TRUST MODELS IN BITCOIN NETWORK

An Interim Project Report

Submitted by

NANDITHA MENON [cb.en.u4cse17043]
SAI SRIHITHA REDDY THUMMALA [cb.en.u4cse17065]
DEVINENI SATYA VANCITA [cb.en.u4cse17220]
SNEHA NANDIGAM [cb.en.u4cse17258]

Under the guidance of

Ms. Vidhya S.

(Assistant Professor (Sr. Gr), Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

Review #3 – November 2020

ABSTRACT

The cryptocurrency was introduced as a way to provide people with the power of controlling their money without having to depend on companies, banks or government institutions. Economic instability and the advent of technology has further increased the need for and utilization of Cryptocurrency. But high volatility, e-wallet thefts and anonymous criminal activities funding make it necessary to have a trust management model.

“Trust” can be seen as a rational form of cooperation under behavioural risk. A trust model is a practical solution to mitigate the risk involved in transacting with anonymous strangers in the bitcoin network. The trustworthiness of a node in the network is evaluated to identify threats to the network. This project aims to compare the existing cryptocurrency trust management models and to implement these models on a signed bitcoin dataset and attempt to develop a hybrid model that derives from the existing model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

KNN	K Nearest Neighbours
DB	Davies Bouldin
CH	Calinski Harabasz
OTC	Over-the-counter

LIST OF SYMBOLS

$g(u)$	Goodness of node u
$f(v)$	Fairness of node v
$w(u, v)$	Weight of the edge between node u and node v
$in(u)$	All the nodes which have incoming edges to u
$out(v)$	All the nodes which have outgoing edges to v
α_i	Number of interactions with outcome i
$\beta_{\bar{i}}$	Number of interactions with outcome $\bar{i}$
G	Shipping goods
L	Shipping lower quality goods
C	Not shipping any goods
$\bar{G}, \bar{L}, \bar{C}$	Represent the converse to G,L,C
$u_{G_iG_j}$	Utility assigned by nodes in group j to nodes in group i
$c_{G_iG_j}$	Cost assigned by nodes in group j to nodes in group i
$T^{reference}_{rG_iG_j}$	Minimum trust value along the most trustworthy reference path
N	No. of paths between agents P and Q
D_i	No. of steps between P and Q on the i^th path
m_i	Q's immediate friend or neighbor on the i^th path. (M = set of Q's friends)
c_ki	Local trust value from peer k towards peer i
$t + k$	Global trust value peer k

Chapter 1

INTRODUCTION

A cryptocurrency is a digital or virtual currency that is secured by cryptography. Bitcoin is the largest and the first block-chain based cryptocurrency by market capacity. Invented in 2008 by Satoshi Nakamoto, this cryptocurrency is now at Rs.7,48,180 per Bitcoin.

The transactions made in the Bitcoin network are irreversible, fast and global. Bitcoin network allows the users to remain anonymous giving the people who do not want to be associated with the transactions a huge advantage. Bitcoin transactions are permissionless even for the ones which take place internationally. They have no third party involved. The transactions directly take place between the two parties.

But Cryptocurrency is still not widely accepted and is illegal to trade in many countries. Financial crimes such as money laundering, cyber theft and illicit transactions have been on the rise. A scheme which evaluates and identifies possible threats to the system based on the trustworthiness is necessary. This project aims to study, implement, compare and design trust management models for a Bitcoin Network.

1.1 Problem Definition

Contemporary cryptocurrencies lack legal, monetary and institutional backing. They provide trust through technology. But bitcoin does not have the status of a legal tender. Nobody is obliged to exchange them for any money or goods.

As such, there is a high chance of fraudulent transactions. A trust model which calculates the trust score of each node can be beneficial and valuable. It can help decrease the number of risky transactions by alerting the user about malicious nodes when participating in a transaction with them.

1.1.1 Obtaining relevant models from different fields:

Evaluate whether the models obtained are relevant and feasible to be applied on the Bitcoin Network.

1.1.2 Implementing the relevant models:

Implementing the models after adjusting the implementations to match the Bitcoin Network data.

1.1.3 Comparing the implemented models with a common metric:

Deciding and then using common metrics to evaluate performance of the models.

1.1.4 Attempting to design and develop a hybrid trust model from the existing models:

Studying the performance of all the models and identifying the most commonly occurring trust factors which are then incorporated into a hybrid model.

1.1.5 Determining the accuracy of the newly designed trust model:

Calculating the performance of the new model and examining the difference in performance when compared to the pre-existing models.

In the following sections, Trust Models from various domains such as Cloud Computing, E-Commerce and Cryptocurrency have been studied. Various applications in these fields have also been studied to find a list of factors that could possibly affect trustworthiness in a bitcoin network.

Chapter 2

LITERATURE SURVEY

2.1 Trust between people

Trust between people is known to be directly proportional to Credibility, Reliability, Intimacy and Inversely proportional to Self-Orientation. Here parallels can be drawn to establish a pattern between various trust factors. It can be concluded that Ratings (Credibility), Total interactions with other nodes(Reliability) and the Total number of interactions with the same node(Intimacy) are all directly proportional to Trustworthiness. It can also be derived that trust of a known malicious node contributes less towards the trust score than the trust derived from a non-malicious node.

2.2 Cryptocurrency Trust Models

Awareness, Legislation, Influence, Availability, Anonymity, Price Volatility , Security are the factors affecting trust in Cryptocurrencies.

There have already been a few implementations of trust models in the Bitcoin network.

2.2.1 Based on blacklisting malicious nodes

1. Creation of blacklist based on historical thefts on bitcoin exchanges
2. Classification task to differentiate honest users from a list of potentially criminal entities
3. Based on the classification, a metric in the form of a risk score is developed to indicate a user's degree of involvement in malicious transactions. The risk score and a revised reputation score are used for this purpose

2.2.2 Based on fairness and goodness of a node

1. After each transaction between A and B , they provide feedback scores on how trustworthy and reliable the other party is.
2. This score is then used to calculate the fairness and goodness of a node based on the below formulas

$$g(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f(u)w(u,v)$$

$$f(u) = 1 - \frac{1}{|out(u)|} \sum_{v \in out(u)} \frac{|w(u,v) - g(v)|}{2}$$

Figure 2.1: Fairness and Goodness of a node

3. Goodness of a node denotes how trustworthy it is and the fairness of a node denotes how honest a node is in their feedback.

2.2.3 Based on time and 1 sigmoid function

The time is taken into account because it is necessary to keep track of the natural changes of every person's evaluation criteria over time by considering the time intervals and frequency of transactions. 1 sigmoid function allows to normalise and set bounds on the scattered values and to transform them into values representing customers propensity. Values are stored in an individual's private space generated by the block chain. New trust values are calculated every time a new feedback is given.

1. Four key metrics: time difference weight (TDW), personal evaluation criteria (PEC), normalized evaluation(NE), and average received normalized evaluation (ARNE)

2. First, the model checks if there are any transactions between user A and B. If so, direct trust will be calculated. Otherwise, it calculates indirect trust between them via C who has done transactions with both A and B, if any (See Figure).

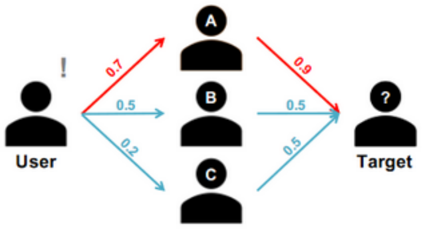


Figure 2.2: Indirect Trust

3. For every update of transactions, it checks again the existence of transactions between the two. If true,indirect trust is replaced by newly calculated direct trust.

2.3 E-Commerce Trust Metrics & Models

From a comprehensive study on the trust Models in E-Commerce ,it has been found that Transaction Cost,Transaction History, Indemnity, Spending Patterns, System Usage, location are all factors affecting the trustworthiness of a node.
It has been found that instead of using trust as a function of credit history, using trust variables could prove to be a better approach

2.4 Cloud Computing Trust Models

2.4.1 Cuboid Trust

This model builds four relations among the three trust factors which include contribution, trustworthiness and quality of resources. It then applies a power iteration of these to compute the global trust value of each peer. This model performs well even in the case of the presence of malicious nodes in the peers.

2.4.2 Eigen Trust

This model is used to evaluate the satisfactoriness of downloads from each peer. It states that S(i,j)=sat(i,j)-unsat(i,j) where S(i,j) is the satisfactoriness of downloads and sat(i,j) is the number of satisfactory downloads and unsat(i,j) is the number of unsatisfactory downloads.This equation can be used to establish the type of relation between Trust score and the trust factors in Blockchain network.

2.4.3 Bayesian Network Based Trust Management(BNBTM)

This model uses Beta probability distribution functions and number of interactions and shipping of goods in an e-commerce application. This equation can also be adjusted to suit the Bitcoin network.
The trust value of peer i is given by

$$\tau_i = \frac{\alpha_i}{\alpha_i + \beta_{\bar{i}}}, (i \in \{G, L, C\}, \bar{\imath} \in \{\bar{G}, \bar{L}, \bar{C}\})$$

Figure 2.3: Trust value of a node i

Where,

$\alpha_i = r_i + 1$

$\beta_{\bar{i}} = r_{\bar{i}} + 1$

2.4.4 GroupRep Model

This model establishes a relationship between Trust Relationships, utility and cost. Trust between any 2 entities is dependent on 3 types of trust : Trust between the 2 nodes, the trust between the two groups to which the entities belong to and the trust between an entity and the group it interacts with.

$$T_{r\,G_iG_j} = \begin{cases} \dfrac{u_{G_iG_j} - c_{G_iG_j}}{u_{G_iG_j} + c_{G_iG_j}} & if\ u_{G_iG_j} + c_{G_iG_j} \neq 0 \\ T_{rG_iG_j}^{reference} & if\ u_{G_iG_j} + c_{G_iG_j} = 0\ and\ \exists\ Trust_{G_iG_j}^{path} \\ T_{r\,G_iG_{strange}} & otherwise \end{cases} \tag{2}$$

Figure 2.4: Trust between Group i and Group j

Where,

$u_{G_iG_j} \geq 0$ and $c_{G_iG_j} \geq 0$

2.4.5 AntRepModel

This model is similar to the Distance Vector Routing algorithm. The Trust Score of each node is evaluated and this information is then communicated to all the peers of the node.Here each node maintains a reputation table and it is updated regularly with the help of forward and backward ants.

2.4.6 Semantic-Web based Model

This model calculates trust by using the number of paths between two nodes and their reliability as factors.
The weight of the path $i(w_i)$ is calculated using:

$$w_i = \frac{\frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}}$$

Figure 2.5: Weight of path i

This gives a higher weight to shorted paths. If agent P and agent Q are friends then P $\rightarrow$ Q, or neighbours then P $\leftrightarrow$ Q then P's trust in Q can be computed directly. Otherwise,

$$T_{P\rightarrow Q} = \sum_{i=1}^{N} \frac{T_{m_i\rightarrow Q} \times \prod_{i\rightarrow j \cup i\leftrightarrow j} R_{i\rightarrow j} \times \frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}}$$

$$= \sum_{i=1}^{N} T_{m_i\rightarrow Q} \times \prod_{i\rightarrow j \cup i\leftrightarrow j} R_{i\rightarrow j} \times w_i$$

Figure 2.6: Trust score of agent Q for agent P

Since, the paths in the case of Bitcoin network are transactions, this model cannot be used directly.But the type of relations between various trust factors and the trust score can still be derived from the equation.

2.4.7 Global Trust

This model acknowledges that trust can further be classified as local trust and global trust.The global trust of a node is calculated as the weighted average of the local trust between two nodes and the number of transactions between those.
The global trust value for node i, t_i is defined as:

$$t_i = \sum_{k} c_{ki} t_k$$

Figure 2.7: Global trust value of node i

2.4.8 Peer Trust

The Peer Trust model computes the trustworthiness of nodes by using the feedback received from peers ,total number of transactions the peer performs,credibility of feedback sources, transaction context factor and community context factor.

2.4.9 Time-Based Dynamic Trust Model

The Time-Based Dynamic Trust Model considers the effect of dilution of trust with time and includes the dilution factor in its equation.
Trust pheromone between nodes i and j at time (t+1) is defined as:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \sigma\tau_{ij}(t)$$

Figure 2.8: Trust pheromone between nodes i,j at time (t+1)

where ρ is the trust dilution fator and $\sigma\tau_{ij}(t)$ is the additional intensity at each inter-operation between entities.
$\sigma\tau_{ij}(t)$ is defined as:

$$\sigma\tau_{ij} = \begin{cases} \dfrac{1}{\frac{1}{1-\tau_{ij}(t)}+1} & \text{if } i \text{ and } j \text{ interact at time } t \\ 0 & \textit{otherwise} \end{cases}$$

Figure 2.9: Additional intensity at each inter-operation

If trust value $p_{ij}(t)$ between nodes i and j at time t is greater than a certain threshold R, they can validate each other's certificate, otherwise not.

2.5 Summary

Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. And this can be done by calculating trust score for each user.

Calculating trust score for each node can be achieved in various ways through different trust models. The literature survey led to the classification of the models into 3 different categories.

1. Cryptocurrency trust models
2. E-commerce trust models
3. Cloud Computing trust models

Awareness, Legislation, Influence, Availability, Anonymity, Price Volatility , Security are found to be the factors affecting trust in Cryptocurrencies. From the e-commerce data, it has been observed that Transaction Cost,Transaction History, Indemnity, Spending Patterns, System Usage, location are all factors affecting trust.

From the above mentioned observations,the trust score is speculated to be directly proportional to the number of transactions a node is involved in,the number of repeated transactions with a particular peer node, the reputation/reviews of a node and inversely proportional to the number of transactions with known risky and fraudulent nodes.

Cryptocurrency trust models mainly speak about calculating the trust score based on the feedback provided from the nodes which participate in a transaction.

1. Fairness and goodness model
   Goodness of a node denotes how trustworthy it is and the fairness of a node denotes how honest a node is in their feedback. Both are calculated based on the feedback score which ranges from -10 to +10.
2. Time and 1-sigmoid function model
   Two types of trust scores - direct and indirect are calculated. Direct trust score is calculated when there is a direct transaction between the nodes. If there is no direct transaction, the indirect trust score is calculated via another node which has done transactions with both the nodes.

From Cloud Computing Trust Literature survey,the following models were found to be useful:

1. Cuboid Trust
   It computes trust score by using power iteration on the four relations between the 3 factors: quality, node and the peer.

2. GroupRep
   It computes trust as a combination of the trust between groups,trust between nodes and the trust between node and group.

3. AntRep
   Each peer stores a reputation table which are continuously updated with the help of forward ants and backward ants.

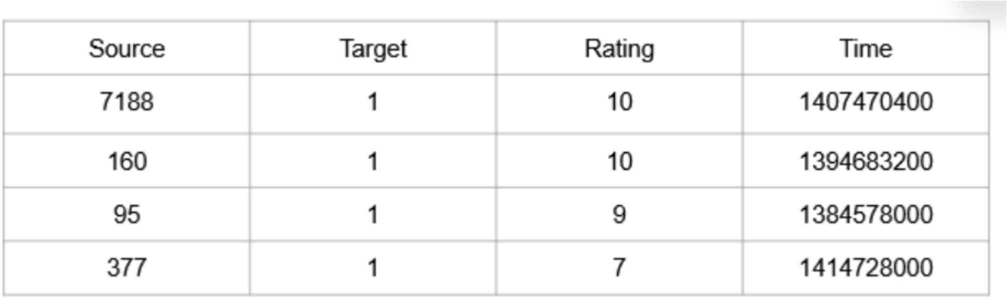
4. Global Trust
   Global Trust is computed as a weighted average of the local trust between two nodes and the number of interactions between them

5. Time Based Dynamic Trust Model(TBDTM)
   This model takes into account the dilution of trust with time.It also considers the additional intensity at each inter-operation between entities as a factor to compute the Trust Score.

All these factors and models can hence be used in the study of Trust in Bitcoin Networks.

2.6 Data Set

1. http://snap.stanford.edu/data/soc-sign-bitcoinotc.html

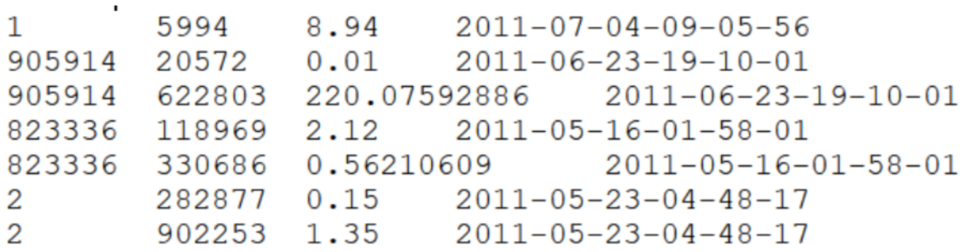


Source	Target	Rating	Time
7188	1	10	1407470400
160	1	10	1394683200
95	1	9	1384578000
377	1	7	1414728000

Table 2.1: Bitcoin OTC dataset

Members of Bitcoin OTC rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1.The signed attribute which signifies the Trustworthiness of a node can be used to implement several networking/machine learning models.

2. http://anonymity-in-bitcoin.blogspot.com/2011/09/code-datasets-and-spsn11.html



```
1       5994    8.94    2011-07-04-09-05-56
905914  20572   0.01    2011-06-23-19-10-01
905914  622803  220.07592886    2011-06-23-19-10-01
823336  118969  2.12    2011-05-16-01-58-01
823336  330686  0.56210609      2011-05-16-01-58-01
2       282877  0.15    2011-05-23-04-48-17
2       902253  1.35    2011-05-23-04-48-17
```

Table 2.2: User Network Edges dataset

The user network represents the transactions between two edges. The datset is in human readable form and can be used for calculating trust scores for models which use the number of transactions as a base.

2.7 Software/Tools Requirements

- **Tensorflow** - a open source library used for machine learning applications
- **Python Networkx** - Python library for creation, manipulation, and study of the structure, dynamics, and functions of complex networks
- **Python Numpy and Pandas** - Data analysis and manipulation tools
- **Python PyPlot and Seaborn** - Data Visualization tools

Chapter 3

PROPOSED SYSTEM

3.1 System Analysis

3.1.1 System requirement analysis

Purpose

The purpose of the project is to explore and understand the existing trust models in Bitcoin Networks.The project further aims to develop a hybrid model derived from all the observed models.

Overall description

The network data is obtained and cleaned, transformed before implementing different Trust models on it. The model performance is computed and critical observations regarding the models are found. The observations thus obtained are used along with the potential trust-influencing factors to develop an improved model which is more robust in determining a node's trust.

Functional Requirements

(1) The study should result in a better understanding of the existing trust models and their applicability to a bitcoin network.
(2) Should identify the factors affecting trust in a bitcoin network.
(3) Should result in the development of a model which offers a better performance than the trust models explored.

Non-Functional Requirements

The model is only applicable to a bitcoin network and might not be useful for any other use-case. The model requires a dataset which explicitly quantifies the trust score a node is given with by its "neighbors".

3.1.2 Module details of the system

1. Data Preprocessing

This module deals with the preprocessing required to be done on the dataset before the dataset can be used for fitting the models.This deals with processes such as Data Cleaning and Data Transformation.

Data Cleaning is required to deal with erroneous and noisy data as well as missing data.

Data Transformation is used to transform the data from one format into another using processes such as encoding categorical data and feature scaling.

The splitting of the dataset into test-data and train-data is also done in this module.

2. Exploratory Data Analysis(EDA)

Exploratory Data Analysis (also known as Data Exploration) is used to analyse the data set using (mostly) visual methods. EDA is helpful in summarizing the relationship between the attributes and also enables to see a hidden pattern in the data and formulate a hypothesis about the dependence of the features and the outcome.

Python libraries such as seaborn, matplotlib have extensive functions for visualisation of datasets which can be used for this module. Networkx library is a rich graph visualisation library which can be used to visualise the bitcoin network and can help in further analysis by using centrality measures like Betweenness, eigen-vector centrality,degree and closeness.

3. Existing Models

3.1. Implementation

From the literature survey done, there are a number of existing models which can be implemented into the bitcoin network. This module will focus on implementing them into the datasets at hand. The models which are taken from others fields are going to be tweaked and adjusted so that they can be applied to the bitcoin network dataset.

3.2. Performance Evaluation

The Performance of each of the models will be evaluated using various Performance metrics such as Accuracy and Confidence Matrix.The results obtained from all the models will be studied as part of this module.

This module will also attempt to justify the errors obtained for the model and try to evaluate if there's any pattern visible in the errors obtained which if present,can be used to improve the model performance.

4. Observations

This module compares the Performance of all the models considered and records all the observations. Since multiple performance metrics are used, a preference order of the metrics needs to be decided upon.This preference order will then be used to establish which models perform better and hence can be used for the Hybrid Model.

5. Hybrid model

5.1. Hybrid model design

From the implementation of the existing models on the bitcoin network datasets in module 3.1 and the observations made in the module 4, the common characteristics/features and patterns are taken into consideration to design a new model. The features which commonly affect the bitcoin network as noted in the literature survey conclusion are also considered while designing the new model in this module. Feature Extraction will also be done to obtain a hybrid of attributes which might result in better Model Performance.

5.2. Hybrid Model Implementation

The hybrid model designed in Module 5.1 will be implemented on the dataset.

5.3. Performance Evaluation

The hybrid Model will be evaluated for its performance based on all the Performance Metrics which were used on the existing models in module 4.2

5.4. Performance Comparison

The Performance Score obtained from module 5.3 will be compared against the performance scores of all the models obtained as part of module 3.2. These performance scores can then be used to establish whether the hybrid model performs better than the existing models for this dataset.

6. Final Observation

After comparing the hybrid model designed as a part of module 5 with the existing models implemented as a part of module 3, the results and inferences will be mentioned in this module.If the newly designed model's performance is not on par with the existing models, the reason and future direction will also be explored in this module.

7. Conclusion

Post the implementation of the modules, a hybrid module would have been developed and the conclusion as to whether it is better than the existing models would be arrived at. A comparison of the existing models performance on the dataset would also be derived as part of this process.

3.2 System Design

3.2.1 Flow diagram of the system

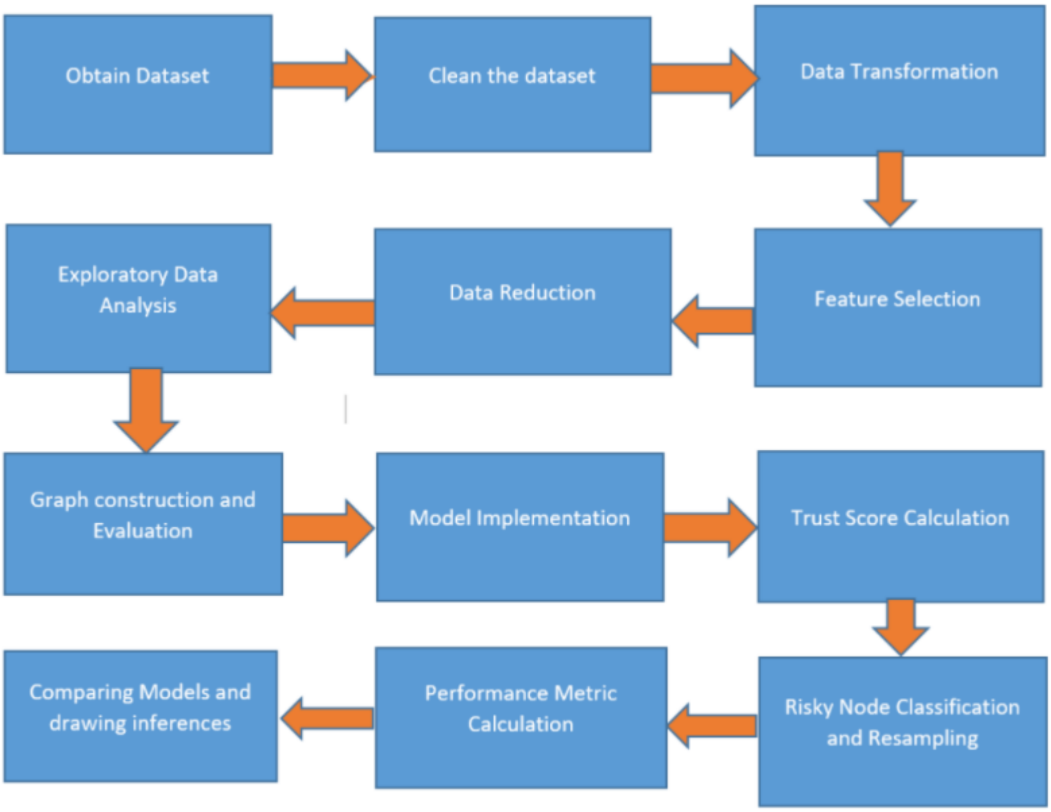

Obtain Dataset
Clean the dataset
Data Transformation
Feature Selection
Data Reduction
Exploratory Data Analysis
Graph construction and Evaluation
Model Implementation
Trust Score Calculation
Risky Node Classification and Resampling
Performance Metric Calculation
Comparing Models and drawing inferences

Figure 3.1: Flow Diagram

3.2.2 Architecture diagram

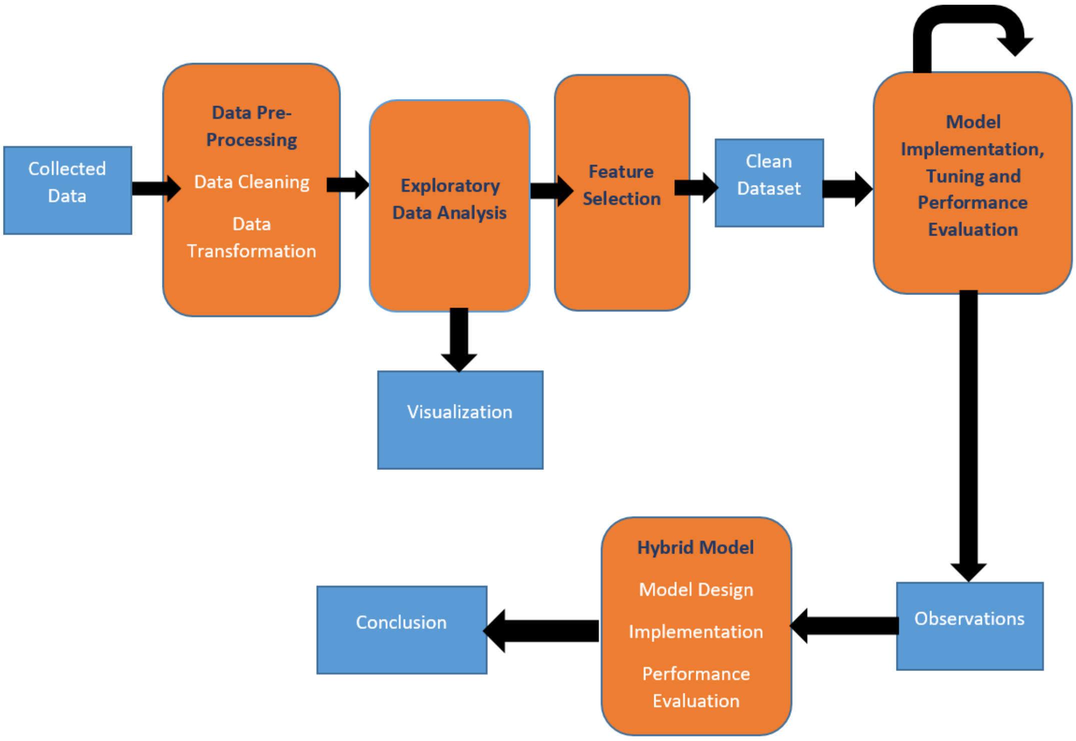


Figure 3.2: Architectural Diagram

Chapter 4

IMPLEMENTATION AND TESTING

4.1 Data preprocessing

The data has been obtained from Kumar et al. (2016). Each line has one rating, sorted by time, with the following format: SOURCE, TARGET, RATING, TIME.

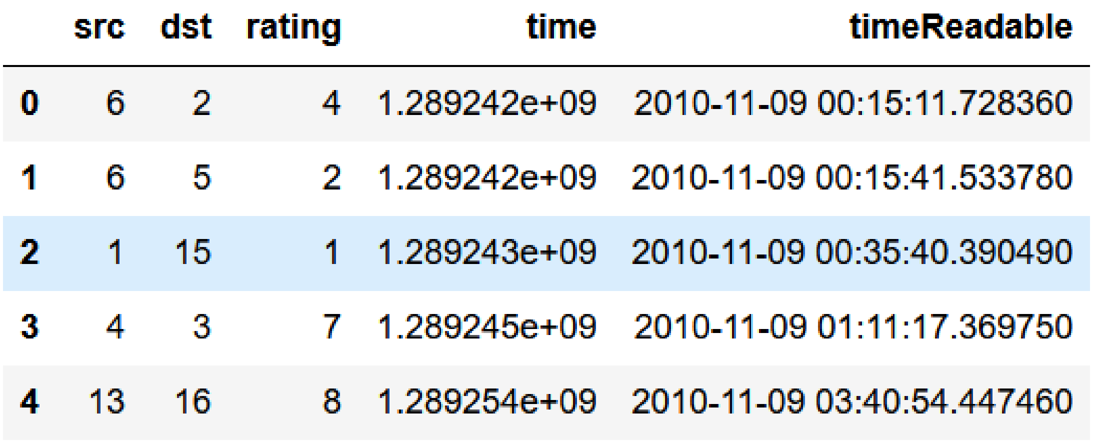
Where,

SOURCE: node id of source, i.e., rater

TARGET: node id of target, i.e., ratee

RATING: the source's rating for the target, ranging from -10 to +10 in steps of 1

TIME: the time of the rating, measured as seconds since Epoch. Which is converted to human readable data.

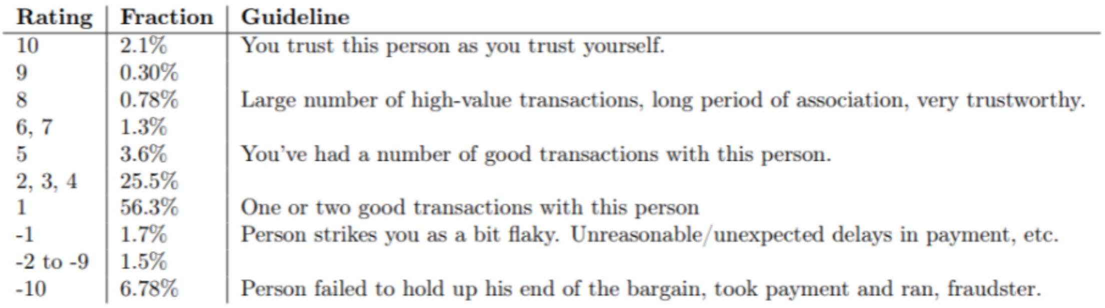


	src	dst	rating	time	timeReadable
0	6	2	4	1.289242e+09	2010-11-09 00:15:11.728360
1	6	5	2	1.289242e+09	2010-11-09 00:15:41.533780
2	1	15	1	1.289243e+09	2010-11-09 00:35:40.390490
3	4	3	7	1.289245e+09	2010-11-09 01:11:17.369750
4	13	16	8	1.289254e+09	2010-11-09 03:40:54.447460

Table 4.1: Snippet of data after preprocessing

4.2 Exploratory data analysis (EDA)

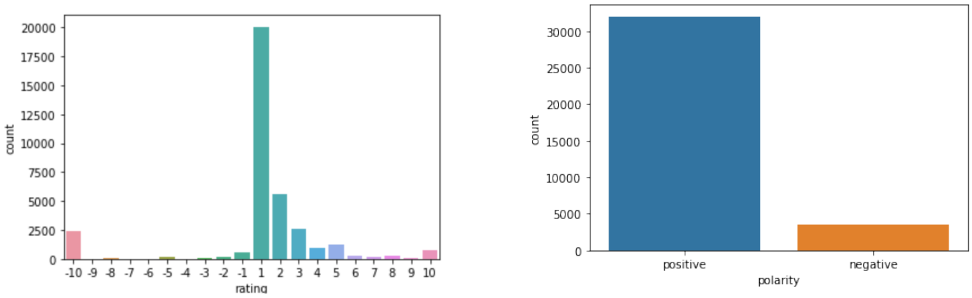
According to Moindrot (2017), the ratings in the dataset follow the below guidelines.



Rating	Fraction	Guideline
10	2.1%	You trust this person as you trust yourself.
9	0.30%	
8	0.78%	Large number of high-value transactions, long period of association, very trustworthy.
6, 7	1.3%	
5	3.6%	You've had a number of good transactions with this person.
2, 3, 4	25.5%	
1	56.3%	One or two good transactions with this person
-1	1.7%	Person strikes you as a bit flaky. Unreasonable/unexpected delays in payment, etc.
-2 to -9	1.5%	
-10	6.78%	Person failed to hold up his end of the bargain, took payment and ran, fraudster.

Figure 4.1: Rating guidelines from the OTC wiki and fraction of the overall ratings for each rating.

And a countplot on ratings reveals that the dataset is heavily skewed towards rating 1, which can be explained because many of the users in the bitcoin network do only one or two transactions with a particular node.



(a) On range of ratings

(b) On the polarity of ratings

Figure 4.2: Countplots on ratings

The dataset has 6000 nodes, 35592 transactions and 89% positive ratings. Using networkx to visualize the network, we find a few interesting observations.

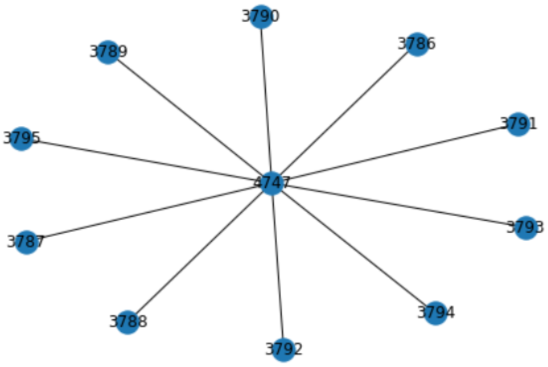


Figure 4.3: Cluster of nodes with -10 rating

The above nodes in the above cluster all have rated each other a -10 indicating mutual distrust.

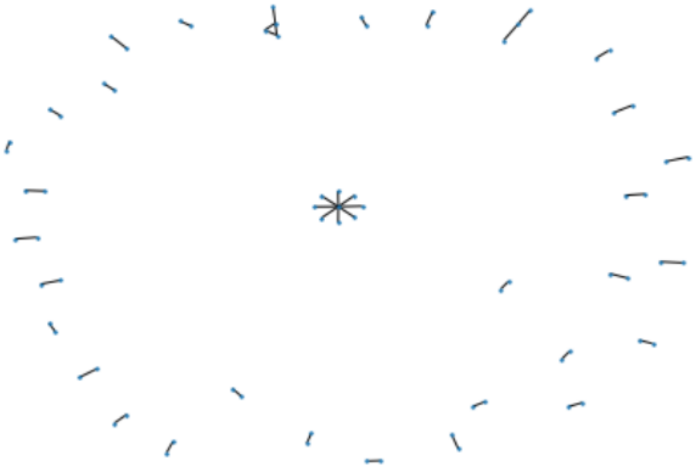


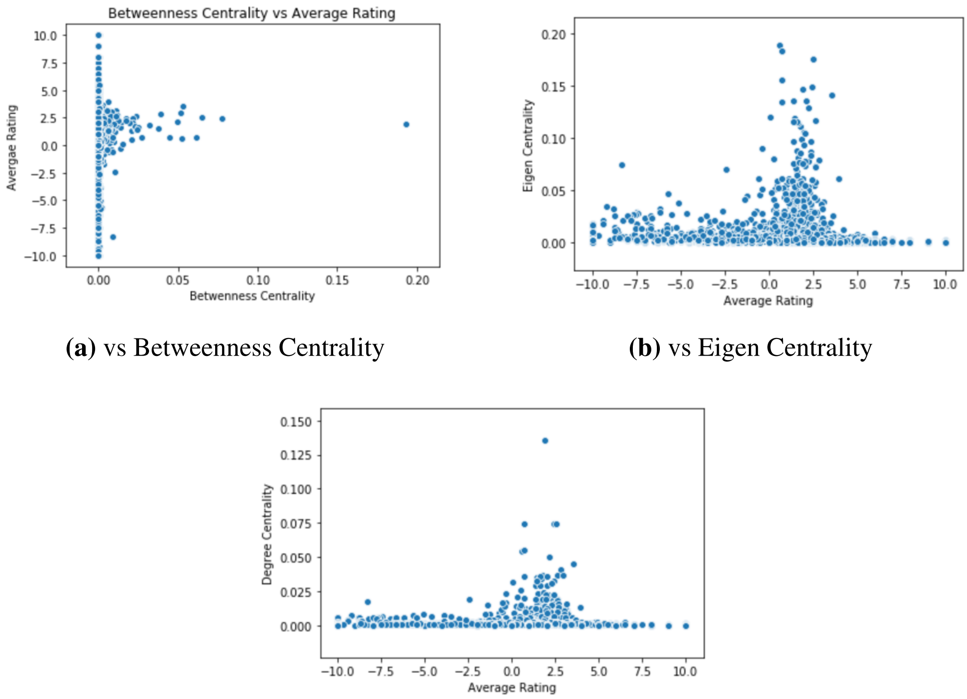
Figure 4.4: Cluster of nodes with +10 rating

Most of the above nodes are disconnected but there is a cluster which rates each other a +10 in the center. Single interaction could be a chance based incident but getting a cluster of +10 rating nodes is extremely dubious and unlikely. It might be that these nodes have rated themselves +ve to boost their trust scores. This suggests that the ratings by themselves aren't an accurate measure of trustworthiness. Considering other measures such as node interactions and time of transaction might provide better grounds for trust score calculations.

We also take a look at various centrality measures like degree centrality, betweenness centrality and eigen vector centrality against the average rating of the nodes in Figure. 4.5.

Average rating of a node is defined as the average of all the ratings it received from other nodes. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. Degree centrality is defined as the number of links incident upon a node. A high eigenvector score means that a node is connected to many nodes who themselves have high scores

The imbalance in the number of ratings resulted in all the centrality measures having their maximum value in the range of [0,5].



(a) vs Betweenness Centrality

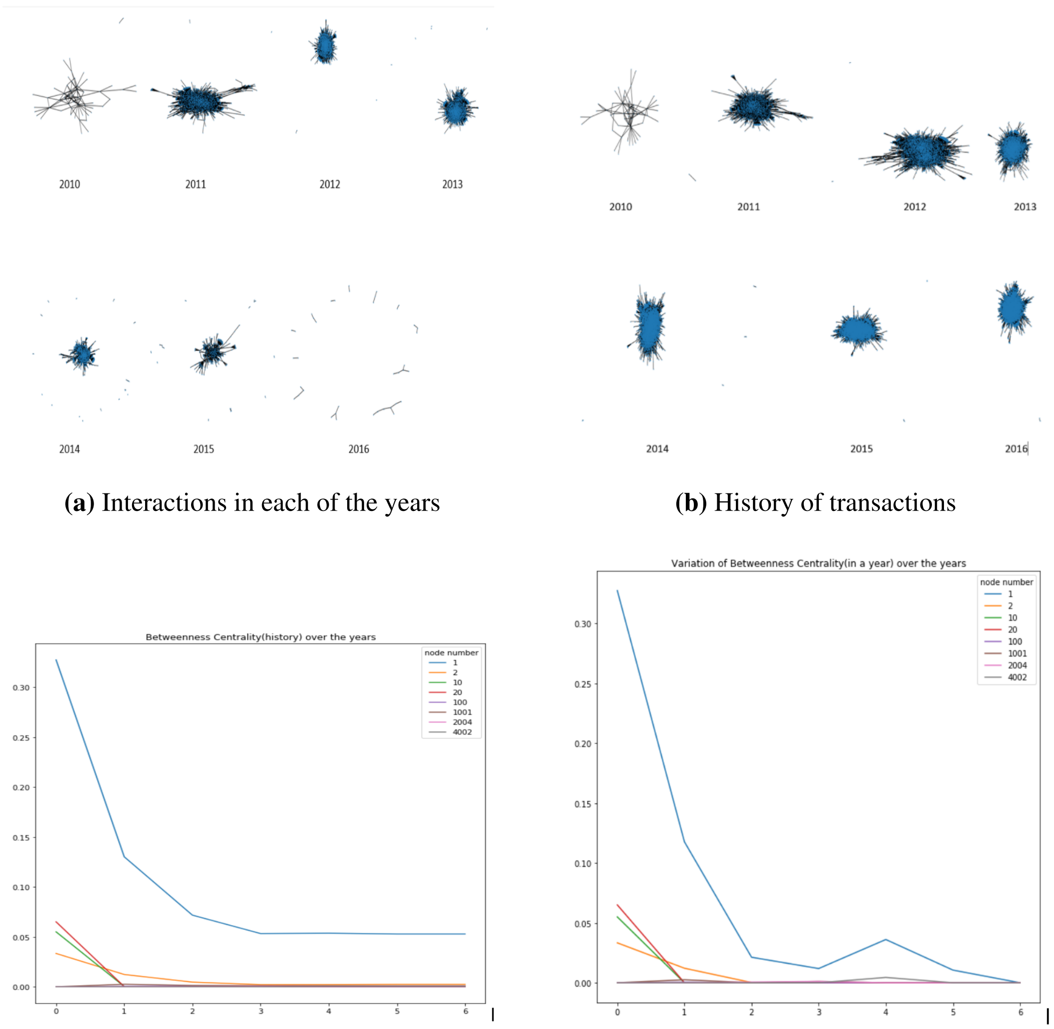
(b) vs Eigen Centrality

(c) vs Degree Centrality

Figure 4.5: Centrality measures against Average ratings

We also try to visualize the centrality measures against the time (Figure.4.6). The dataset is further divided based on the year of the transactions for this.

We observe that the Betweenness Centrality of a node decreases with time. Graph suggests that the nodes with lower node value had a high betweenness centrality in the beginning which reduced over time. This shows that node activity varies with time and shows the need for a time-adjusted model to compute trust. Also, the number of transactions per year seems to be reducing. There is also an increase in the outlier points which do not belong to any major cluster.



(a) Interactions in each of the years

(b) History of transactions

(c) Change of Betweenness Centrality of nodes over the years

(d) Betweenness Centrality Values in each year

Figure 4.6: Time based visualizations

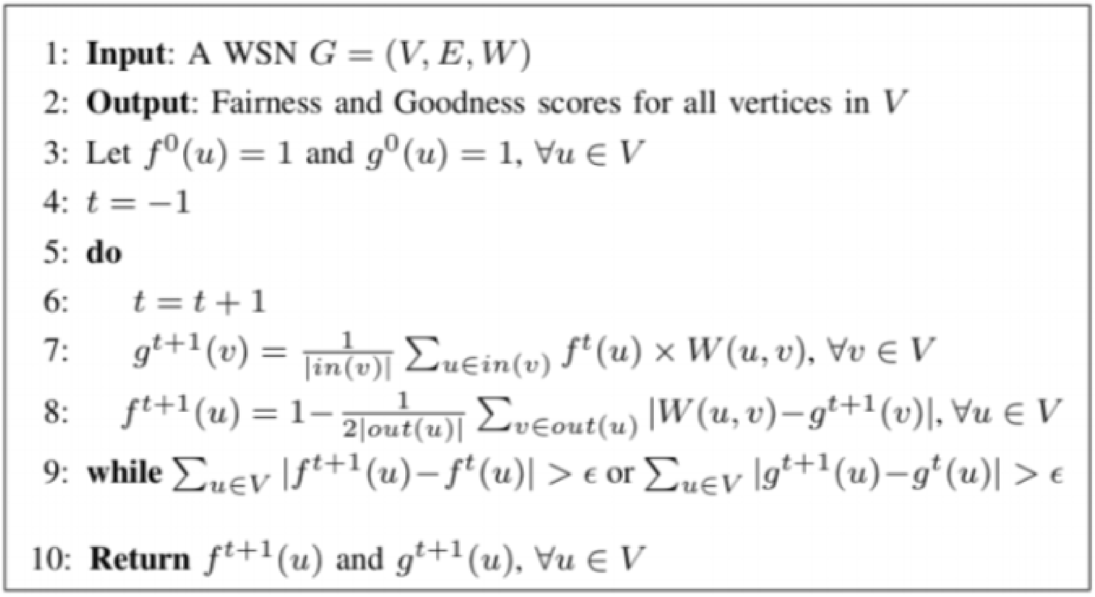
4.3 Existing Models

4.3.1 Implementation

Fairness and Goodness Trust Model

Based on the goodness and fairness formulas (Figure.2.1) and the below algorithm, we wrote code to calculate the fairness and goodness of a node iteratively.

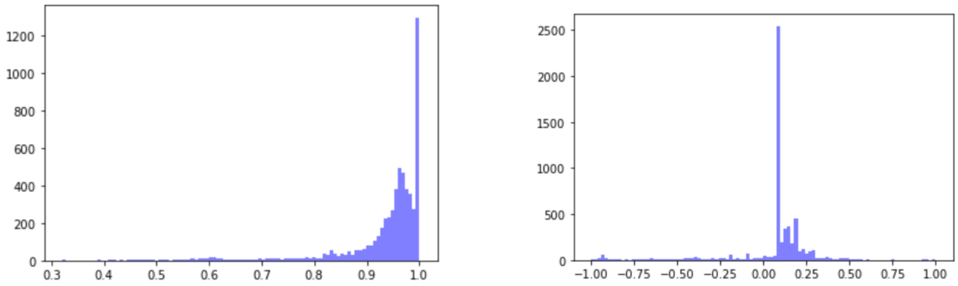
[H]



```
Input: A WSN $G = (V, E, W)$
Output: Fairness and Goodness scores for all vertices in $V$
Let $f^0(u) = 1$ and $g^0(u) = 1, \forall u \in V$
$t = -1$
do
    $t = t + 1$
    $g^{t+1}(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f^t(u) \times W(u,v), \forall v \in V$
    $f^{t+1}(u) = 1 - \frac{1}{2|out(u)|} \sum_{v \in out(u)} |W(u,v) - g^{t+1}(v)|, \forall u \in V$
while $\sum_{u \in V} |f^{t+1}(u) - f^t(u)| > \epsilon$ or $\sum_{u \in V} |g^{t+1}(u) - g^t(u)| > \epsilon$
Return $f^{t+1}(u)$ and $g^{t+1}(u), \forall u \in V$
```

Figure 4.7: Fairness and Goodness algorithm (From Kumar et al. (2016))

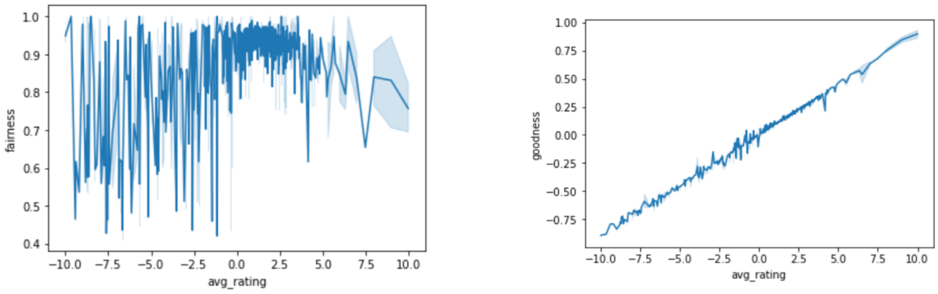
The fairness and goodness of the 6000 nodes have been plotted in histograms (Figure 4.8) and visualised against the average ratings (Figure 4.9).



(a) Fairness

(b) Goodness

Figure 4.8: Histograms on fairness and goodness of nodes



(a) vs Fairness

(b) vs Goodness

Figure 4.9: Against Average Ratings

The fairness of the nodes in the dataset is concentrated from [0.8,1], while goodness of nodes is concentrated around [0,0.3].

To classify a node into "more trustworthy (1)", "ambiguous (0)" and "less trustworthy (-1)", we use the mean fairness score (= 0.936153) and mean goodness score (= 0.070098) as the thresholds.

This classification results in the below countplot -

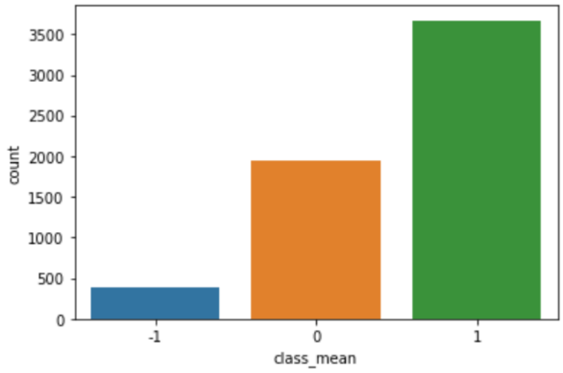


Figure 4.10: Classification of nodes based on mean score thresholds

Plotting the fairness of a node against the goodness of a node with the size of the node proportional to the average rating of the node, we get an interesting plot as shown below.

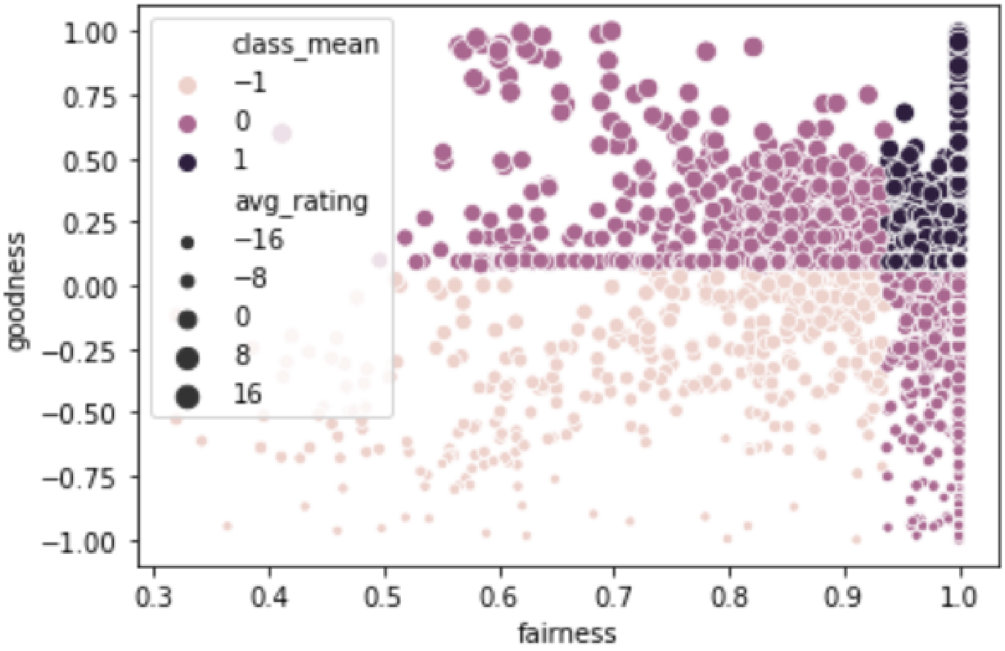


Figure 4.11: 2D plot of Goodness vs Fairness

The clustering can be seen even more clearly with the 3D plot -

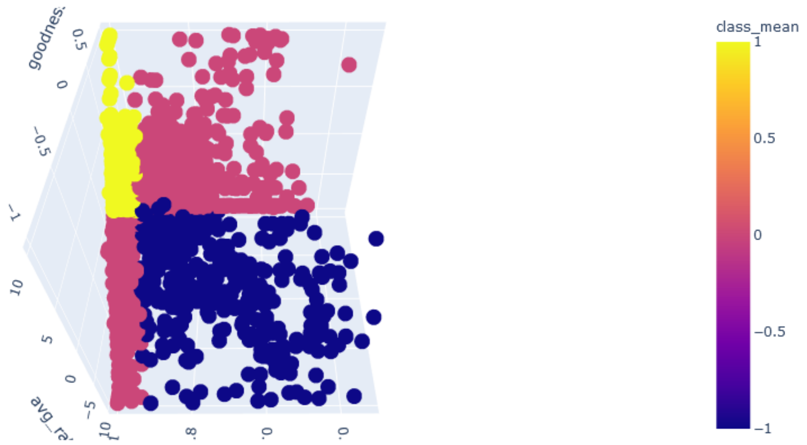


Figure 4.12: 3D plot of Goodness vs Fairness

K Nearest Neighbours (KNN) model using centrality measures

We also try to classify the nodes based on the centrality measures found earlier in section Exploratory data analysis (EDA) (Figure 4.5). The true labels are defined based on the sign of the average rating of each node. After splitting the whole dataset into training set and testing set, we perform KNN clustering using betweenness centrality and degree centrality to predict rating class of the node. The performance evaluation of this KNN model is discussed in the next subsection - Performance Evaluation.

4.3.2 Performance Evaluation

Fairness and Goodness Model

To evaluate the Fairness and goodness trust model which does not have any true labels - Davies Bouldin (DB) score and Calinski Harabasz (CH) score are selected to find out if the clustering is a fit or not.

The CH score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. A lower DB index relates to a model with better separation between the clusters.

The DB score results a 2.23 and CH score results a 799.08. The CH score is quite high because the clusters are very dense (especially cluster 1 in Figure.4.11).

KNN model using centrality measures

The KNN model implemented in the previous subsection (KNN model using centrality measures) is evaluated using the confusion matrix metrics. The KNN model is to predict 0 or 1 as the rating class of the node where 0 indicates overall negative average rating of the node while 1 indicates positive average rating. The model provides a satisfactory 70.4% accuracy but a disappointing 31% of f1-score for negative rating prediction.

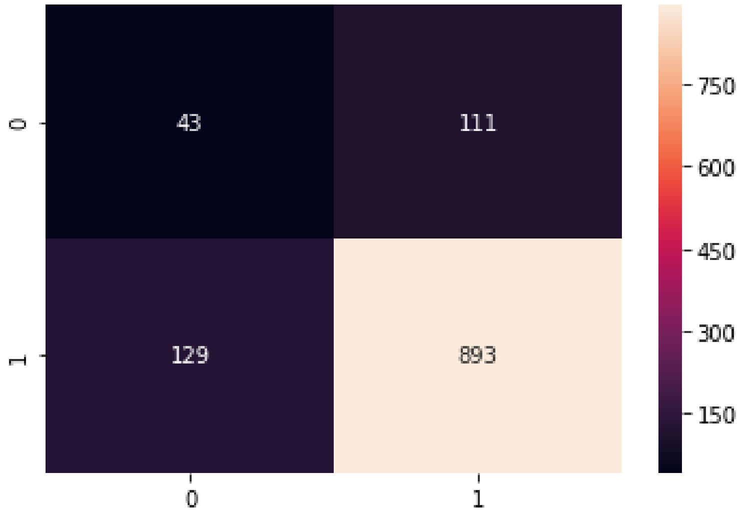


Figure 4.13: Confusion matrix for KNN model

Chapter 5

RESULTS AND DISCUSSION

Fairness and Goodness Model

The graphs in figure.4.9 indicate that fairness and the average ratings of the nodes are not related. While goodness and average ratings of the score are directly proportional to each other as the goodness of a node is defined as "how good a node is".
In figure.4.12, though the "more trustworthy (1)" nodes are high in number, they are all concentrated in one corner of the graphs. While the ambiguous nodes and the less trust worthy nodes are spread across the whole plane/space.

KNN model using centrality measures

The low f1-score of indicates that the model is able to recognise the trustworthy nodes (nodes with positive average rating) but fails to recognise the untrustworthy nodes (nodes with negative average rating). But identifying the untrustworthy nodes is the goal of a trust model. This model fails at this important task, making it unfit.

Chapter 6

CONCLUSION

From the implementation of the two trust models it has been found that the goodness and fairness of a node can predict the rating of a node. It has also been found that a KNN model using the betweenness centrality and degree centrality values as the distance parameter manages to attain a good accuracy. The KNN model doesn't predict the class 0 well and this results in low f1-score.

From the implementations we can conclude that fairness, goodness, degree centrality and betweenness centrality are the factors influencing Trust.These factors are identified to be used in the hybrid model. It has also been found that there's a correlation between centrality measures and the time. This observations makes it necessary for the consideration of time as a factor influencing trust.

Chapter 7

FUTURE ENHANCEMENT

The project further plans to implement the other models identified during the Literature Survey. The models will be implemented and their results will be evaluated to obtain the observations. The role of time in determining the trust score will also be explored. From the observations, a set of potential factors influencing the trust score will be implied. A hybrid using these factors will be designed and it's performance will be evaluated carefully to obtain optimum results.

The hybrid model is expected to perform better than all the implemented models and will be helpful to other researchers. The study is expected to serve as an introduction to Trust models used in Bitcoin Networks and provide a breakthrough in this field.

REFERENCES

1. Decker, C. and Wattenhofer, R. (2013). “Information propagation in the bitcoin network.” *13th IEEE Conference on Peer-to-peer Computing*, 1EEE.

2. Kumar, S., Spezzano, F., Subrahmanian, V., and Faloutsos, C. (2016). “Edge weight prediction in weighted signed networks.” *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, IEEE. 221–230.

3. Moindrot, O. (2017). “Trust in bitcoin exchange networks.” Project for cs224w-2017, Stanford University.

4. Sadhya, V., Sadhya, H., Hirschheim, R., and Watson, E. (2018). “Exploring technology trust in bitcoin:the blockchain exemplar.” *26th European Conference on Information Systems, Research papers*, Vol. 5, AIS Electronic Library (AISeL).