**Fake – News DApp**

**Assignment-3 : CS765**

Venkatesh K S - *210050164*

Veeresh B Patil - *210050163*

Vinu Rakav S – *210050165*

**AN OVERVIEW OF OUR ALGORITHM**-

* Anybody can become a voter in the DApp, by providing valid identity proofs at the start (e.g. government ID, biometric proof, phone number, email-id, verified social accounts, etc.)
* No two accounts can have the same identity proofs, the verification of validity of the provided proof is done off the chain
* Once verified, the voter is registered and allowed to vote on any news
* At the beginning, a voter enters a probation period for a while, where they vote with reduced weightages, not affecting the outcome significantly
* After the probation period, if the voter has gained enough trust, they become full-time voters
* To cast a vote, the voters must deposit some initial amount as a safe-trust
* After the voting period ends, the weighted score of the news article is calculated, and based on this result, the voters who voted correctly are rewarded, whereas the voters who voted incorrectly are punished
* The incorrect voters do not get their deposits back
* Whereas the correct voters get their deposits back, and additionally, the deposits of the incorrect voters are distributed among the correct voters proportional to their reputation scores
* We have utilized 2 kinds of weightage scores, one is for the weightage in the voting (weights), and the other is for the weightage in reward collection (reputation score), both capturing different essences of voting patterns.
* We also have implemented a **periodic reputation decay** for the following reasons --
  + *Avoid Stagnation*: Without decay, initial reputation scores might remain unchanged indefinitely.
  + *Adaptability*: Participants’ behaviour can change over time. Decay ensures that outdated reputations are adjusted.
  + *Encourage Active Participation*: Fact-checkers who consistently contribute should maintain their reputation, while inactive ones see a decline.

**HANDLING VARIOUS ISSUES-**

1. *Sybil Attack:*

We have introduced multiple layers of security to prevent Sybil Attack-

* *Personal identification at the time of registering:* We can implement various personal identity matching mechanisms. This identity verification process can be implemented off-chain with appropriate resources. Even though this almost guarantees against the Sybil attack, it might be slightly more expensive to implement
* *Probation period:* During the probation period, the voting patterns of the fact-checkers are observed carefully and reputation score adjusted accordingly. If the fact-checker demonstrates honesty, consistency, and expertise during this period, their reputation increases, and they gain full access to voting and other privileges. If not, they are terminated from the DApp.
* *Starting with a low reputation score:* The fact checkers start off with a weight of 0.5 but a low reputation score of 1, hence de-motivating the sybil attack.
* *Initial deposit for voting:* Enforcing an initial deposit to be eligible to vote, punishing for incorrect voting, and poorly incentivising for correct behaviour during probation period plays an important role almost in mitigating the sybil attack.

1. *Method to evaluate or re-evaluate the trustworthiness of voters:*

We experimented with 2 different ways of updating trustworthiness –

* Method 1: this captures the all-time voting accuracy of the fact-checkers

**Score = (#correct) / (#correct + #incorrect)**

* Method 2: This emphasizes on consistent voting behaviour in the recent past

**New\_Score = Old\_score \* c + current\_vote\_score \* (1-c)**

Here, c is hyperparameter which can be altered to give different weightages to the recent past, and the past past, as is suitable.

This score is used to weigh the different votes of the fact-checkers.

We also have a reputation score which increases for every correct vote, and decreases for every wrong vote – this score is used for weighted distribution of rewards. It is calculated as –

New\_score = max(0, old\_score-0.5) -- for wrong votes

New\_score = min(100, old\_score+0.5) -- for correct votes

1. *The opinions of more trustworthy voters should be given more weight:*

To achieve this, the fakeness\_score of a news article is calculated as the weighted average of the votes of all fact\_checkers, where weight represents the trustworthiness of the voters. We also have independent weights for different news categories, so that they can build trustworthiness in different categories independently, and for every news, the appropriate experts are majorly considered.

The reputation score, on the other hand, is common over all the categories for a given voter. This makes sure that people do not cast their votes on topics that they know not of, and adversely reduce their overall reputation score.

1. *Rational voters are to be incentivised:*

Rational voters can accumulate higher reputation scores, through repeated accurate voting. Higher reputation score implies higher weightage in the reward collection.

1. *Uploading a news item:*

We can store news on the blockchain, or on some other external media. We can have a corresponding front-end for the DApp, where news articles can be written, and can be identified by article numbers on the website.

Or, we can do without a front-end as well, where users can read any news stored with the corresponding article no.

To make our DApp more flexible, we can accept URL of any news on any website as an identifier. But appropriate measures must be taken to check for the maliciousness of uploaded URL, as every other fact\_checkers must visit the website to read the news and vote on it.

1. *Bootstrapping:*

To handle the issue of lack of initial lack of trustworthy ratings, we can start with a small set of trusted fact-checkers (e.g., experts, journalists). And we can gradually expand the pool as new fact-checkers prove their reliability, until a point where the DApp can truly run in a de-centralized fashion.

**EXPERIMENTS & OBSERVATIONS**

We have the following equations which determine the effect of malicious behaviour,

based on the values of p and q:

q -> fraction of malicious voters

p -> fraction of honest voters with accuracy of 0.9

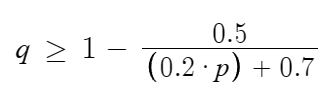
1-p -> fraction of honest voters with accuracy of 0.7

To calculate the fluctuation point, we have the following equation on an average,

during the longer run (assuming all news are real) –

q\*0 + (1-q) \* [p\*0.9 + (1-p)\*0.7] = 0.5

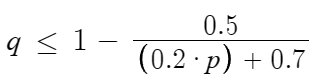
This leads to the following constraints ---

**CASE 1:** 

=> malicious miners dominate, and all news are incorrectly labeled.

This causes the reputation scores to converge to (1, 0.1, 0.3) {in the order defined above}

– a complete opposite of their honesty

**CASE 2:**  

=> malicious miners do not dominate, and all news are mosty labeled correctly.

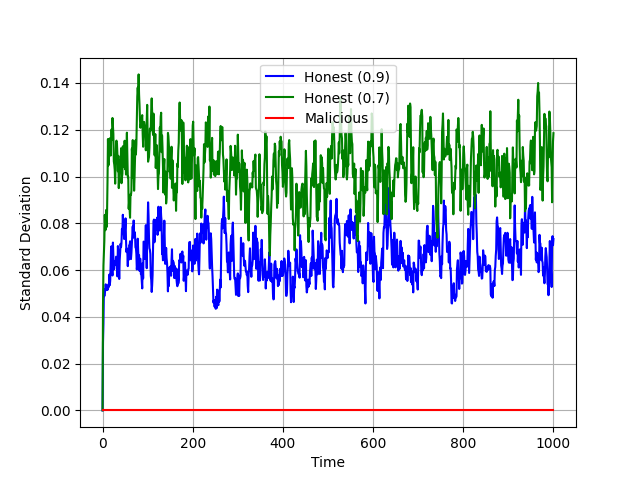
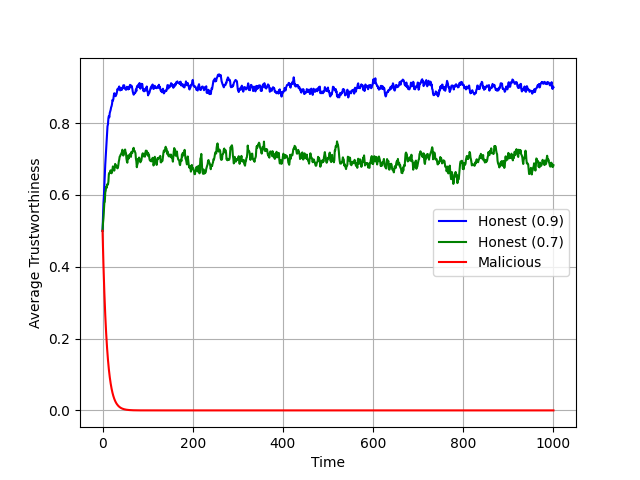
This causes the reputation scores to converge to (0, 0.9, 0.7) {in the order defined above}

– as expected, in the ratio of their honesty

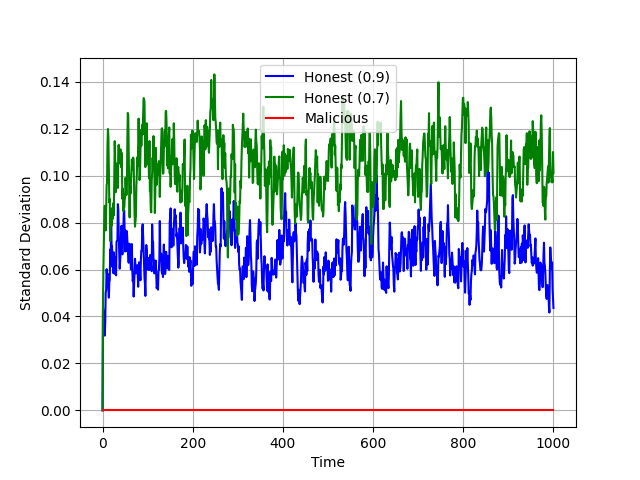
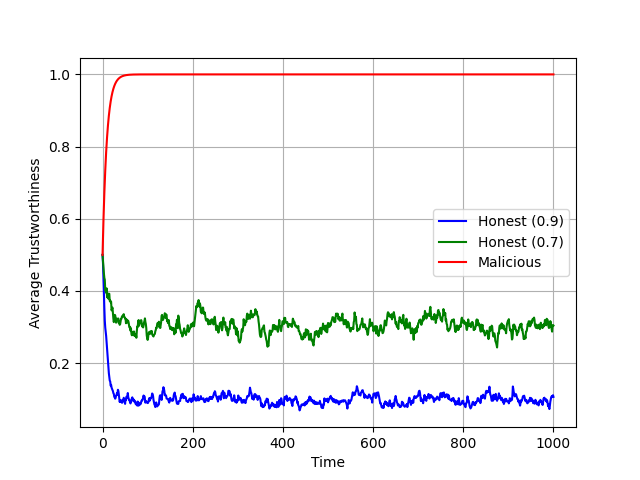
This claim can also be shows experminentally –

We shall show the experiments done for both the varities of weighting systems as shown earlier, {and for the fluctuation point of (p,q) = (0.5, 0.375)}

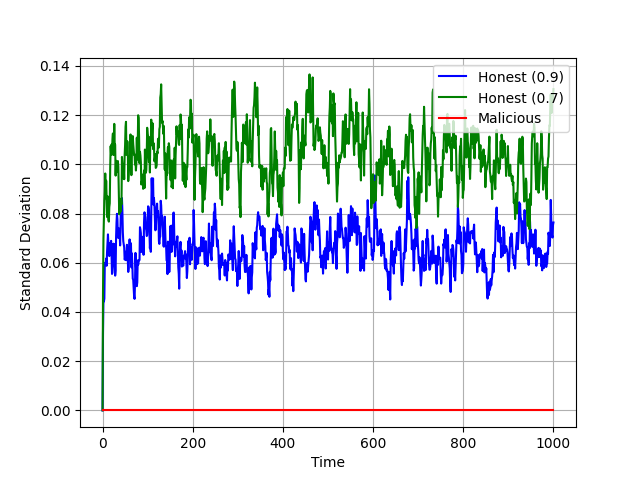
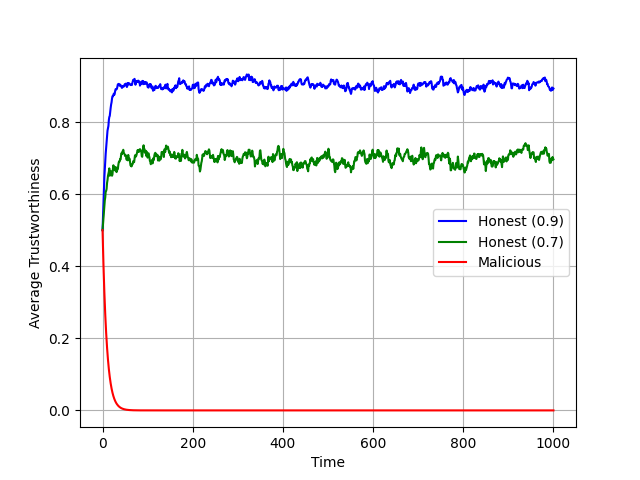
* **Method -1:**
  + Case1 : p=0.5, q=0.37 (safe side of fluctuation)



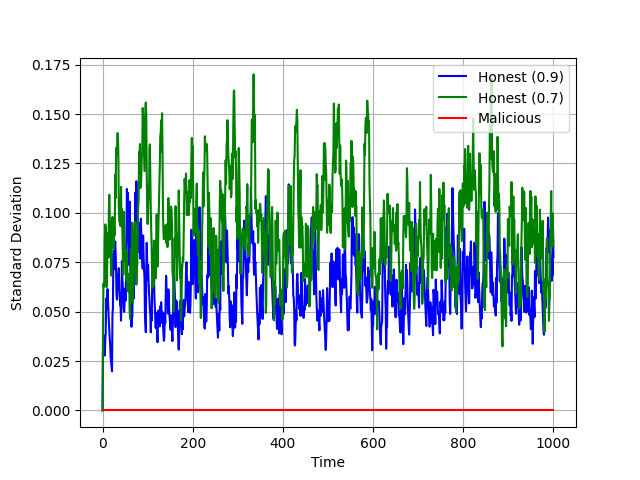
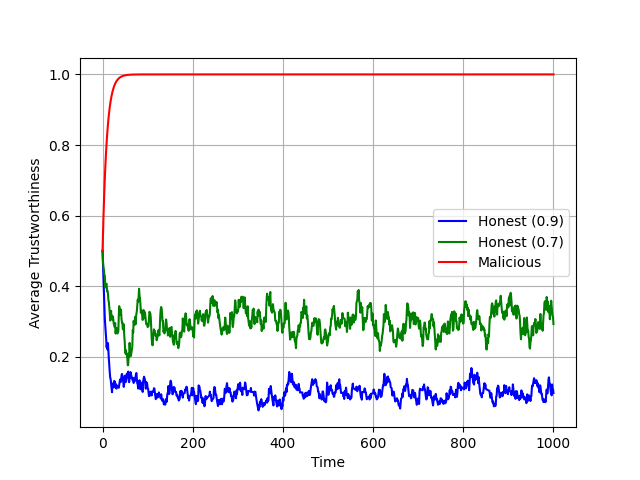
* + Case2 : p=0.5, q=0.38 (unsafe side of fluctuation)



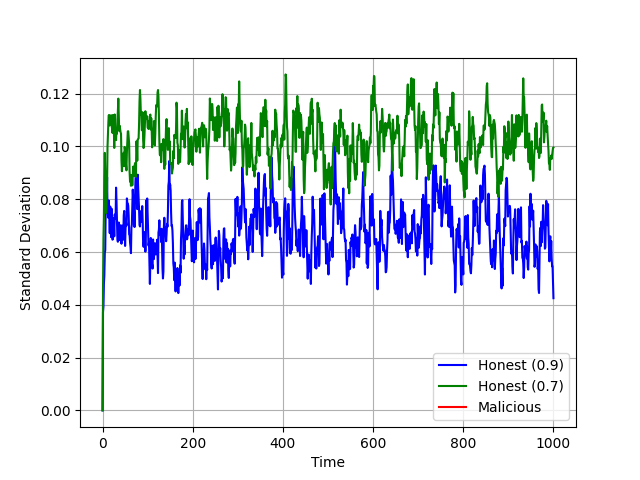
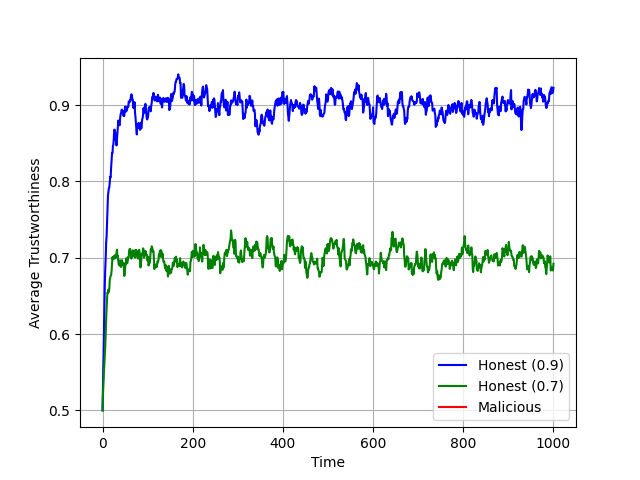
* + Case3 : p=0.5, q=0.1 (negligible maliciousness)



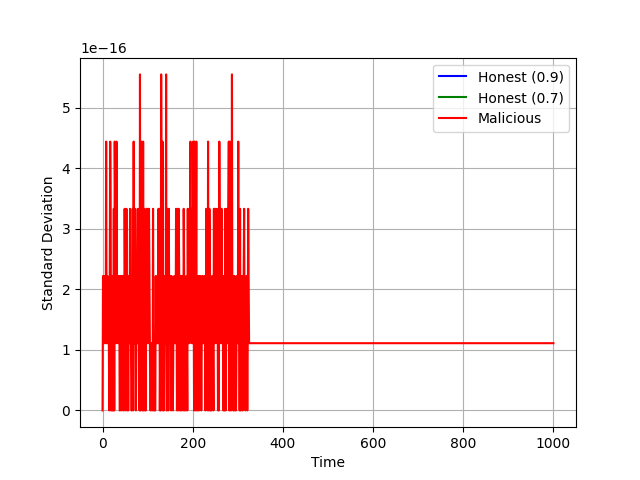
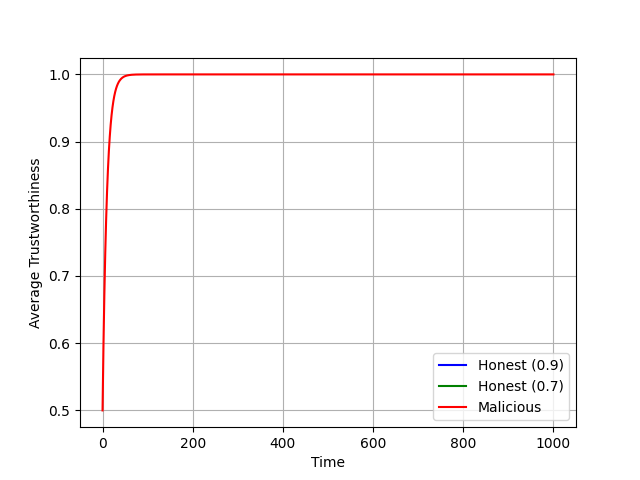
* + Case4 : p=0.5, q=0.8 (dominant maliciousness)



* + Case5 : p=0.3, q=0.0 (no maliciousness)



* + Case6 : p=0.3, q=1.0 (full malicious)

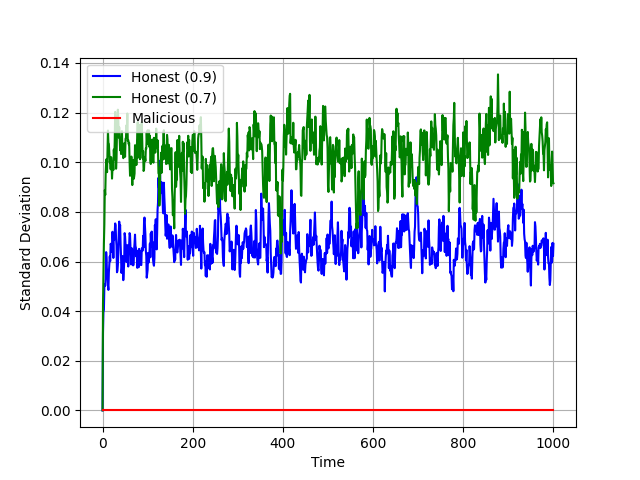
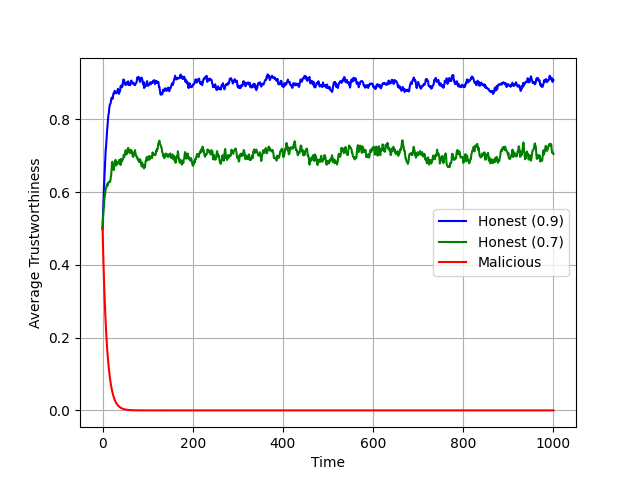


* **Method -2:**

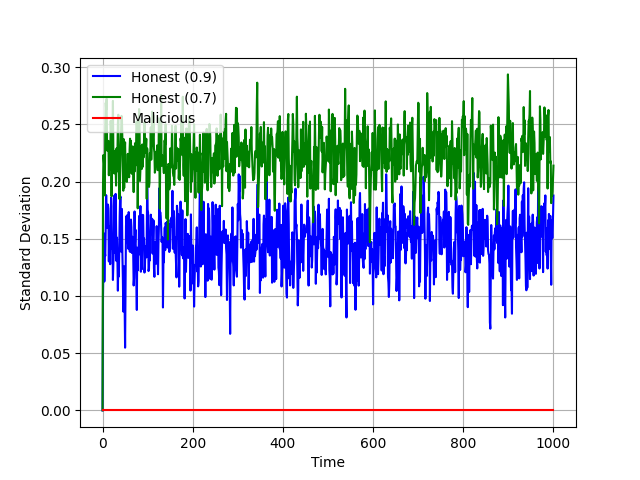
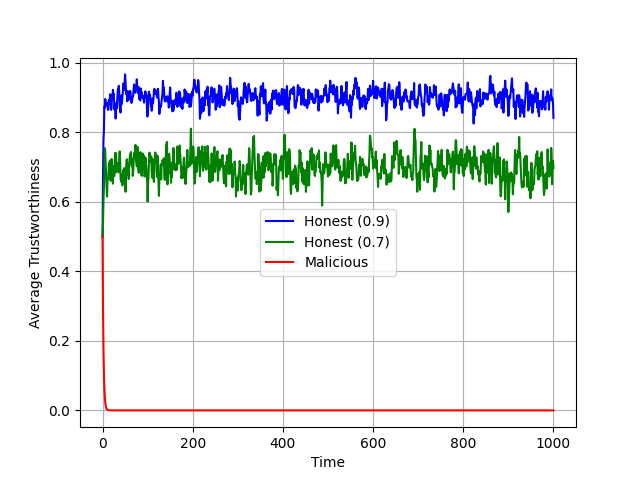
In this method too, we observed similar trends of convergence

Hence, we won’t discuss that:  
But, we have a more interesting study on the hyperparameter x --

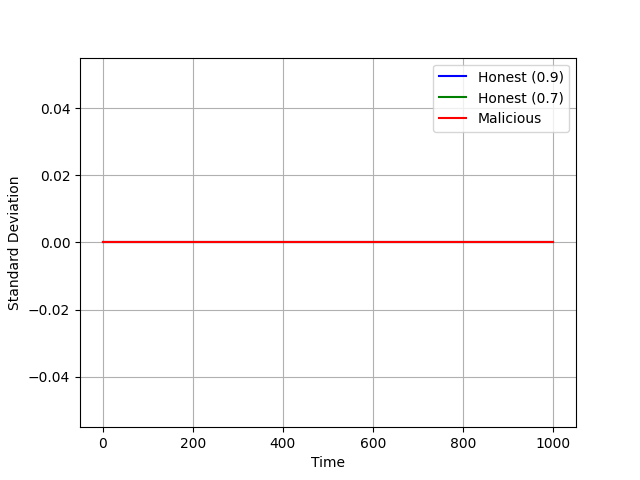
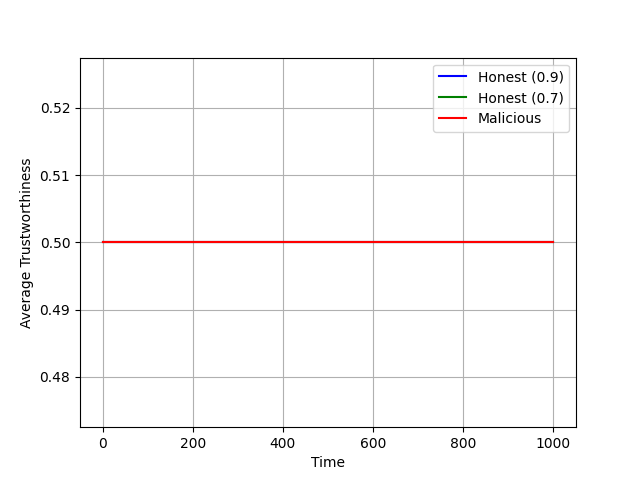
* + Case1 : x = 0.9, p = 0.5, q = 0.1



* + Case2 : x = 0.6, p = 0.5, q = 0.1



* + Case3 : x = 1.0, p = 0.5, q = 0.1



We can obersver that increasing the value of ‘x’, decreases the standard deviation in the convergence across time.

This is because ‘x’ is the weight of the long run previous reputation score, which takes into account all previous votings, and thus is a lot more stable.

Whereas, ‘1-x’ is the weight of the current vote which can be very unstable.

Thus, increasing the value of ‘x’ helps capture the longer run consistency of the voter. But, the value cannot be too large that it neglects the newest voting patterns. Finding the exact balance between how much weight to give consistency and how much weight to give latest behaviour, is a matter of design choice, and practically determinable.