

# An Implementation of Fake News Prevention by Blockchain and Entropy-based Incentive Mechanism

Chien-Chih Chen<sup>1,2\*</sup>, Yuxuan Du<sup>1</sup>, Richards Peter<sup>1</sup>  
and Wojciech Golab<sup>1,2</sup>

<sup>1</sup>Electrical and Computer Engineering, University of Waterloo,  
University Avenue West, Waterloo, N2L 3G1, Ontario, Canada.

\*Corresponding author(s). E-mail(s): [j2255che@uwaterloo.ca](mailto:j2255che@uwaterloo.ca);  
Contributing authors: [d9du@uwaterloo.ca](mailto:d9du@uwaterloo.ca); [r2peter@uwaterloo.ca](mailto:r2peter@uwaterloo.ca);  
[wgolab@uwaterloo.ca](mailto:wgolab@uwaterloo.ca);

## Abstract

Fake news is undoubtedly a significant threat to democratic countries nowadays because existing technologies can quickly and massively produce fake videos, articles, or social media messages based on the rapid development of artificial intelligence and deep learning. Therefore, human assistance is critical if current fake news prevention systems desire to improve accuracy. Given this situation, prior research has proposed to add a quorum, a group of appraisers trusted by users to verify the authenticity of digital content, to the fake news prevention systems. This paper proposes an Entropy-based incentive mechanism to diminish the negative effect of malicious behaviors on a quorum-based fake news prevention system. In order to maintain the Safety and Liveness of our system, we employed Entropy to measure the degree of voting disagreement to determine appropriate rewards and penalties. Moreover, we use Hyperledger Fabric, Schnorr signatures, and human appraisers to implement a practical prototype of a quorum-based fake news prevention system. Then we conduct necessary case analyses and experiments to realize how dishonest participants, crash failures, and scale impact our system. The outcomes of the case analyses and experiments show that our mechanisms are feasible and provide an analytical basis for developing fake news prevention systems. Furthermore, we have added six innovative contributions in this extension work compared to our previous workshop paper in DEVIANCE 2021.

**Keywords:** Proof of Stake, Security, Deviance, Social network, Social media

# 1 Introduction

## 1.1 Motivation

In recent years, the influence of fake news has increased, and it has caused great harm to democracy, the press, and freedom of speech. For example, fake news significantly influenced the 2016 U.S. presidential election and the Brexit referendum [1]. Many countries currently enact laws requiring technology companies, including Facebook and Google, to prevent fake news on their platforms [2]. Therefore, technology companies are actively combating fake news to respond to the compliance requirements from governments. For example, Facebook cooperates with news experts to identify fake news on its platform [3]. Google also has actively established policies to prevent the spread of fake news on its products [4]. Even the IEEE [5] is aware of the seriousness of fake news proliferation and has formulated a guideline to advocate human welfare.

However, due to the platforms' social media characteristics, such as Facebook, Twitter, and Instagram, the rapid delivery of information makes it challenging to prevent fake news. Modern users receive countless messages daily, so confirming each news item's authenticity is difficult. Moreover, fake news might come from authoritative news websites as well, such as CNN, BBC, or The Wall Street Journal. In addition, with the development of data technology and artificial intelligence (AI), experts can quickly analyze Internet users' habits and then manipulate the public to achieve their desired results by circulating fake news [6]. Therefore, solely relying on AI and other technologies to automatically identify fake news is difficult.

Bitcoin, first launched by Satoshi Nakamoto in 2008 [7], has received tons of attention and brought about the trend of cryptocurrencies. Today, when people mention blockchain, the first thing that comes to mind is cryptocurrencies. However, the blockchain's ability goes beyond just supporting the development of cryptocurrencies. Because of blockchains' several significant characteristics, it is helpful for people to use them in many scenarios requiring authenticities. These scenarios include food source, resume verification, voting calculation, and many other fields [8][9]. Combating fake news is exactly an application scenario suitable for using blockchain [9].

At present, some researchers have begun to apply blockchains to AI-based fake news prevention systems [10][11][12]. However, even with the assistance of blockchains, fake news prevention systems that rely solely on AI still cannot overcome the technical limitations of AI in catching bogus messages or fake videos. Several studies have already suggested that although AI can detect fake news to a certain extent, human participation is equally important in identifying disinformation rather than merely depending on AI [13][14].

Jaroucheh, Alissa, Buchanan, and Liu (2020) proposed a conceptual software framework that uses a blockchain and digital signatures to verify online content by leveraging humans’ knowledge, not just relying on AI [15]. With the help of digital signature and blockchain technology, their framework calculates online content’s *trustworthiness level* based on the *trustiness values* of each appraiser in users’ appraiser lists. However, the *trustworthiness level* of [15] is generated by appraisers’ trust levels set by users. This approach lacks objectivity because the *trustworthiness level* of the same news verified by the same group of appraisers will lead to different *trustworthiness levels* if various users set different trust levels for the same group of appraisers. Then, the *trustworthiness level* will lose its correspondence to the authenticity of the news. Unfortunately, in [15], no mechanism is designed to avoid this situation. Instead, although our system allows users to define their own trusted appraisers, they cannot set trust levels for appraisers. We use credit points and the confidence of content creators and appraisers to calculate trust scores for users. The credit points of each content creator and appraiser are calculated based on a *Credit Point Tuning Process* that is introduced in Section 4.2.1. Referring to trust scores, users could decide whether they desire to trust a piece of news or not. In this way, our system could offer objective trust scores generated from the crowd intelligence to contribute to curbing fake news.

## 1.2 Our Contributions

The main contribution of our previous workshop paper in Deviance 2021 [16] is that we completed the following improvements for the concept that was offered by [15]:

### ***Offering a proof of stake entropy-based incentive mechanism to diminish effects from malicious nodes***

In this paper, we design a novel incentive mechanism on top of the proof of stake [17] concept to decrease the possibility of generating false information from content creators. Besides, our incentive mechanism could encourage content appraisers to contribute their effort to inspect content to provide trust scores based on their credit points and confidence toward a specific appraisal opinion they offer. Moreover, we adopt entropy [18] to calculate the rewards and punishments of appraiser results to adjust each appraiser’s stakes and credit points. Depending on the incentive mechanism we propose, our system’s trust score has more credibility than the *trustworthiness level* in [15] and possesses the capability to eradicate ineligible appraisers. In sum, our system mainly focuses on offering a well-designed incentive mechanism to curb selfish behaviors from content creators or appraisers rather than providing ground truth.

### ***Providing a concrete implementation to prove our concept***

We not only use Hyperledger Fabric [19] to implement the blockchain network but also apply the Schnorr signature [20] to verify appraising results and confidence sent by human appraisers. In contrast, [15] did not state the detail of implementing the blockchain network. In brief, our implementation transfer the concept of [15] to a real-world application. With the immutability and traceability that blockchains provide, the content creators could utilize our system to prove the authenticity of their content. Users also could retrieve trust scores to realize whether they could trust target news.

### ***Proposing a feasible crash failure handling and scalability mechanism***

It is indispensable for a distributed system to work without any disturbance when any node in its network fails. On the other hand, scalability is also essential for a distributed system. Therefore, in this work, we design a mechanism to prevent crash failure with scalability. However, in [15], the authors only suggested high-level architecture and did not discuss their crash failure handling and scalability mechanism.

### ***Demonstrating abundant case analyses and experiment results to facilitate future research***

We provide various case analyses and experimental evaluations on our system’s incentive mechanism, crash failure prevention, and scalability, while [15] did not mention any experimental data. The outcomes of analyses and experiments offer an analytical benchmark for future quorum-based fake news prevention systems.

Moreover, we added *six novel contributions* compared to our previous workshop paper in Deviance 2021 [16]. **First**, we added the factor of *confidence* in the formula of calculating the **Reward of Content (RoC)** and **Punishment of Content (PoC)** in (7) as well as a new category of appraising result, *Neutral*, to the original two types of appraising results in the formula of calculating the *Entropy(S)*. **Second**, we conduct different sensitivity analyses of our entropy-based incentive mechanism based on the new three types of appraising results. **Third**, we propose a new process regarding how to use secret keys of AES to secure appraising actors (AAs) voting opinions to keep AAs appraising content independently. **Fourth**, We discuss in more detail how our system keeps the two critical factors: Safety and Liveness of a distributed system. **Fifth**, we propose a strategy to optimize the *BRRate* and *BPRate* to keep **Safety** and **Liveness** at the same time. **Finally**, we propose an approach to avoid the content creator sending duplicate content to get undeserved rewards. In conclusion, our primary goal is to provide **trust scores** to assist users in judging news by themselves rather than offering ground truth.

Concerning the remainder structure of this paper, we will explain related works and background to which we refer in Section 2 and Section 3. Afterward,

our methodology and implementation will be described in Section 4, including architecture, incentive mechanism, business logic layer, data access layer, and how we use the *AES* algorithm to make AAs vote independently. Then the explanations of case analyses and experiments are shown in Section 5. Finally, the conclusion and future work is stated in Section 6.

## 2 Related Works

Nowadays, many content-sharing platforms, such as Medium or Reddit, allow users to edit and share all kinds of content freely. However, the audit-free content publishing mechanism can effortlessly become a platform for spreading fake news because no one, even administrators of the content-sharing forum, takes the initiative to verify content [15]. In addition, many social media platforms, such as Twitter, Facebook, or Google, have possibly become hotbeds for fake messages because users can publish content at will as long as they do not violate social media norms. Regrettably, the social platforms do not offer reliable enough verification mechanisms for the authenticity of each content as well [3][4][15].

According to the major international economic and political crises in recent years, it is indisputable that fake news threatens national security [1][15]. Thus, governments of various countries have formulated policies to curb the unhealthy trend of producing or spreading false information. In order to prevent the spread of fake news on content sharing or community platforms, national policies tend to require platforms to be responsible for the content. If platforms plan to verify the authenticity of digital content, some measures must be taken to ensure that the platform's AI exhibits the correct behavior [15]. However, these platforms' measures lack users' involvement. For instance, although Facebook has its internal authenticity verification algorithms for the content and tries to flag fake news, those algorithms are black boxes to the public. Platforms' logic of distinguishing false news is debatable because the users do not participate in the algorithm design, and the fake news identified by the platform does not necessarily conform to the user's definition of fake news [21].

In response to government regulations, technology companies have been working hard to improve platform content quality and increase the speed and accuracy of detecting fake news [15]. For example, Google mentioned in its white paper that AI would be used to improve the algorithm's accuracy in detecting fake news for detecting fake news [4]. To enhance the quality of content, Facebook hires authorities in various fields to review content and actively build an ecosystem that is comprised of professional content production organizations and individuals [3]. Nevertheless, these efforts are not enough. As [15] mentioned, manual mechanisms need to be added to the AI fake news detection system to remove blind spots of AI. Because of recent developments in AI and deep learning, only using algorithms to check for fake news automatically

is difficult. It is unreliable and impractical to rely entirely on algorithms to prevent fake news because there are always blind spots in AI [15].

The root cause of blind spots in AI is that if planning to build an AI-based fake news detection model, we must analyze false news attributes [15]. For example, Sirajudeen, Fatihah, Adamu, and Abubakar (2017) proposed a set of algorithms that can analyze the network packets of the content to collect different types of false news [22]. Another solution was provided by “Reality Defender” to analyze fake news patterns in content [23].

However, AI models mainly adapt voice recognition, image recognition, or machine learning algorithms to detect various false news. As we mentioned, those algorithms possess inherent technical limitations to hinder them from ideally screening incorrect information. In other words, the correctness of detecting fake news of AI models relies on massive data that people could collect. Nevertheless, we cannot guarantee that we will collect enough samples for AI training plans every time. Consequently, the technical limitations of underlying AI technologies are why automatic false news detection accuracy is always lower than manual detection [15].

In addition, since the characteristics of blockchains have been widely used in various next-generation Internet application services [8][9], researchers began to try to add blockchains to fake news prevention systems [9]. The most intuitive form is to apply AI techniques on blockchain nodes or pre-defined smart contracts to detect fake news. The so-called *smart contract* is a pre-defined code deployed in each node on the blockchain network to execute automatically according to the contract content [24]. For instance, Jing and Murugesan (2018) proposed a blockchain-based fake news detection system. The digital content that individuals publish will be recorded on a blockchain as transactions, and each block is connected to the other through hash pointers. Users in their blockchain network can apply any AI techniques to detect whether any one of those transactions in blocks is fake news [10]. Also, Torkey, Nabil, and Said (2019) define the credibility attribute for news in their blockchain-based fake news detection system. The credibility is mainly calculated based on the number of times the news content has been shared and the total number of past sharing times and followers of the news source. When a user broadcasts Tweets or Posts worth sharing to each node in the blockchain network, each node will calculate the credibility value of the news according to the pre-defined smart contract. If the credibility of a Tweet or Post is less than a threshold value defined in advance, the Tweet or Post will be regarded as false news [11]. Then, Dhall, Dwivedi, Pal, and Srivastava (2021) used the Hyperledger Fabric to record and track the news source and determine whether a piece of information is fake by observing the forwarding rate of the information exceeds a pre-defined threshold value. Similar to [10], any node of [12] can also use any AI techniques.

Though, as we mentioned earlier, AI technology has its limitations. Even with the powerful features of blockchain, it is unreliable to use AI alone without human intervention. For example, Huckle and White (2017) use Ethereum

and AI to build a system that can only verify sources of digital content rather than verifying content. They mentioned in their research that the ability to verify the authenticity of digital content belongs to humans, so people cannot rely on AI techniques alone to develop fake news detection systems [13]. Similarly, Ansari, Azhar, and Akhtar (2022) conducted a comprehensive analysis of 34 studies related to misinformation and disinformation. Their research also pointed out that AI can indeed help identify fake news, but it still needs the help of human knowledge [14].

Due to AI's limitations in verifying content authenticity, researchers have begun to leverage human capabilities in fake news prevention systems. Wahane and Patil (2022) focus on manual inspection of news sources, believing that news media with high reputations will not produce fake news [25]. However, we cannot trust that only low-reputation news sources produce fake news. Even for high-reputation news sources, care must be taken to guard against issues of careful subjectivity, bias, and malicious behavior [26]. Also, Pawlicki and Jahankhani (2022) pointed out that reputable news sources do not mean that they will absolutely not produce fake content and must be further verified by humans [27].

To mitigate the shortcomings of only checking sources, researchers have begun to focus on leveraging human power for checking news content [28][29][30][31][32][33][34]. However, in those system architectures in [29],[30],[31], and [34], only news media can publish messages, and ordinary people cannot post any articles or videos. These architectures aim to reduce the chances of misinformation generated by news publishers by managing the quality of news publishers. However, it is not suitable for social media platforms to only allow news media to publish news because, based on the essence of social media platforms, everyone should be able to publish digital content freely.

In [15], their system allows user to define their appraising quorum, a group of verifiers, to assist users in verifying the authenticity of the content. Crowd intelligence could alleviate blind spots from appraisers. In addition, being decentralized, immutability and traceability, it is suitable for blockchains to be utilized to form an anti-counterfeiting mechanism. Accordingly, [15] recommends storing appraising results of digital content via blockchains as well.

Moreover, as [15] mentioned, manual verification is divided into centralized (expert-based) and decentralized (crowd-sourced). Centralized is performed in platforms like PolitiFact [35] or HoaxSlayer [36]. Those platforms have a group of experts with the background to verify the authenticity of the content. Still, as mentioned about Facebook's controversial internal censorship mechanism [21], experts' judgment to detect fake news differs from that of the users. The decentralized method distributes the content to multiple individuals for content review, but this is back to a state of no control mechanism. Furthermore, it is impossible to confirm the reviewer's educational background, whether it is fair without any partiality, and whether they have malicious

motives [15]. We believe that the decentralized method aligns more with the spirit of blockchains’ decentralization. Still, there must be a set of incentives for each content verifier to work on improving personal background knowledge and review content honestly. As [26][37] pointed out, designing a fake news prevention system requires a well-designed incentive mechanism to encourage participants to verify digital content honestly.

Because it is essential to apply incentive mechanisms to assure humans honestly verify digital content, several studies have tried introducing incentive mechanisms into fake news prevention systems [38][39][40]. Chen, Srivastava, Parizi, Aloqaily, and Al Ridhawi (2020) propose a credibility-based score system. News publishers are rewarded with increased credibility for publishing real news or reporting fake news and less credibility when they are identified as publishing fake news. On the other hand, news verifiers increase their credibility when they successfully report fake news; otherwise, their credibility decreases [38]. Zen, Hong, Mohan, and Balachandran (2021) use Ethereum form tokens to reward news validators for correct opinions and exponentially deduct Ethereum form tokens from validators if they offer incorrect news verification results [39]. Farooq, Ashraf Makhdomi, and Altaf Gillani (2022) proposed a dynamic reputation system as its incentive mechanism. When a verifier makes a correct decision, its reputation score will increase. Instead, the verifier’s reputation score will decrease. When a user’s reputation score is too low, that user will not be allowed to verify digital content [40].

Although [38][26][39] all have design incentive mechanisms, there are two main problems with these incentive mechanisms. First, the verification difficulty of each news is different, and it is unfair to provide the same reward or punishment for each news. Second, since blockchain is one type of distributed system, as mentioned in [37], we must design an appropriate reward and punishment adjustment mechanism to ensure that the system maintains sufficient **Safety** and **Liveness**. For keeping **Safety** in an incentive mechanism, punishments should remain at a reasonable level so that validators could not earn rewards by only relying on guessing rather than doing their best to verify digital content. However, to maintain **Liveness**, rewards must be high enough and must be greater than punishments so that users are willing to participate in verification.

Consequently, our system uses entropy to design a mechanism that can adjust rewards and punishments in a quorum-based fake news prevention system. To the best of our knowledge, this is the first proposed work to apply the concept of entropy to design the ratio of reward and punishment in a fake news prevention system. Moreover, in Section 5.3, we propose a strategy to set the ratio of BR and BP to ensure that the system maintains **Safety** and **Liveness**. Then, Section 5.4 will introduce a more detailed analysis of how our incentive mechanism assists our system in maintaining both **Safety** and **Liveness**.

## 3 Background

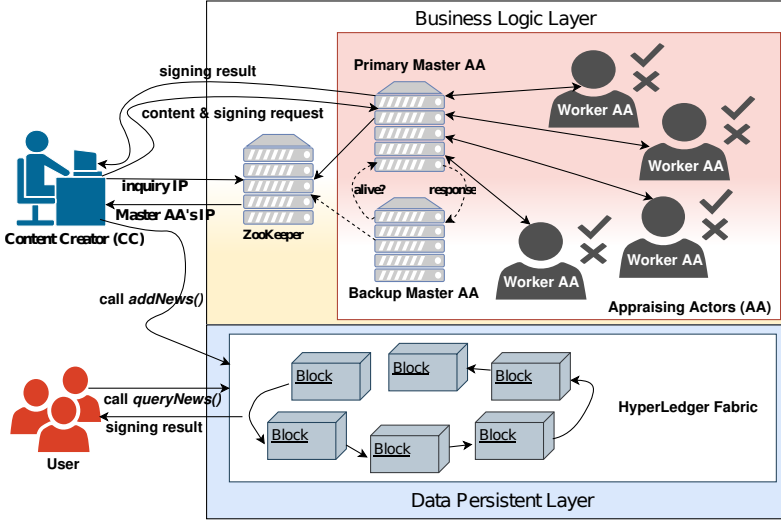
Hyperledger Fabric is one of the projects under the Linux foundation’s Hyperledger umbrella. Today, it is one of the popular private and permissioned Distributed Ledger Technologies (DLT). Hyperledger Fabric [19] states that it was designed for solving several enterprise use cases that required permissioned networks, manageable identities, high transaction throughput, and low latency of transaction confirmation while preserving privacy and confidentiality requirements of transactions and data associated with the transactions. Because of its design, Hyperledger Fabric does not require a native currency such as Bitcoin to perform its normal operations. Also, it eliminates the cryptographic mining process, usually performed on permissionless blockchains.

On the other hand, Hyperledger Fabric enables developers to use the corresponding Software Development Kit (SDK) to write Smart Contracts (also called *Chaincode* in Hyperledger Fabric) in various programming languages and then deploy Smart Contracts on Hyperledger Fabric’s network. Further, these Smart Contracts can be invoked using the command line interface of Hyperledger Fabric or clients’ codes developed using the SDK. In its internal design, Hyperledger Fabric lets organizations, also known as *members*, form consortiums on a high level, and these consortiums are responsible for owning and maintaining the Distributed Ledger within their network. Further, Hyperledger Fabric allows organizations to deploy smart contracts to perform transactions within the network without a central authority. These smart contracts are packaged into a chaincode in Hyperledger Fabric [19]. Each participating organization can execute smart contracts implemented by chaincodes and perform transactions with the distributed ledger.

## 4 Methodology and Implementation

### 4.1 Architecture

As the first contribution we mentioned in Section 1, we use Hyperledger Fabric as our underlying Blockchain network infrastructure, while [15] did not state how they implement their blockchain network. Moreover, to enhance performance, we use the Schnorr signature instead of the two-round trip collective signature that takes more time to finish the signing process [15] adopted. Besides, as shown in Fig. 1, we divided the architecture presented in [15] into two main layers, Data Persistent Layer and Business Logic Layer, which we will describe in this section. According to this paper’s case analysis and experiment results, we believe our system could fully support the most consequential function that a quorum-based fake news prevention system requires: providing objective trust scores to content. Next, we will introduce the implementation detail and main system flow of two critical parts: the Business Logic Layer and Data Persistent Layer in our quorum-based fake news prevention system.



**Fig. 1** One of the content creators (CCs) and users. Two layers: Business Logic Layer and Data Persistent Layer that contain their components

#### 4.1.1 Business Logic Layer: Client applications

As the name implies, the business logic layer (BLL) includes the logic for fake news verification in this layer. In this section, we will describe all of the components in the BLL.

##### *Content Creator (CC)*

CC is responsible for creating news content, broadcasting the content to AA and user nodes, obtaining the Schnorr signature, and sending a request to the blockchain network to write the content hash and collective signature to Hyperledger Fabric. Besides, **CC must pay the rewards** that our system incentive appraisers as a fee for asking appraising its content. The approach to calculating rewards and punishments will be introduced in Section 4.

##### *Appraising Actors (AA)*

In our system, the quorum is comprised of appraising nodes with two Master AA nodes and four Worker AA nodes. In the real world, appraising behaviors should be executed by humans. However, the primary purpose of this paper is to build a prototype to prove a quorum-based fake news prevention system. Thus, we simulate human behaviors by assigning different appraising results to AAs. Besides, AAs verify the news' content rather than sources of information because even reputation sources still possibly produce false information.

Master AA's responsibilities are making asynchronous RPC calls to worker AA nodes when CC sends content. Then sends appraising results back to the CC after collecting all the worker AAs' opinions. On the other hand, Worker AA receives a CC's hash content from the Master AA, reviews content, and

sends back their views to the Master AA using their digital signatures. In other words, our system simulates behaviors in which human appraisers review content and transmit their viewpoints to the Master AA.

As shown in Fig. 1, there is one primary Master AA and one backup master AA in the BLL. A Master AA that is executed first will create its znode to put its IP address and port number in the znode. Then, it will set itself as the primary Master AA node in the ZooKeeper. Like the solid arrow line from the primary Master node to the ZooKeeper in Fig. 1. Afterward, worker AAs and the CC could get the primary Master AA node’s IP address and port number to start communicating by sending an inquiry to the ZooKeeper. The double-arrow lines between the primary Master AA and worker AAs in Fig. 1 represent that the worker AAs already know the primary Master AA’s IP address and port number. Hence, the primary Master AA and worker AAs could begin to send RPC calls to each other.

Furthermore, we let the backup Master AA send an RPC call to verify whether the primary Master AA is still alive or not every 50 ms. After waiting longer than the timeout time, 100 ms, if the backup master AA cannot receive the response from the primary Master AA, the backup Master AA will consider the primary Master AA is already crashed. Then the backup Master AA will take over the responsibility of dispatching signing tasks to worker AAs. The double-arrow dot lines in Fig. 1 between the ZooKeeper, primary Master AA, and backup Master AA describe how the backup Master AA set itself to the primary Master AA. Whenever any one of the Master AAs is executed when the other master AA already exists will be our system’s backup Master AA. On the other hand, the backup Master AA will only build its znode under the same path as the primary Master AA rather than setting it as the primary Master AA in the ZooKeeper. After getting the primary Master AA’s IP address and port number, the backup Master AA will regularly check whether the primary Master AA is alive or not. Once the primary Master AA fails, the backup master AA will set it as the primary Master AA.

### *User*

Receives the content from CC and uses the hash value of the content as a querying parameter to retrieve the digital signature stored in the blockchain. After obtaining the signature concatenation string from the blockchain, the user node can know which Worker AAs agree or disagree with the content. Specifically, users could realize each appraiser’s confidence, credit points, and trust scores in their decision regarding the authenticity of the content from the signature concatenating string. In Section 4.2, we will explain confidence, credit points, and trust scores more thoroughly.

Our system comprises all the CCs, AAs, blockchain, and user nodes. Each CC, AA, and blockchain node is on different servers but in the same private network. The experiment results in Section 5 illustrate that our design could prevent crash failure and possess scalability for multiple appraisers. Moreover,

by providing the proof of stake incentive mechanism we proposed in Section 4.2 of this paper, we could raise the motivations of CC and AAs to contribute their best in conducting their tasks with benign behaviors.

#### 4.1.2 Data Persistent Layer: Hyperledger Fabric

We named it a Data Persistent Layer (DPL) because it receives data and provides functionalities to record data on the Hyperledger Fabric network. Also, DPL provides APIs for users to query specific signatures via the content’s hash. The DPL’s APIs that are implemented by chaincode will be introduced in Section 4.4.

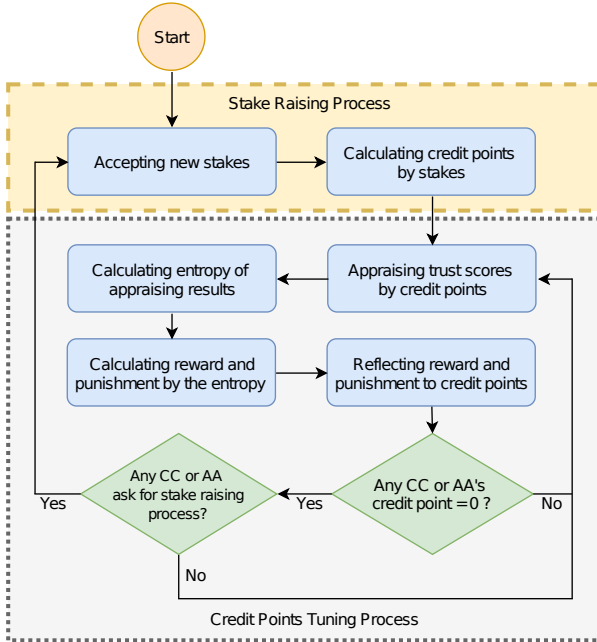
### 4.2 Proof of Stake Entropy-Based Incentive Mechanism

Based on the concept of proof of stake [17], any entity who desires to become a content creator (CC) or appraiser actor (AA) in our fake news prevention system must hold a certain amount of digital tokens as their stakes. At present, we did not integrate any specific digital token in our system because our priority goal in this paper is to build a pilot system to apply an entropy-based incentive mechanism to a quorum-based fake news prevention system. However, any digital token, like bitcoin or ETH, could be integrated into our quorum-based fake news prevention system for paying rewards or punishments for each appraising outcome.

We define *credit points* to represent the reputation of CCs and AAs in our system. Anyone who desires to be a CC or an AA must pay a certain amount of digital token to get their initial credit points. According to the core spirit of proof of stake, credit points could motivate each CC or AA not to be a malicious node because the more they spend, the higher their credit points. Moreover, we design a reward and punishment process to augment the willingness of CC and AA to contribute to the quorum-based fake news prevention system with ethical behaviors. Fig. 2 shows the main flow of calculating and adjusting credit points of CCs and AAs. Besides, we also define the *trust score* that is calculated by credit points of CCs and AAs to indicate how many percentages a piece of news is authentic or fake. Now, we will introduce the proof of stake entropy-based incentive mechanism that we propose in Fig. 2.

#### 4.2.1 Stake Raising Process

Stake raising process is that our system sets a period to let CCs or AAs increase their credit points, calculated by stakes, via paying digital tokens by themselves. Another case to initiate the stake raising process is CCs or AAs could request a new round of stake raising processes whenever their credit points are equal to zero. In our system, we treat CCs as one type of appraisers. However, CCs only could support their content because they should be fully confident that the content they generate is one hundred percent authentic. Although CC may also be a malicious node and send fake content to AAs and users,



**Fig. 2** Main flow of the proof of stake entropy-based incentive mechanism

our proof of stake entropy-based incentive mechanism could combat malicious CCs through severe punishments for dishonest behavior in our system.

Next, we define the *credit point of appraisers (CPoA)* in (1), where  $n$  is the total number of entities that desire to increase credit points,  $i$  is the number of an individual entity,  $1 \leq i \leq n$ , and  $S_i$  is the digital tokens that an  $entity_i$  would like spend to be an appraiser. Moreover, the stake an entity could spend will be limited by an upper and lower bound that our system sets.

$$CPoA_i = \frac{S_i}{\sum_{i=1}^n S_i} \quad (1)$$

**Example 1:** Assume that there are five entities, and  $entity_1$  would like to be a CC in our system as well as  $entity_2$  to  $entity_5$  plan to join our system to be AAs by holding stakes that our system recognizes. Their stakes are 5000, 1000, 2000, 10000, and 3000 relatively. Therefore, we could calculate their initial *CPoA* are 0.24, 0.05, 0.10, 0.48, and 0.14, respectively, where we round results of (1) to two decimal digits.

#### 4.2.2 Credit Points Tuning Process

Our system utilizes *trust scores* that are calculated by credit points of CCs and AAs to indicate whether a piece of news is fake or not to users. The credit

point of each CC or AA will be increased or diminished based on the outcomes of their jobs: CCs will get rewards if they produce authentic content based on evaluations from AAs. Similarly, AAs will also receive rewards if the side, fake or authentic, they support has a more significant trust score than the other side. Having introduced an overview of the credit point tuning process, we will now discuss each step that the credit point tuning process contains.

### *Appraising trust scores by credit points*

We calculate two different trust scores in our system. First, we define the **Score of authentic (SoA)** in (2), which means how much possibility that a content is real, where  $c$  means the content that is currently appraised by AAs,  $a_i$  is an individual appraiser including the CC,  $Conf_i$ ,  $0 < Conf_i \leq 1$ , represents how much confidence the appraiser has in its judgement. Nevertheless, each appraiser **only could put their  $Conf_i$  on supporting either authentic or fake rather than endorsing both sides**. Besides, although being one type of appraiser, the CC could only give 1.0 to its  $Conf$  because, as a content creator, the CC should trust its content one hundred percent. Moreover,  $A_{app}$  is the group of appraisers who approve specific information, and  $n$  is the total number of appraisers in  $A_{app}$ . If one appraiser considers a piece of news is authentic, the appraiser should approve it, or appraisers could reject content as long as they believe they receive false information.

$$SoA(c) = \sum_{i=1}^n CPoA_i \times Conf_i, \forall a_i \in A_{app} \quad (2)$$

The second type of trust score we define, on the other hand, is the **Score of Fake (SoF)** in (3).  $SoF$  represents how much possibility that a content is fake, where  $a_j$  means an individual appraiser, including the CC as well,  $A_{rej}$  is the group of appraisers who reject specific information, and  $n$  is the total number of appraisers in  $A_{rej}$ . Besides, like  $SoA$  of (2), the  $c$  in (3) represents a particular content being appraised.

$$SoF(c) = \sum_{j=1}^n CPoA_j \times Conf_j, \forall a_j \in A_{rej} \quad (3)$$

**Example 2:** The credit point of the CC is 0.24, and the other four appraisers' credit points are 0.05, 0.10, 0.48, and 0.14, according to Example 1. A case in point is if the CC produces a content ( $c_1$ ) and then sends it to the four appraisers we mentioned in Example 1. Suppose that  $a_1$  and  $a_3$  approve  $c_1$  with  $Conf_{a_1} = 0.7$  and  $Conf_{a_3} = 0.8$  respectively. On the contrary,  $a_2$  and  $a_4$  reject  $c_1$  on the basis of  $Conf_{a_2} = 0.8$  and  $Conf_{a_4} = 0.7$ . We could obtain the **SoA** and **SoF** of  $c_1$  that are rounded to the second decimal place based

on (2) and (3):

$$\begin{aligned} SoA(c_1) &= 0.24 \times 1.0 + 0.05 \times 0.7 + 0.48 \times 0.8 = 0.66 \\ SoF(c_1) &= 0.10 \times 0.8 + 0.14 \times 0.7 = 0.18 \end{aligned}$$

When users query trust scores from the blockchain, they will realize that the  $c_1$  has been appraised on 0.66 real and 0.18 fake instead, like Example 2. To sum up, the appraising result:  $SoA$  and  $SoF$  play a pivotal role in calculating rewards or punishments to adjust the credit points of CCs and AAs in our system. What follows is an account of how our system tunes credit points by appraising results.

### ***Calculating entropy of appraising results***

Nowadays, Shannon entropy [18], which is defined in (4) [41], is one of the most widely used methodologies to measure disorder [42].

$$Entropy(S) = -\sum_{i=1}^k p_i \log_2 p_i \quad (4)$$

In (4), we define  $S$  as a set that contains the appraising results of all appraisers. Compared to our previous work [16], we modified  $k = 3$  because there will be three classes of appraising results: ***Authentic***, ***Fake***, and ***Neutral*** in our system. Moreover,  $p_i$  means the fraction of appraising results in class  $i$ . In particular, we define  $p_1$  as the proportion of appraising results that are equal to ***Authentic*** in  $S$  while  $p_2$  as the proportion of appraising results that are equal to ***Fake*** in  $S$ . Besides, we define  $p_3$  as the proportion of appraisers who put their opinion on the ***Neutral***. In sum, we could calculate the  $p_1$ ,  $p_2$ , and  $p_3$  in (5). Note that we already mentioned the definition of  $n$  in (1) as well as  $A_{app}$  and  $A_{rej}$  in (2) and (3), respectively.

$$\begin{aligned} p_1 &= \frac{\sum_{i=1}^k Conf_i}{n}, \forall a_i \in A_{app} \\ p_2 &= \frac{\sum_{i=1}^k Conf_i}{n}, \forall a_i \in A_{rej} \\ p_3 &= 1 - p_1 - p_2 \end{aligned} \quad (5)$$

Consequently,  $Entropy(S)$  will equal zero if all appraisers in our system have the same opinion to consider whether certain content is real or fake. In this case,  $Entropy(S) = 0$  means either  $p_1 = 1$  or  $p_2 = 1$  with 100% *confidence*. On the other hand,  $Entropy(S) \approx \log_2(3)$  if the  $p_1$ ,  $p_2$ , and  $p_3$  are extremely close to each other, which means the viewpoints on the authenticity of a piece of content are thoroughly divergent.

**Example 3:** According to Example 2, we know that CC,  $a_1$ , and  $a_3$  approve the  $c_1$ . Conversely, the  $a_2$  and  $a_4$  reject the  $c_1$ . Based on (5), we could figure out the  $p_1 = 0.5$ ,  $p_2 = 0.3$  and  $p_3 = 0.2$ . Hence, the  $Entropy(S) = 1.49$ . Note that we round  $Entropy(S)$  to the second decimal place.

### *Calculating reward and punishment using entropy*

In our system, we compare the  $SoA$  and  $SoF$ , and then if  $SoA > SoF$ , our system will reward appraisers who approve a particular content ( $c_1$ ) and punish other appraisers that reject  $c_1$  and vice versa. Furthermore, the CC would be punished if its content has been appraised as false information. Namely, the CC cannot get back its collateral if it provides fake content. We will discuss this case more in Section 4.5.

Our calculation approach of reward and punishment is inspired by Ethereum [43][44]. We define how we calculate our **Basic Reward (BR)** and **Basic Punishment (BP)** in (6). In this section, for a straightforward understanding of calculating reward and punishment via entropy in our system, we set **Basic Reward Rate (BRRate)** to 0.1% and **Basic Punishment Rate (BPRate)** to 10% of each content as an example. However, we provide a discussion of a strategy to tune the value of  $BR$  and  $BP$  for keeping *Safety* and *Liveness* in our system. Besides, the  $BPRate$  is higher than  $BRRate$  because we desire to raise the cost of producing malicious behaviors. However, Section 5.3 provides a formula to suggest the optimal values for configuring the  $BR$  and **Basic Punishment (BP)**. Further, our design offers appraisers receive more rewards and punishment if the  $Entropy(S)$  is smaller because **lower  $Entropy(S)$**  means the appraisers' opinions are **highly consistent**. In summary, we use  $Entropy(S)$  to design our entropy-based incentive mechanism to reward the presumptively correct appraisers and punish the wrong ones. The appraisers who are presumptively correct are the ones who vote for the winning standpoint of higher scores between  $SoA$  and  $SoF$ .

$$\begin{aligned} BR &= total\ stakes \times BRRate \\ BP &= total\ stakes \times BPRate \end{aligned} \tag{6}$$

In light of this thought, we define our **Reward of Content (RoC)** and **Punishment of Content (PoC)** in (7), where **Extra reward (ER)** is a number of stakes that CC pays for raising rewards to encourage AA willing to verify its content. Besides, the definition of  $p_1$  and  $p_2$  are the same as their meaning in (5). As a result, based on our incentive mechanism, if  $SoA > SoF$ , appraisers who vote to agree on the content will receive the  $RoC$ , whereas the  $PoC$  will locate to appraisers who approve contents when  $SoF > SoA$ . An extreme case worth mentioning is that when the  $SoA(c_1) = SoF(c_1)$ , users could still get the trust scores, but appraisers will not receive any rewards or punishments. Although we set the  $BR$  and  $BP$  as a fixed ratio of total stakes, the  $BR$  and  $BP$  could be calculated dynamically based on the ratio

based on our optimization strategy in Section 4.3, like Ethereum, to maintain reasonable **BR** and **BP**. Note that  $k$  represents the number of categories that our system has. As we mentioned in (4),  $k$  would equal 3.

$$\begin{aligned} RoC &= (\log_2 k - Entropy(S)) \times (BR + ER) \times p_1 \\ PoC &= (\log_2 k - Entropy(S)) \times BP \times p_2 \end{aligned} \quad (7)$$

**Example 4:** Based on Example 1, we know the total stake is 21,000. Thus, the **BR** would be  $21,000 \times 0.1\%$ , and the **BP** would be  $21,000 \times 10\%$ . Besides, we also know the  $Entropy(S) = 1.49$  according to Example 3. Then, here we set the **ER** to 100 in this example. Hence, we could get the **RoC** = 5.75 and **PoC** = 59.83 that are rounded from the result of (7) to the second decimal place. Also for the  $c_1$  in Example 2, the  $SoA(c_1) > SoF(c_1)$ . Therefore, our system will reward appraisers who approve  $c_1$  and punish appraisers who reject  $c_1$ .

### ***Reflecting reward and punishment to credit points***

After getting the rewards and punishment of certain content, our system will add or reduce credit points of appraisers based on how much percentage they contributed to the  $SoA$  or  $SoF$ . Hence, we define the **Ratio of SoA (RSoA)** and **Ratio of SoF (RSoF)** in (8). Note the **RSoA** and **RSoF** are rounded to the second decimal place.

$$\begin{aligned} RSoA_i &= \frac{CPoA_i \times Conf_i}{SoA}, \forall a_i \in A_{app} \\ RSoF_j &= \frac{CPoF_j \times Conf_j}{SoF}, \forall a_j \in A_{rej} \end{aligned} \quad (8)$$

For instance, in Example 2, we know that the  $SoA(c_1) = 0.66$  and the  $CPoA_1 \times Conf_1 = 0.24$ . As a result, the  $RSoA_1$  would be 0.36 (rounded to the second decimal). Following (8), we could calculate all **RSoA** and **RSoF** for all appraisers. Thus, for  $c_1$  in Example 4, we will reward stakes of 2.09, 0.31, and 3.35 ( $RoC \times RSoA_i$ ) to  $CC$ ,  $a_1$ , and  $a_3$ , respectively, but slash 26.59 and 32.57 ( $PoC \times RSoF_j$ ) from the stakes of  $a_2$  and  $a_4$  relatively as punishments. Besides, we rounded ( $RoC \times RSoA_i$ ) and ( $PoC \times RSoF_j$ ) to the second decimal place. Finally, we recalculate credit points according to modified stakes after each round of appraisers. The updated credit point in this example would be:  $CC = 0.24$ ,  $a_1 = 0.05$ ,  $a_2 = 0.09$ ,  $a_3 = 0.48$  and  $a_4 = 0.14$ .

### ***Inspecting whether any credit point of CC or AA is equal to zero***

After recalculating credit points based on updated stakes after each round of appeasement, we will inspect whether an appraiser has zero credit points.

Then, the smart contract of our system will notify appraisers whose credit points are equal to zero. Otherwise, the status of our system will return to accept another round of appraising.

***Checking whether any CC or AA asks for a new stake raising process***

Any entity that desires to be an appraiser must have a non-zero credit point. So, appraisers whose credit point equals zero could ask for another stake raising process to increase their stakes to earn extra credit points, or they would not be allowed to be appraisers.

Until now, we have already introduced the entropy-based incentive mechanism proposed in this paper. Furthermore, when each AA is ready to send its appraising result back to the smart contract, it must adapt ***AES***, with a 128-bits secret key, to encrypt its voting opinion and necessary data. Those essential data include two critical values that will be used to verify its *Schnorr signature* by our smart contract.

Further, our system will ask each appraiser to use its ***Schnorr signature*** via a 128-bits secret key of ***AES*** to encrypt its decision. Then, appraisers must send their encrypted appraising result directly to the blockchain to affirm that the individual appraiser will conduct its appraising task independently. When the CC finishes adding the news to the blockchain, our system will decrypt the appraising results. We will describe how we keep appraising results secret between CC, AA, and Smart Contract(chain code) in Section 4.6.

### **4.3 Implementation of BLL: CC, AA and User**

In the following paragraphs, we present our implementation of Client Applications: CC, AA, and User nodes.

#### **4.3.1 CC—Create content**

In a real-world scenario, the content should be an article, an image, or a video piece. However, any article, video, or image is encoded into a binary string to be transferred to another endpoint. Therefore, we generated content with fixed characters composed of alphabets. Another advantage of this approach is that it is easier to control the content size, which is an essential parameter in the system throughput testing. To get the accurate size of the content we generated, we transform the string into a bytes array, and the size of the bytes array is the actual size of the content. A unique id and hash value are generated for each piece of content and are sent out along with the content to the primary Master AA node.

### 4.3.2 CC–Initiating the signing process

The CC plays the role of initiating the signing process and storing signatures from AAs on the blockchain. The CC exerts RPC calls (Thrift) to send content(message) to AAs and receives signatures from AAs. As we mentioned before, our internal architecture of AA nodes’ cluster comprises two Master AA nodes; one is the primary Master AA, and the other is Backup Master AA. We also have four worker AA nodes (see Fig. 1). Besides, we adopt the ZooKeeper, a centralized node for maintaining configuration information, coordinating services, and providing distributed synchronization [45], to achieve the mechanism of preventing the crash failure of AA nodes.

As shown in Fig. 1, the CC will contact the ZooKeeper to get the available primary Master AA node’s IP address and port. After getting the primary Master AA nodes’ connection information, the CC will send an RPC call to trigger the primary Master AA to issue asynchronous RPC calls to worker AA nodes responsible for signing tasks. Worker AA nodes will sign the content passed from Master AA nodes and return the result upon receiving the RPC call. On top of that, to maintain a healthy connection between Worker AA nodes to the primary Master AA node, we let Worker AA nodes send a heartbeat to ZooKeeper to get the address of an existing primary Master AA node. The double arrow connecting worker AAs to the primary Master AA represents that the Master AA sends RPC calls to assign a signing task to worker AAs, and the worker AAs reply to the Master AA after completing sign tasks.

When a Master AA receives a request from CC, the Master AA will use the current date and time as a random seed and leverage the seed as an input parameter to the built-in Java random function to select appraisers in candidates that users specify. For example, the random seed will equal 20220101230000 if the Master AA receives the CC request at **11:00 pm on January 1st, 2022**. Then if users specify three appraisers in ten candidates, our system will randomly select appraisers by different random seeds each time. This selection process of the Worker AA can ensure that candidates who conduct appraising tasks are randomly selected each time, thereby reducing the possibility of success that attackers know Worker AA in advance and control them one by one. As we mentioned in the paragraph *Appraising Actors (AA)* in Section 4.4.1, if the Master AA crashes, the backup Master AA will take over the Master AA’s job to keep our system running.

### 4.3.3 AA–Executing the signing process

With the help of the *Schnorr signature*, Worker AAs sign the content and return a concatenation string composed of a flag representing whether a Worker AA agrees on the content or not, another flag whether a Worker AA believes the content is duplicate or not, and the signature data.

#### 4.3.4 CC–Storing content to Hyperledger Fabric

As mentioned, we adopt the *Chaincode* to design two smart contracts to achieve the DPL’s functionalities. These two smart contracts are *addNews()* and *queryNews()*. As shown in Fig. 1, the CC will call *addNews()* function to store the id, content, and the concatenation string after the CC gets the encrypted data from worker AAs. The concatenation string is appraising results (the first number of the string, 1 means an AA agrees to the content, and 0 represents the AA disagrees instead) and signature data.

#### 4.3.5 User–querying content from Hyperledger Fabric

Similarly, the other smart contract: *queryNews()*, is used by the user node to query a specific message’s signing result. In Fig. 1, we describe the flow of how the user node gets the signature result. The user node calls *queryNews()* to inquire about the blockchain by sending a message’s hash value. Then the user node could retrieve a signature concatenation string if the message exists in the blockchain. Afterward, the user will parse the signature concatenation string and examine which worker AAs approved the content.

### 4.4 Implementation of DPL: HyperLedger Fabric node

We mainly refer to the configuration suggestions from the HyperLedger Fabric [19] to configure our blockchain network. This section will introduce the implementation details of the Hyperledger Fabric network used in this work in three main points.

Firstly, we deployed the Hyperledger Fabric network on only one server. Our Hyperledger Fabric contains two Peer Organizations and an Orderer Organization. Besides, each organization has its own Certificate Authority. Moreover, Peer Organizations have two users – Admin and User. On the other hand, Orderer Organization has a single Orderer and only one user: Admin.

Secondly, we use Fabric-CA Server in order to generate the crypto materials. The network used in this work spawns three Fabric-CA Server Docker containers, one for each peer organization and one for the orderer organization. After starting the Fabric-CA Server containers, the network enrolls a CA Admin using Fabric-CA Client and then registers and enrolls every entity using that CA Admin. Fabric-CA Server will generate self-signed certificates and also a secret key. Once the crypto materials are generated, they are moved to the entities’ appropriate directory structure for use. Finally, this work also uses an application channel to communicate between peer and orderer organizations.

Lastly, concerning the Chaincode design, the transaction that needs to be recorded in the distributed ledger includes the content created by the Content Creator and the signature collected from the Appraisal Actors by the Content Creator. Although we use Java to create objects to support our application’s smart contract, we believe our main workflow in Fig. 1 could be implemented in any other programming language supported by the Hyperledger Fabric. [19].

## 4.5 Adapt *AES* algorithm to make AAs vote independently

Keeping each AA's opinion secret in a certain time period is essential in our system because an AA will rely on others' appraising results to make decisions if one AA could realize others' thoughts. Then, the appraising results will not be independent and objective. Like Fig. 3, although the existing component that *Hyperledger Fabric* has, *Channel* [19], could restrict participants to only access data in its belonging channel, all nodes in the same channel still could see each node's proposed opinion. It is negative for corporations to build applications that must keep transactions secret in a certain time period via *Hyperledger Fabric*. Accordingly, we design a process to support keeping each AA's viewpoint not being known by others until the end of an appraising period as the following procedure:

### ***Step 01: CC set extra reward and sent content to chaincode and the Master AA***

Except for the *BR* that the system offers, *CC* could put more stakes as an *ER* to encourage more AAs to express their opinion on its content. When a *CC* wants to send a piece of content to our system, it could set the *ER* and send it to the Master AA and the *Smart Contract* in the blockchain. When the *Smart Contract* receives this content, they will record it on the blockchain. Moreover, the *CC* must put the same amount equal to *BP* as collateral. If one content is appraised as fake, the *CC* will lose its collateral.

### ***Step 02: The Master AA notifies all AAs with a time limitation***

When the *CC* sends the content to the Master AA, the Master AA will notify all appraisers and set a time limitation, which we set to **24** hours. Before the time limitation, AAs could decide whether to join the appraising task via putting an amount stake as collateral. Our system will reimburse if their opinion finally lands in the majority.

### ***Step 03: AAs put their stakes as collateral and start voting***

Once a worker AA decides to join an appraising task, it needs to put the same amount of the *BP* as collateral. Then, a Worker AA must generate a 128 bits secret key and use the *AES* to encrypt its appraising result. After that, it sends the appraising result and *Schnorr signature* data via *AES* back to the Master AA. Then, when *CC* receives the encrypted data from the Master AA, the *CC* will send the encrypted data to the *Smart Contract*.

### ***Step 04: Smart Contract asks for secret keys to decrypt appraising results***

After the end of the time limitation (24 hours in this work), the Master AA will ask all AAs who appraise current content to provide their secret keys to the *Smart Contract*. Then the *Smart Contract* will use secret keys that

AAs provide to decrypt the appraising results to calculate the reward and punishment we already described before.

These four steps above are our primary process of independently affirming each AA appraise content. Now, let us focus on two special scenarios in this process. **Firstly**, if no AA decides to evaluate the content within the time limit, *Smart Contract* will notify CC whether to increase the Rewards. If the CC decides to increase the rewards, the Master AA will ask all AAs if they are willing to evaluate the same content with the new amount of *ER* and reset the time limitation. If the CC is unwilling to increase the rewards, the system will note that the content has expired in the Blockchain and inform the CC that if they want to propose the same content, they must wait for **24** hours.

In addition, AA will review whether the content is duplicated or not in its evaluation opinion to ensure that CC will not obtain rewards by repeatedly providing the same content. If the AA thinks that the content is similar to others tremendously, the content will be considered duplicated item. CC will not get any reward and will lose its collateral to pay *BP* to be punished for its dishonest behavior: sending duplicate content.

## 5 Cases Analysis and Experiments

Our main contributions are to design an entropy-based incentive mechanism and conduct various experiments to verify the feasibility of a quorum-based fake news prevention system. This section first analyzes our entropy-based incentive mechanism to demonstrate we could effectively resist the 51% and Sybil attacks. Then we discuss the experiment results of crash failure prevention and scalability of our system. Our analysis and experiment results assure that our work’s incentive mechanism and system architecture are robust and feasible. In other words, this paper provides the essential basis for strategic planning for further development in quorum-based fake news prevention systems.

### 5.1 Case Analysis of Proposed Incentive Mechanism

It is essential for a quorum-based to possess an effective incentive mechanism to give impetus to appraisers to complete content evaluations reliably. Our proof of stake entropy-based incentive mechanism raises punishment to dishonest CCs or AAs to deter malicious behaviors, which represents appraisers who offer an opposite opinion of the authenticity of news in our system. Table I presents all possible scenarios of our incentive mechanism in Fig. 2. Compared to our previous workshop paper in Deviance 2021 [16], there will be only four cases in our system because CCs only could vote their content to *REAL*. We define two primary types of cases in our system. The first one is **Regular Cases**, which means the *majority of AAs (Majority)*, appraisers who vote for the winning standpoint of higher scores between SoA and SoF, identify the authenticity of content correctly, such as the cases that we marked in gray

in Table I. In contrast, the *minority of AAs (Minority)* means the appraisers that hold a different opinion than the *Majority*. The second type is the cases with no background color representing the ***Exotic Cases***, which indicate the *Majority* failing to judge the authenticity of a piece of news.

Note that the CC does not be counted in the *Majority*, although the CC is one type of appraiser in our system. Furthermore, the *N/A* in Table I states that all appraisers have the same viewpoints on appraising. Next, we will dive into discussions of ***Regular Cases*** and ***Exotic Cases***. Note that although we add the *confidence* factor into the process of our incentive mechanism, the *confidence* factor will not affect the original analysis in this section the same our previous work [16].

**Table 1** Reward and Punishment in different scenarios

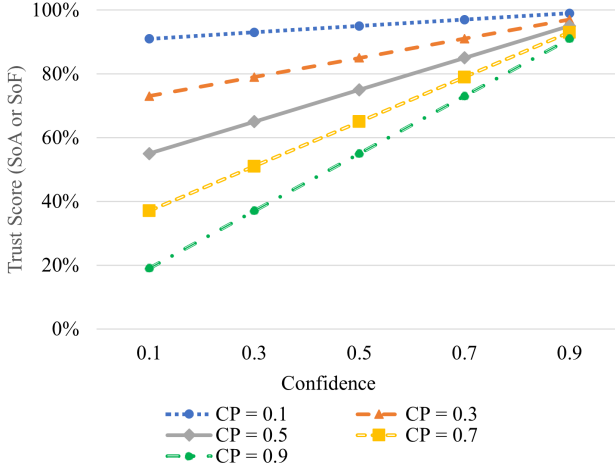
Case No.	Viewpoints of Content			Reward (+) or Punish (-)		
	<i>CC</i>	<i>Majority</i>	<i>Minority</i>	<i>CC</i>	<i>Majority</i>	<i>Minority</i>
1	REAL	REAL	N/A	+	+	N/A
2	REAL	REAL	FAKE	+	+	-
3	REAL	FAKE	N/A	-	+	N/A
4	REAL	FAKE	REAL	-	+	-

### 5.1.1 Regular Cases

As all the cases that possess a gray background in Table I indicate, our incentive mechanism could motivate appraisers to correctly specify the authenticity of information because our system punishes dishonest appraising and rewards reliable entities. On the other hand, each appraiser would enhance their ability to judge the authenticity of news because of the high cost of judging news falsely.

### 5.1.2 Exotic Cases

Exotic cases show a critical point: only malicious nodes could be rewarded, whereas other benign appraisers are punished. The root cause of exotic cases is that several AAs unite to commit deception, but these cases will hardly happen in our system. Our incentive mechanism facilitates that it is almost impossible for a hacker to control more than 51% of stakeholders in our system, much less to collect all stakes. It is also difficult to collect more than 51% stakes in our system. Also, our system could resist the Sybil attack because it would be a considerable cost if one entity desires to impact trust scores in our system significantly. However, one possibility of our system, like all other incentive-based blockchain systems that have just been initiated, is that if the amount of stake that we hold is tiny, our system would be vulnerable to 51% attacks. To sum up, according to the analysis result, our incentive mechanism indeed owns the ability to motivate the CCs to provide dependable news and induce AAs to contrive to judge fake news honestly.



**Fig. 3** Sensitivity analysis of confidence and credit points of an AA to SoA/SoF

## 5.2 Sensitivity analysis of the proof of stake entropy-based incentive mechanism

The most significant outcomes in our incentive mechanism are *trust scores (SoA or SoF)*, *reward (RoC)* and *punishment (PoC)*. In this section, we will conduct two sensitivity analyses to realize variations that variables caused to trust scores, rewards, and punishments.

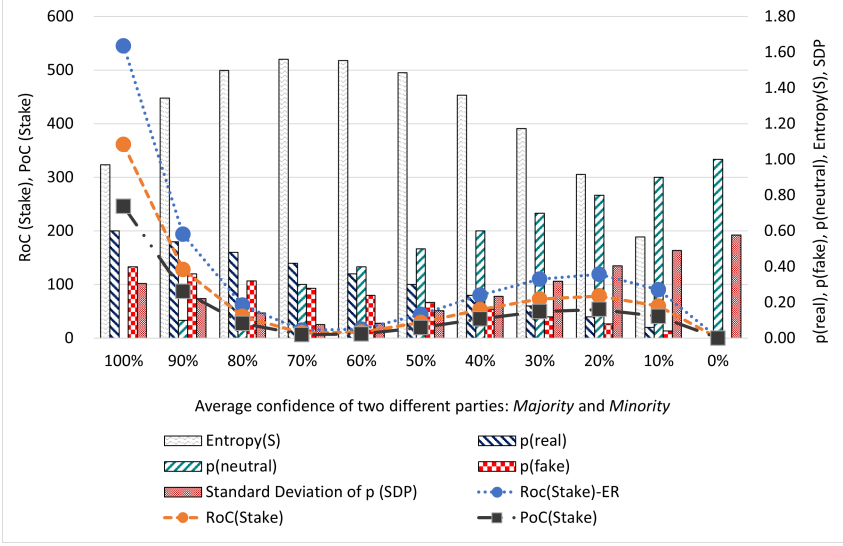
### 5.2.1 From confidence and credit points to trust scores

In Fig. 3, we set an experiment with only one CC and one AA. Therefore, the summation of credit points of CC and AA would be 1.0, and we could calculate *SoA* or *SoF* by (2) or (3) in Fig. 3. based on different pairs of credit points and the confidence of the AA. An obvious pattern is shown in Fig 3. That is, if AA’s credit point is getting higher, the variation of its confidence will affect the *SoA/SoF* more. This behavior is reasonable because any entity with tremendous credit points would have more power to influence the trust score based on the calculating approach of trust scores.

### 5.2.2 From entropy to rewards and punishments

*a) Different sides of viewpoints of appraisers have the same confidence:*

We use an example:  $p(\text{real}) = 60\%$  and  $p(\text{fake}) = 40\%$  with 100% confidence that all appraisers. The definitions of  $p(\text{real})$  and  $p(\text{fake})$  here and in the following paragraphs are the same as  $p(1)$  and  $p(2)$ , respectively, in (5). This case represents that the percentages of all appraisers who believe the content is real are higher than the percentage of verifiers who give their opinion that the content is fake. In other words, appraisers who believe this content is real



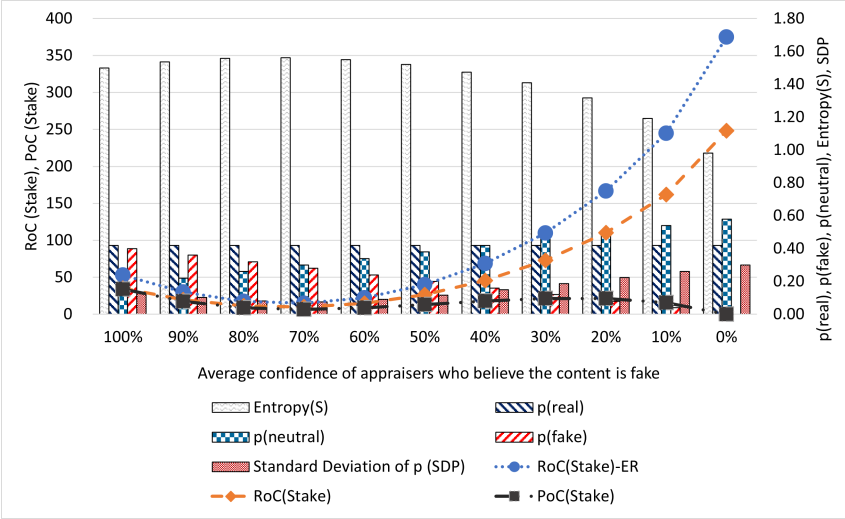
**Fig. 4** Sensitivity analysis of *confidence* to (a)  $p(\text{real})$ ,  $p(\text{neutral})$  and  $p(\text{fake})$ , (b)  $Entropy(S)$  and (c)  $RoC$  and  $PoC$ . Also, the impact of  $ER$  to  $RoC$  and  $PoC$  is shown in this figure as well. In this figure,  $p_1$  is equal to  $p_2$ .

will get rewards. Besides, appraisers have the same confidence in their opinion. That is, whether the content is determined to be true or false, the verifier has 100% confidence. In Fig. 4, we can see that the maximum entropy value appears when the confidence on 70%. When all appraisers have 70% *confidence* in their viewpoints, we could get:

$$\begin{aligned}
 p(\text{real}) &= 0.6 \times 0.7 = 0.42 \\
 p(\text{fake}) &= 0.4 \times 0.7 = 0.28 \\
 p(\text{neutral}) &= 1 - 0.42 - 0.28 = 0.3
 \end{aligned}$$

Therefore, the  $Entropy(S)$  in this case would be 1.56, which is higher than the  $Entropy(S) = 1.00$  when the *confidence* = 100%. This result means that the content is more difficult to be examined as real or fake, so appraisers do not have such as much *confidence* as another content of  $Entropy(S) = 1.00$ . Also, another interesting point from Fig. 1 is when the *confidence* is down to zero of both sides of appraisers, there would be no reward or punishment to all appraisers because there is no majority on top of this case.

On the other hand, as shown in Fig. 4, we could observe four interesting points: **First**, both  $p(\text{real})$  and  $p(\text{fake})$  will decrease when the *confidence* goes down because low *confidence* represents the hesitation of appraisers in the face of some content that is difficult to distinguish between true and false. Based on the hesitation of appraisers, the  $p(\text{neutral})$  will increase with the decrease of *confidence*. **Second**, the standard deviation of  $p(SDP)$  in Fig. 4 is the standard deviation of  $p(\text{real})$ ,  $p(\text{fake})$ , and  $p(\text{neutral})$ . We could observe



**Fig. 5** Sensitivity analysis of *confidence* to (a)  $p(\text{real})$ ,  $p(\text{neutral})$  and  $p(\text{fake})$ , (b)  $Entropy(S)$  and (c)  $RoC$  and  $PoC$ . Also, the impact of  $ER$  to  $RoC$  and  $PoC$  is shown in this figure as well. In this figure,  $p_1$  is **not** equal to  $p_2$ .

that when the *confidence* reduces from 100%, the  $SDP$  will decrease until the *confidence* reaches 70%. After that, the  $SDP$  goes up from *confidence* lower than 70% until *confidence* is equal to 20%. This phenomenon is because the reducing *confidence* leads to divergent appraising opinions to make  $SDP$  lower. However,  $Entropy(S)$  will get lower when the *confidence* reaches 70% because viewpoints of appraisers will get more and more converge on neutral. **Third**, it is obvious in (7) that lower confidence also makes  $RoC$  and  $PoC$  lower. Nonetheless, the  $RoC$  and  $PoC$  will get higher when the confidence is lower than 70%, and that is because  $Entropy(S)$  is also getting lower after its peak. **Finally**, in Fig. 4, the  $RoC(\text{Stake})-ER$  means that the  $RoC$  is in a situation where a CC puts  $ER = 500$  to its content. By doing so, the CC could encourage more worker AA to verify its content because the curve of reward:  $RoC(\text{Stake})$  is close to the curve of punishment:  $PoC(\text{Stake})$  if the CC adds  $ER$  to a piece of content.

**b) Different sides of viewpoints of appraisers have the different confidence:**

In Fig. 5, we set  $p(\text{real}) = 60\%$  with 70% *confidence*. Also, the  $p(\text{fake}) = 40\%$  with different confidence in this example. As shown in Fig. 5, with the confidence of appraisers who disagree with the content decrease, the  $Entropy(S)$  will also reduce. Therefore, the  $RoC$  and  $PoC$  will reach the highest when the confidence of appraisers who disagree with a piece of content equals zero.

Based on Fig. 5, there are still four points to which we could pay attention: **To begin with**, because the  $p(\text{fake})$  is less than  $p(\text{real})$ , we could readily see that when the  $p_2$  goes down,  $p(\text{fake})$  will reduce as well, but the  $p(\text{neutral})$

rises. This result is very intuition because some appraisers keep their opinion on being neutral. **Next**, with the reduction of  $p_2$ , the  $Entropy(S)$  goes up until  $p_2$  reaches 70%. After that,  $Entropy(S)$  begins to reduce because appraisers converge their opinion on keeping neutral. **Then**, in Fig. 5, the  $Entropy(S)$  of each interval of the x-axis is higher than the  $Entropy(S)$  on the same location of Fig. 4’s x-axis. This result is because the  $SDP$  in each location of the x-axis is smaller to those  $SDP$  at the same scale of the x-axis. **Finally**, similar to Fig. 4, if the CC puts  $ER$  to its content, the  $RoC(Stake)-ER$  will increase evidently to motivate appraisers willing to verify its content.

***c)The minimum stand deviations between  $p(real)$ ,  $p(fake)$ , and  $p(neutral)$ :***

When the X-axis in Fig. 4 and Fig. 5 is 70%, the *Stand Deviation of  $p(SDP)$*  reaches a minimum value, *Entropy* comes to the maximum value, and *Reward* and *Punishment* fall to their minimum values. This situation does not mean that our system failed to identify news. On the contrary, it is a helpful appraising result because a split pattern of opinion among appraisers is an excellent sign to indicate a piece of news that is hugely difficult to be identified its authenticity. Moreover, a divergent opinion also means this news is highly controversial. Journalists could collect critical information that needs more effort to work on and then get exclusive news if they could verify the authenticity of that news that gets split opinions from our system. Verifying fake news is not a black and white problem because, in most cases, some parts of one content are authentic, and some are fake. Our system does not aim to instruct users on whether a piece of news is precisely authentic or fake. Instead, we offer objective trust scores as a reference for users to decide how much they could trust certain information.

### **5.3 A strategy for tuning the value of $BR$ and $BP$ to keep *Liveness* and *Safety* at the same time**

There are two critical parameters to calculate rewards and punishments in (7). In Section 4.2.2, we simply set  $BP = total\ stake \times 0.1\%$  and  $BR = total\ stake \times 10\%$ . However, for a real-world application, we need an approach to adjust  $BP$  and  $BR$  to reach two primary goals: (1) Raise  $PoC$  to curb dishonest behavior and (2) Reduce  $RoC$  to keep the cost of appraising content low. However, When trying to achieve these two goals, we will encounter a dilemma: we cannot set the punishment too high; otherwise, no CCs or AAs will be willing to take any action in our system. Nonetheless, we cannot let the punishment be too low to prevent malicious behaviors. Thus, this section will propose a reasonable range for tuning  $BR$  and  $BP$  to reach the two primary goals we mentioned above.

We first introduce assumptions in our optimization model of tuning  $BR$  and  $BP$  for further discussion. **First**, we assume each AA has 100% *confidence*

in its appraising opinion. **Second**, the opinion of the *Majority* is correct. If the majority believe a piece of news is authentic, the news could be considered accurate. **Third**, we assume all honest appraisers will try their best to offer a correct opinion. On the other hand, all malicious appraisers will get rewards by cheating as much as possible.

On top of these assumptions, we could define several variables in our BR/BP tuning model. To begin with, the  $PHA_{correct}$  means the proportion of honest AAs with correct opinions. Therefore, we could also get  $PHA_{incorrect} = (1 - PHA_{correct})$ . On the other hand,  $PDA_{correct}$  represents the proportion of dishonest AAs whose opinion is the same as the majority as well as we then could get  $PDA_{incorrect} = (1 - PDA_{correct})$ , which means the proportion of dishonest AAs whose viewpoints are different compared to the *Majority*. Here we summarized these four variables in (9).

$$\begin{aligned} PHA_{incorrect} &= (1 - PHA_{correct}) \\ PDA_{incorrect} &= (1 - PDA_{correct}) \end{aligned} \quad (9)$$

There is a core assumption we made in our system: "honest AAs always try to offer correct viewpoints, as well as dishonest AAs prefer to just offer the same opinion to the majority." Hence, we could assume  $PHA_{correct} = 0.9$  and  $PDA_{incorrect} = 0.9$  for further analysis in our BR/BP tuning model. We could also set  $p(real) = 0.51$  and  $p(fake) = 0.49$  because, in our assumption, AAs should have highly divergent standpoints between two groups of honest and dishonest AAs. Note that the definitions of  $p(real)$  and  $p(fake)$  are the same as those in (5). Moreover, we assume the  $ER = 0$  in BR/BP tuning model. Then, based on (7), we could get four equations to get all types of rewards and punishments that our system has in (10). Note that  $K = 3$  that we already mentioned in (7), and we ignore the constant:  $(\log_2 k - \text{Entropy}(S))$ .

$$\begin{aligned} RoC(HA_{correct}) &= PHA_{correct} \times BRRate \times 0.51 \\ PoC(HA_{incorrect}) &= (1 - PHA_{correct}) \times BPRate \times 0.49 \\ RoC(DA_{correct}) &= PDA_{correct} \times BRRate \times 0.51 \\ PoC(DA_{incorrect}) &= (1 - PDA_{correct}) \times BPRate \times 0.49 \end{aligned} \quad (10)$$

In (10),  $RoC(HA_{correct})$  represents the rewards that the honest AAs whose opinion is the same as the *Majority* gets. Instead,  $RoC(DA_{correct})$  means that the rewards the dishonest AAs who express the same opinion as the *Majority* earn. In our tuning model, we could ignore  $RoC(DA_{correct})$  because it means that some dishonest AAs do not act as malicious nodes toward a certain piece of news. Besides,  $PoC(HA_{incorrect})$  indicates the punishments that are suffered from the honest AAs whose opinion is different from the *Majority*, which means that they make mistakes in an appraising take. Also,  $PoC(DA_{incorrect})$  means the punishment that dishonest AAs will be slashed whenever their opinion does not be the same as the *Majority*.

Furthermore, for one side, if we desire to obtain Liveness, we should maximize  $RoC(HA_{correct})$  and minimize the  $PoC(HA_{incorrect})$ . On another side, to maintain Safety, we must maximize  $PoC(DA_{incorrect})$  to curb malicious behaviors. Based on these goals, we could form two critical inequalities of our BR/BP tuning model (11):

$$\begin{aligned} RoC(HA_{correct}) &\geq PoC(HA_{incorrect}) \\ PoC(DA_{incorrect}) &\geq RoC(HA_{correct}) \end{aligned} \quad (11)$$

Based on (10), and let us put all numbers we set to all the variables in (11), we could transfer (11) to (12)

$$\begin{aligned} 0.51 \times 0.9 \times BRRate &\geq 0.49 \times 0.1 \times BPRate \\ 0.49 \times 0.9 \times BPRate &\geq 0.51 \times 0.9 \times BRRate \end{aligned} \quad (12)$$

To sum up, we could conclude a final inequality (13) as the guideline to tune BR/BP in our system. The (13) indicates that if we desire to keep both Safety and Liveness in our system, the BPRate should be at least 1.04 times the BRRate. For example, if we set the  $BRRate$  to 5%, our model will suggest setting  $BPRate$  to 5.2%.

$$BPRate \geq 1.04 BRRate, BPRate > 0 \quad (13)$$

## 5.4 Analysis for the Safety and Liveness of our system

**Safety** and **Liveness** are two critical factors to a distributed system, like our fake news prevention system. In this section, we will do some analysis to discuss the Safety and Liveness of our work.

### 5.4.1 Safety

In Section 5.1, we discuss some potential threats to our system and provide sufficient analysis to prove our system could resist those threats we already mentioned. Except for the difficulties in controlling more than 50% AAs, our system could check whether the content is duplicate enough to avoid the CC sending duplicate content repeatedly to gain rewards dishonestly. Moreover, collaterals (stakes) that AAs put in whenever they decide to join an appraising task could encourage AAs to deal with their job more honestly.

### 5.4.2 Liveness

In our system, CC could raise their  $ER$  to motivate AAs to verify news to increase our system's liveness. Also, even though there are no AAs to verify

specific content, our system will still let CC resend the content after 24 hours to keep the system from halting on certain news that cannot get any appraising results.

Moreover, compared to our previous work [16], we add the *confidence* factor to equation (7), which calculates the *PoC* and *RoC*. As shown in Fig. 1 and Fig. 2, AAs could raise or reduce their *confidence* based on the difficulties of judging a piece of content. In sum, the *confidence* factor assists our system in keeping liveness because AAs could spend their effort on improving their abilities to verify content to make themselves more confident in judging content to get more rewards. Besides, the *confidence* factor could also help AAs avoid punishment in some content they cannot determine the authenticity.

On the other hand, our system will request AAs to set their *confidence* to at least 60% because we believe AAs only guess arbitrarily without putting their effort into verifying content if our system does not set this rule. Once an AA decides to express its opinion on content, it must stand on one side and express a value of *confidence* greater than 60%. This limitation could assure that AAs will not get rewards without giving any opinion or judging without any reasons for specific content.

## 5.5 Interactions between CC and AAs

Although we added new features into this extension work, we did not re-conduct our experiments because we believe our previous experiments conducted in [16] have already let us understand the basic performance and scalability of our architecture.

Essentially, message size significantly affects the performance of distributed systems, including latency and throughput. In this paper, we design six different sizes of content:  $2^3$  KB (8 KB),  $2^5$  KB (32 KB),  $2^7$  KB (128 KB),  $2^9$  KB (512 KB),  $2^{11}$  KB (2 MB), and  $2^{13}$  KB (8 MB) to observe the effect of different sizes of contents for the five experiments we will discuss in this section. Moreover, considering prevailing use cases of social media, people always send short text messages to others, like a tweet or a post on Facebook. According to users' preferences to make messages as brief as possible when using social media, we set the maximum message size to 8 MB. If people desire to transmit long videos or huge videos on our system, they could put links in their messages to enable others to download their content. Therefore, we believe 8 MB is a suitable upper limitation of our system.

In experiments I to III, we let the CC send messages to the Master AA node for one minute five times to record the throughput and latency. Then we summarized the average throughput and latency in each message size. For experiments I, II, and III, we define our throughput and latency as below.

### *Latency*

The average waiting time is required for the cycle from CC sending a message to AAs until CC receives the response message sent back from AAs.

### *Throughput*

We define throughput as the average number of requests sent by CC that an AA node could process per second. To get more accurate figures of experiment result in a single thread with synchronous requests, we use only one client of CC to send requests one by one when the previous request ends.

Note that we assume all workers agree on the content that our CC sent in experiments I to III with one hundred percent confidence (*Conf*) because the main goals of these experiments focus on measuring how the communications of different nodes affect the throughput and latency in our system. Also, human appraisers in the real world might take a few hours to several days to evaluate content. Moreover, we do not use AES in these experiments because we desire to avoid other factors to affect the performance result of our architecture.

Note that we assume all workers agree on the content that our CC sent, in experiments I to III, with one hundred percent confidence (*Conf*). The main goals of these experiments focus on measuring how the communications of different nodes affect the throughput and latency in our system. Also, human appraisers in the real world might take a few hours to several days to evaluate information. Still, in our experiments, we let worker AAs send back their appraising results once they finish the verification task. Moreover, we do not use *AES* in these experiments because we desire to avoid other factors affecting our architecture’s performance.

#### **5.5.1 Experiment I: Normal Master AA and Worker AA nodes**

In the first experiment, we plan to realize the optimal performance that our system could achieve. As a result, we leave no failures on any AA nodes and only one master AA and worker AA node in this experiment. As shown in Fig. 6, our throughput for 8 KB contents is 602 messages per minute and 63 messages per second for 8 MB contents. As we can see, the throughput drops slightly from 8 KB to 512 KB and steeply falls when the size reaches 2 MB. This phenomenon meets our expectations because CC and AAs need more time to transmit larger messages to each other and process the content. Therefore, the latency increase and the throughput decrease result from the augmentation of message size.

#### **5.5.2 Experiment II: Scalability of Worker AA nodes**

We focus on the Worker AA nodes regarding our system’s scalability because the number of worker AA nodes may be tremendous in the real world. Therefore, it is worth understanding the impact of different numbers of worker AA on our system’s performance to increase the number of worker AAs in our system. Our experiment result of scalability is shown in Fig. 6. We could observe that, in each message size, the latency increases slightly with the augmenting numbers of Worker AA nodes. This increase in latency is because, in our system, Master AA will confirm whether all workers AA have completed their

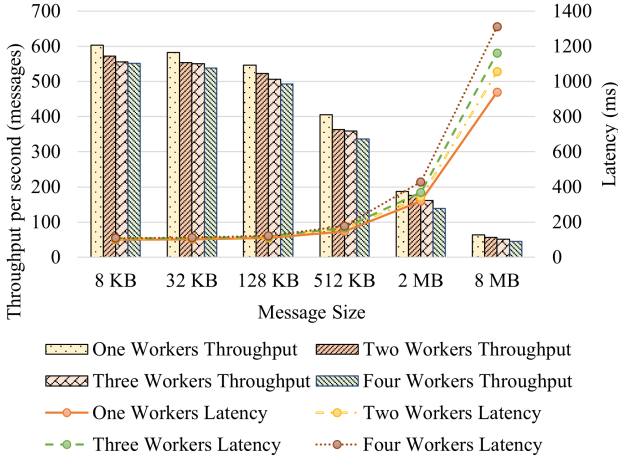


Fig. 6 Throughput and latency of scalability experiments

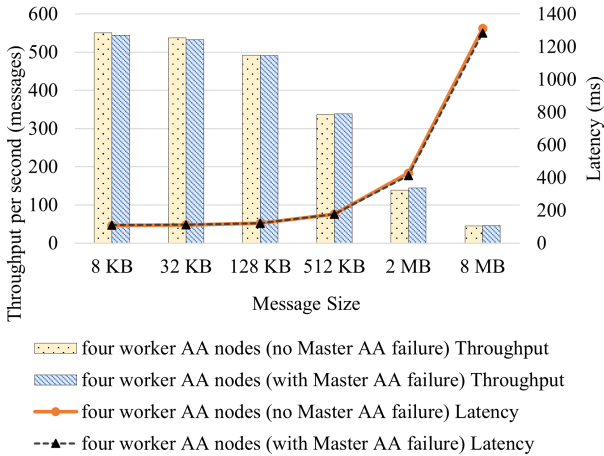


Fig. 7 Throughput and latency when the Master AA fails

work or not at every two ms interval. Therefore, the total latency will be determined by the slowest worker AA. Moreover, Fig. 6 displays that the latency gap is not tiny between different numbers of worker AA nodes in each message size, which means the increase of worker AA nodes will not lead to latency growth. However, further work to scale our system to a hundred or thousand worker AA nodes would be necessary to verify the scalability limitations of worker AA nodes.

A straightforward question for our system is: *how do human appraisers handle a large amount of content generated every second on social media?* Indeed, it is unrealistic to make our system evaluate every news item on social media. However, our goal in scalability is not to verify each news on

social media but to evaluate a specific number of false information highly similar to real news. It is reasonable that our system mainly supports certain kinds of fake news that are difficult to distinguish because general users have essential fake news identification ability and can easily distinguish information that is obviously false news. But when it comes to immensely controversial news, such as political or movie celebrity scandals, or news that requires expertise in healthcare or information technology, the average person cannot tell the truth from a fake. In fact, remarkably disputing news is what our system desire to assist users in verifying. When encountering vastly contentious news that cannot be identified, the user can submit the news to our system for human appraisers identification and finally get trust scores as the basis for judging news' trustworthiness.

In addition, one situation that needs to improve scalability is that the number of false information will increase significantly in a certain period, such as the US presidential election or the outbreak of regional wars. At this time, each working AA can also hire more people to assist in fact-checking because any reporters, bloggers, or people capable of fact-checking can become an appraiser of our system. On the other hand, each appraiser can train its Machine learning (ML) classifier to identify fake news. When ML classifiers reply that some false information is difficult to be judged, it is an appropriate scenario in which human appraisers could show their talent. And using ML classifiers does not mean we do not need the incentive mechanism we designed. Instead, we use *entropy* to calculate the consistency of opinions across appraisers, including humans or AI, to identify the extent of difficulty of judging a piece of news is fake or not. According to the *entropy* value among different appraising opinions, we could determine appropriate rewards and penalties to encourage honest behavior and discourage selfish behaviors.

### 5.5.3 Experiment III: Crash Failure Handling of Master AA nodes

Providing that there are no Master AA nodes, our system could not normally work because the Master AA node is responsible for receiving CC's content and assigning signing tasks to Worker AA nodes. Hence, we offer another critical contribution to designing a fault tolerance mechanism to prevent Master AA node failures. Fig. 7 states a comparison of latency and throughput of our system without Master AA failures and when encountering Master AA failures. This experiment executes the Primary Master AA and lets all four Worker AA nodes connect to the primary Master AA nodes. Then CC sends messages to the primary Master AA nodes, and we stop the Master AA nodes at the 10, 20, 30, 40, and 50 seconds within one minute that the CC sends messages to the Master AA nodes. As Fig 7 shows, our system could still operate normally even during the primary Master AA failures.

We adopt ZooKeeper to implement the fault tolerance mechanism. When the primary Master AA fails, the CC will get the backup Master Worker node's address, and then the CC could send messages to the backup Master

AA node to make the signing process continue. Fig. 7 states that the failure of the Master AA node produces trivial influence on our system. We could observe that the average throughputs of the normal case (no Master AA node fails) and abnormal conditions (the primary Master AA node fails) are close. Also, the two latency lines of both scenarios approximately overlap. This result demonstrates that our mechanism is highly feasible because, based on this result, we even could ignore the impact of a failure of the primary Master AA node. We could be confident to ignore the influence of the failures of the primary Master AA because we do not execute any extra work when the ZooKeeper transfers the signing tasks for the primary Master AA Node to the backup Master AA node.

## 5.6 Interactions between CC/User to the blockchain

### 5.6.1 Experiment IV: Time taken when CC and User node save/retrieve data to/from the blockchain

Before sending data to the blockchain node, CC will use the BKDR hash function [46] to convert the content to a hash value. The CC then sends the hash value to the blockchain nodes. As a result, in this experiment, all the content sizes are the same. In other words, The smart contract: *addNews()* and *queryNews()* are executed once per message size and calculate the average time taken between all different message sizes. The result indicates that it took CCs on average 13.6 ms to finish the *addNews()* operation and took Users 11.3 ms on average to complete the *queryNews()* function. The reason for the difference between the time taken to process *addNews()* and *queryNews()* is that when Hyperledger Fabric adds a new transaction to a block, the Hyperledger Fabric needs to update the CouchDB and the blocks. Moreover, the performance of executing smart contracts of our system is similar to IBM's performance benchmark [47].

## 6 Conclusion and Future work

Our previous work [16] offers four principal contributions:

- Proposing a proof of stake entropy-based incentive mechanism that could diminish the negative impact of malicious behavior to enhance the reliability of a fake news prevention system.
- Practically implementing a blockchain via Hyperledger Fabric to prove the feasibility of our design of a quorum-based fake news prevention system.
- Designing mechanisms of crash failure prevention and scalability to enhance the system's robustness.
- Providing various experiment results of system performance for further reference.

Moreover, we add six significant novel contributions compared to our previous work [16]:

- Adding the factor of *confidence* in calculating the *RoC* and *PoC* and adding a new category of appraising result: *Neutral* to the original two types of appraising results in the formula of calculating the *Entropy(S)*.
- Conduct different sensitivity analyses of our entropy-based incentive mechanism based on the new three types of appraising results.
- Proposing a new process regarding how to use secret keys of AES to secure AAs voting opinion to keep AAs appraising content independently.
- Discuss in more detail how our system keeps the two critical factors: *Safety* and *Liveness* of a distributed system.
- Proposing a strategy to optimize the *BRRate* and *BPRate* to obtain both the *Safety* and *Liveness* of our system.
- Offering an approach to avoid the CC sending duplicate content to get rewards illegally.

Several essential case analyses are provided in this paper to demonstrate that the proof of stake entropy-based incentive mechanism we proposed is effective in decreasing the negative effect of malicious participants on our system. In addition, the experimental results show that, as we proposed in this work, it is feasible to use Hyperledger Fabric to implement the blockchain network and ZooKeeper to design the mechanism of tolerating crash failure and supporting scalability. Last but not least, the outcomes of analyses and experiments in this work can be used as a significant reference for future research.

Nonetheless, we list two limitations of our system:

- Our system does not aim to scale along with the amount of data like AI-based fake news prevention systems. Instead, it is suitable to apply our system to verify that highly controversial news is difficult for AI to identify.
- Rather than providing ground truth, our system contributes by offering a *trust score* to let users figure out their appraisers' opinions regarding whether digital content is true or false. Based on our *trust score*, users must decide if they believe a piece of news.

Finally, as explained below, we recognize that several future works could still strengthen this implementation.

- There are still two centralized components, ZooKeeper and Master AAs, in our system. It would be better to decentralize these two components.
- This system could be deployed to **Amazon Web Services (AWS)** to measure latency and throughput more similar to the real-world application.
- Applying some public fake news data sets to measure the similarity between our appraising results to those data sets.

## Declarations

- **Data availability:** All data generated or analyzed during this study are included in this published article.

## Compliance with Ethical Standards

**Funding:** This study was funded by Ripple and the Natural Sciences and Engineering Research Council of Canada (NSERC).

## Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- [1] Zhou, X., Zafarani, R.: A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.* **53**(5) (2020). <https://doi.org/10.1145/3395046>
- [2] Hacıyakupoglu, G., Hui, J.Y., Suguna, V.S., Leong, D., Rahman, M.F.B.A.: Countering fake news: A survey of recent global initiatives. S. Rajaratnam School of International Studies, Nanyang Technological University, Jurong West, Singapore, 2018, Accessed: August 16, 2021, [Online]. Available:[https://think-asia.org/bitstream/handle/11540/8063/PR180307\\_Countereng-Fake-News.pdf?sequence=1](https://think-asia.org/bitstream/handle/11540/8063/PR180307_Countereng-Fake-News.pdf?sequence=1) (2021)
- [3] Guess, A., Nagler, J., Tucker, J.: Less than you think: Prevalence and predictors of fake news dissemination on facebook. *Science Advances* **5**(1), 4586 (2019) <https://www.science.org/doi/pdf/10.1126/sciadv.aau4586>. <https://doi.org/10.1126/sciadv.aau4586>
- [4] How google fights disinformation. Google Inc., CA., USA, White paper, February 2019, Accessed: August 15, 2021. [Online]. Available: [https://blog.google/documents/37/How\\_Google\\_Fights\\_Disinformation](https://blog.google/documents/37/How_Google_Fights_Disinformation) (2019)
- [5] Chatila, R., Havens, J.C.: The IEEE global initiative on ethics of autonomous and intelligent systems. *Robotics and Well-Being* (2019)
- [6] Isaak, J., Hanna, M.J.: User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer* **51**(8), 56–59 (2018). <https://doi.org/10.1109/MC.2018.3191268>
- [7] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260 (2008)
- [8] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. *International journal of web and grid services* **14**(4), 352–375 (2018). <https://doi.org/10.1504/IJWGS.2018.10016848>

- [9] DiCicco, K.W., Agarwal, N.: Blockchain technology-based solutions to fight misinformation: A survey. In: *Disinformation, Misinformation, and Fake News in Social Media*, pp. 267–281. Springer, New York (2020). [https://doi.org/10.1007/978-3-030-42699-6\\_14](https://doi.org/10.1007/978-3-030-42699-6_14)
- [10] Jing, T.W., Murugesan, R.K.: A theoretical framework to build trust and prevent fake news in social media using blockchain. In: *International Conference of Reliable Information and Communication Technology*, pp. 955–962. Springer, New York (2018). [https://doi.org/10.1007/978-3-319-99007-1\\_88](https://doi.org/10.1007/978-3-319-99007-1_88)
- [11] Torky, M., Nabil, E., Said, W.: Proof of credibility: A blockchain approach for detecting and blocking fake news in social networks. *International Journal of Advanced Computer Science and Applications* **10**(12) (2019). <https://doi.org/10.14569/IJACSA.2019.0101243>
- [12] Dhall, S., Dwivedi, A.D., Pal, S.K., Srivastava, G.: Blockchain-based framework for reducing fake or vicious news spread on social media/messaging platforms. *Transactions on Asian and Low-Resource Language Information Processing* **21**(1), 1–33 (2021). <https://doi.org/10.1145/3467019>
- [13] Huckle, S., White, M.: Fake News: A Technological Approach to Proving the Origins of Content, Using Blockchains. *Big Data*, 5 (4), 356-371 (2017). <https://doi.org/10.1089/big.2017.0071>
- [14] Ansari, J.A.N., Azhar, M., Akhtar, M.J.: The spread of misinformation on social media: An insightful countermeasure to restrict. *Studies in Economics and Business Relations* **3**(1) (2022)
- [15] Jaroucheh, Z., Alissa, M., Buchanan, W., Liu, X.: Trustd: Combat fake content using blockchain and collective signature technologies, pp. 1235–1240 (2020). <https://doi.org/10.1109/COMPSAC48688.2020.00-87>
- [16] Chen, C.-C., Du, Y., Peter, R., Golab, W.: 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). <https://doi.org/10.1109/BigData52589.2021.9671778>
- [17] King, S., Nadal, S.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August **19**(1) (2012)
- [18] Shannon, C.E.: A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review* **5**(1), 3–55 (2001)
- [19] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K.,

- De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. EuroSys '18. Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3190508.3190538>
- [20] Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) Advances in Cryptology — CRYPTO' 89 Proceedings, pp. 239–252. Springer, New York, NY (1990). [https://doi.org/10.1007/0-387-34805-0\\_22](https://doi.org/10.1007/0-387-34805-0_22)
- [21] Ross, B., Jung, A., Heisel, J., Stieglitz, S.: Fake news on social media: The (in) effectiveness of warning messages. In: 39th International Conference on Information Systems, p. 16 (2018). Association for Information Systems
- [22] Sirajudeen, S.M., Azmi, N.F.A., Abubakar, A.I.: Online fake news detection algorithm. Journal of Theoretical and Applied Information Technology **95**, 4114–4122 (2017)
- [23] Reality Defender 2020. Reality Defender, <https://rd2020.org/index.html> (accessed August 16, 2021) (2020)
- [24] Lauslahti, K., Mattila, J., Seppala, T.: Smart contracts—how will blockchain technology affect contractual practices? Etna Reports (68) (2017). <https://doi.org/10.2139/ssrn.3154043>
- [25] Wahane, A., Patil, B.: Blockchains to curb fake news in an online world. In: 2022 International Conference for Advancement in Technology (ICONAT), pp. 1–6 (2022). <https://doi.org/10.1109/ICONAT53423.2022.9725933>. IEEE
- [26] Fraga-Lamas, P., Fernández-Caramés, T.M.: Fake news, disinformation, and deepfakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality. IT Professional **22**(2), 53–59 (2020). <https://doi.org/10.1109/MITP.2020.2977589>
- [27] Pawlicki, M., Jahankhani, H.: Advancing governance of news provenance posted on social media platforms with the use of blockchain technology. In: Social Media Analytics, Strategies and Governance, pp. 1–30. CRC Press, Boca Raton, Florida, USA (2022)
- [28] Paul, S., Joy, J.I., Sarker, S., Ahmed, S., Das, A.K., *et al.*: Fake news detection in social media using blockchain. In: 2019 7th International Conference on Smart Computing & Communications (ICSCC), pp. 1–5

- (2019). <https://doi.org/10.1109/ICSCC.2019.8843597>. IEEE
- [29] Saad, M., Ahmad, A., Mohaisen, A.: Fighting fake news propagation with blockchains. In: 2019 IEEE Conference on Communications and Network Security (CNS), pp. 1–4 (2019). <https://doi.org/10.1109/CNS.2019.8802670>. IEEE
- [30] Qayyum, A., Qadir, J., Janjua, M.U., Sher, F.: Using blockchain to rein in the new post-truth world and check the spread of fake news. *IT Professional* **21**(4), 16–24 (2019). <https://doi.org/10.1109/MITP.2019.2910503>
- [31] Christodoulou, P., Christodoulou, K.: Developing more reliable news sources by utilizing the blockchain technology to combat fake news. In: 2020 Second International Conference on Blockchain Computing and Applications (BCCA), pp. 135–139 (2020). <https://doi.org/10.1109/BCCA50787.2020.9274460>. IEEE
- [32] Ush Shahid, I., Anjum, M.T., Hossain Miah Shohan, M.S., Tasnim, R., Al-Amin, M.: Authentic facts: A blockchain based solution for reducing fake news in social media. In: 2021 4th International Conference on Blockchain Technology and Applications, pp. 121–127 (2021). <https://doi.org/10.1145/3510487.3510505>
- [33] Koly, W.S., Jamil, A.K., Rahman, M.S., Bhuiyan, H., Bhuiyan, M.Z.A., Al Omar, A.: Towards a location-aware blockchain-based solution to distinguish fake news in social media. In: International Conference on Ubiquitous Security, pp. 116–130 (2021). [https://doi.org/10.1007/978-981-19-0468-4\\_9](https://doi.org/10.1007/978-981-19-0468-4_9). Springer
- [34] Singh, R.R., Thakral, M., Kaushik, S., Jain, A., Chhabra, G.: A blockchain-based expectation solution for the internet of bogus media. In: Intelligent Data Communication Technologies and Internet of Things, pp. 385–397. Springer, New York (2022). [https://doi.org/10.1007/978-981-16-7610-9\\_28](https://doi.org/10.1007/978-981-16-7610-9_28)
- [35] “PolitiFact. PolitiFact, <https://www.politifact.com/> (accessed August 17, 2021)
- [36] Adair, B., Stencil, M., Guess, C., Ryan, E., Luther, J., Royal, A.: Fact checking. Duke Reports’ LAB, <https://reporterslab.org/fact-checking/> (accessed August 17, 2021)
- [37] Han, R., Yan, Z., Liang, X., Yang, L.T.: How can incentive mechanisms and blockchain benefit with each other? a survey. *ACM Computing Surveys (CSUR)* (2022). <https://doi.org/10.1145/3539604>

- [38] Chen, Q., Srivastava, G., Parizi, R.M., Aloqaily, M., Al Ridhawi, I.: An incentive-aware blockchain-based solution for internet of fake media things. *Information Processing & Management* **57**(6), 102370 (2020). <https://doi.org/10.1016/j.ipm.2020.102370>
- [39] Zen, T.H.Y., Hong, C.B., Mohan, P.M., Balachandran, V.: Abc-verify: Ai-blockchain integrated framework for tweet misinformation detection. In: 2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 1–5 (2021). <https://doi.org/10.1109/SOLI54607.2021.9672392>. IEEE
- [40] Farooq, M., Ashraf Makhdomi, A., Altaf Gillani, I.: Crowd sourcing and blockchain-based incentive mechanism to combat fake news. In: *Combating Fake News with Computational Intelligence Techniques*, pp. 299–325. Springer, New York (2022). [https://doi.org/10.1007/978-3-030-90087-8\\_15](https://doi.org/10.1007/978-3-030-90087-8_15)
- [41] Silberschatz, A., Korth, H.F., Sudarshan, S.: In: *Database System Concepts* (6th Ed.), pp. 897–898. McGraw-Hil, New York, USA (2010)
- [42] Lesne, A.: Shannon entropy: a rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Mathematical Structures in Computer Science* **24**(3) (2014). <https://doi.org/10.1017/S0960129512000783>
- [43] Wood, G., *et al.*: Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* **151**(2014), 1–32 (2014)
- [44] Wang, W., Hoang, D.T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., Kim, D.I.: A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access* **7**, 22328–22370 (2019). <https://doi.org/10.1109/ACCESS.2019.2896108>
- [45] Apache ZooKeeper. he Apache software foundation, <https://zookeeper.apache.org/> (accessed August 17, 2021)
- [46] Kernighan, S.W., Ritchie, D.M. (eds.): *The C Programming Language*. Prentice Hall Professional Technical Reference, Oboken, New Jersey, USA (1988)
- [47] Thakkar, P., Nathan, S., Viswanathan, B.: Performance benchmarking and optimizing hyperledger fabric blockchain platform. In: 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 264–276 (2018). <https://doi.org/10.1109/MASCOTS.2018.00034>