CS765 : ASSIGNMENT 3

A Decentralized Application for Detecting Fake News

Submitted On :

Sunday, April 14, 2024

Submitted By :

Swetha M (23M0756)

Chaitra Gurjar (23M0831)

Index

1 Sybil Attack

A malicious actor can launch several fictitious identities, nodes, or accounts in a peer-to-peer network as part of a Sybil assault. A typical Sybil attack aims to flood the network with a high number of controlled fictitious entities in order to obtain disproportionate influence or control over it. We intend to avoid the sybil attack by forcing voters to make a deposit. This is a requirement for casting a vote. For a sybil attack, the attackers would have to pay an extremely high cost to enter the voting pool. We also penalize money for every incorrect vote that is cast. This would discourage them from performing the attack.

2 Evaluation of Trustworthiness

The trustworthiness or the weight of a voter is determined by the number of correct votes until the current news item. We reduce the trustworthiness of the voter for every vote that is wrong - i.e. the voter voted contradictory to the consensus in the network. For example, if a voter predicts the correct answer for a news item with a probability of ‘p’, then their trustworthiness must converge to ‘p’ if the initial trustworthiness is set to one. For this, we use the quadratic decay formula depending on the previous value of trustworthiness.

$$T_{i+1} = \texttt{max(0, } T_i \texttt{ - (} T_i \texttt{ - (1 - wrong\_votes/total\_votes))}^{2}\texttt{*0.01)}$$

3 Weight of Votes

We consider weighted votes based on trustworthiness, instead of simply adding up all the votes cast. This keeps the malicious voters in check, as their trustworthiness decays to minimum based on the formula above. Hence, their votes are lightweight and do not contribute to the consensus. Voters vote the news as (0 or 1) and the consensus is calculated as :

```
total_votes = total_votes + trustworthiness
consensus = total_votes > sum_of_weights/2
```

4 Incentive for Correct Votes

We pay a certain amount to the voters who predict the correct output. However, we maintain the trustworthiness, without depending on the correctness of the vote. Hence honest voters participate and vote truthfully to the best of their ability.

5 Upload News

The news item is uploaded with three parameters : the news id, the category and the content. This is because some voters may have an expertise in some categories but not in the others. We maintain the trustworthiness of each category for each voter.

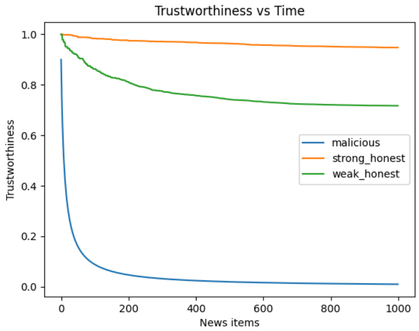
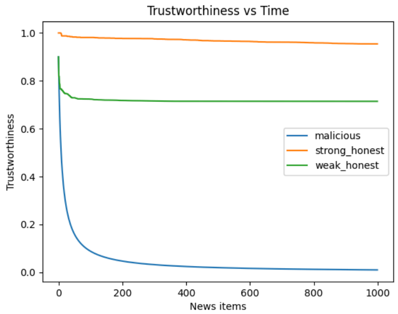
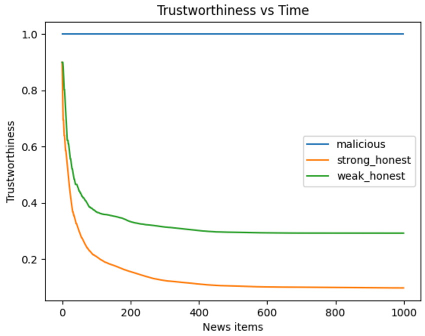
6 Bootstrapping

We initialize the trustworthiness of all voters (we set it as 1) to give them an equal chance.

7 Experimental Analysis

We change the parameters, N : number of voters, p : fraction of honest voters who vote correctly with a probability of 0.7, q : fraction of voters who are malicious.

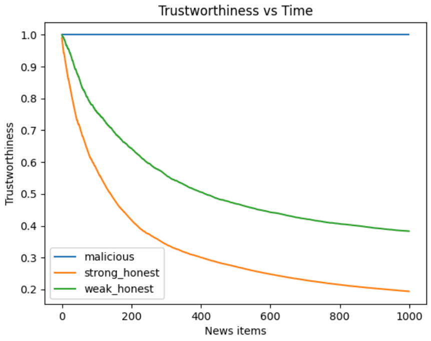
1. **Changing 'q' (N=500, p=0.5)**

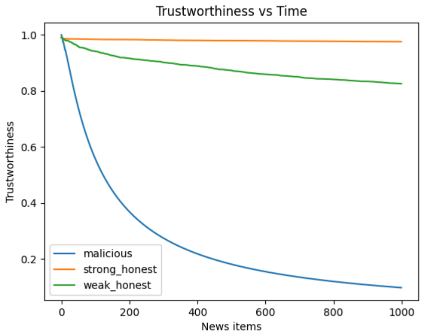
q	graph	wrong consensus
0.2		0
0.3		0
0.5		1000

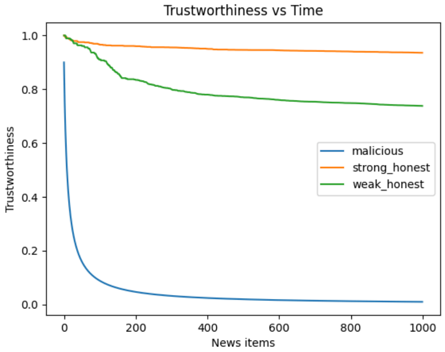
Insights :

- If the value of q is 0.5 or more, majority voting always gives the wrong consensus.
- Even in some cases of q being 0.4, we get the wrong consensus.
- But if we assume the number of malicious nodes is less than 40%, we always converge to the right trustworthiness score. After convergence, only the votes of honest nodes (>0.5) matter in the weighted vote.

2. Changing ‘p’ (N=500, q=0.4)

<table>
<tr><th>p</th><th>graph</th><th>wrong consensus</th></tr>
<tr><td>0.2</td><td>


</td><td>1000</td></tr>
<tr><td>0.7</td><td>


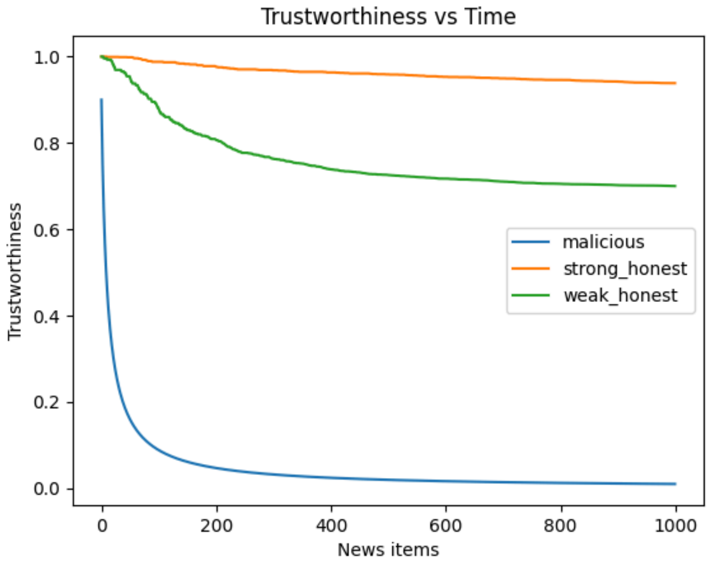
</td><td>5</td></tr>
<tr><td>0.9</td><td>


</td><td>0</td></tr>
</table>

Insights :

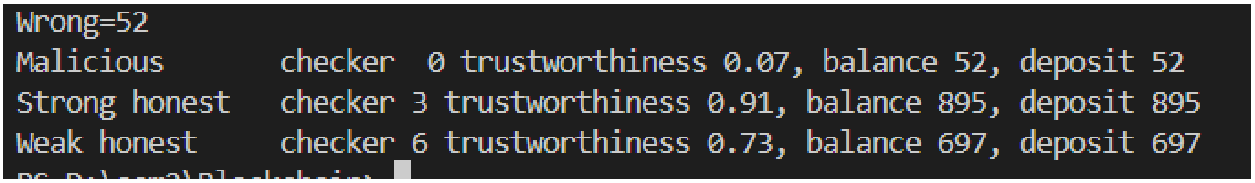
- Changing p doesn’t change the wrong consensus considerably in normal situations when malicious nodes are less.
- When malicious nodes are more, it is important for having more honest nodes which give the right answer (0.9 probability for right answer).
- When p is less for the same N,q, we might converge to wrong results.

3. Changing ‘N’ (p=0.5, q=0.33)

N	graph	wrong consensus
10		34
100		4
1000		0

Insights : Though minor, the number of wrong detections is less when N is large. When a large number of peers joined the network, the consensus is more stable.

8 Sample Output



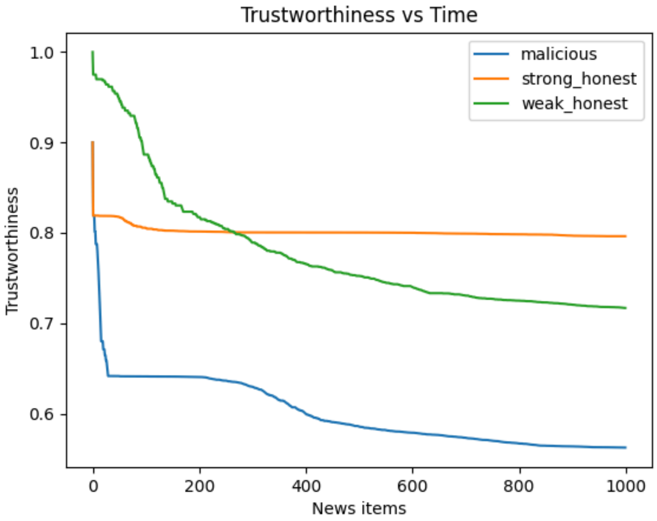
```
Wrong=52
Malicious       checker  0 trustworthiness 0.07, balance 52, deposit 52
Strong honest   checker 3 trustworthiness 0.91, balance 895, deposit 895
Weak honest     checker 6 trustworthiness 0.73, balance 697, deposit 697
```

Insights :

- Converged to right trustworthiness.
- Balance of honest nodes indicate the number of right votes they did.
- Deposits of malicious nodes are completely swept out.

9 Other Adversary Attacks

We assume here that the malicious voters always vote incorrectly. In another attack, the attacker may voter correct and wrong with a probability of 0.5. Even when malicious nodes are 80%, the 50% right nodes amongst them make our system stable and correct. The malicious nodes are able to save deposits and also gain some money periodically. Better than when malicious nodes vote wrong always.



****