outcome. In order to highlight the importance of the evaluations, both the penalties and the rewards are scaled by the confidence score. In this way the system encourages users to carefully consider their assessments and discourages hasty or uninformed decisions. The penalties assigned for incorrect evaluations are intentionally designed to be more severe then the rewards:

$$Punishment_i = AF_i * Conf_i * (1.0 - Entropy) \tag{6}$$

In contrast, user rewards for correct evaluations are inversely proportional to their trust level. This design choice reduce the risk of centralization, eliminating the possibility for users to cumulate too much trust: users with the higher trust level will receive smaller increments to their trust score compared to less reliable users. The reward for each correct evaluator is computed as:

$$Reward_i = \frac{(100 - AF_i) * Conf_i * (1.0 - Entropy)}{M} \tag{7}$$

In this equation, M is a constant (e.g., 2, 2.5, or 3) that ensures rewards remain significantly smaller than penalties, maintaining a stringent system against fraudulent behavior.

VI. Evaluation

In this Section we provide the evaluation of the SOCIALTRUSTR solution that is twofold: on one side we analytically analyzed the behavior of the system focusing on the distribution of users' trust and TRS tokens from a global perspective, on the other side we tested SOCIALTRUSTR in a real chat application for mobile devices. For the first part we used a Monte Carlo simulation to demonstrate how the SocialTrustr mechanism incentivizes honest behavior and penalizes manipulation within a decentralized digital ecosystem. It is important to note that the simulation script is based on pseudo-random generation of evaluations, which does not accurately simulate a real-world application of the system. In real usage, user evaluations would be influenced by complex factors such as biases or personal knowledge about the content being assessed and we leave the test and collection of real data as future work.

The simulation was conducted with 10 simulated users over 200 iterations. The initial state of the system is represented by a list of trust levels (set to 50 initially) and a list of TRS token quantities (set to 500 initially) for each user. In each iteration, a user is randomly selected to share content and stake a certain number of tokens (set to 20 TRS). Only users with sufficient tokens can share and evaluate content.

Other users are simulated to stake a specified amount of TRS (set to 10 TRS) and evaluate the reliability of the shared content, with 70% and 30% proportions respectively for True and False evaluations, and a confidence score within the range of 40% to 100%. This decision was made to simulate a more realistic environment where users are motivated to share honest content. At the end of each iteration, we collect the content reliability and TRS tokens for each user.

Figure 4(a) examines the evolution of reliability within the simulation. The outcome of a validation is determined by a weighted average of evaluations, meaning the system will always reward the majority of users. This is evident in the graph due to the increasing trend of the sum of reliability values. Additionally, in the final set of iterations, it is noticeable that no user holds a significantly higher reliability compared to others. In other words, it is impossible to autonomously increase one's reliability, which prevents a centralization of decision-making power. It is also evident that the impact of penalties is almost doubled (M=2.5 in Equation 7). However, this does not prevent a user from increasing their reliability through subsequent correct behavior. Two examples visible in the graph are users number 8 and 9, who have a lower

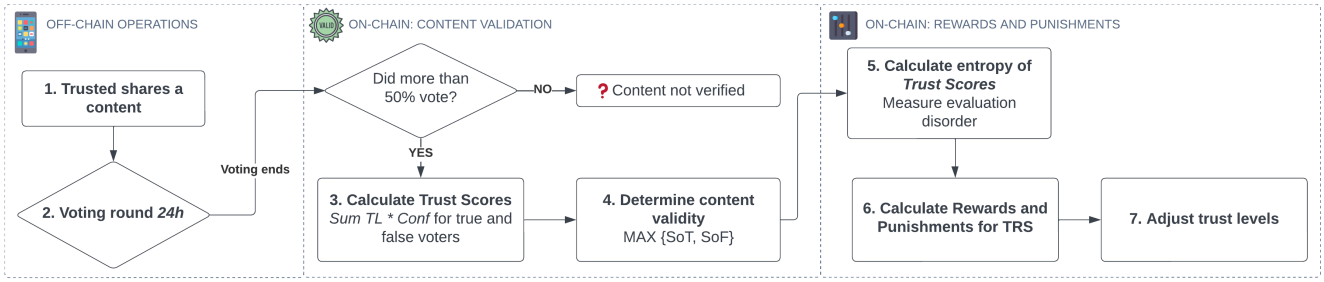


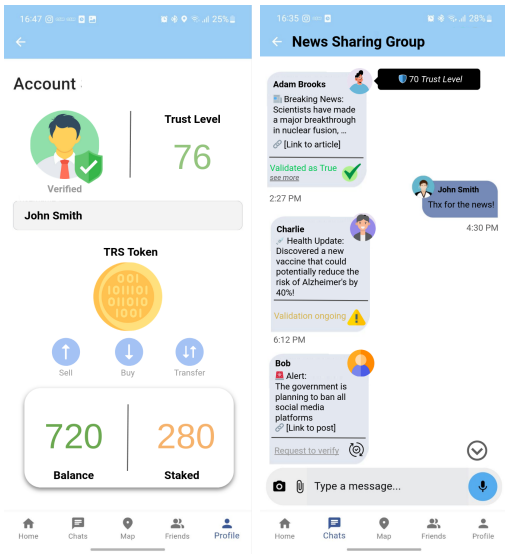
Fig. 2: Content evaluation flow.

reliability than others at iteration 100 but conclude with a reliability above 80 by the end of the simulation.

Figure 4(b) reveals the dynamics of the system's economic factors. The graph shows that, over each group of iterations, the total quantity of TRS remains unchanged. This occurs because the TRS lost is redistributed as rewards to users who evaluated content correctly. This mechanism allows the system to resist hoarding and manipulation. It is important to note that, with only ten users in the simulation, the earnings are not substantial. In fact, after 200 iterations, the user with the highest number of TRS achieved a total gain of only about 100 TRS. The introduction of a token cost for sharing and evaluating content acts as a deterrent mechanism, discouraging malicious activities such as spam and falsification.

Entropy is a crucial element in the tuning mechanism of SOCIALTRUSTR. Figure 5 illustrates the fluctuations in the delta related to increases and decreases in user reliability for each entropy value calculated during the first iteration of a simulation repeated 10,000 times. Low entropy indicates an orderly system with predictable behavior and high reliability of evaluations. Conversely, high entropy may signal attempts at manipulation or uncertainty in evaluations. The most significant feature observed in the plot is the decrease in the delta of increases and decreases as entropy rises. This reduces both the rewards and penalties for reliability when evaluation disorder is high, as shown in Equations 6 and 7. This mechanism not only minimizes reliability adjustments in cases of uncertain validations but also enables harsher penalties for users who evaluate anomalously compared to the majority. By using entropy as a tuning mechanism, the system can automatically adjust the parameters related to reliability, strengthening defenses against manipulation and maintaining high collective honesty.

These analyses reveal that the SOCIALTRUSTR system has a robust and sophisticated tuning mechanism based on entropy, capable of incentivizing honesty and containing manipulations. For the second part of the evaluation, we integrated the SOCIALTRUSTR functionalities into Livechat [4], a mobile messaging application with advanced social features such as real-time chat, location sharing and a leaderboard based on daily step counts. Figure 3(a) shows the user's profile, where the information about her trust level and the TRS balance in its wallet is displayed; figure 3(b) shows a detail of a group chat, in which some messages have been verified and validated as *true*, while others are still pending or not asked to be validated.



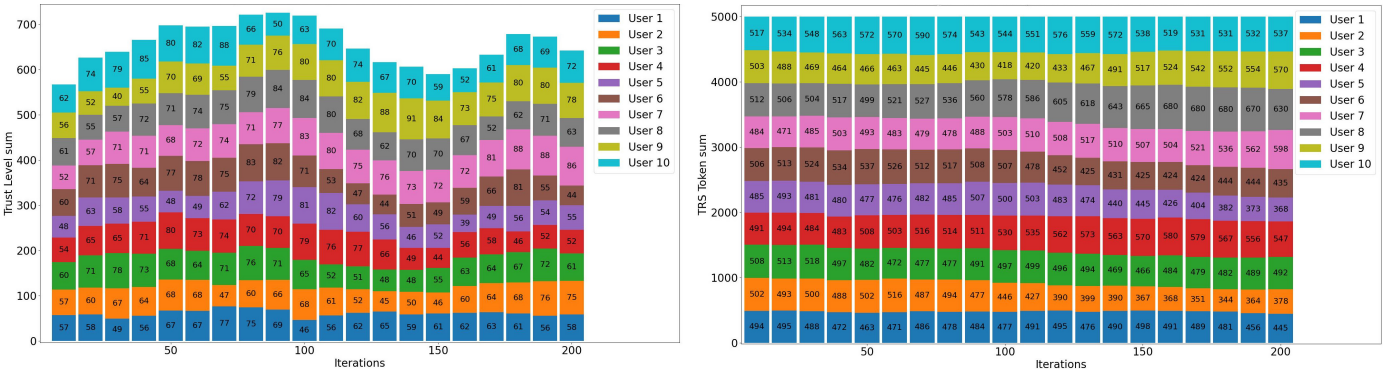
(a) User's profile (b) Chat detail

Fig. 3: Screenshot of Livechat mobile app, where SOCIALTRUSTR has been integrated.

VII. Conclusion

In this paper we explored the concept, design, and implementation of SOCIALTRUSTR, a blockchain-based system aimed at enhancing the authenticity and traceability of content shared online to tackle the spread of misinformation and fake news. SOCIALTRUSTR addresses this challenge by introducing a digital token and an entropy-based incentive mechanism to minimize manipulation and reward honest behavior. User evaluations directly influence the online reputation of both users and shared content, fostering a decentralized and peer-driven approach to content validation. Simulation analyses demonstrate the system's effectiveness in promoting honesty and penalizing deceptive behaviors.

[4]https://github.com/ManuelArto/LiveChat



(a) (b)

Fig. 4: 4(a) Analysis of user's trust across 10 users and 200 iterations and 4(b) analysis of TRS redistribution across 10 users and 200 iterations

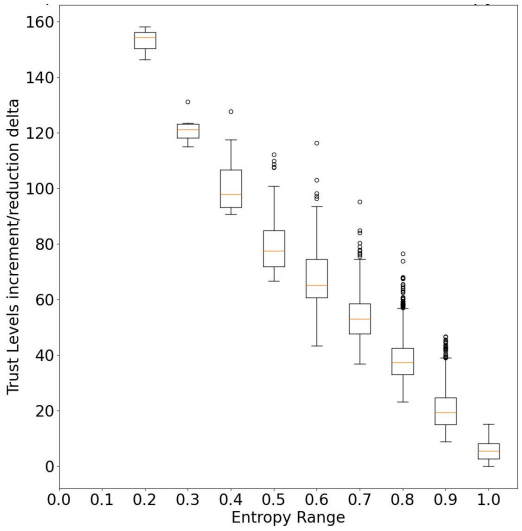


Fig. 5: Analysis of Entropy and Delta in Reliability Increase/Decrease Across 10 Users and 10,000 Iterations.

Future developments of SOCIALTRUSTR will focus primarily on testing its practical effectiveness in real-world scenarios with actual users, collecting meaningful behavioral data to refine and optimize its reward and penalty mechanisms. These tests will provide opportunities to enhance the system's efficiency and scalability. Further, efforts will be directed towards expanding SOCIALTRUSTR application to broader platforms to observe its impact on a larger scale. Lastly, interoperability with multiple blockchains will be explored to extend the project's reach, improve security, and optimize on-chain transaction costs.

REFERENCES

[1] T. Hayward, "The problem of disinformation," *Social Epistemology https://www. tandfonline. com/journals/tsep20*, 2023.

[2] N. Walter, J. Cohen, R. L. Holbert, and Y. Morag, "Fact-checking: A meta-analysis of what works and for whom," *Political communication*, vol. 37, no. 3, pp. 350–375, 2020.

[3] J. Allen, A. A. Arechar, G. Pennycook, and D. G. Rand, "Scaling up fact-checking using the wisdom of crowds," *Science advances*, vol. 7, no. 36, p. eabf4393, 2021.

[4] L. Gigli, I. Zyrianoff, F. Montori, C. Aguzzi, L. Roffia, and M. Di Felice, "A decentralized oracle architecture for a blockchain-based iot global market," *IEEE Communications Magazine*, vol. 61, no. 8, pp. 86–92, 2023.

[5] P. Fraga-Lamas and T. M. Fernandez-Carames, "Fake news, disinformation, and deepfakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality," *IT professional*, vol. 22, no. 2, pp. 53–59, 2020.

[6] T. Yilmaz and Ö. Ulusoy, "Modeling and mitigating online misinformation: a suggested blockchain approach," *arXiv preprint arXiv:2303.10765*, 2023.

[7] Z. Jaroucheh, M. Alissa, W. J. Buchanan, and X. Liu, "Trustd: Combat fake content using blockchain and collective signature technologies," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1235–1240, IEEE, 2020.

[8] C.-C. Chen, Y. Du, R. Peter, and W. Golab, "An implementation of fake news prevention by blockchain and entropy-based incentive mechanism," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 114, 2022.

[9] E. Sengupta, R. Nagpal, D. Mehrotra, and G. Srivastava, "Problock: a novel approach for fake news detection," *Cluster Computing*, vol. 24, pp. 3779–3795, 2021.

[10] X. Wang, H. Xie, S. Ji, L. Liu, and D. Huang, "Blockchain-based fake news traceability and verification mechanism," *Heliyon*, vol. 9, no. 7, 2023.

[11] S. Zhang and J.-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5715–5722, 2019.

[12] C.-C. Chen, Y. Du, R. Peter, and W. Golab, "An implementation of fake news prevention by blockchain and entropy-based incentive mechanism," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 2476–2486, 2021.

[13] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.