Design Document and Report

Sarthak Sharma (23M0789)
Kumaran Kartikeyan (23M0803)
Pulkit Suhag (23M0782)

Brief Introduction:

We can start with a randomly connected voters network implemented as a real world peer to peer network. Initially there are certain numbers of voters, each having a trustworthiness array which is present with every other voter in the network as well. This array gives the trustworthiness of the voter in that particular news field. For ex, for a voter V:

[20, 85, 90] [Physics, Geopolitical, Sports]

Signifies that voter V has rating of 20 in physics, 85 in Geopolitical news and 90 in sports news. Higher the rating, higher is the trustworthiness of that voter.

The model works on the Byzantine rule that about 2/3rd of the voters are Honest for the proper working of the model. There is a leader election if a leader is down by the voters with initially a voter, where any 2 random voters can stand as candidates. Only those voters who have an overall rating of 200 can stand as candidates initially. And the other voters vote for them. Every voter has a random amount of coins.

The user who sends the news to verify along with the news category (Physics…) , Sends the news to at least 1/3rd+1 of the voters, who then sends the news to the leader and then the leader broadcasts it to the network. The voters can vote for 0 (fake) or 1 (true) or if they are malicious they can also not decide to send their vote. The leader checks the majority of the votes and decides whether the news is fake or not. The voters who gave the correct vote will get their rating as well as coins increased by 5 in that particular domain. Or if they gave the wrong vote their rating as well as coins decreased by 5.

After the voting the client is given a response by the voters and it needs to get a reply from at least 1/3rd +1 of the voters to be guaranteed that he got the right response.

1) **Sybil Attack:** In this type of attack an attacker creates more than 1 identity to affect the decision making.All the voters have a unique userID. There can be 2 possible solution to this attack, the first, we can create a dictionary present with everyone and the dictionary content would be of the form:

   {UserID: Aadhar Card number or other unique number}

   Which means the new voter has to verify their identity using any government procedure such as KYC, This is a foolproof solution. Another one would be that a new voter can join with the voting [0,0,0] in the network and his vote won't count until it is more than 25 in that particular field. Then only his vote will count. This procedure still has some flaws, that if the voter can initially give right votes but it might give wrong one later, one possible solution to this might be that if a voter gives 3 consecutive wrong votes then he can be penalised for some time and also with the coins.

2) **Method to evaluate or re-evaluate the trustworthiness of voters:** As mentioned the voters are rated from 0 to 99 in that particular news domain. After every news vote their ratings are modified. This is done by the leader broadcasting a dictionary to all the other users who update their dictionary matrix. The dictionary would be of the following format:

{UserID1: -5, UserID2: +5}

Thus all the users have the same dictionary matrix after every news voting of the example format:

{UserID: [20,85,90]}

3) More trustworthy voters should be given more weight: As mentioned every voter has a news domain array of the format [_,_,_]. Now the voter rating is updated according to his vote, as mentioned before. We can then apply weighted voting of the following format:

User Rating	Vote
0<=v<25	0
25<=v<50	1
50<=v<75	2
75<=v<100	3

The voting can't go less than 0. And the less rating voters are also incentivised to vote for reward as well as for them to become a trustworthy voter.

4) **Rational voters are to be incentivised:** it is simple to implement as when the dictionary is arrived at by the leader it also sends the user who gave right voting some amount of coins and can also penalise who gave wrong voting.

5) **Uploading a news item**: The user uploads a news item and sends the news article to at least 1/3rd +1 number of voters (Rule governed by the byzantine rule). At Least one honest node will send the news article to the leader and it will broadcast the news to the rest of the network.

6) Bootstrapping: The network needs to start with initial trustworthy voters for it to initially work properly. For the new user they start with a [0,0,0] rating and grows as such, initially their voting does not count, but it counts later.

Report regarding parameters n,p,q

n= number of voters
p= probability of being it more trustworthy
q=probability of it being malicious

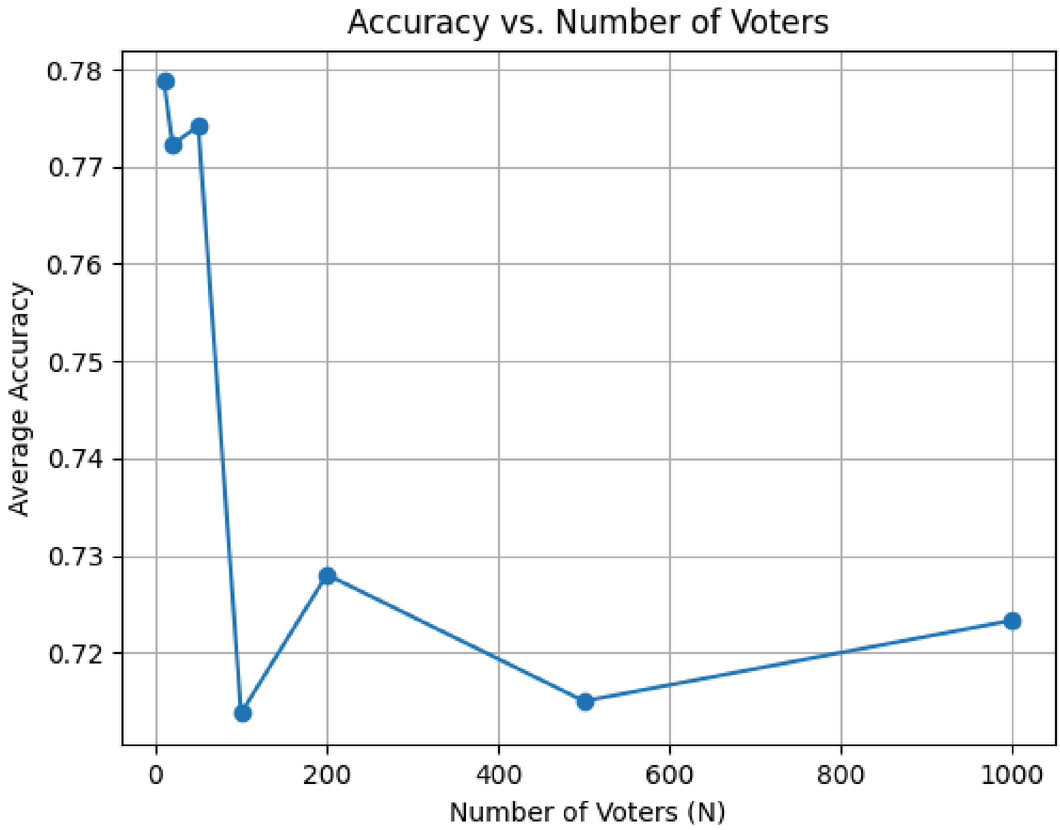
Introduction: We consider all the news to be true for the sake of calculating the accuracy with respect to n, p, q and plotted graph with respect to change in n, p and q. We update the trust and coins accordingly as in this report. We are giving some news and calculating the average of the news accuaracies. And accuracy vs n or accuracy vs p and accuracy vs q is plotted. We calculated the voting accuracy as:

correct_votes/(correct_votes+wrong_votes)

1) **Change with respect to n (voters) :**

Experimentation: We experimented with different values of n that are `10, 20, 50, 100, 200, 500, 1000`

We fixed our values of p and q to be 0.8 and 0.25 and observed the accuracy with respect to it as shown:

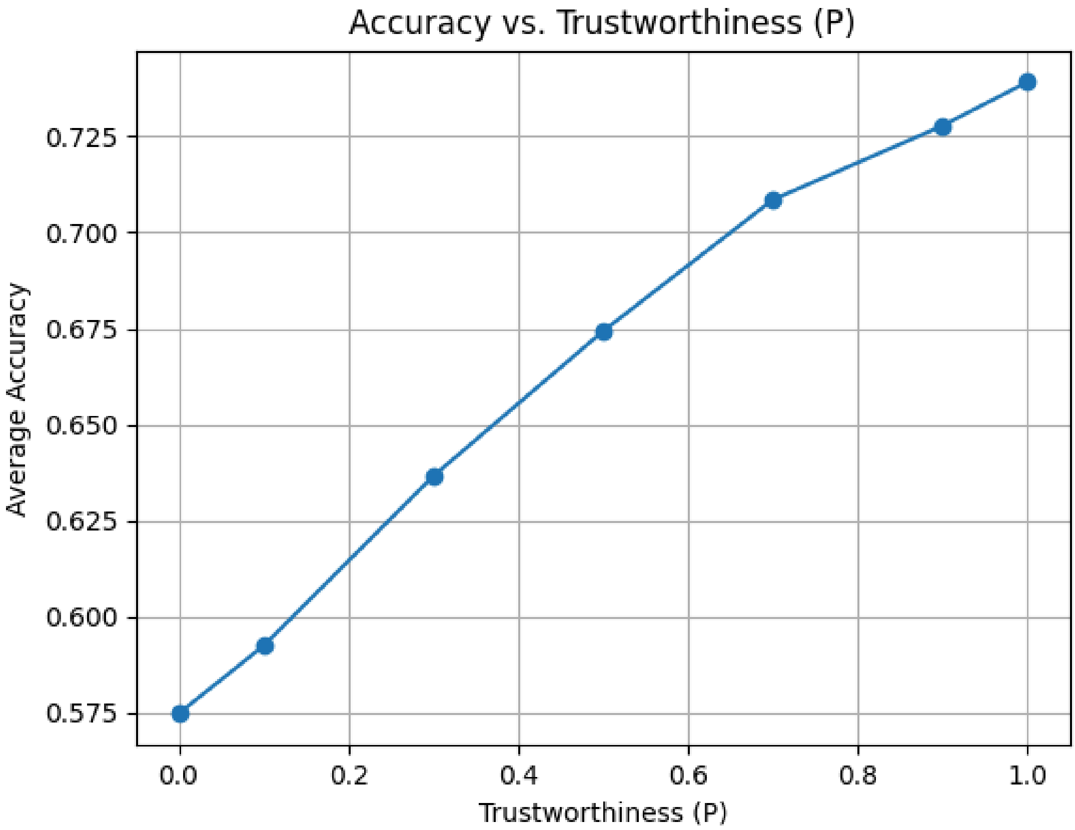


Observations and Explanation: We observed that accuracy doesn’t change drastically and comes out randomly each time we run with it being always between 0.8 and 0.7 in each value of n (voter) stating that the accuracy doesn’t change with respect to the number of voters.

2) **Change with respect to p (trustworthiness) :**

Experimentation: We experimented with different values of p that are: `0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0`

We fixated our values of q and n as 0.25 and 100 respectively, The results are shown below:

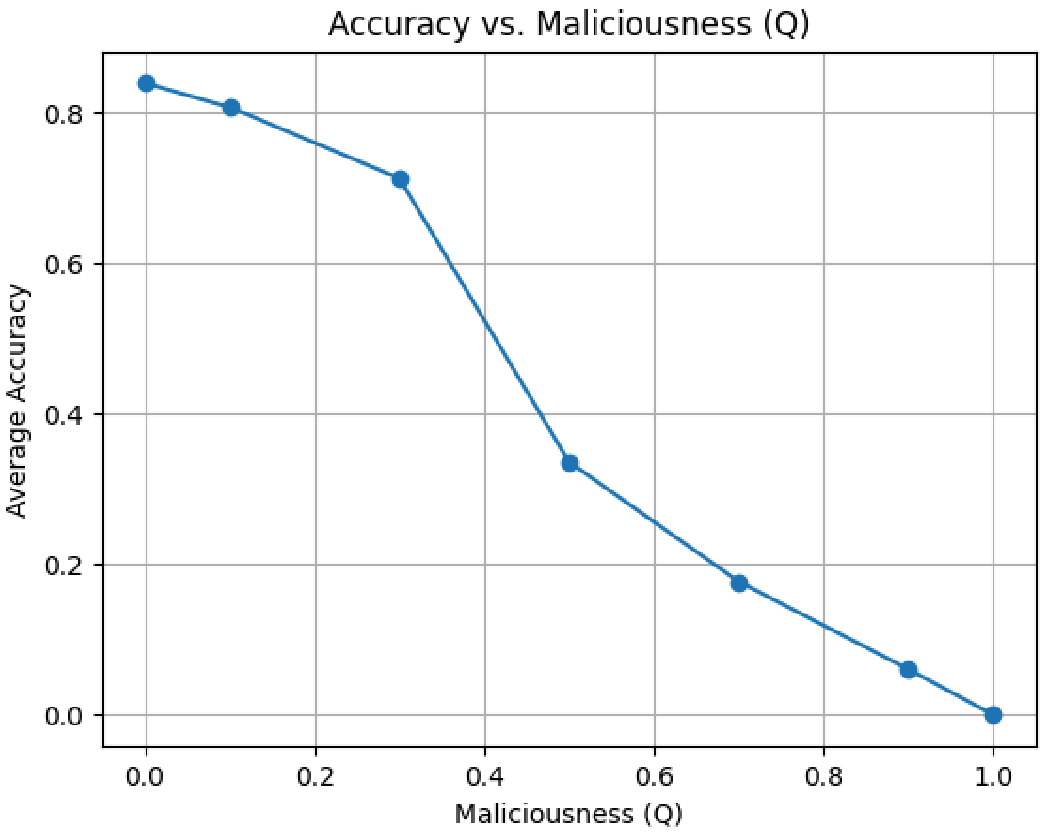


Observations and Explanation:: We see almost a linear growth in accuracy with some amount of randomness by it growing from 0.575 accuracy to 0.725 , As it is expected that if the trustworthiness is increased then the amount of voters giving the correct votes with a high probability also increases.

3) **Change with respect to q (maliciousness) :**

Experimentation: We experimented with different values of q that are `0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0`

We fixated our values of p and n to be 0.8 and 100 respectively. The results are shown below:



Observations and Explanation: We observed a linear decline with the increase in the value of maliciousness that is the percentage of malicious voters. It gradually decreased with some amount of randomness from high as 0.8 to low at 0 when all the voters are malicious.

This is clearly observable as if the number of malicious voters increases the number of wrong votes also increases reducing the accuracy.

Thank You