FAKE NEWS DETECTION IN SOCIAL MEDIA USING BLOCKCHAIN

Kaustubh Katkar, Nikhilesh Reddy Tummala and Santosh Kannan

(3147-0922 8350-1593 9095-5971)

CISE University of Florida, 2020

***Abstract* – Fake news has become a long-standing problem on social media platforms. Such false information is used by malicious attackers to defame people, manipulate elections and much more. Over the course, many different methods have been proposed to tackle this problem. In this paper, we present a blockchain based model, which can help detect fake news in social media. We leverage the peer to peer architecture of the public blockchain network in deeming a news authentic or fake. This is achieved by a rating scheme we propose for each post, wherein, end users of the system are encouraged to rate a post as valid or invalid. All such functionalities are achieved using the smart contracts system provided in the Ethereum blockchain model. This paper also presents a prototype model for the ideology and backs the proposal with some experimental results. Furthermore, this paper identifies the challenges our implementation might face as well as acknowledges the technologies whose collaboration can improve the efficiency of this research.**

I. INTRODUCTION

Blockchain in social media is novel concept which has numerous possible applications. Our model leverages the security advantages of a blockchain network to authenticate shared news on social media. We provide an Ethereum blockchain based prototype to effectively distinguish true and false information shared on the social media platform.

Social media is an easy and quick source for all kinds of global and local news. However, it backfires when malicious individuals spread false information. These individuals / organisations seek to take advantage of people's tendency to share appealing information without verifying the authenticity of the news. Such acts could potentially influence elections negatively, defame well-known people and much more.

The rise of fake news highlights the abrasion of long-standing institutional firewall against misinformation in today's world. However, much remains unknown regarding the vulnerabilities of individuals, institutions, and society to manipulations by malicious actors. A new system of safeguards is needed. To deeply understand the issue regarding fake news, a recent paper titled "*The science of fake news*" [1] has attempted to study the science behind fake news. This answers the questions, what is fake news? how is fake news generated? what are the essential elements in spreading false information? Solutions proposed based on media reliability which classifies the level of truth for the news in the question answering system based on modified CNN deep learning model are not practical in use. These model work by training the model with an input dataset which contains the truthfulness of each media as well as that of the proposition. But keeping the media dataset updated and maintaining hundred percent truthfulness is still a huge challenge. The organization which deploys this model might be in danger of government interventions in the matter of categorizing certain news if they are of sensitive issue. None the less, there is always a drawback of these organizations who took the task of identifying fake news on the internet might be biased to some of the data posted online and might design a model that could benefit them in any manner. Some proposed models aim to leverage machine learning and AI related approaches to tackle this problem. For instance, authors Youngkyung Seo, Deokjin Seo, Chang-Sung Jeong provided a fake news detection model based on text classifier system [2].

Although researches have proposed few ideas which are being implemented in the real-world. Some of these real-world applications that certain organizations deploy to identify fake news includes programs like Machine Learning algorithms combined with Linguistic Analysis [9] and Knowledge Engineering [8]. These type of programs, analyse the data posted online by running data mining and information retrieval algorithms which then are linked to linguistic analysis to identify the tone of the news and later verify this data using some fact checking techniques like Knowledge Linker (Ciampaglia et al. 2015), PRA (Lao et al. 2011), PredPath (Shi et al. 2016) etc. Some approaches which use predictions algorithms to check the fact are Degree Product (Shi et al. 2016), L. Katz (1953), Adamic & Adar (Adamic et al. 2003) and Jaccard coefficient (Liben et al. 2016. Fact checking mainly focuses on checking the fact of the news by verifying with a stored reservoir of information containing all the facts. There are some fact checking organizations who provide online fact checking services like: Snopes3, PolitiFact4 & Fiskkit5.

We thus aim to provide a practical solution utilising the advantages of blockchain. Primarily we focus on leveraging the below advantages:

- Enhanced security and privacy
- Decentralized network architecture
- Robust authentication and anonymity

In this paper, we propose a model to detect fake news from social media using user input. However, there are over thousands of original news being published every day and forcing user feedback on all this news is not practical. We therefore classify the news into two categories: high priority and low priority based on criterion discussed later. High priority news is authenticated using the user input. These posts are deemed as valid or invalid based on a rating scheme defined in the later sections. For low priority news feeds, we

suggest an automatic fake news detection scheme to avoid any user validations. Along with this, we create additional blockchain called the "Fake news blockchain" to which stores the count of fake news detected. We discuss the design of our model and the functioning of our prototype in the following sections.

II. RELATED WORK

Due to exponential growth of information online, there are many existing detection programs to identify whether any data posted online is fraudulent. Researchers over the last decade have proposed multiple ways to nullify the effects of fake news. Few creators recently explored the Blockchain route to develop a smart contract based fake news detection model [3]. Another paper demonstrates a model based on leveraging user inputs to verify the authenticity of the news [4]. However, these approaches only the recommend an approach without backing it up with required analysis.

Hoaxy6 is another platform for fact checking operation. Collection of data, detection of fake data and analysis of this to check online misinformation is part of Hoaxy. The criteria they followed is to check the news is false or not, by simply referring it to some domain experts, individuals or organizations on that particular topic and that could be a disadvantage because there would be absolutely no way for anyone to determine the credibility of those domain experts or individuals or organizations when it comes to fact checking. There could be a possibility where they could not obtain all the facts to store in their repository or even if they did, we still cannot guarantee whether they would bias over some facts of sensitive issues.

Keeping this in mind, our proposed solution uses blockchain to provide decentralized peer-to-peer network and that would eliminate the threats faced by these centralized fake news detection applications. Having a decentralized peer-to-peer network would present a great advantage when it comes to credibility because instead of relying on any domain experts or organizations or a bank full of facts, we ask the general public and some authorized evaluators to validate the newsfeed by motivating them with promising incentives and these evaluators would be financially punished if their validation towards any newsfeed is biased or they will be financially rewarded if they give genuine news validation.

III. SYSTEM ARCHITECTURE

A. System Overview

The model proposed in this paper presents the use of blockchain for publishing the news and computing the validity of the news. The news is classified into high priority and low priority based on the criterion in the smart contracts. The "smart contracts" are used to track the published news and change in ratings (such as origin of a news article, users who repeatedly publish false information). We use the Ethereum blockchain [10] model to log time and id of the publisher, Vote count and the corresponding ratings on external databases. The system further maintains two blockchains, one which the hashed value for the published text to maintain data integrity and reduce the overhead for data storage. The other blockchain called "fake news blockchain" adds the blocks whenever a fixed number of fake news have detected. This aids in performing various analysis on blockchain.

The first smart contract monitors the news publishing process. Each new publisher address is assigned credentials and the identity is confirmed through the public keys.

As soon as the publishers publish the news, the smart contract determines whether the news is high priority or low priority. For high priority news, validators will provide a corresponding rating for each of the posts. The ratings from the validators are calculated with a weighted model, which ensures the integrity of the system. For low priority items, the algorithm provided in the functionality section authenticates the news.

In the following sections, we present the design and working of our model

B. System Description

The primary focus of our design is to build a model which leverages blockchain to nullify the effect of false information on the social media platforms. The design for our model broadly assumes two flows in the system: *publishing a news and validating a news.* The system is deployed on a peer to peer network with Ethereum Blockchain client. At a high level, this design contains three components: Ethereum Client, Database and frontend. These are typically hosted on servers in a social media platform. In this report, we provide a model based on Twitter ecosystem assisted by Ethereum Client which acts as an interface for the blockchain and a social media application. This model can further be utilised in any other social media system by integrating the blockchain model into the existing ecosystem.

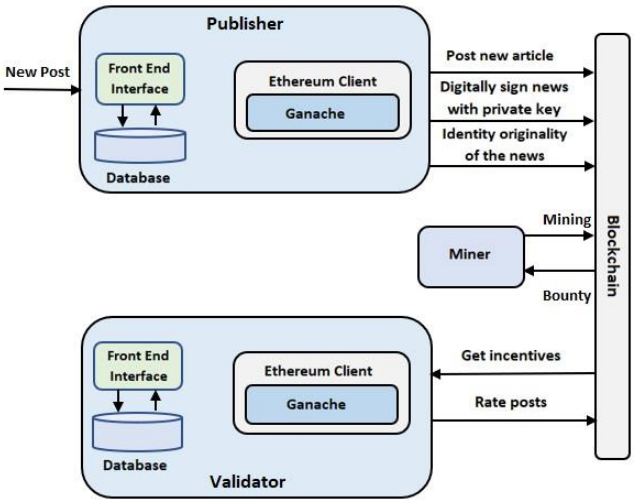


Fig 1. High level view of Ethereum blockchain client

Front End Interface: Typically, a front-end system consists of a user interface which allows the user to access the data. In our application, the front-end system would assist the user with features such as login, publishing and rating the news feeds. These actions are performed on the blockchain through the Ethereum client interface with the public / private keys.

Database: The database is a data repository for the entire distributed network. All the posts and images would be physically stored in the databases, designed with data marts.

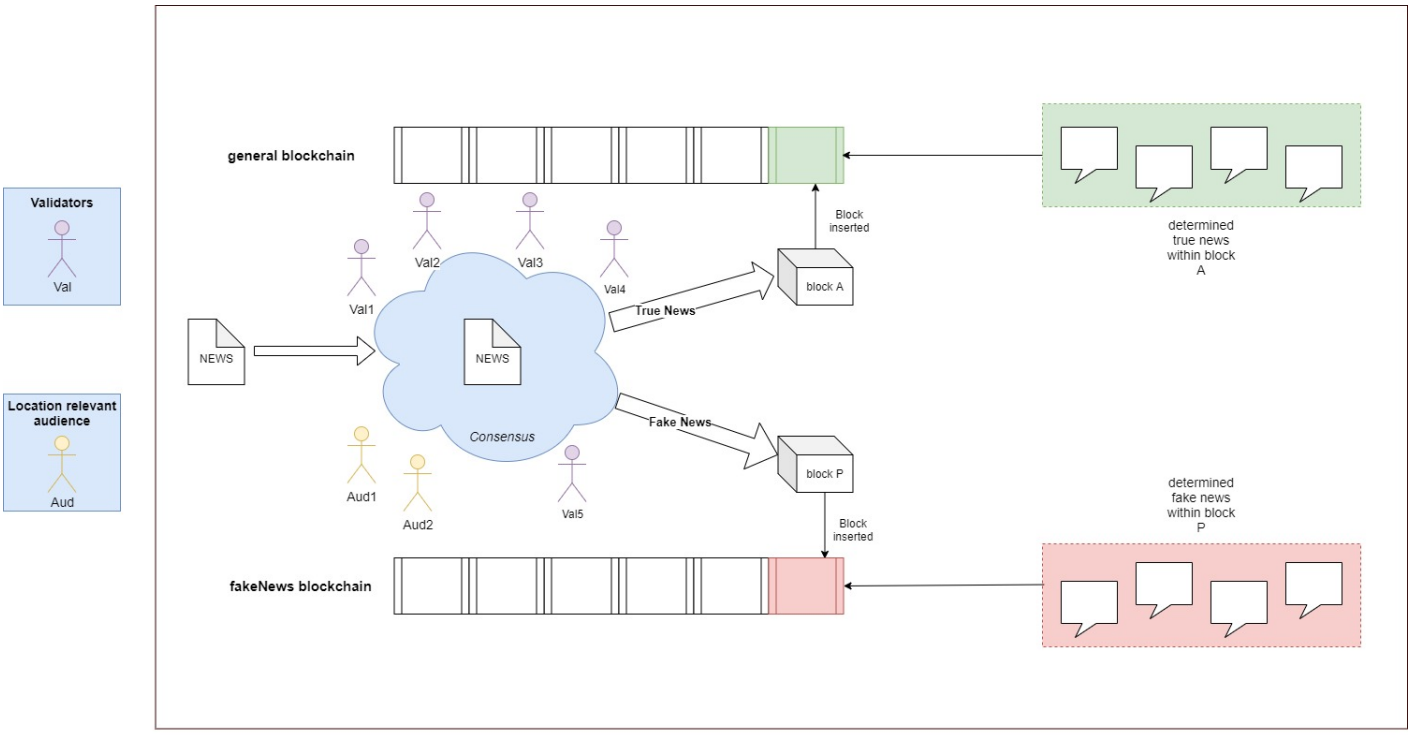


Fig2. Structure of the blockchain in fake news detection

The databases along with the front-end system are responsible for the client and server-side communication of the network.

Ethereum Client: This component is the primary module for implementing the functionality of the blockchain [12]. Briefly, this component is tasked with connecting to the peer to peer network, sending encoded ratings to the network and maintaining a local copy of the blockchain. In our prototype, we use Truffle and Ganache client. The smart contracts [13] are deployed through this client and the addresses of participants are resolved. The smart contracts for publishers and validators would be continuously running within the client. In any event, these contracts would be triggered, and corresponding transactions would be executed on the blockchain.

Blockchain: Blockchain is a continuous chain of blocks. This model utilises two separate blockchains. In the general blockchain, each block stores a transaction which occurs in the network. Smart contracts are designed and deployed on the blockchain. These contracts are continuously running within the system and are triggered every time a publisher publishes news, or a validator provides rating. The other blockchain called the "fake news blockchain" adds a block to the blockchain every time a fixed number of fake news have been detected. As we obtain legitimacy scores on a numeric scale of 1 to 10, the scores less than 5 will be added to the fakeNews blockchain otherwise they get added in the general blockchain. This bifurcation will allow us to maintain a consistent record of the blocks in terms of the news type. Also queries over the blockchain will take less amount of time as both can run the getTransaction function in parallel when identifying if a post already exists and has been reposted. This event is helpful in the situation of a repost so that the post is not again passed through the consensus mechanism and instead receives the existing legitimacy rating.

Blockchain plays a key role in two perspectives in the network. For a publisher, the blockchain will store the address of the publisher, timestamp of the post and hashed value of the post. Once a news feed has been posted, validators would be able to access the post through their public keys and rate the feed accordingly. Each transaction within the blockchain will be a collection of data which will include the address of the publisher, the rating of the post, the addresses involved in the consensus and the priority of the post. Initially the high priority will be assigned to news posted by verified account and low priority will be assigned to news posted by non-verified accounts (general audience). However, there will be instances where a low priority post gets updated to a high priority based on the general audience reaction. As there is no third party determining the priorities for each post, we successfully ensure that decentralization is satisfied. These posts will then be queued for consensus by validators.

Proof of Stake consensus: Every validator participates with a stake. This stake will impact the final incentive received. When all the transactions are captured within the block the miner will add the block to the blockchain and the reward gets distributed between the participating validators as incentive. Incentivization scheme of our model is still dictated by the time and rating of the validator. The percent of incentivization will simply apply to the stake of a validator. In the case that a validator is dishonest he/she will lose the stake. Dishonesty will be identified as the deviation from the consensus and the degree of deviation will be accounted for which results in reduction of the validator rating. This is to ensure long term honesty of the validator as we do not want a validator to occasionally vote honestly and reap most benefits for that session. Stake will also enforce accurate voting discipline and validators with confidence in the reliability of the post will be participating with higher stakes to achieve corresponding incentives. The participation of the validator will also be checked within the validator mapping to deny validators trying to cast multiple votes. A robust approach of a weighted Proof of stake consensus can be utilized as in [11].

C. Smart Contracts:

Our model incorporates smart contracts for publishing posts, voting mechanism for validators and audiences, and providing incentives to the validators.

Publisher: Every news published by a publisher is subject to the terms provided in the smart contract for publishing. The smart contract determines whether the news published was an original or a duplicate post, since it is unnecessary to spend time validating redundant posts. The smart contract would identify the original post and assign the same ratings to the duplicate post. The contract would also calculate the rating for a publisher based on the votes received for each post published by the publisher. This is essential in flagging malicious users who spread false information.

*Voting Validators***:** The validators are verified and authentic accounts who receive incentive for their input. To determine whether an account is authentic will be the responsibility of the social media platform. They can achieve this by associating accounts with their businesses or brands and provide our module with that information. The validators are tasked with rating all the news posts. These ratings are processed through smart contracts to compute the overall rating of the post, rating of the publisher and the individual rating of the validator. After each consensus, the ratings of validators, publisher and post is updated. The smart contract will dictate the window of the voting period. This window will rely on two factors: an 18-hour time period or the count of validators participating.

Voting Audiences: The general public or unverified accounts will also be allowed to vote albeit their votes will be displayed in a different format. These ratings will be displayed immediately on the post and will not be an indicator of whether the post is legitimate or not unless it satisfies a geographic constraint. General audiences are not rewarded for their input in our model. Our primary focus will be to determine what format of news attracts the general public to assist analysts of the social media platform.

The geography of an account is an important factor to determine the accuracy of a news story as people belonging to the region can attest to it. In this regard there is a research towards a protocol named FOAM [5] which addresses how they obtain geospatial data and can be incorporated to determine the location of an address. This protocol has matured over 5 years, at the time of writing this report, which is why we are comfortable recommending it. Thus, based on the location we can add the unverified votes to the legitimacy score as well however with a low multiplier. These votes will also be added to the audience votes.

D. Accessing Location Information

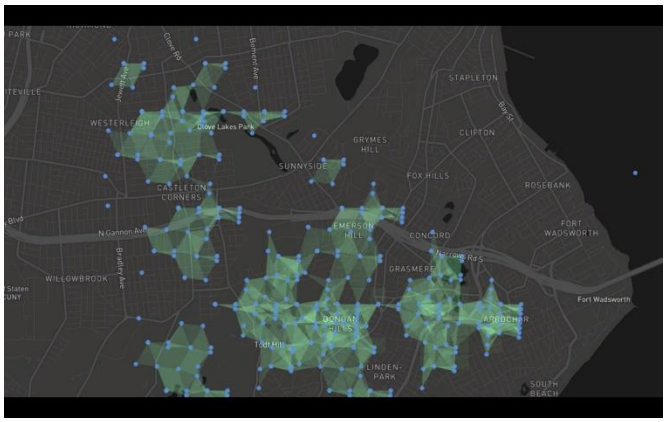
Our model aims to leverage the user geography information to prioritize feedback based on geography. To provide location-based services, the system requires to have location information of all the users who are involved in the application. There are lot of satellite based, mobile geo location-based approaches to read the location of the user. However, we recommend a blockchain based model for geo location services called FOAM. This model has been proposed in a recent research for accessing geo location using blockchain. This protocol works towards getting the precise geo-spatial location of the users through a decentralized application while still preserving their privacy.

FOAM is a geo-spatial protocol built on an Ethereum blockchain. The protocol functions by adding an entire new layer to the Ethereum stack which is the location layer standard. We can use any of the different types of 2-dimensional location encoding standards listed below:

Geolocation Standards

	example	unique	not proprietary	deterministic	verifiable	cryptospatial
postal	Times Square, Manhattan, NY 10036	no	yes	no	no	no
long/Lat	40.758895,-73.9873197	yes *	yes	yes	yes	no
GEO+ETH	XrCNFtAaz5x1HUw6o5GLbtMDqc1Nn4qX	yes	yes	yes	yes	yes
CSC	r5x34ru7i	yes	yes	no	yes	yes
geohash	dr5ru7k	yes*	yes	yes	yes	no
what3words	rocky.silver.funded	yes	no	no	no	no
xaddress	2399 OUT CASTS	yes	no	no	no	no
open location code	Q257+H3	yes	yes	yes	yes	no
makaney code	WWJT-89GN	yes	yes	yes	no	no
what3emojis		yes	yes	yes	no	no

FOAM protocol introduces a new blockchain spatial dimension called CSC - Crypto-Spatial Coordinate and operates on the proof of Location consensus. CSC functions by assigning a unique mathematically determined location address to any place on the map and the size of this place can be as small as a single building in the city. The smart contracts can reference the built-in environment and obtain proofs of events for the locations termed as Proof of Location. This protocol requires people to verify the location information.



As depicted in the image above, zones are created in this system. Whenever a user walks into any of these zones, he requests a “presence claim”. The authorities verify this claim against other zones and store the information on the blockchain. As users travel to different locations, the “presence claim” protocol is triggered, and the system can access this information. This information can be leveraged to achieve the goal of prioritising users based on geographical locations.

IV. FUNCTIONING OF THE MODEL

The functionality of the model varies for high and low priority news. Thus, we distinguish the functionalities and describe them in two separate sections:

1. HIGH PRIORITY NEWS

As mentioned earlier, the model relies on the communication across the application system (front and back end) and the blockchain ecosystem. Our prototype for high priority news items is deployed using Ganache Ethereum client and Truffle development environment. A detailed working of the model is presented below:

A. Publishing a News

Each publisher is identified as a verified / non verified publisher by the social media organization. This information is used by the smart contract to assign them corresponding user ids. Once a publisher has registered with the system, they can use their credentials to login into the system. This would navigate them to the user interface of the social media application. The publisher can then upload the post and publish it on the peer to peer network. Based on whether the publisher is a verified or non-verified, the news articles are distinguished as high priority and low priority.

B. Publisher Smart Contract

Once the publisher uploads a post, the Ethereum client is responsible for interfacing the social media application and the blockchain. The post will include a digital signature of the publisher, signed with its private key. The post goes through the *'publisher'* smart contract which identifies whether the post is original. This is achieved using the language *now* which returns the current Unix Epoch time of the transaction that increments each second. So, the smallest value is always the earliest. We maintain the timestamp as a constant variable because if all the blocks are not up to date and if within a vast network a post appears where the blocks updated are not yet synchronized, when the entire network gets updated, the timestamp of the post will be checked against same post if such a post exists and the smaller timestamp value wins. This will require another mapping of post to its Epoch time which will be in uint256 format. A private function is used to update this mapping with access privileges only to the owner of the contract.

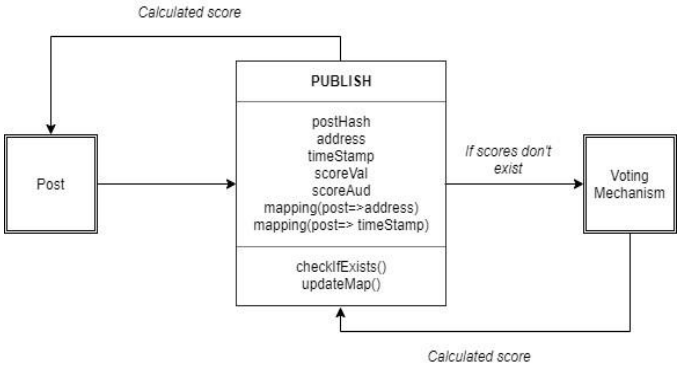


Fig 3. Flow of operations through Publish smart contract

Since we are not concerned with how many times the post has been republished, we will identify the first instance when the post is published and maintain a hash value of the post. Solidity provides *keccak256()* function to achieve this. We map the hash value with the address of the source. At any instance when a post is published the hash value of the newly published post will be calculated and checked if it already exists in the hash. If it does, then the already acquired audience and validator ratings are displayed over the post. Otherwise, the new hash is added to the map along with the address of the sender. The voting mechanism is initiated to compute the scores. We will discuss this in more detail in the Voting smart contract section.

Thus, this smart contract will store the cryptographic hash value of the post, the timestamp and the address of the publisher on the blockchain.

C. Voting by Validators

After a news has been successfully published on the blockchain, the validators can access these posts through their public key and send their ratings. There are three types of ratings associated with this model: rating for a post, rating for the publisher (*rpub*) and rating for the validator (*rval*).

Considerations:

18-hour limit: This will be the window for validators to vote over the legitimacy of a news. This period ensures that across the world every validator had an opportunity to participate in the consensus if they wish to. The incentivization however is strictly based on the timeliness of the vote. This can be seen in the section (III.C.2).

Count of the validators: We can maintain a threshold for validators (eg. 40% of all validators) and when that number is reached, we can end the consensus process and update ratings. This ensures that under high participation the legitimacy scores for the news will be posted faster. This function provides the possibility that some critical piece of news posts will receive their legitimacy score much faster and most of the audiences will be able to view the legitimacy score when they come across the news within the initial hours. Thus, a quick consensus will be very beneficial scenario which is what this aims to achieve.

These two factors are events in the programming context and can be used with JavaScript web3.js collection of libraries. These events are triggered when the voting window ends, or the predefined count of voters is attained. After this the owner of the contract will be accountable to release the ratings that will determine individual performances and incentives. To ensure that validators are not able to call the contract functions and get the current score during the voting window the solidity contract has a access restricted private function that computes the average of each post. After the end of the voting mechanism the owner of the contract calls another private function releaseScores() with access modifier that assures only the owner node of the contract will be able to call the function.

```
address private owner;

function votingMec() public{
    owner = msg.senderl
}

modifier onlyOwner(){
    require(msg.sender==owner);
    _;
}
```

D. Rating for the post

Each validator can rate the authenticity of a post on a scale of 10. These ratings are then calculated with a weighted scheme according to the individual rating of the validator. The validators rating is taken into consideration to factor in the legitimacy of their vote. A validator with a higher *rval* will have a greater influence on the rating of a post compared to a validator with a lower *rval*.

Post Rating Scheme: Consider a news article is published and two validators A and B with respective *rvals* 5 and 10 decide to vote on the article. A voted 5 whereas B voted 6. Our scheme calculates the average rating as weighted scheme as follows:

$$\textit{Average Rating} = \frac{\sum (\textit{rval for user} * \textit{rating by user})}{\sum \textit{rval for user}}$$

Although, the ratings 5 and 6 are not far apart, the weighted value (*rval* for user * rating by user) i.e. 25 and 60 are significantly different. Since B had a *rval* of 10, it has a higher influence on the overall rating of the post.

Each vote is stored in local variables of functions, which are not visible to the public. This is essential in calculating the *rval* of the validator.

E. Rating for the validator (*rval*)

As presented in the above section, the validator's rating plays an important role in computing the rating of the post. Thus, it is important to implement a method to efficiently calculate the validator rating. Along with this, it is equally important to provide the validators an incentivization scheme for rating as many posts sincerely.

Our smart contract efficiently computes the *rvals* for the validators based on the rating of the posts. For each post, the smart contract will consider a score bracket of + and – 1 of the average rating for the post. All validators who rated within this range will a positive effect on their rating. The ratings of all validators who rated outside the bracket will be negatively affected by a factor of how far their rating was off.

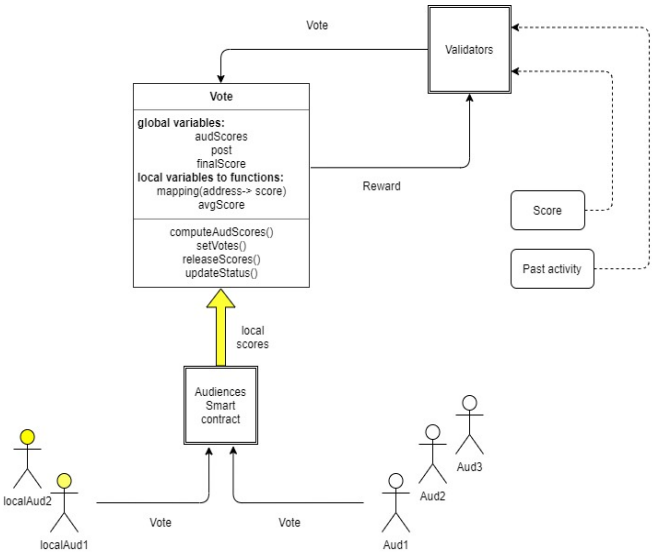


Fig 4. Flow of operations with respect to the voting smart contract

Rval Rating Scheme: Suppose a news article receives an overall rating of 7 after the 24hr voting window. A validator who voted 8 would have an increase in *rval*, whereas, validator who voted 4 would have a decrease in *rval*.

Incentivization: Each rating by the validator will qualify for an incentive. Since the unit of currency will depend on the organization, we propose the incentivization in terms of factors. Suppose the maximum reward for providing a rating is 1 unit, a validator with a *rval* of 8.5 will qualify for a maximum incentive for 0.85 units. For fairness, we will award the scores of 9 and above full amount i.e. 1 × max reward, as maximum reward as it will be very unlikely for any account have a perfect score. Initially, all validators will begin with a rval of 0.7. This should motivate the validators to provide sincere and accurate ratings to qualify for higher incentives. Do note that the time specific multiplier, which we discuss ahead, with still affect the maximum reward value for each validator.

Since we are tackling spread of fake News, we need to limit the duration within which the voting takes place. Having people vote when the news becomes irrelevant is not a practical scenario. So, we maintain a time limit within which voting will take place. Through our simulations, we decided a decay pattern as below:

All votes received in the first hour after the news is published are eligible for full rewards. Votes thereafter decay at a rate of 1.2 which is equivalent to a multiplier of 0.833 every hour. It follows the following pattern:

0.8, 0.667, …, 0.036

The below chart shows reward eligibility per hour for a validator with *rval* 8.

Hour	Reward Multiplier
1st	0.8
2nd	0.8 ÷ 1.2 = 0.667
3rd	0.667 ÷ 1.2 ≈ 0.556
4th	0.556 ÷ 1.2 ≈ 0.463
5th	0.463 ÷ 1.2≈ 0.385
6th	0.385 ÷ 1.2 ≈ 0.322
…	…

** This decrement scheme does not affect the *rval* score

After the end of the voting period the rval scores for the validator are updated. The scores and voting average is stored in a private function with access only to the owner of the smart contract and the scores are updated and released after the status of voting period changes to *false*.

This distribution scheme for rewards sufficiently ensures that there is a reasonable drop off in the reward scale after early hours and especially after the first hour. As this approach can be easily modified to scale up or down according to the time period of voting, it provides favourable control to the social media platform to make a judgement over it. We tried a formula driven approach as well to compute the reward for each time period. However, this added computation over the network increases overhead due to which performance might degrade with increasing scale of operations.

Local Audiences: The location-based services (section III.D) will help us add weight to the general audiences voting from the region of post. Their reaction will contribute towards the final rating of the legitimacy of the post. These votes however will have a lesser impact as compared to that of the validators. Out of the possible 100 percent of the votes towards a post 10 percent could be entrusted to the audiences inhabiting that location. A consideration here will be that under less participation we should ideally not imply that it is the mutual vote for the locality. To this effect our model defines the localCount parameter which will maintain the count of locale audiences and compute the local vote average in the localAvg function. Once the localCount crosses 1000 the localAvg function will return the score to the releaseScores() function else it returns null. After the releaseScores() calls getLocalAvg() and there exists a value ≠ null, then the votes will have a 90:10 split; else the validators votes account for the total score.

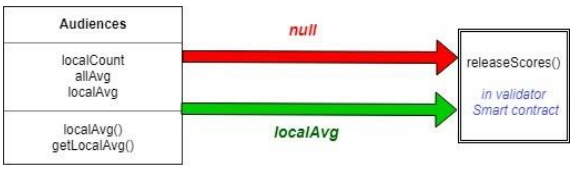


Fig 5. audience smart contract

F. Rating for the Publisher

Like the rating for each validator, each publisher is rated based on rating history of their news. As seen above, every post is rated by validators and these votes are stored on the blockchain. These scores help determine the authenticity of the uploader as well. This is done by storing the validator score of the post against the address of the uploader. Recurring behaviour of malicious or misleading uploads will gradually determine the status of the account for which we propose a three-strike policy. This is a scheme where a yellow, orange and red tag is incorporated for strike 1, strike 2 and strike 3 respectively. Initially, we keep a buffer of five posts where the publisher will not be assigned any tags.

2. LOW PRIORITY NEWS

A. Publishing a News & Publisher smart contract

Similar to the high priority news flow, the smart contract for publishing governs the publishing process. Based on whether the publisher is a verified or non-verified, the news articles are distinguished as high priority and low priority. Once the news feed is classified as low priority, it goes into the low priority news smart contract.

The hash values for these news headers are then compared against old news to determine whether the news is a duplicate news. In this case, the old ratings are replicated for the news. If news is an original news, the rating of the publisher is compared against the average ratings of all publishers to determine the authenticity of the news. Upon comparison, the news is deemed as fake or authentic and is added to the low priority queue. Another algorithm processes this low priority queue. For the particular news, it looks up the retweets and likes on the post and determines whether the news should be classified as high priority. If not, the news is added to the "Fake News Blockchain".

The validation of low priority news can be implemented as shown below in Algorithm 1 and 2.

Algorithm 1

1. Input: News = $\{N_1, N_2, N_3 \dots N_m\}$
2. Input: Time = $\{T_1, T_2, T_3 \dots T_m\}$
3. Output: Low Priority queue
4. Procedure: Create low priority queue
5. While (News.hasNext())
6. for each $N_i \in$ News Do
7. if (N_i.is_duplicate == TRUE) then
8. Rating (N_i) = Rating (Original_post)
9. else
10. Rating (N_i) = Publisher_Rating (N_i.publisher)
11. Threshold = $\frac{\sum \text{Publisher\_Rating(N.publisher)}}{\sum \text{Count(N)}} * 0.9$
12. if Rating (N_i) > Threshold then
13. Ignore (N_i)
14. else
15. Add Ni and Ti to Low Priority Queue
16. End if
17. End for each
18. End While
19. Return Low Priority queue
20. End Procedure

Algorithm 2

1. Input: Low Priority Queue
2. Output: Fake News Blockchain (FNB) = $\{B_1, B_2, B_3 \dots B_m\}$
3. Procedure: Fake news detection for Low priority queue
4. While (Low_priority_queue.has_next())
5. While (Ti – Current timestamp > 6hrs)
6. for each $N_i, T_i \in$ Low_Priority_Queue Do
7. if (N_i.publisher.followers) > 100,000

```
8.  if ((Ni.Likes) / (Ni.publisher.followers))  OR
    ((Ni.Retweets) / (Ni.publisher.followers)) > 50%
9.  Add Ni to High Priority Queue
10. else
11. Bj  ←  Ni   // Add News to block
12. FNB ← Bj    // Insert block into Blockchain
13. if (count (Bj)) == 100)
14. Open new block Bj+1
15. End if
16. End if
17. End if
18. End While
19. Delete Ni from Low Priority queue
20. Return FNB
21. End While
22. End Procedure
```

V. EXPERIMENTS AND RESEARCH

Separate experiments and analysis were conducted for High priority and Low priority news due to the difference in the validation process of the model.

A. HIGH PRIORITY NEWS

Upon running our preliminary tests on a model on Ganache , we attained an approximate legitimacy with score 6 and issued a bracket of +1 and -1 which deemed the scores 7 and 5 valid as well. In our implementation we rounded off the average score to its nearest integer for the bracket. The validators with scores within this bracket i.e. 5,6 and 7 will thus be receiving full credit for their votes increasing their individual score.

We determined that the scale of 10 works better to identify instances of prejudice and bias as a vote further apart from the consensus will have reasonable doubt.

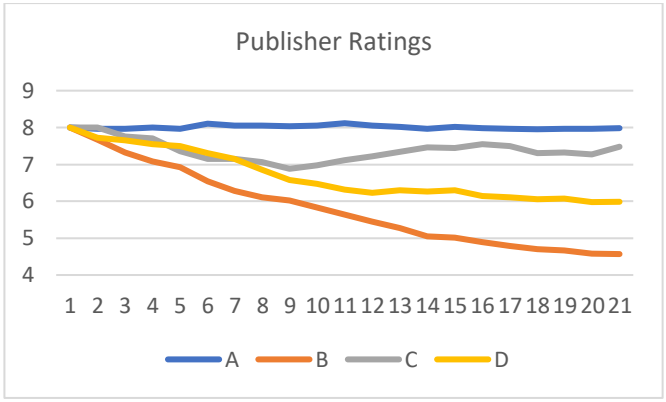
Weighing the scores: In this setting we had to make sure that validators with a good record of honest votes should be allowed to have a bigger contribution to the voting process than their counterparts with a lower score. Thus, we added weights to their score when averaging the total score.

The test example along with the description of its result is mentioned with its formulation in the FUNCTIONING OF THE PROTOTYPE section (section III).

To validate whether the proposed model meets its objective of identifying fake news and discouraging biased validators, we created a mock environment with 200 validators and 500 news articles. We assigned an initial rating of 8 for both validators and publishers and neglected the first 10 posts for each publisher. For the analysis, we followed the journey of 4 categories of publishers and validators as mentioned below:

- Publisher A – Genuine news publisher
- Publisher B – Repeated fake news publisher
- Publisher C – Publisher posting fake news occasionally
- Publisher D – Inconsistent publisher

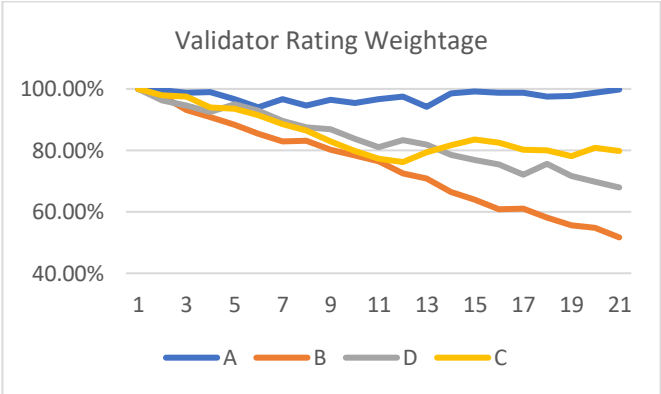
First, we observed the publisher's behaviours through posts 10-30 and tracked the ratings as show in figure below:



From the above figure we can infer that a genuine publisher's ratings are not affected through the system. On the other hand, a publisher (B) who is a repeated offender of fake news, is adversely affected through the models rating scheme. The case is only somewhat better for an inconsistent publisher (D). As for publisher (C), whenever a fake news is published, the rating takes a hit and is stabilized with genuine posts later. Thus, apart from publisher A, the system does not encourage any other publisher. With strict rules on the ratings and licenses on the publishers, the system could prove to be a robust solution for the fake news problem.

Secondly, we monitored the behaviour of validators and their influence on the validation process. Like the publishers, we analysed the scenarios for 4 validators A, B,C and D respectively. Furthermore, since the ratings, influence on the validation process and the incentives are positively correlated, we only provide the chart for validators influence. Figure 2 shows the validator's rating weightage to a particular rating through the first 20 posts.

FIGURE 2: Percentage of Validator's vote considered in the system



The observations are similar to the publisher ratings. The percentage of a vote considered from a genuine validator is ~100%. However, it decreases as we go through other categories of validators. The trend is similar for the individual validator ratings and incentives received by validators for each rating. Thus, the model discourages fake news publishers as well as biased validators.

Rewarding System: Our knowledge of rewards for the input of validators is based on the real-world instances where Google, Facebook and Amazon are rewarding users for their data either through questionnaires or other programs.

As in our case since the validators are professionals in their fields, the reward should be more lucrative. However, we

ourselves refrain from taking any liberties and suggesting the reward as we are aware that each industry will have varying economic standing and different allotment of funds therein. Hence, we believe this decision is best left with the social media platform itself.

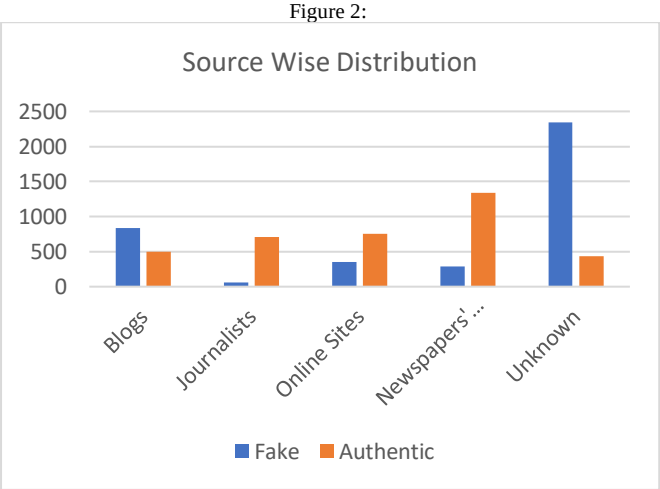
B. LOW PRIORITY NEWS

To validate the accuracy and the practicality of the proposed algorithm for low priority news, we applied an experiment on a dataset collected from BuzzFeed assuming all sources as non-verified. The dataset contains several disaster relevant news posted by various sources of information. Microsoft Excel tool has been used for analysing the data. Since we do not have any estimates on the publisher ratings, we used the ratings from Buzzfeed's statistics on News Trust and scaled it between 1- 10 as shown below in Table 1.

News Source	Publisher Rating
Newspapers' websites	5.8
Journalists	5.2
Blogs	3.4
Online Sites	3.5
Unknown	1.5

The distribution of the dataset across different data sources is shown below:

Figure 2:



On applying the algorithm, the simulation detected 4105 news posts as fake and 3508 posts as authentic. On comparing the results obtained with the Buzzfeed provided results, it could be determined that the algorithm failed to recognize 705 (18%) news posts as fake and 927 (24%) posts as authentic. The breakdown for detected fake news is shown below in table 3.

TABLE 3: Detected Fake and Authentic News

News Source	Fake News	Authentic News
Newspapers' websites	0	1339
Journalists	0	712
Blogs	836	0
Online Sites	0	752
Unknown	2342	0

TABLE 4: Total Fake and Authentic News in the Dataset

News Source	Fake News	Authentic News
Newspapers' websites	291	1339
Journalists	61	712
Blogs	836	496
Online Sites	353	752
Unknown	2342	431

Thus, based on these numbers, the True Positive, True Negative, False positive, False Negative, Precision, Recall and Accuracy for the low priority model can be calculated using equations:

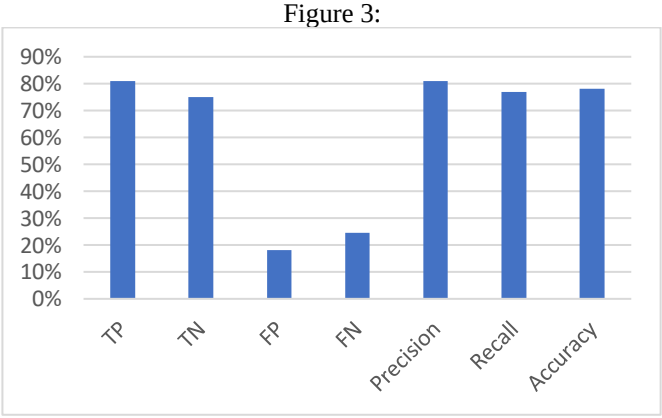
Precision = TP / (TP + FP)

Recall = TP / (TP+ FN)

Accuracy = TP+TN / P + N

Figure 3 represents the calculated values for the obtained results:

Figure 3:



The model as shown below, obtains an accuracy ~78% for the low priority queue. However, as mentioned before we have taken all the ratings for publishers from Buzzfeed and assigned the same rating for a class of publishers. Our model on the other hand, will assign individual ratings for publishers as per mentioned in the functioning section, which will improve the accuracy of the model. Furthermore, as the algorithm suggests, any low priority news feed which receives high amount of user interaction which will be moved to the high priority queue. Thus, we can see that the numbers obtained in our analysis is only the benchmark. With the recommendations provided in this paper, the accuracy for low priority news posts will improve.

VI. MODEL APPLICATION

The suggested model can be applied on existing social media platform such as Facebook, Instagram, Twitter etc. A brief scenario is explained below on incorporating the fake news detection model on Twitter platform.

On a high level, Twitter ecosystem consists of cloud storage which persists the tweets and different APIs are used to access these tweets. To incorporate blockchain based fake news detection, a new API for accessing these tweets and carrying out the required actions is required. Thus, once the tweet is posted, the API could read the information and interface it with the Ethereum client or any other blockchain clients. This client will then use the public key of the publisher and create an entry in the local ledger and then can be written into the blockchain. Once the blockchain contains the entry for the published news, another API could fetch each of the news IDs and display on the social media's website.

After a tweet is visible, the validators after logging in with their validation credentials will be able to rate the posts through the public keys. These ratings will again be interfaced with the backend systems through APIs. For Twitter ecosystem, metrics such as retweets or followers or likes can be used in the proposed algorithm as per the requirements. Thus, the goal of maintaining privacy and anonymity while mitigating the risk of fake news in the ecosystem can be achieved

VII. LIMITATIONS

Since our proposed solution leverages user feedback to detect fraudulent news on social media through blockchain, the model is susceptible to any security limitations the blockchain inherently possess, such a partitioning attacks [7], de-anonymization attacks [6], DDos[8] etc. Although the paper does not address these attacks, future researchers can provide enhancements on this system.

As we do not want validators to have access of the average rating, our model releases the rating only after the voting period ends or the max validator count is reached. This is to ensure that the validators will not be leaning towards a certain rating based on some available indicator. Although the latter is an attempt to speed up the voting process in certain instances, we still will not be able to provide the earliest of the audiences of the news with any indication over whether the news would be genuine of false.

Another drawback lies with deciding the priorities of the post. Our model determines this based on the account types being verified or non-verified. The model does review low priority posts with overwhelming audience reaction and treats it as high priority which addresses this situation partially. However, certain posts maybe high priority but are neither made from a verified account, nor have received much reaction. A case in this scenario could be the whistle-blowers that release posts from newly made accounts. In many cases such posts have been discovered late and similarly in our model it would not be recognized as an honest or dishonest post due to low priority. There may also arise an event where opinion pieces attract a big reaction and are then involved in the voting mechanism. Since it is an opinion piece it does not make any sense to include a true or false rating. This could be solved if the platform includes a tag that specifies the type of post. This solution from the platform's end will ensure that resources of validator is not wasted in voting over these posts and instead the voting mechanism is used on more pressing news articles.

Since our model has an upper bound for the max validator count it could be manipulated to vote in a certain way if prejudiced validators achieve that number which would exclude the late arriving honest validators from participating. This is highly unlikely as our model suggests a bound based on the total global validators but there still remains a slim possibility.

The model design presented only identifies replicas of news posted on the social media. It falls short in detecting posts of similar news, written in different tones. Also, providing incentives through cryptocurrencies does not guarantee that the voters would participate in all voting processes.

VIII. CONCLUSION AND FUTURE SCOPE

Although the limitations presented in the previous section, the proposed model will help tackle the fake news detection problem in any social media ecosystem. Leveraging anonymous user input for determining the authenticity of the news and providing incentives for accurate and honest feedback allows the model to maintain its integrity. Through our analysis we were able to successfully verify the results of our approaches.

We were also able to ensure that the voting mechanism is using resources judiciously by being utilized on high priority posts. The social media platform will therefore not be enforced to incentivize for every other post. Our model presents the provision where low priority posts can also be updated as high priority which adds a layer of fairness. Additionally, the max validator count equips our model with the possibility of faster verification of news.

Future advancement to this current proposed solution would be to incorporate Machine Learning Algorithms to identify duplicate posts which are written in a different tone or format and categorize it in a fashion that would prevent redundant voting for the same information which was already posted on our system. This could also save the organization from over-spending the incentives to the evaluators for validating the same newsfeed repeatedly

IX. REFERENCES

[1] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. Science, 359(6380):1094–1096, 2018.

[2] FaNDeR: Fake News Detection Model Using Media Reliability -IEEE Conference Publication. [Accessed 22 Mar. 2019].

[3] Adnan Qayyum, Junaid Qadir, Muhammad Umar Janjua, and Falak Sher. Using Blockchain to Rein in The New PostTruth World and Check the Spread of Fake News. Information Technology University (ITU), Lahore, Pakistan, 2019

[4] Shovon Paul, Jubair Islam Joy, Shaila Sarker, Amit Kumar Das, Sharif Ahmed, Abdullah - Al - Haris Shakib. Fake News Detection in Social Media using Blockchain. 7th International Conference on Smart Computing & Communications (ICSCC), 2019.

[5] E. Leka, L. Lamani, B. Selimi and E. Deçolli, "Design and Implementation of Smart Contract: A use case for geo-spatial data sharing," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 1565-1570

[6] G. Wondracek, T. Holz, E. Kirda and C. Kruegel, "A Practical Attack to De-anonymize Social Network Users," 2010 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, 2010, pp. 223-238.

[7] M. Saad, V. Cook, L. Nguyen, M. T. Thai and A. Mohaisen, "Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 1175-1187.

[8] Sajjad Ahmed, Knut Hinkelmann, Flavio Corradini, "Combining Machine Learning with Knowledge Engineering to detect Fake News in Social Networks-a survey" Department of Computer Science, University of Camerino, Italy; FHNW University of Applied Sciences and Arts Northwestern Switzerland Riggenbachstrasse 16, 4600 Olten, Switzerland.

[9] Kelly Stahl, "Fake news detection in social media" B.S. Candidate, Department of Mathematics and Department of Computer Sciences, California State University Stanislaus, University Circle, Turlock, CA 95382.

[10] Mohammed Torky, Emad Nabil, Wael Said, "Proof of Credibility: A blockchain Approach for Detecting and Blocking Fake news in social networks", International journal of Advanced Computer Science and Applications, Vol. 10, No.12,2019

[11] Leonardos S, Reijsbergen D, Piliouras G. Weighted Voting on the Blockchain: Improving Consensus in Proof of Stake Protocols. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Blockchain and Cryptocurrency (ICBC), 2019 IEEE International Conference on. May 2019:376-384. doi:10.1109/BLOC.2019.8751290.

[12] Wood, G. (2014) Ethereum: A Secure Decentralised Generalised Transaction Ledger, Ethereum Project Yellow Paper

[13] Hegedus P. Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts. 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Emerging Trends in Software Engineering for Blockchain (WETSEB), 2018 IEEE/ACM 1st International Workshop on, WETSEB. May 2018:35-39.