Tampere University

Laura Lepomäki

PEER REVIEW NETWORKS BETWEEN BITCOIN TRADERS

Faculty of Engineering and Natural Sciences
Master of Science Thesis
October 2019

ABSTRACT

Laura Lepomäki: Peer review networks between Bitcoin traders
Master of Science Thesis
Tampere University
Science and Engineering
October 2019

Bitcoin is a cryptocurrency that can be traded online. Some of the online Bitcoin trading platforms allow traders to give trust ratings to each other. Trust ratings are meant to indicate with whom to trade. Given and received trust ratings between Bitcoin traders form a Bitcoin trader peer review network. Understanding the functionality of Bitcoin peer review networks is crucial due to counter-party risk in Bitcoin transactions. This work studies the social aspects of Bitcoin trading. Trust rating data from two online Bitcoin trading platforms, Bitcoin OTC and Bitcoin Alpha, is used.

Bitcoin trader behaviour in peer review networks is reduced to five behavioural features: attention, reputation, activity, fairness and goodness. The first three are derived from the data in a straightforward way. The last two are determined by using a state-of-the-art algorithm designed for trust/distrust networks. Trader types are extracted by clustering the traders based on the behavioural features. Due to timestamped data it is possible to define how the behaviour of Bitcoin traders evolve over time. Bitcoin peer review networks are represented as chronological aggregated snapshots of the underlying temporal system. Per each aggregated network, traders are clustered based on their behaviour. Cluster transitions provide information about how Bitcoin trader behaviour evolves over time. This work focuses especially on adverse behaviour. Adverse behaviour refers to giving unfair trust ratings to others or being distrusted by other traders, especially fair ones. The impact of receiving unfair ratings on a trader's behaviour is studied. In addition, it is studied if adversely behaving traders form communities. A community is a group of traders who have been rating each other. Behavioural clusters are also studied in relation to the most and the least central traders. The most central traders substantially contribute to the peer review network while the impact of the least central ones is negligible.

The behavioural clusters show clear similarities between the datasets. There are trader types for which behaviour is exceptionally persistent. For well behaving traders it is common to remain as they are. Distrusted traders are likely to remain distrusted or disappear from the network, which can partly be explained by unfair negative treatment. Unfairly negatively rated traders can react to unfair treatment by becoming unfair themselves. Some of the most reputable traders have received their reputation from unfair positive ratings. Active and noticed traders with medium reputation behave in various ways in the future and are likely to stay in the network. In addition, it is observed that communities of unfairness and distrust emerge in Bitcoin peer review networks the same time with a burst of negative trust ratings. Surprisingly, the results on centrality show that the most well behaving traders become the least central. The most central traders in Bitcoin peer review networks are active and noticed traders who do not behave adversely.

Keywords: peer review networks, Bitcoin traders, behavioral clusters

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Laura Lepomäki: Vertaisarviointiverkot Bitcoin kaupankäyjien välillä
Diplomityö
Tampereen yliopisto
Teknis-luonnontieteellinen tutkinto-ohjelma
Lokakuu 2019

Bitcoin on yksi kryptovaluutoista, ja niitä voidaan ostaa ja myydä internetissä. Osa internetissä olevista Bitcoin kaupankäyntialustoista tarjoaa mahdollisuuden antaa luottamusarvoja toisille Bitcoin kaupankäyjille. Luottamusarvot on tarkoitettu osoittamaan, kenen kanssa kannattaa käydä kauppaa. Saadut ja annetut luottamusarvot Bitcoin kaupankäyjien välillä muodostavat vertaisarviointiverkon. Bitcoin vertaisarviointiverkon ymmärtäminen on tärkeää, koska kaupankäyntiin liittyy vastapuoliriski. Tässä työssä tutkitaan Bitcoin kaupankäynnin sosiaalisia puolia. Työssä käytetään luottamusarvodataa kahdelta Bitcoin kaupankäyntialustalta, Bitcoin OTC:ltä ja Bitcoin Alpha:lta.

Kaupankäyjien käyttäytyminen on redusoitu viiteen käyttäytymisominaisuuteen: huomio, maine, aktiivisuus, reiluus ja hyvyys. Ensimmäiset kolme on suoraviivaisesti johdettu datasta. Jälkimmäiset kaksi on määritetty käyttäen viimeisintä menetelmää edustavaa algoritmia. Kaupankäyjätyypit on määritetty ryhmittelemällä kaupankäyjät klustereihin käyttäytymisominaisuuksien perusteella. Aikamerkityn datan johdosta on mahdollista määrittää, kuinka Bitcoin kaupankäyjien käyttäytyminen muuttuu ajan myötä. Ajan kanssa muuttuva vertaisarviointiverkko on esitetty kokoamalla verkko kronologisiksi tilannekatsauksiksi. Jokaista koottua verkkoa kohden kaupankäyjät on ryhmitelty käyttäytymisen perusteella. Klusterisiirtymistä saadaan informaatiota käyttäytymisen muuttumisesta. Työssä keskitytään erityisesti epäsuotuisaan käytökseen. Epäsuotuisa käytös tarkoittaa, että kaupankäyjä antaa epäreiluja luottamusarvoja tai on epäluotettu erityisesti reilujen kaupankäyjien mielestä. Tässä työssä tutkitaan epäreilujen luottamusarvojen vastaanottamisen vaikutusta kaupankäyjän käytökseen. Lisäksi, työssä tutkitaan muodostavatko epäsuotuisasti käyttäytyvät kaupankäyjät yhteisöjä. Yhteisöllä tarkoitetaan kaupankäyjien ryhmää, jossa kaupankäyjät ovat antaneet toisilleen luottamusarvoja. Käyttäytymisklustereita tutkitaan myös keskeisimpiin kaupankäyjiin nähden. Keskeisimmät kaupankäyjät vaikuttavat merkittävästi vertaisarviointiverkkoon, kun taas vähiten keskeisten kaupankäyjien vaikutus on merkityksetön.

Käyttäytymisklustereissa on selkeitä samankaltaisuuksia datajoukkojen välillä. Osa käyttäytymistyypeistä on poikkeuksellisen pysyviä. Hyvin käyttäytyville on yleistä säilyä sellaisina. Epäluotetuille kaupankäyjille on todennäköistä pysyä epäluotettuina tai lähteä verkosta. Osaltaan se voidaan selittää epäreilulla negatiivisella kohtelulla. Epäreilun negatiivisesti arvioidut kaupankäyjät voivat reagoida epäreiluun kohteluun muuttumalla itse epäreilummiksi. Osa maineikkaimmista kaupankäyjistä on saanut maineensa epäreilun positiivisista luottamusarvoista. Aktiiviset ja huomioidut kaupankäyjät, joiden maine on keskivertoa, käyttäytyvät eri tavoin tulevaisuudessa ja heille on todennäköistä pysyä verkossa. Lisäksi, epäreilujen ja epäluotettujen yhteisöjä ilmenee samanaikaisesti negatiivisten luottamusarvojen ryöpyn kanssa. Yllättävästi, suotuisimmin käyttäytyvät kaupankäyjät päätyvät vähiten keskeisiksi. Keskeisimmät kaupankäyjät ovat aktiivisia ja huomioituja kaupankäyjiä, jotka eivät käyttäydy epäsuotuisasti.

Avainsanat: vertaisarviointiverkot, Bitcoin kaupankäyjät, käyttäytymisklusterit

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

PREFACE

This thesis was conducted independently as a separate project from work and school projects. The general topic of complex networks was given by Juho Kanniainen, one of the thesis supervisors. Datasets and the specific topic were decided independently. The research questions were reviewed together with the supervisors before proceeding further. This work was conducted as exploratory data science, where a given data is approached by applying various mathematical methods in order to conclude which research questions and methods are the most reasonable in the context. Exploratory phase of the thesis project was partly done at work. The majority of the work was conducted outside working hours. The supervisors gave feedback all the way during the process.

I would like to thank my thesis supervisor Henri Hansen for helping me with mathematical notations and academic writing, and last but not least for encouraging me during the busy months. I greatly thank also my other thesis supervisor, Juho Kanniainen, for leading me to the interesting world of complex networks and for giving me essential feedback regarding the topic. I would also like to thank Niklas Jahnsson for giving me feedback for the final version of the thesis. The greatest thanks goes to my family for supporting me during the process. Special thanks to Jesse who significantly contributed to the balance between the work and spare time.

In Tampere, Finland, on 25 October 2019

Laura Lepomäki

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

HITS	Hyperlink-Induced Topic Search
WCSS	Within cluster sum of squares

$\|A\|$	The number of elements in A
a	Attention
α	Significance level
$\hat{\alpha}$	Bonferroni corrected significance level
$\underset{C_k}{\operatorname{argmin}}(\cdot)$	Cluster C_k that minimizes a given criteria
β	Trader
$\beta_n^{(j)}$	The j:th trader who has rated trader n
C	Cluster
c	Activity
$d_{\text{in}}(n)$	In-degree of node n
$d_{\text{out}}(n)$	Out-degree of node n
ΔX	Change in the value of variable X
$\delta(x,y)$	Function that has the value 1 if $x = y$ and 0 otherwise
E	Set of edges
e	Edge
ε	Error tolerance
η	Sum of edge weights
f	Fairness
G	Network
G_t	Aggregated network corresponding to a time-stamp t
g	Goodness
γ	Trader
$\gamma_n^{(j)}$	The j:th trader rated by trader n
$h : E \longrightarrow \mathbb{R}$	Mapping from edges to real numbers in a weighted network
K	Integer
k	Non-negative integer, the number of nodes, the number of clusters
M	Integer
m	Non-negative integer, the number of edges
μ	Mean
$\max(x)$	Maximum value for x
N	Set of nodes
$N_{n_i,\text{out}}$	Set of nodes node n_i points to
$N_{n_i,\text{in}}$	Set of nodes that point to node n_i
n	Node, trader
Ω	Community
ω_i	Community of node n_i
$P(X = M)$	Probability that a variable X has the value M
ϕ	Hub value
Q	Modularity

$\mathbb{R}$	Set of real numbers
r	Reputation
ρ	Pearson correlation
$\sum_{i=0}^{M}$	Sum of M elements indexed by i
$\sum_{G,n_i}$	Sum of the weights of edges between node n_i and its adjacent nodes in network G
$\sum_{\Omega}$	Sum of edge weights within community Ω
$\sum_{G,\Omega}$	Sum of the weights of edges in network G that are incident to nodes in community Ω
$\sum_{\Omega,n_i}$	Sum of the weights of edges between node n_i and its adjacent nodes in community Ω
σ^2	Variance
σ	Standard deviation
t	Time
τ	The number of certain occasions
θ	Authority value
V	Set of feature values
W	Within cluster sum of squares
$w_{i,j}$	Edge weight between nodes n_i and n_j
$X_{n,t}$	List of trust ratings received by trader n in aggregated network G_t
$Y_{n,t}$	List of trust ratings given by trader n in aggregated network G_t
Z	Set of feature vectors
z	Feature vector, sample
$\times$	Multiplication, Cartesian product
$x \iff y$	x if and only if y
$x \in X$	x belongs to X
$\vert\vert \cdot \vert\vert$	Distance, similarity measure
$\forall$	For all
$\binom{m}{i}$	The number of i-combinations of a set of m elements
$\{x_1, \ldots, x_m\}$	Set of m elements
$(x_1, \ldots, x_m)$	List of m elements

1. INTRODUCTION

Interacting with strangers has become more common due to the internet. On the internet, there are platforms where transactions are made between anonymous participants. To approach the risk of fraudulent behaviour in such systems, many online platforms allow participants to give feedback to each other. The feedback a participant has received on a platform forms the participant's reputation. This kind of reputation system is stated to introduce trust between participants and to encourage participants to behave well. [2, 3] In this work, a group of participants who give feedback to each other is called *a peer review network*.

Emotions and intuition play a role in human reactions. In a peer review network a participant can give feedback to someone who has given feedback to the participant. This allows one to retaliate negative or unfair feedback by giving back negative feedback. Previous studies provide results on reactions to unfairness and low reputation. Restaurants with low reputation are more likely to create fake reviews on an online customer review platform than restaurants with good reputation [4]. Cheating in a game is more likely for those who experience unfair treatment in the game [5]. In a game where accepting an offer is always economically beneficial, people tend to reject strictly unfair offers but accept offers that are only slightly unfair [6]. In an organization, a group of employees can react to unfair treatment by dishonest behaviour, if the one behind the unfair treatment is external to the group [7].

This work studies how participants in peer review networks give and receive feedback. The peer review networks studied in this work are formed by Bitcoin traders, who have given trust ratings to each other on online Bitcoin trading platforms. Due to anonymity, a counterparty risk is present in Bitcoin transactions. To avoid trading with distrustful traders, some Bitcoin trading platforms allow traders to give trust ratings to each other. Time-stamped trust ratings can be used to describe Bitcoin trader behaviour and its evolution. In this work, trader behaviour refers to giving and receiving feedback in a Bitcoin peer review network. Data on actual trades is not used, and trader behaviour does not refer to how Bitcoin traders buy and sell Bitcoins. The trust ratings are used to define behavioural features: *attention*, *reputation*, *activity*, *fairness* and *goodness*, using simple methods and a state-of-the-art algorithm.

The focus of this work is on *adverse behaviour*. By adverse behaviour it is meant that a trader is distrusted by others or gives unfair trust ratings to other traders. Based on the above listed behavioural features, traders are divided into clusters. A cluster is a group of similarly behaving traders. In this work, a behavioural cluster refers to a group of traders whose behaviour in terms of giving and receiving feedback is similar. Behavioural clusters are used to study how trader behaviour evolves over time and how the traders react to unfairness. This work searches for answers to the following questions about Bitcoin trader behaviour:

1. What kind of behavioural clusters are formed from Bitcoin traders?
2. How does a trader's behaviour in a Bitcoin peer review network change over time?
3. What is the impact of receiving unfair ratings on a trader's behaviour in a Bitcoin peer review network?

In addition, topological matters of Bitcoin peer review networks are studied in this work. If a trader receives unfair or negative ratings from another trader, a reaction might be to give back similar ratings. This would indicate mutual distrust or retaliation. To examine if adversely behaving traders have been rating each other, traders are divided into communities. A community of traders is a group of traders who have been densely rating each other while rating only few traders from other groups. Communities differ from clusters in that they are related to network topology, while clusters are related to the behavioural features. Another network topology related quantity studied in this work is centrality. Centrality is a measure of a trader's importance. The most central traders in a peer review network substantially contribute to the network while the impact of the least central ones is negligible. This work searches for answers to the following questions about Bitcoin peer review network topology:

1. Are there communities of adversely behaving traders in Bitcoin peer review networks?
2. How does centrality relate to traders' behaviour in Bitcoin peer review networks?

The main results show that trader behaviour does not change drastically over time. Traders who are distrusted by others remain distrusted or disappear from peer review networks. This type of behaviour can partly be explained by unfair negative feedback. Trusted traders remain as they are exceptionally often. Traders who actively give trust ratings and receive many ratings from other traders in a network can behave in various ways in the future but are likely to stay in the network. In addition, peer review networks contain communities of adversely behaving traders. When the proportion of highly negative trust ratings increases, it is observed that unfair and disreputable traders form communities. Furthermore, the results on centrality are partly counter-intuitive. That is, the most well behaving traders become the least important unusually often in Bitcoin peer review networks. To become and remain an important contributor in Bitcoin peer review network one needs to be active, noticed and not behaving adversely.

1.1 Social Networks

A network is a set of components called *nodes* and connections or links between them, called *edges*. Also the word *vertice* can be used in place of the word *node*, the latter being chosen to be used in this work. There are many structures, phenomena and systems in real life that can be represented as a network, one of the most common of which is the internet. In the internet, computers or groups of computers are represented by nodes whereas physical links between them are represented by edges. [8]

One type of a network is a *social network*. A social network is a network where actors, e.g. individuals, groups or communities, are connected to each other according to some relationship [9]. This relationship could be for example friendship, co-authorship or employment, just to mention some. As the name suggests, social networks model social phenomena. Nodes in a social network represent actors or groups of actors, and edges represent interactions or relationships between the actors. [8] Typical examples of social networks are systems such as Facebook and Twitter. Yet the topic covers also networks that are not necessarily designed to be social networks. [9] For example, online platforms where people can buy and sell currencies are not related to social relationships as such. All the same, human interaction is present in such buyer-seller networks, and these networks can be categorized as social networks. Social networks facilitated by the internet are called *online social networks*. Online social networks have gained a lot of attention as a result of increased supply of data. [9] Networks studied in this thesis can be categorized as online social networks.

Social network analysis can be divided into two parts: structural analysis and content-based analysis. Structural analysis refers to understanding arrangement and linkage of the network, including but not limited to investigating communities and centrality. Communities and centrality are explained in more detail in the next chapter. Also, the evolution of the network over time can be part of structural analysis. In comparison to structural analysis, content-based analysis is related to additional information of the network. Many social networks include large amount of information that can be advantageous for understanding the nature of the network. For example, a social network platform such as Facebook contains pictures, text and games that provide a lot of additional data into the analysis. [9] It is common to combine structural and content-based analysis [9]. Both topological and social aspects can be taken into account [10, e.g.], which is the approach taken in this work.

1.2 Bitcoin Peer Review Networks

Cryptocurrency means digital money that is secured by cryptographic procedures. One of the cryptocurrencies is *Bitcoin*. Bitcoin was created 2008 by Satoshi Nakamoto and it is currently probably the most known cryptocurrency [11]. Cryptocurrencies have gained a lot of attention due to their relatively recent upcoming and revolutionizing nature. There are other cryptocurrencies than Bitcoin such as Etherium and XRP, yet data used in this work is related to Bitcoin. Bitcoins are formed in a process where complex mathematical problems are solved by a network of computers. This is called *Bitcoin mining*. Those who lend their computational power to the system are called *Bitcoin miners*. [12] People who in turn buy and sell Bitcoins are referred to as *Bitcoin traders*. In this thesis social networks of Bitcoin traders are examined.

Unlike traditional digital currencies and central banking systems, Bitcoin trading does not have any third party intermediary assuring appropriate handling of transactions. Instead, Bitcoin is implemented using distributed ledger technology, more specifically block-chain. In addition, Bitcoin traders are anonymous in a sense that they are not identifiable by other traders. Thus, Bitcoin trader network functionality is based on a peer-to-peer network, where transaction history is maintained and verified by Bitcoin miners using block-chain technology [12, 13, 14]. A known vulnerability of such system is '51% attack' where a group of Bitcoin miners covering more than half of the computational power of the Bitcoin mining network would be able to control transactions [15].

Technology-wise, Bitcoins cannot be 'double spent'. This is due to a time-stamp server storing blocks of irreversible transactions to keep track on transaction history verified by Bitcoin miners. The concept of trust in block-chain based finance is far from being trivial. It is considered to be shifted instead of excluded in such systems. [12] It is crucial to understand how trust is embedded in Bitcoin peer review networks. The topic being extensive, this work is scoped to focus on social aspects. Consequently, technological matters are not included in the analysis.

There are many platforms where Bitcoins can be traded, of which Bitcoin OTC and Bitcoin Alpha are the ones whose data is used in this thesis. The two datasets used in this work are referred to as *Bitcoin OTC* and *Bitcoin Alpha* according to the source of the data. On these platforms traders can give ratings based on how they trust other traders. This type of rating system is a way to tackle the problem of hiding fraudulent behaviour behind anonymity. In other words, it gives information about with whom to trade. Naturally there is a risk of fraudulent traders affecting Bitcoin trading by giving false trust ratings to other traders. The nature and evolution of Bitcoin peer review networks can be studied from time-stamped trust rating data. This is important as both technical and social aspects need to be studied in order to guarantee proper functionality of Bitcoin trading. To approach the latter, this thesis continues the work of [16, 17] by studying in more detail the behaviour of Bitcoin traders. Trader behaviour is captured into features derived from trust rating data. The evolution of trader behaviour over time is possible to analyze due to time-stamped data. Because other type of data such as executed trades is not included, trader behaviour in this work is related to given and received trust ratings in a peer review network of Bitcoin traders. Other type of behaviour is outside the scope of this work.

Datasets used in this work are available online in [18, 1] and introduced in [16, 17]. Links to the actual trading platforms where the datasets are from are provided in [18, 1]. Unfortunately, the link to Bitcoin Alpha trading platform is no longer valid. The datasets used in this work are considered valid as they are used in previous publications. Both datasets contain traders numbered by positive integers and trust ratings ranging from -10 up to 10 with a step 1 excluding 0. Rating value 10 represents the highest possible trust, while -10 means severe distrust. In the datasets there are no repeating ratings meaning

Table 1.1. *The number of traders and trust ratings in Bitcoin OTC and Bitcoin Alpha datasets are shown in the first columns. 'Time Range' presents the first and the last time-stamp in the data. The average number of received, μ_{in}, and given, μ_{out}, ratings as well as their variances ($\sigma^2_{in}, \sigma^2_{out}$ resp.) rounded to the closest integer are shown in the last columns.*

Dataset	Traders	Ratings	Time Range	μ_{in}	σ^2_{in}	μ_{out}	σ^2_{out}
Bitcoin OTC	5881	35592	2010-11-08 - 2016-01-25	6	313	8	533
Bitcoin Alpha	3783	24186	2010-11-08 - 2016-01-22	6	271	7	378

that trader n_i gives a trust rating to trader n_j once if ever in the data. A trader cannot give a trust rating to himself/herself. The total number of nodes and edges as well as the time ranges covered by the datasets are presented in Table 1.1. The mean and variance of the number of received and given ratings over the traders are also presented in the table. Accordingly, datasets cover 6 years in total and include thousands of Bitcoin traders. The number of given and received trust ratings varies significantly between traders.

2. MATHEMATICAL METHODS

Mathematical methods used in answering the research questions are explained in this chapter. In the first section, the concept of network is introduced with examples. Properties of networks, and quantities related to nodes are discussed. After that, the concepts of clustering, community detection and centrality are presented. The methods chosen in this work are described in more detail. There are many other methods and algorithms related to the topics of this chapter yet they are left outside the scope of this work. Comparing for example clustering methods is considered a natural and interesting extend to this work. The methods used in this work are chosen by their efficiency and suitability, and they are considered to be a reasonable start for a more elaborate research.

2.1 Network Theory

Networks can be applied to multiple scientific fields ranging from biological food chains and chemical reaction networks to internet and power grids. The approach in network theory is to model the underlying system as a network, and to use mathematical methods to understand its nature. [8] Recently, the study of complex networks has gained importance due to increased supply of data and computational power. Complex networks are used to model large, complex systems that change over time. The study of complex networks has been developed in the context of real networks. [19] The word *complex* in this context usually refers to the size of the network but also to the complex nature of the underlying system. Network components can include additional information, for example there can be multiple types of edges or various attributes associated to nodes. Complex networks can be used to model large complicated systems.

Definition 2.1.1 (Network) *A network $G = (N,E)$ is a collection of nodes $n_i \in N = \{n_1, n_2, ..., n_k\}$ and edges $e_{i,j} \in E \subseteq N \times N$. An edge $e_{i,j} = (n_i, n_j)$ represents a connection from node n_i to n_j. Nodes n_i and n_j are called adjacent if $(n_i, n_j) \in E$ or $(n_j, n_i) \in E$.*

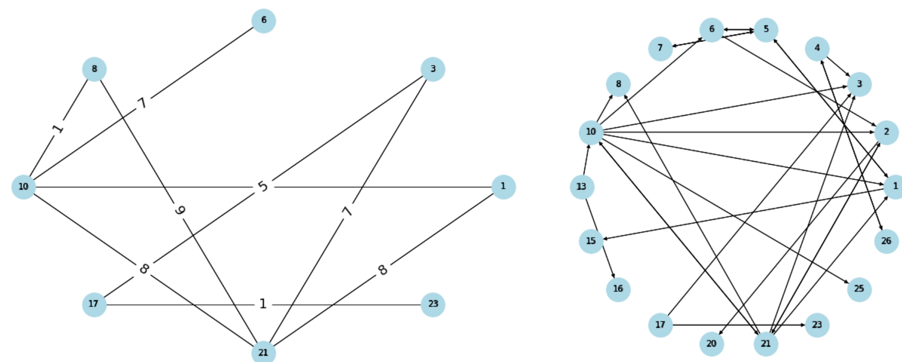
Definition 2.1.2 (Undirected Network) *An undirected network is a network $G = (N,E)$ such that*

$$(n_i, n_j) \in E \iff (n_j, n_i) \in E.$$

Definition 2.1.3 (Weighted Network) *Let $G = (N,E)$ be a network. G is weighted if there exists a mapping $h : E \longrightarrow \mathbb{R}$*

$$h((n_i, n_j)) = w_{i,j},$$

that assigns a weight for each edge.



(a) Undirected, weighted network **(b)** Directed network

Figure 2.1. *An illustration of (a) an undirected, weighted network and (b) a directed network using real data from Bitcoin OTC trading platform [1].*

Mathematical definition of a network is presented in Def. 2.1.1. An edge e is defined by two nodes, n_i, n_j, representing that there is a connection from n_i to n_j. Nodes that have an edge between them are called *adjacent*. A network can be directed or undirected. In a directed network, an edge has a direction, i.e. it is going from one node to another. A network is directed by definition. In an undirected network, connections work both ways as defined in Def. 2.1.2. Some networks include additional information on top of nodes and edges. One type of network that is often used is called *weighted network*. In a weighted network every edge has a *weight* associated to it as expressed in Def. 2.1.3. Weight of an edge or *edge weight* is usually a real number. Sometimes it is particularly noted that an edge can have a negative weight. Such networks are called *signed networks*. Edge weight can represent for instance the strength of the connection. [8] Networks studied in this work are directed, signed and weighted.

An example of an undirected, weighted network is shown in Fig. 2.1(a). In the figure, numbered spheres represent nodes, and lines between them represent edges. Fig. 2.1(b) presents an example of a directed network. Examples of real world directed networks include World Wide Web in computer science, protein-protein metabolic network in biochemistry, citation network in scientific research, and pray-predator food network in ecology. Collaboration network and road maps are examples of undirected networks. Some networks can be interpreted as directed or undirected depending on the aspect the phenomenon is studied from. [8] For instance, a network where people are linked if they are friends might be considered undirected but could also be interpreted as directed.

Networks are used to model systems related to various scientific fields, as demonstrated by the above listed examples. It is beneficial to develop general methods and quantities that can be applied to networks in various contexts. One commonly used quantity of a node in a network is its *degree*.

Definition 2.1.4 (Degree) *Let $G = (N, E)$ be a network. The degree of a node n, $d(n)$, in* *an undirected network is the number of edges adjacent to that node,*

$$d(n) = |\{\gamma : (n,\gamma) \in E\}|,$$

where $|\cdot|$ is the number of elements in a set. In a directed network, the in-degree of a node n is the number of in-going edges,

$$d_{in}(n) = |\{\beta : (\beta,n) \in E\}|,$$

whereas the out-degree of a node n is the number of out-going edges,

$$d_{out}(n) = |\{\gamma : (n,\gamma) \in E\}|.$$

The degree of a node in a network is defined in Def. 2.1.4. In an undirected network, the degree of a node can be thought as the number of its neighbors. In a directed network there are two types of degrees: *in-degree* and *out-degree*. The in-degree of a node is the number of edges coming to the node. Analogously, the out-degree of a node is the number of edges going from that node to other nodes. [8]

Definition 2.1.5 (Self Loop) *Let $G = (N,E)$ be a network. A self loop at node $n \in N$ is $(n,n) \in E$.*

A basic property of a network is whether it contains *self loops* or not. A self loop at a node means that the node is connected to itself, as defined in Def. 2.1.5. A network that contains no self loops and has at most one edge from any node n_i to another node n_j is called *a simple network*. [8] Networks studied in this work are simple networks.

Networks can be used to model both static and dynamic systems. Static networks are networks that do not change over time. They are often used to model systems that evolve over time relatively slowly. Networks that model slowly changing systems can be represented as a sequence of static networks. In such an approach, chronological aggregated snapshots of the network are used to model the underlying slowly evolving system. Networks that evolve over time are called dynamic or *temporal networks*. Temporal networks allow studying how the structure and properties of the underlying dynamic system changes over time. [9]

In network theory there are various quantities to describe the nodes of a network. Many times it is necessary to order nodes based on how important and central they are in a network. This type of *node hierarchy* can be used for example in finding the most cited publications in a citation network. Accordingly, one quantity in network analysis is the *centrality* of a node. Centrality is a measure of a node's importance in terms of how connected the node is to other nodes in a network. Centrality measures are often based on the degree of a node. In its simplest form, degree can be used as a centrality measure. In many networks there are a small number of nodes that have exceptionally high degrees. Such nodes are called *hubs*. Recent empirical and theoretical studies argue that the hubs of a network can have an excessive impact on the behaviour of the network. [8] In a directed network, centrality can be separated into two types: being central in terms of out-going and in-going edges. This division allows extracting important source nodes and destination nodes. An important source node points to many important nodes. Important source node is called a hub in this context. Important destination node, a so called *authority*, is pointed at by many hubs. [20] Centrality is further discussed in section 2.4.

Networks are applied to various problems such as link prediction [16, 21, 22, 23], anomaly or fraud detection [17, 24, 25, 26] and complex contagion [27, 28, 29]. Link prediction refers to predicting the emergence, signs and weights of edges in a network [16]. Anomaly and fraud detection refers to detecting anomalous or fraudulent nodes in a network. For example, in [26] a method for finding users who give false reviews on an online commerce platform is developed. Complex contagion is a phenomenon where connected nodes have an influence on each other. It can mean for example rumours or news spreading in a network of people [29]. One of the typical tasks related to networks is *community detection*. In community detection, nodes are partitioned into groups so that nodes within a group are highly connected, while nodes in different groups share only few links [8]. Community detection is discussed in more detail in section 2.3.

Systems that are modeled by complex networks usually include a large number of components and connections [19]. Such systems are not feasible to analyze without increasing the level of abstraction. Complex networks are a mathematical tool and thus an abstraction of the underlying system. Higher abstraction level allows applying general methods and quantities to a wide range of systems.

2.2 K-Means Clustering

Clustering means grouping samples based on their similarity. It is used to recognize patterns in data. Clustering is applied in various contexts such as biological analysis, image processing, and data compression [30, 31, 32]. There are many methods for clustering, yet on a general level they apply the same idea. At first, a similarity measure needs to be defined. For instance, Euclidean distance can be used as a similarity measure. Samples are then partitioned into groups called *clusters* so that similar samples are placed into the same cluster. The goal is to find a partition of the samples so that the samples within a cluster are similar to each other and differ from the samples in other clusters. Representing the samples in a dataset by clusters is a simplification and thus part of the information is lost in clustering. [33]

There are various types of clustering such as probabilistic clustering and hierarchical clustering. In this work, clustering refers to *partitioning relocation clustering*. In this type of clustering, samples are grouped into disjoint subsets and the optimal set of subset is found in an iterative way. Starting from some initial set of subsets, clustering algorithm reassigns each sample into a cluster based on some criterion. In each iteration the clusters are modified based on the reassigned samples. In this way the clustering result is gradually improved. [33]

The most popular partitioning relocation clustering method is called *K-means clustering*. [32] K-means clustering is a fast and simple method that is usually the first one to apply due to quick implementation [31]. It scales well to large datasets and it is guaranteed to converge. K-means clustering is technically applicable only with numerical features [33]. In K-means clustering, samples are divided into k clusters, $C_1, C_2, ..., C_k$, where k is a predefined positive integer. Each cluster is represented by a so called *centroid* that is the center of the cluster. Each cluster center lies in the same space with the samples but is not necessarily any of the samples. Initial cluster centers can be randomly chosen or defined based on a more advanced technique. Each sample is clustered based on the closest cluster center to the sample. After dividing the samples into clusters, each cluster center is changed to be the mean of the samples in the cluster. The approach in K-means clustering is to minimize the squared distance of samples to their closest cluster center. [34]

A commonly used solution for finding a local minimum in K-means clustering is introduced in [35] and referred to as *Lloyd's algorithm*. In Lloyd's algorithm cluster centers are initialized by randomly uniformly choosing k samples from the total sample set. The procedure in Lloyd's algorithm can be described by the following steps [34]:

1. Define the number of clusters, k.
2. Initialize the cluster centers, $\mu_k^{(0)}$.
3. Consider the j:th iteration. Assign each sample z to the cluster of its closest cluster center:
$$z \in C_k = \underset{C_k}{\operatorname{argmin}} ||z - \mu_k^{(j-1)}||,$$
where $||\cdot||$ is a distance measure.
4. Update each cluster center to be the mean of the samples in that cluster:
$$\forall k : \mu_k^{(j)} = \frac{1}{|C_k|} \sum_{z \in C_k} z,$$
where $|C_k|$ is the number of samples in cluster C_k.
5. Repeat steps 3 and 4 until the cluster centers do not change significantly:
$$\sum_k \left| \mu_k^{(j)} - \mu_k^{(j-1)} \right| < \varepsilon,$$
where ε is a predefined tolerance.

The above described algorithm can be efficiently run with different values for initialization parameters and the best result can be picked in order to avoid ending up in a poor local minimum [31, 30]. Also, more advanced cluster center initialization methods have been developed as random initialization may lead to a poor clustering result [32]. Initialization method introduced in [34] increases the speed and the accuracy of K-means algorithm. This modified version of K-means is referred to as *K-means++*. In K-means++ cluster centers are initialized according to specific probabilities. That is, a sample is chosen as a cluster center with a probability proportional to the distance from the sample to the existing cluster centers. The first cluster center is arbitrarily chosen among the samples. This initialization method is used in this work.

Definition 2.2.1 (Within Cluster Sum of Squares) *Denote a list of M samples by $Z = (z_1, z_2, ..., z_M)$. Let the samples be clustered into K clusters, $C_1, C_2, ..., C_K$. Within cluster sum of squares, W, is*

$$W = \sum_{k=1}^{K} \sum_{z_i \in C_k} ||z_i - \mu_k||^2,$$

where μ_k is the cluster center of the k:th cluster and $||\cdot||$ refers to a chosen distance/similarity measure.

Clusters are not defined in advance. It is not known beforehand how the samples should be divided into clusters. K-means clustering can end up in a different local minimum depending on the initialization, which points out that there are multiple ways to cluster the samples. Hence there needs to be a way to measure the validity of the clustering result. One way to measure the similarity of samples within clusters is *within cluster sum of squares* (WCSS). WCSS is defined in Def. 2.2.1. The smaller the value for WCSS, the more similar the samples within clusters are. K-means clustering is designed to minimize WCSS [31, 32]. By design of the K-means method, WCSS decreases as the number of clusters increases. Increasing the number of clusters comes with a cost of complexity. Because the goal of clustering is to simplify data, increasing the number of clusters complicates analysis and interpretation. The need to define the number of clusters in advance is one of the drawbacks in K-means clustering. One way to decide k is to plot WCSS against the number of clusters. Usually the curve drops rapidly at the beginning but decreases only a little when the values for k increase. Based on the curve, one can see the trade off between accuracy and complexity. The number of clusters can then be decided on a case-by-case basis.

2.3 Community Detection

Community detection is one of the most central topics in the study of complex networks [36]. Community detection means partitioning nodes in a network into groups or *communities* based on their linkage. The idea is that nodes within a community are highly linked to each other while nodes in different communities share only few links. Community detection is used to discover structurally related units not known in advance. [9] An example of a community is a group of friends in a social network of acquaintances. In community detection nodes are divided into disjoint subsets, a subset representing a group of actors or items that are densilty connected to each other. Community detection algorithms are designed to discover the optimal set of subsets based on some criterion. A common criterion is to maximize a so called *modularity*. Modularity is a measure of how inter-connected communities are compared to the connections between communities. Optimizing modularity is known to be computationally hard. Consequently, approximation algorithms are used for modularity based community detection for large networks. [37]

One of the often used modularity based community detection method is *Louvain* community detection. This method was first introduced in [37] to detect communities in large, weighted, undirected networks. There are solutions for other types of networks too, e.g. signed network community detection is discussed in [38]. In this work, the signs, weights and directions of edges are dropped in community detection. This is to extract communities of traders purely from the structural perspective. Knowledge of given trust ratings is discarded and edges are used to represent interactions between traders. In other words, peer review networks of Bitcoin traders are simplified to show which traders interact with each other without referring to the type of the interaction. Dropping the edge weights means in practice that each edge has a weight 1. Due to this simplification, more complex community detection methods are not needed. Louvain community detection is chosen in this work based on its efficiency, suitability and easy implementation.

Consider an arbitrary undirected, weighted network $G = (N, E, h((\cdot,\cdot)))$, where h is a mapping that assigns a weight for each edge (see Def.2.1.3). In our case

$$\forall (n_i, n_j) \in E : h((n_i, n_j)) = w_{i,j} = 1.$$

Denote the number of nodes by $|N|$. Louvain community detection has two iterative phases. First, each node is assigned to a different community. Hence there are $|N|$ communities at the beginning. The following notations are used in defining modularity:

- ω_i: the community of node n_i,
- $\delta(\omega_i, \omega_j)$: a function having value 1 if $\omega_i = \omega_j$ and 0 otherwise,
- η: the sum of all the edge weights, $\eta = \sum_{(n_i,n_j)\in E} w_{i,j}$,
- $\sum_{G,n_i}$: the sum of the weights of the edges between node n_i and its adjacent nodes in network G.

Modularity is calculated as $Q = \frac{1}{2\eta} \sum_{i,j} \left[w_{i,j} - \frac{\sum_{G,n_i} \sum_{G,n_j}}{2\eta} \right] \delta(\omega_i, \omega_j)$.

Per each node $n_i \in N$ the gain in modularity is calculated in cases of moving the node to the communities of its adjacent nodes. Node n_i is then moved to the community of the adjacent node that maximizes modularity gain. If modularity cannot be increased, node n_i is left in its current community. [37]

The above described notations together with the following notations are used in defining the gain in modularity:

- Ω: a community,
- $\sum_{\Omega}$: the sum of the edge weights within community Ω,
- $\sum_{G,\Omega}$: the sum of the weights of the edges in network G that are incident to the nodes in community Ω,
- $\sum_{\Omega,n_i}$: the sum of the weights of the edges between node n_i and its adjacent nodes in community Ω.

The gain in modularity, ΔQ, when a node n_i is moved to community Ω, can be calculated efficiently by

$$\Delta Q = \left[\frac{\sum_{\Omega} + \sum_{\Omega,n_i}}{2\eta} - \left(\frac{\sum_{G,\Omega} + \sum_{G,n_i}}{2\eta}\right)^2\right] - \left[\frac{\sum_{\Omega}}{2\eta} - \left(\frac{\sum_{G,\Omega}}{2\eta}\right)^2 - \left(\frac{\sum_{G,n_i}}{2\eta}\right)^2\right].$$

After moving each node according to the maximum modularity gain, the process starts again. Iterations stop when no more modularity gain can be achieved. In this way, a local maximum for modularity is found in the first phase. [37]

In the second phase, a new network is formed. In the new network nodes are the communities constructed in the first phase. Edge weights between the nodes in the new network are formed by summing up the weights of the edges between the communities. Since each community contains inter-connections, the new network contains self loops. The weight of the self loop edge of community Ω is the sum of the weights of the inter-connections in Ω. Using the above notations, the self loop weight of community Ω is $\sum_{\Omega}$. The first phase is then applied to the new network. That is, communities of communities are formed by making single changes at a time and stopping when modularity cannot be increased anymore. The two phases together are referred to as a *pass*. Louvain community detection iterates the passes until modularity changes no more. In this way, a local maximum for modularity is achieved in an iterative way. Louvain community detection is applicable to large datasets due to its efficiency. One of the drawbacks in Louvain community detection is that the communities formed in the first phase are sensitive to the order of the nodes. The impact of the order of the nodes on modularity is stated to be insignificant but computational time is assumed to be affected. [37]

2.4 Centrality

Centrality is a quantity that describes how important a node is in a network. Nodes can be ranked by centrality, and the received node hierarchy can serve as a tool for network analysis. Especially in social networks, centrality is used to highlight the most influential nodes. There are many ways to define centrality. *Degree centrality* refers to determining a node's importance based on how connected it is to other nodes. *Eigenvector centrality* advances degree centrality by considering how connected a node is to important nodes. [8] Some of the centrality methods have been developed in the context of internet search engines. Search engines rank web pages based on how well they fit the entry given by a web user. Algorithms such as *Hyperlink-Induced Topic Search* (HITS) [20], *Page Rank* [39] and its relatively recently modified version *Quantum Page Rank* [40] have all been developed for search engines. Yet, they are used as centrality measures in other contexts too [41, 42, e.g.]. There are also centrality measures that use other information in addition to network structure to define the most central nodes. For example, in the context of internet search engines, a modified version of Page Rank and HITS methods includes web page topics into centrality method [43].

In this work, only the directed edges are used to rank Bitcoin traders by their importance. The underlying rationale is that highly connected traders have the greatest influence on the peer review network because they have given and/or received many trust ratings. Incorporating edge weights, namely trust rating values, is not desired as the aim is to see how structural matters relate to the behaviour of the traders. Centrality measures that include additional content are not relevant in this case. Many centrality measures such as Page Rank assign one centrality value per each node in a network but HITS centrality measure divides centrality into two categories: hubs and authorities. Because the networks in this work are directed, HITS centrality method is suitable in this context. HITS is considered to be more informative than e.g. Page Rank because it provides two types of centrality measures.

HITS is introduced in [20] as a method for ranking web pages according to an input given by an internet search engine user. The idea behind HITS algorithm is that there are two types of important web pages: authorities that contain relevant information and hubs that point to authorities, i.e. tell where to find the information [9]. In the context of Bitcoin peer review networks authorities are traders that have received many trust ratings, especially from hubs. Authorities are known traders and a lot of information about their trustworthiness is available. Hubs in Bitcoin peer review networks are traders that have given ratings to many others, especially authorities. Hubs are traders that considerably contribute to the discussion of "with whom to trade".

HITS algorithm defines an authority and a hub values for each node in a network in an iterative way. The higher the hub (authority) value of a node is the more important hub (authority) the node is. Denote the hub value of node n_i by $\phi(n_i)$ and the authority value by $\theta(n_i)$. Denote by $N_{n_i,\text{out}}$ the set of nodes node n_i points to, and by $N_{n_i,\text{in}}$ the set of nodes that point to node n_i. A predefined threshold ε determines the upper limit for how much the values can change between consecutive iterations for the algorithm to stop. With these notations the HITS procedure finds hub an authority values per each node in a network the following way:

- Consider a directed network $G = (N, E)$.
- Initialize all hub and authority values to 1:

$$\forall n_i \in N : \phi^{(0)}(n_i) = \theta^{(0)}(n_i) = 1.$$

- Consider the k:th iteration. Per each node $n_i \in N$, the authority value of n_i from the $k-1$:th iteration is increased by the hub values of the nodes in $N_{n_i,\text{in}}$:

$$\forall n_i \in N : \theta^{(k)}(n_i) = \theta^{(k-1)}(n_i) + \sum_{n_j \in N_{n_i,\text{in}}} \phi^{(k-1)}(n_j).$$

- Per each node $n_i \in N$, the hub value of n_i is increased by the new authority values of the nodes in $N_{n_i,\text{out}}$:

$$\forall n_i \in N : \phi^{(k)}(n_i) = \phi^{(k-1)}(n_i) + \sum_{n_j \in N_{n_i,\text{out}}} \theta^{(k)}(n_j).$$

- Before the next iteration, the hub (authority) values of the nodes are normalized by dividing the values by the current maximum hub (authority) value.
- The algorithm iterates until the values change less than a certain tolerance ε. When the stopping criterion is met, hub (authority) values are normalized by the sum of the hub (authority) values over all nodes. The output of the HITS algorithm is the normalized hub and authority values from the most recent iteration.

Above described HITS procedure is presented after NetworkX Python package's implementation of HITS algorithm [44]. HITS algorithm that includes normalization is guaranteed to converge [45].

2.5 Hyper-Geometeric Test

Hyper-geometric test is a statistical test related to hyper-geometric distribution. Hyper-geometric distribution is a discrete distribution that presents probabilities for a number of successes. Following the example in [46], hyper-geometric distribution can be explained by considering an urn of k balls of which m are blue and $k-m$ are white. If K balls are drawn without replacement, the number of blue ones, M, is a a hyper-geometric random variable with parameters k, m and K. The probability of M successes, namely M blue balls, is

$$P(X = M) = \frac{\binom{m}{M}\binom{k-m}{K-M}}{\binom{k}{K}}.$$

The probability of observing M blue balls signals how exceptional the result is. In this sense, hyper-geometric distribution can be used to point out exceptional observations. Hyper-geometric test refers to testing over-representation or under-representation of a certain type of objects under the null-hypothesis of random occurrence [47] In case of over-representation, hyper-geometric distribution is used to determine the probability of having at least M successes,

$$P(M \leq X) = 1 - \sum_{i=0}^{M-1} P(X = i) = 1 - \sum_{i=0}^{M-1} \frac{\binom{m}{i}\binom{k-m}{K-i}}{\binom{k}{K}}.$$

In the example, hyper-geometric test has the null-hypothesis that at least M blue balls are observed as a result of randomly sampling K balls without replacement from an urn that contains m blue balls and k balls in total.

In statistical tests, a *significance level*, α, is the tolerance for falsely rejecting a null-hypothesis. For instance, $\alpha = 0.05$ means that there is less than 5% chance the null-hypothesis holds true even if it is rejected. Comparing probability $P(\cdot)$ to a given significance level determines if the null-hypothesis is rejected or not. In this example, $P(M \leq X) < 0.05$ would result in the rejection of the null-hypothesis. Under-representation is tested in a similar manner by calculating the probability of observing at most M samples of a certain type,

$$P(X \leq M) = \sum_{i=0}^{M} P(X = i) = \sum_{i=0}^{M} \frac{\binom{m}{i}\binom{k-m}{K-i}}{\binom{k}{K}}.$$

Hyper-geometric test can be used to determine if a result is statistically significant. Statistical significance in this context means rejecting the null-hypothesis that the observed result is due to random sampling from the population.

In case there are multiple hyper-geometric tests of the same event, it is crucial to adjust the significance level accordingly. This is to avoid falsely rejecting null-hypotheses. There are multiple ways to adjust the significance level, one of which is Bonferronig correction. [48] In Bonferroni correction the significance level, α, is divided by the number of tests [47]. In case the above described example is repeated 5 times, Bonferroni corrected significance level, $\hat{\alpha}$, would be

$$\hat{\alpha} = \frac{0.05}{5} = 0.01.$$

Bonferroni correction is stated to be conservative [47] in a sense that it quite significantly decreases α with respect to the number of tests. In this work the focus is on strong evidence due to the large extend of the topic. Research questions are answered from the perspective of highly exceptional observations. Therefore, Bonferroni correction is considered suitable.

3. FEATURE EXTRACTION

In this chapter the representation of the peer review networks and features derived from the trust rating data are explained. The features are used to assess the behaviour of Bitcoin traders in peer review networks. More advanced features are explained in more detail. The research methodology used in this thesis is explained and the methods introduced in the previous chapter are put into the context of answering the research questions.

3.1 Aggregated Network

One of the main questions in network analysis is whether the network is considered static or dynamic. Having time-stamped data does not necessarily mean that a dynamic interpretation would be the most suitable. The majority of social network analysis uses static networks. If a network is evolving over time relatively slowly it might be useful to interpret it as consecutive snapshots of the changing network. [9] That is, consecutive events of edges and nodes emerging and disappearing would be batched into a snapshot representing a dynamic network as a static aggregation over a certain time interval. A reason to choose such an approach is that analysis of slowly changing networks might not benefit from incorporating its dynamics to the extend to which it adds complexity.

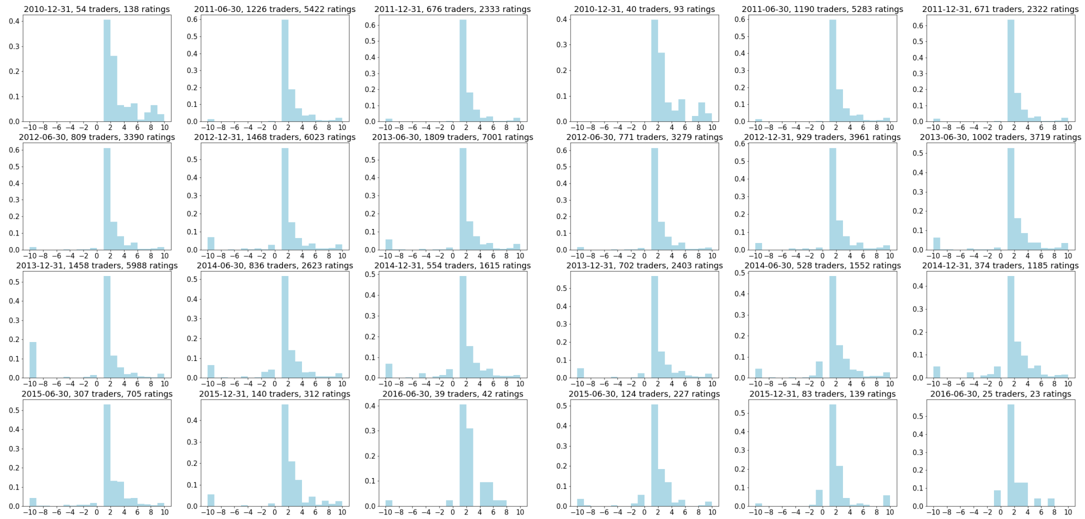
In this work, it is decided to aggregate the time-stamped data per half a year and represent the evolution over time of the network as a sequence of snapshots. In this way, both datasets cover 12 time-steps in total, starting from the second half of 2010 and ending after the first half of 2016. For example, aggregated data on date 2012-12-31 includes all ratings given from the 1st of July 2012 until and including 31st of December 2012. The time-steps are enumerated from 1 to 12. Each time-step corresponds to a time-stamp. For instance, the fifth time-step has a time-stamp 2012-12-31. In this way, the 12 aggregated networks correspond to non-overlapping time periods denoted by the end dates of the time periods.

Definition 3.1.1 (Aggregated Network) *Aggregated network corresponding to a time-stamp t is defined by*

$$G_t = (N_t, E_t, h_t),$$

where E_t is the set of edges and N_t is the set of traders present in the peer review network on a half a year time period ending at date t. Mapping $h_t((n_i, n_j))$ assigns a rating value for each edge $(n_i, n_j) \in E_t$.

Aggregated network is defined in Def. 3.1.1. The concept of weighted network (see Def.2.1.3) is applied in defining aggregated network. That is, there is a function associated to the network that maps each edge to a trust rating. In other words, $h_t((n_i, n_j))$ is the



(a) Bitcoin OTC **(b)** Bitcoin Alpha

Figure 3.1. *The sub-figures present the rating distributions on each time-step in (a) Bitcoin OTC and (b) Bitcoin Alpha. Trust ratings are integer numbers from -10 to 10 with a step of 1 excluding 0. The titles of the sub-figures show the number of traders, the number of ratings and the time-stamp of the aggregated network.*

rating value given by trader n_i to trader n_j in an aggregated network corresponding to a time-stamp t. As the two datasets used in this work are handled separately, there are two sequences of 12 aggregated networks.

Figure 3.1 shows the rating distribution in each aggregated network. It is evident that the overall rating behaviour does not change dramatically over time although the number of traders varies significantly. For example, in Bitcoin OTC there are less than 50 traders at the lowest and around 1800 at the highest. Highly negative values, namely -10 ratings, start to appear after the first half a year, increasing quite remarkably during the second half of 2012. In Bitcoin OTC -10 ratings cover nearly 20 percent of all the ratings during the second half of 2013 as seen in Fig. 3.1(a). In Bitcoin Alpha, the proportion of -10 ratings stays below 10% over all time-steps according to Fig.3.1(b). Slightly positive values, namely +1 and +2, seem to dominate in both datasets covering always more than 50 percent of the ratings.

3.2 Feature Vector

The datasets used in this thesis contain nodes and time-stamped weighted edges. Therefore they provide only the basic information to build the networks in the first place. However, there is a way to derive additional information from weighted edges. In recent studies, node behaviour is reduced to two novel features defined by the in-going and out-going weighted edges of a network. The same datasets that are used in this work are studied in the context of fraudulent user detection and link prediction. [16, 17] This work builds on top of the existing research by adding more features into the analysis. This thesis contributes to the research on online social networks by studying how the features evolve over time and relate to the topological quantities of peer review networks.

Features are derived from the trust rating data to extract trader behaviour. Trader behaviour refers to how traders rate each other and get rated by others in a peer review network. The numbers and the values of given and received ratings are used to describe the behaviour of the traders. Trader behaviour consists of five features, and the features are calculated separately for each trader in each aggregated network.

First, the lists of given and received ratings are defined.

Definition 3.2.1 (Given and Received Ratings) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. Denote the traders who have given ratings to n by $\beta_n^{(1)}, \beta_n^{(2)}, ..., \beta_n^{(M_n)}$, where $M_n = d_{in}(n)$ (see Def. 2.1.4). The list of ratings n has received from other traders is*

$$X_{n,t} = (h_t((\beta_n^{(1)}, n)), ..., h_t((\beta_n^{(M_n)}, n))).$$

Denote the traders who have received ratings from n by $\gamma_n^{(1)}, \gamma_n^{(2)}, ..., \gamma_n^{(K_n)}$, where $K_n = d_{out}(n)$ The list of ratings n has given to other traders is

$$Y_{n,t} = (h_t((n, \gamma_n^{(1)})), ..., h_t((n, \gamma_n^{(K_n)}))).$$

Definition 3.2.1 presents how the lists of given and received ratings are denoted. These lists are used in defining some of the features. One of the five features is *attention*.

Definition 3.2.2 (Attention) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. The attention of trader n is defined by the number of ratings n has received,*

$$a_t(n) = d_{in}(n).$$

The attention of a trader is defined by the number of ratings the trader has received as presented in Def. 3.2.2. Attention is used to describe how noticed a trader is by other traders. A trader with a high attention value has got feedback from many others about how trusted the trader is. Attention is zero at the lowest and $|N_t| - 1$ the highest, where $|N_t|$ is the number of traders in G_t. In case trader n has only given ratings to others and not received any, $a_t(n) = 0$. If trader n has been rated by all other traders in the network, $a_t(n) = |N_t| - 1$.

Another feature based on the received ratings is *reputation*. Reputation describes how trusted a trader is by other traders.

Definition 3.2.3 (Reputation) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. Let $X_{n,t}$ be the list of ratings n has received as defined in Def. 3.2.1. The reputation of trader n is defined by the average of the ratings n has received,*

$$r_t(n) = \begin{cases} \frac{1}{d_{in}(n)} \sum_{x \in X_{n,t}} x, & \text{if } d_{in}(n) \neq 0 \\ 0, & \text{otherwise}. \end{cases}$$

The reputation of a trader, defined in Def.3.2.3, is the average of the ratings the trader has received. As the rating values range from -10 to 10, reputation is -10 the lowest and 10 the highest. The higher the reputation the more trusted the trader is. If a trader has no received ratings, reputation is set to zero.

Per each aggregated network all the ratings given by a trader define the trader's *activity*. Activity measures how actively a trader gives ratings to others.

Definition 3.2.4 (Activity) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. The activity of trader n is defined by the number of ratings n has given,*

$$c_t(n) = d_{out}(n).$$

As defined in Def.3.2.4, the activity of a trader is the number of ratings the trader has given to other traders. Activity is zero the lowest and $|N_t| - 1$ the highest. In case a trader has only received ratings, activity is zero.

The above described features are intuitive in a sense that they are easy to calculate and interpret, and they provide information about traders' behaviour in peer review networks. In addition to attention, reputation and activity, more advanced features are used to capture how in line the rating behaviour of a trader is with other traders. Fairness and goodness reflect how a trader's received and given ratings associate to the *general opinion* [16]. The general opinion in this context means taking all the ratings in an aggregated network into account. The underlying assumption is that *the wisdom of crowd* is not biased. In other words, taking all the ratings into account, it is assumed to be possible to extract how fair and good the traders are.

Definition 3.2.5 (Fairness and Goodness) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. Let $X_{n,t}, Y_{n,t}$ be the lists of received and given ratings respectively as defined in Def. 3.2.1. In accordance with Def. 3.2.1, the number of elements in $X_{n,t}$ and $Y_{n,t}$ is M_n and K_n respectively. Let the ratings be scaled to range from -1 to 1. The fairness, f_t, and the goodness, g_t, of n are defined by [16]*

$$f_t(n) = 1 - \frac{1}{K_n} \sum_{i=1}^{K_n} \frac{\left| h_t((n, \gamma_n^{(i)})) - g_t(\gamma_n^{(i)}) \right|}{2},$$

$$g_t(n) = \frac{1}{M_n} \sum_{i=1}^{M_n} f_t(\beta_n^{(i)}) h_t((\beta_n^{(i)}, n)).$$

Fairness and goodness are defined in 3.2.5. Fairness measures how fairly a trader has given ratings to other traders. Goodness presents *fairness weighted reputation*. By definition the value range for goodness is from -1 to 1 and for fairness from 0 to 1 due to scaled rating values. As shown in the definition, fairness is used to define goodness and the other way around. These features are calculated in an iterative way from one another. [16]

The algorithm for calculating fairness and goodness values is presented in [16] and its Python implementation can be downloaded from [49]. In this work, the ratings are scaled to range from -1 to 1, and the algorithm takes an aggregated network as input. The algorithm returns the fairness and goodness of each trader in a network. At the beginning, all traders are given the maximum fairness value. Therefore, the underlying procedure is to interpret traders as maximally fair unless otherwise can be stated based on the given ratings. The goodness of a trader is initialized by the average of the ratings received by the trader. If a trader has only given ratings and not received any, the value for goodness is zero. In this way, the initial values for goodness equal reputation, when reputation is calculated using scaled ratings. The algorithm iterates until the sums of the differences of fairness and goodness values to the values on the previous iteration are less than a predefined tolerance, ε. In this work, $\varepsilon = 10^{-6}$.

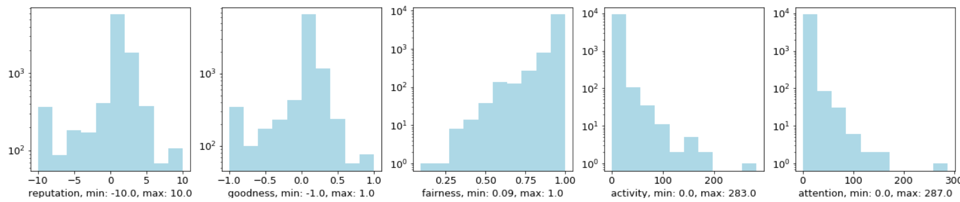
Per each iteration, goodness of trader n is the average over the ratings n has received weighted by the raters' fairness from the previous iteration. The updated goodness values are then used to update each trader's fairness. That is, each rating trader n has given is compared to the goodness values of the rated traders. The fairness of trader n is then the maximum fairness value, namely 1, decreased by the average amount of which the ratings given by n differ from the goodness values of the rated traders. The goodness of a trader captures how trusted the trader is especially by fair traders. The fairness of a trader describes how in line with the general opinion the trust ratings given by the trader are. In case a trader has no received ratings, goodness is set to zero, thereby in line with reputation. In case a trader has not given any ratings, fairness is set to 1.

The above described five features are gathered per each trader in an aggregated network to form traders' feature vectors.

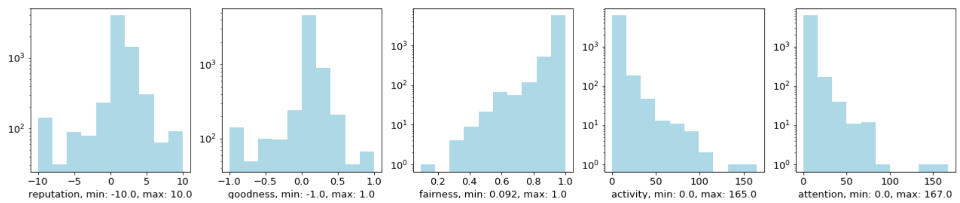
Definition 3.2.6 (Feature Vector) *Consider an aggregated network $G_t = (N_t, E_t, h_t)$. Let n be an arbitrary trader, $n \in N_t$. The feature vector of trader n is:*

$$z_t(n) = \begin{bmatrix} a_t(n) \\ r_t(n) \\ c_t(n) \\ f_t(n) \\ g_t(n) \end{bmatrix},$$

where $a_t(\cdot)$, $r_t(\cdot)$, $c_t(\cdot)$, $f_t(\cdot)$ and $g_t(\cdot)$ are the attention, reputation, activity, fairness and goodness of the trader respectively.



(a) Bitcoin OTC



(b) Bitcoin Alpha

Figure 3.2. *The sub-figures present the unscaled feature values over all traders and time-steps for (a) Bitcoin OTC and (b) Bitcoin Alpha. The feature value distributions are shown separately per each feature: reputation, goodness, fairness, activity and attention starting from left. The captions show the names, and the minimum and maximum values of the features. The feature values are shown on the x-axis and the number of traders is shown on the y-axis.*

Feature vector is defined in Def. 3.2.6. If a trader is present on more than one of the time-steps, there is a sequence of chronological feature vectors that represent how the trader's behaviour evolves over time. Representing traders by feature vectors serves as a method to study how the traders behave with respect to each other and how their behaviour evolves over time.

The unscaled feature values over all traders and all 12 aggregated networks are shown in Fig. 3.2. Based on Fig. 3.2, the most of the traders share similar feature values. First impression is therefore, that the most of the traders behave similarly. Yet, it is the combination of features that describes trader behaviour. Also, there are values pointing out differing and adverse behaviour. That is, based on e.g. Fig. 3.2(b) traders with highly negative goodness are observed in the data. In addition, a small group of traders are notably more active than the rest. This is also the case with attention.

Trader behaviour is reduced to a 5 dimensional feature vector (Def. 3.2.6) derived from the trust rating data. Due to anonymity and having data only on trust ratings this is a problem of partially hidden information. As there is no ground truth available about the behaviour of the traders, the analysis is based on relative rather than absolute values. Feature value quantiles are used to extract different types of traders. For instance, unfair traders are those among the lowest with respect to fairness. Active traders in turn are those among the highest with respect to activity. Dividing feature values into quantiles is discussed more in the next chapter.

3.3 Feature Correlations

As a first step to address the behavioural features, feature correlation is calculated. This is to see if there exists linear dependence between the features.

Definition 3.3.1 (Pearson Correlation between Features) *Denote the lists of the values of two features by $V_i = (v_{i,1}, v_{i,2}, ..., v_{i,M})$ and $V_j = (v_{j,1}, v_{j,2}, ..., v_{j,M})$. Pearson correlation, ρ, between the features is*

$$\rho(V_i, V_j) = \frac{\sum_{k=1}^{M}(v_{i,k} - \mu_i)(v_{j,k} - \mu_j)}{\sigma_{v_i}\sigma_{v_j}},$$

where μ_i and μ_j are the means,

$$\mu_i = \frac{1}{M}\sum_{k=1}^{M} v_{i,k},$$

and σ_i, σ_j are the standard deviations,

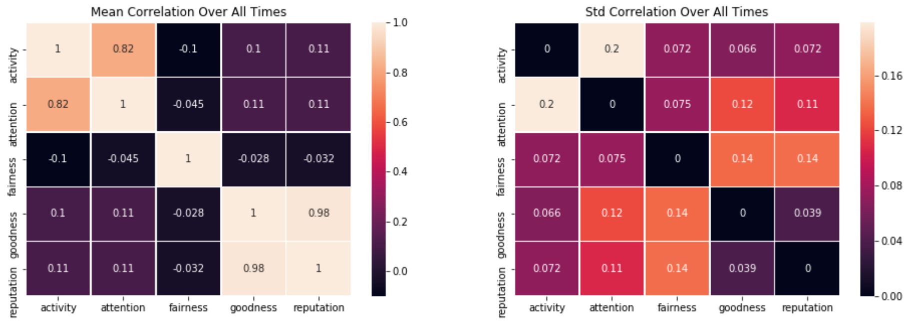
$$\sigma_i = \sqrt{\sum_{k=1}^{M}(v_{i,k} - \mu_i)^2},$$

of the feature values in V_i and V_j respectively.

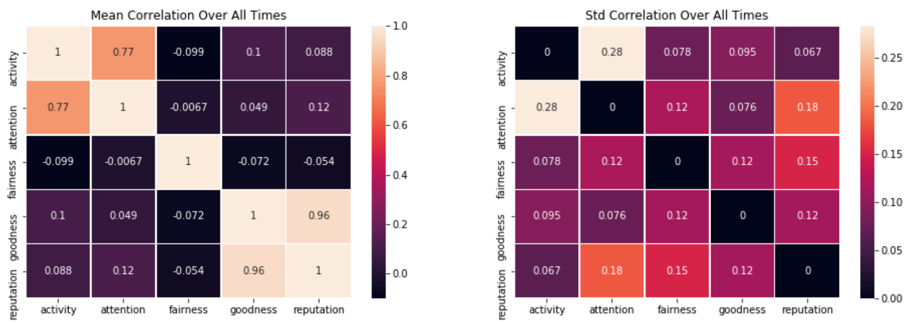
Pearson correlation, defined in 3.3.1, is a continuous measure ranging from -1 to 1. It represents how much variables linearly depend on each other. Correlation value 0 means that the variables do not show any linear relationship. Velues 1 and -1 refer to the highest possible positive and negative linear dependency respectively. Pearson correlation does not uncover nonlinear relationships. [50]

Feature correlations on the same time-step, averaged over all time-steps, are shown in Fig. 3.3. The correlations are similar between the datasets. Reputation correlates highly positively with goodness, which is expected from the way goodness is constructed. Also, attention and activity show high positive correlation, yet correlation between them varies the most. Interestingly, the mean correlation between fairness and the other features is negative. Negative correlation between fairness and activity might stem from construction since traders with no given ratings are given the maximum fairness.

Correlations of the current feature values with the previous ones are shown in Fig. 3.4. The correlation values are calculated per each time-step by taking the traders who exist also on the next time-step and averaging correlations over time. As shown in the Fig. 3.4, the current feature values mainly correlate with their own previous values. The correlations with other features do not differ significantly from the correlations in Fig. 3.3 that show the current feature correlations. The fact that the a features on time-step *i* correlate with their own values on time-step $i+1$ indicates that trader behaviour is persistent as the corresponding time periods are not overlapping. Due to averaging over all traders and



(a) Bitcoin OTC



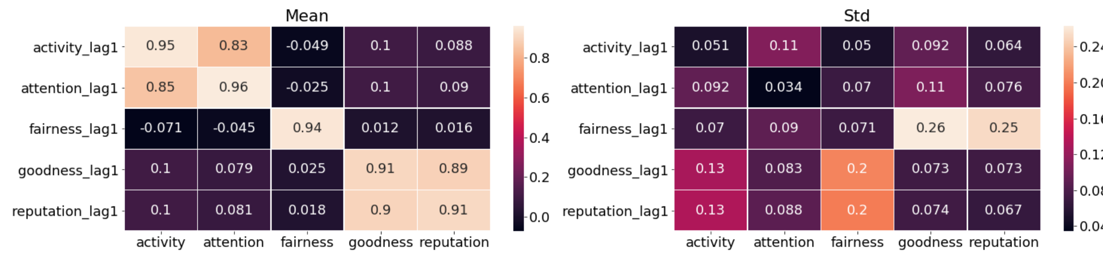
(b) Bitcoin Alpha

Figure 3.3. *Correlation between the features is calculated per each of the* 12 *time-steps. Heat maps show the mean (left) and the standard deviation (right) of the correlations over the time-steps for (a) Bitcoin OTC and (b) Bitcoin Alpha. The color bars next to each sub-figure show how the colors correspond to the correlation values.*

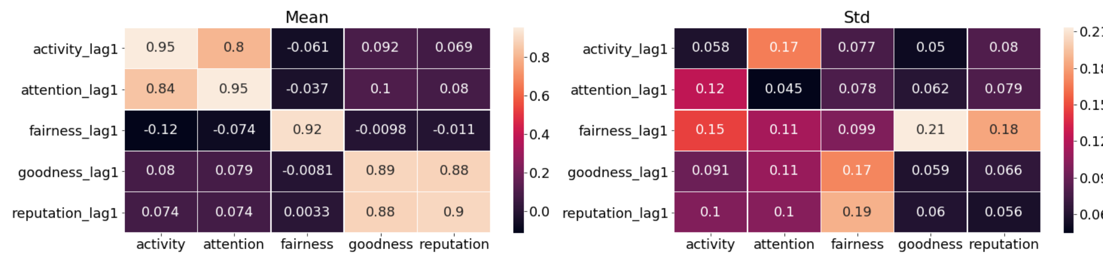
time-steps, correlation can only be thought as the first glimpse into the behaviour of the traders. To conclude, based on the correlations, most of the features do not depend linearly on each other. An exception that does not stem from the way the features are constructed, is that attention positively correlates with activity. The results are similar between the datasets.

3.4 Methodology

Modeling can be about *explaining* or *predicting* [51]. The focus of this thesis is on explaining. This work contributes to the research on social trust and distrust networks. These types of networks have been studied in the context of for example link prediction [16, 52] and fraud detection [17, 26]. One a general level, the focus of this work is on studying human behaviour in the context of receiving and giving feedback. More specifically, this work analyzes Bitcoin trader behaviour in peer review networks. The analysis focuses on observing how trader behaviour, especially adverse behaviour, evolves over time. By adverse behaviour it is meant that a trader gives unfair ratings to others or receives negative ratings from other traders, especially from fair traders. The main



(a) Bitcoin OTC



(b) Bitcoin Alpha

Figure 3.4. *Correlation between feature values on consecutive time-steps is shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The columns show how each feature correlates with the previous values of the features. Suffix '_lag1' is added to the names of the previous features. The sub-figures show the mean (left) and the standard deviation (right) of the correlations. The color bars next to each sub-figure show how the colors correspond to the correlation values.*

trader types are extracted by clustering the feature vectors using K-means clustering. A behavioural cluster is assigned to each trader in each aggregated network. Clustering serve as a method to raise abstraction level in order to understand trader types without explaining individual cases. Part of the information is lost in clustering as a trader belonging to a certain cluster might not be fully represented by the properties of the cluster. Yet, this method allows to extract more general observations on trader behaviour.

The consequences of traders' behaviour in Bitcoin peer review networks are analysed from the perspective of the behavioural clusters. Over the aggregated networks, traders can move from one cluster to another, stay in the same cluster or leave the peer review network. Per each trader a *cluster pattern*, that is a chronological sequence of cluster transitions, is recorded. The consequences of trader behaviour are then analyzed from the perspective of exceptional cluster transitions. Exceptional cluster transitions are found using hyper-geometric test. For example, a consequence of adverse behaviour could be that an unfair trader loses his/her own reputation. This would indicate that unfairness is punished by reputation lost. The functionality of Bitcoin peer review networks can be questioned in case the results show no consequences for adverse behaviour.

It is also studied how receiving unfair ratings from others impacts a trader's behaviour. The difference between a trader's reputation and goodness is used to determine if the trader is comparably unfairly rated. Exceptional cluster transitions among the unfairly rated traders are then analysed to see how these traders react to unfair treatment. A reaction could be for example that an unfairly rated trader rates others more unfairly afterwards. This would indicate that unfairly rated traders are likely to retaliate unfairness. In an online Bitcoin trading platform traders only see the given trust ratings, which provide information about traders' reputation. Based on the trust ratings, a trader can then decide with whom to trade. Fairness of a trader cannot be seen from the trust ratings as clearly as reputation, which in a sense makes it a hidden feature for traders. It is interesting to explore, how receiving unfair ratings affects a trader's behaviour, as it points out the impact of unfair traders.

In addition, it is studied how the behavioural clusters relate to certain topological quantities of peer review networks. An emotional and intuitive response of a trader to receiving unfair or negative ratings could be that the trader gives back similar ratings. Reasons for such a reaction could be mutual distrust or the need to retaliate undesired treatment. It is explored if there are communities of adversely behaving traders, as it would mean these traders have been rating each other. A community of adversely behaving traders would indicate that unfair or negative ratings go both ways. In case adversely behaving traders do not form communities, the datasets would not provide evidence for mutual distrust or retaliation. Communities are extracted per each of the 12 aggregated network using Louvain community detection algorithm. The fourth research question: *"Are there communities of adversely behaving traders in Bitcoin peer review networks?"*, is answered by analyzing how traders in the clusters of adverse behaviour are divided into communities. Hyper-geometric test is used to highlight the communities with exceptional cluster proportions. For example, consider a cluster C_j contains highly disreputable traders. In case a community containing unusually many traders from C_j existed in a network, a community of mutual distrust would be found in the network.

It is also studied how centrality relates to the behavioural clusters. HITS centrality algorithm is used to define hub and authority values per each trader in each aggregated network. The impact of a trader's previous behaviour on centrality in the next time-step is assessed by studying the cluster proportions among the most and the least central traders. By this way it is possible to evaluate if a certain type of behaviour serves as a strategy to gain centrality. For instance, one could expect that reputable traders would end up among the highest in centrality unusually often. Also, the current clusters of traders with the highest and the lowest possible hub and authority values are studied to see how centrality relates to traders' current behaviour. By construction, traders with high activity and attention values are likely to be the hubs and the authorities in the network. It is interesting to see what kind of a role for example fairness and goodness play in centrality. Consider that clusters C_i and C_j contain active traders so that traders in C_i are ungood and traders in C_j are good. If the hubs in a network were unusually often from cluster C_j but not so for cluster C_i, it would indicate that goodness is positively related to centrality. The fifth research question: *"How does centrality relate to traders' behaviour in Bitcoin peer review networks?"*, is answered by finding the over-represented clusters among the most and the least central traders.

In short, trust rating data is used to extract features that form a trader's feature vector. The feature vectors are used in clustering the traders based on their behaviour. The focus of this work is mostly on adversely behaving traders and unfairly rated traders. Clusters in aggregated networks provide a way to analyze exceptional cluster transitions. Communities in each aggregated network are extracted. Cluster proportions in communities are reviewed to see if there exists communities of adversely behaving traders. Cluster proportions are also studied to see if a certain type of behaviour leads to being among the most or the least central traders. Statistical significance of the results is tested using hyper-geometric test. In the next chapter, the implementation of the methods is explained, and the results are presented.

4. TRADER BEHAVIOUR IN BITCOIN PEER REVIEW NETWORK

The results of clustering the traders in the peer review networks based on their feature vector values are presented and discussed in this chapter. Transition matrices are formed from the cluster patterns of the traders to see how the behaviour of the traders change over time. Hyper-geometric test is used to highlight exceptional cluster transitions. The results stem from the two datasets, Bitcoin OTC and Bitcoin Alpha, introduced earlier.

4.1 Trader Clustering

The concept of clustering is previously discussed in section 2.2. In this section, clustering is applied in practice. As mentioned earlier, clustering is about grouping samples based on their similarity according to a similarity measure. Euclidean distance is used as a similarity measure in this work.

Definition 4.1.1 (Features' Within Cluster Sum of Squares) *Calculate a feature vector (see Def. 3.2.6) per each trader in each aggregated network. Let* $Z = (z_1, z_2, ..., z_M)$ *be the list of all feature vectors enumerated from* 1 *to* M*. Denote the* j*:th component in the* i*:th feature vector by* $z_{i,j}$*. Let the feature vectors be clustered into* K *clusters,* $C_1, C_2, ..., C_K$*.*

In accordance with Def. 2.2.1, within cluster sum of squares, W*, is*

$$W = \sum_{k=1}^{K} \sum_{z_i \in C_k} \sum_{j=1}^{5} (z_{i,j} - \mu_{k,j})^2,$$

where $\mu_{k,j}$ *is the average of the* j*:th feature of the traders in cluster* C_k*.*

The value for WCSS is used to decide the number of clusters. WCSS is defined in the context of the feature vectors in Def. 4.1.1. In Def. 4.1.1 WCSS is defined using squared Euclidean distance measure as done in [53].

To answer the research question:

> What kind of behavioural clusters are formed from Bitcoin traders?,

K-means clustering is applied to extract behavioural clusters among traders. All the features in this work are numerical and thus K-means clustering can be used. K-means clustering is not applicable to very high dimensional data due to convergence of distance. In this case, data is five dimensional, thus K-means can be used. In the rest of this work, K-means clustering with K-means++ initialization is used and referred to as *K-means*.

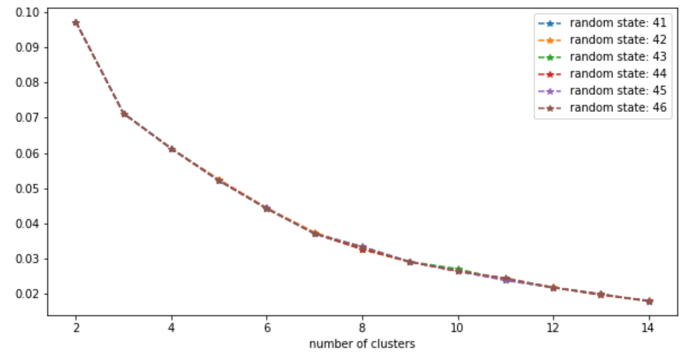


Figure 4.1. *The curves show WCSS averaged over all samples with respect to the number of clusters. Different curves correspond to different values for random state as presented in the legend. The results are derived using Bitcoin Alpha dataset.*

quantile	reputation	activity	attention	goodness	fairness	label
0.05	-0.580000	-1.000000	-0.993031	-0.558544	0.524775	very low
0.15	0.000000	-1.000000	-0.993031	0.000000	0.800690	low
0.25	0.100000	-0.992933	-0.993031	0.093781	0.867104	medium low
0.50	0.100000	-0.992933	-0.986063	0.098185	0.936838	medium
0.75	0.200000	-0.978799	-0.979094	0.181127	0.999255	medium high
0.85	0.233333	-0.957597	-0.958188	0.214986	1.000000	high
0.95	0.400000	-0.893993	-0.902439	0.374131	1.000000	very high

(a) Bitcoin OTC

quantile	reputation	activity	attention	goodness	fairness	label
0.05	-0.266667	-1.000000	-0.988024	-0.334045	0.633753	very low
0.15	0.100000	-1.000000	-0.988024	0.086080	0.818928	low
0.25	0.100000	-0.987879	-0.988024	0.095274	0.875416	medium low
0.50	0.100000	-0.987879	-0.976048	0.098936	0.935318	medium
0.75	0.200000	-0.951515	-0.952096	0.190639	0.995952	medium high
0.85	0.260000	-0.927273	-0.928144	0.237776	1.000000	high
0.95	0.450000	-0.818182	-0.843114	0.400849	1.000000	very high

(b) Bitcoin Alpha

Figure 4.2. *Labeled feature value quantiles in (a) Bitcoin OTC and (b) Bitcoin Alpha. Feature values are scaled to range from -1 to 1.*

To decide the number of clusters, k, multiple values for k are tried. Also the value for *random state* parameter, that affects cluster center initialization, is varied. Due to K-means++ initialization method it is expected that the random state parameter does not have a significant impact on clustering result. WCSS is calculated per each clustering result. As expected and shown in Fig. 4.1, WCSS decreases as the number of clusters increases, while random state does not have an effect. Increasing the number of clusters complicates analysis and decreases distinctiveness of the clusters. After trying various numbers of clusters, 10 clusters are considered to provide sufficient amount of information while maintaining easy interpretation.

Clustering is done separately for both datasets. In each aggregated network, a feature vector is calculated for each trader according to Def. 3.2.6. All the feature vectors are used in clustering. The features are scaled to range from -1 to 1 as K-means is sensitive to feature value ranges. Without scaling, features such as attention and activity would end up having the most impact on the clustering result due to their larger value range compared to the other features.

In order to describe each cluster, the cluster centers are labeled using labeled feature value

cluster	reputation	activity	attention	goodness	fairness	number
0	low	medium high	high	low	very low	139
1	medium	medium	medium	medium	medium	4399
2	very low	medium	medium	very low	medium	367
3	high	medium	medium high	high	medium	906
4	low	medium	medium high	low	medium low	313
5	medium	medium	medium	medium	very low	315
6	medium	very high	very high	medium	medium	84
7	very high	medium low	medium	very high	low	150
8	medium	high	high	medium	medium low	2001
9	low	medium	medium	low	medium	702

(a) Bitcoin OTC

cluster	reputation	activity	attention	goodness	fairness	number
0	medium	medium	medium	medium	medium	3266
1	very low	medium	high	very low	very low	43
2	very low	medium	medium	very low	medium	174
3	very high	medium	medium	very high	low	124
4	medium	high	high	medium	medium low	1236
5	medium	very high	very high	medium	medium	125
6	medium	medium	medium	medium	very low	146
7	medium low	medium	medium	low	medium	504
8	high	medium	medium	high	medium	680
9	very low	medium low	medium	very low	medium	141

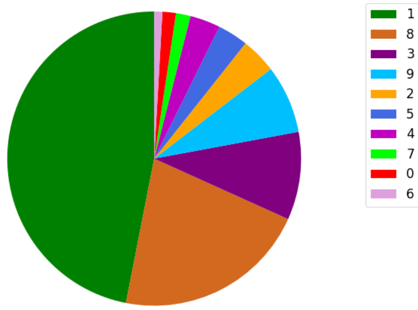
(b) Bitcoin Alpha

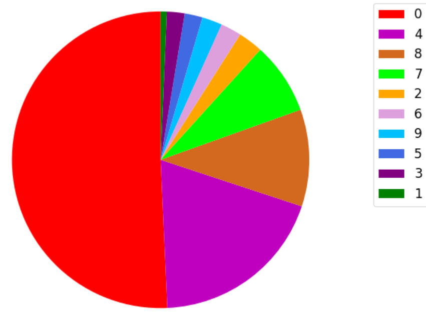
Figure 4.3. *K-means results with 10 clusters for (a) Bitcoin OTC and (b) Bitcoin Alpha. The clusters are enumerated from 0 to 9. Cluster centers with respect to each five features are labeled based on the labeled quantile values in Fig. 4.2. 'Number' shows the number of traders in each cluster over all time-steps.*

quantiles. Figure 4.2 shows how the quantile values correspond to the labels. Consider the j:th feature and the k:th cluster. Cluster center labeling is done the following way:

- Cluster center value $\mu_{k,j}$ is given the label *'medium'*, if it is greater than or equal to the 0.25 quantile and lower than or equal to the 0.75 quantile of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *medium low*, if it is lower than the 0.25 quantile and greater than or equal to the 0.15 of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *low*, if it is lower than the 0.15 quantile and greater than or equal to the 0.05 quantile of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *very low*, if it is lower than the 0.05 quantile of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *medium high*, if it is greater than the 0.75 quantile and lower than or equal to the 0.85 quantile of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *high*, if it is greater than the 0.85 quantile and lower than or equal to the 0.95 quantile of the j:th feature.
- Cluster center value $\mu_{k,j}$ is given the label *very high*, if it is greater than the 0.95 quantile of the j:th feature.

If there are repeating quantile values, as is the case for the 0.05 and 0.15 quantiles of activity in Bitcoin OTC (see Fig. 4.2(a)), the milder label is used. In other words, the label corresponding to the quantile that is closer to the 0.5 quantile of the feature is used. In this example, cluster center labels for activity cannot have the label *very low* in Bitcoin OTC, and the label for value -1 is *low*. This is to assure that the labels corresponding to the top and the least most quantiles do mean extreme feature values. In the rest of the analysis of the clusters, being disreputable refers to belonging to a cluster which label for reputation is 'medium low', 'low' or 'very low'. Being reputable refers to belonging to a cluster which





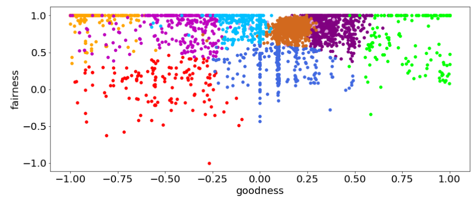
(a) Bitcoin OTC **(b)** Bitcoin Alpha

Figure 4.4. *Proportion of traders in each cluster in (a) Bitcoin OTC and (b) Bitcoin Alpha. The color of each cluster is shown in the color bars.*

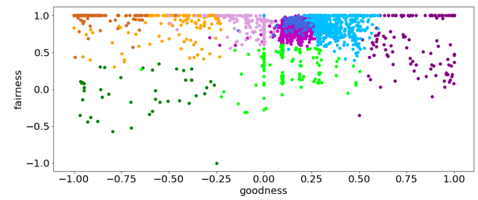
label for reputation is 'medium high', 'high' or 'very high'. Similar logic applies to other features too, e.g. being ungood versus good. Because none of the clusters have fairness label 'medium high', 'high' or 'very high', traders in the clusters of medium fairness are called *not unfair* to distinguish then from unfair traders. The labeled cluster centers as well as the number of traders in each cluster are shown in Fig. 4.3. To better illustrate the sizes of the clusters, pie chart representations are shown in Fig. 4.4.

As can be seen in Figure 4.4, in both datasets there is one main cluster into which roughly half of the traders belong to. The four largest clusters cover more than 75% of the traders. The clusters show similarities between the two datasets. Expectedly, the largest cluster has *medium* labels. The largest cluster is referred to as *the main cluster* and traders belonging to it are referred to as *medium traders*. As seen in Fig. 4.3, the second largest cluster in both datasets includes active, unfair traders who gain attention. The third largest cluster contains reputable and good traders but differs a bit in terms of attention between the datasets. In Bitcoin OTC, this cluster has traders who gain attention, while in Bitcoin Alpha, cluster center for attention is 'medium'. Again, the fourth largest cluster has ungood, disreputable traders in both of the datasets. Interestingly, the cluster centers lie on the low side of the fairness values. Considering the number of possible cluster centers given that there are 7 labels and 5 features, the cluster centers as substantially similar between the datasets.

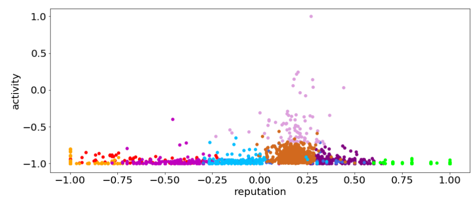
To illustrate the clustering result, some of the features are plotted against each other in Fig. 4.5. In the figure, samples belonging to the same cluster have the same color. The main cluster is excluded from the figure to add clarity and to introduce trader types that differ from the medium behaviour. As can be seen in Fig. 4.5, the most of the clusters are relatively distinct from each other, yet some overlapping can be seen e.g. in Fig.4.5(a). Due to 5-dimensional feature vectors it is not possible to draw the clustering result in total. Based on the labeled cluster centers in Fig. 4.3, the clusters are distinct in both datasets.



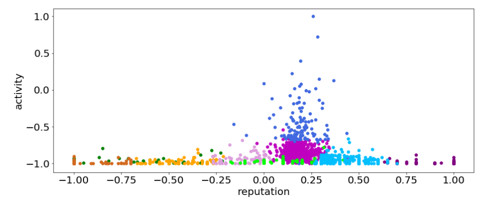
(a) Fairness and goodness, Bitcoin OTC



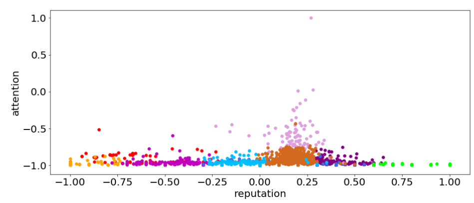
(b) Fairness and goodness, Bitcoin Alpha



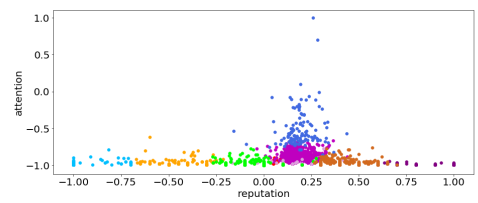
(c) Activity and reputation, Bitcoin OTC



(d) Activity and reputation, Bitcoin Alpha



(e) Attention and reputation, Bitcoin OTC



(f) Attention and reputation, Bitcoin Alpha

Figure 4.5. *K-means clustering result with respect to the features denoted in the sub-figure captions. The main cluster that contains medium traders is excluded.*

To compare the clusters between the datasets in terms of *fairness* and *goodness*, they are ordered by size and shortly described in Table 4.1. This is to extract interesting clusters from the perspective of adverse behaviour. In this way, it is easier to see that the six largest clusters have similar trader types in terms of goodness and fairness. That is, there are clusters of unfair, ungood and good traders. Although the remaining four clusters differ by the order between the datasets, one can see from the table that similar trader types are present. Due to describing the clusters on a high level this is somewhat expected. The table is used to facilitate interpretation of the cluster transition results presented in the next section.

4.2 Cluster Transitions

Cluster transition means that a trader switches from a cluster to another or stays in the same cluster. Consider an arbitrary trader n who is present in a peer review network on time-steps i and j so that $i < j$. Trader n is in cluster C_k on the i:th time-step and in cluster $C_{k'}$ on the j:th time-step. There is then a cluster transition from cluster C_k to cluster $C_{k'}$. Note that it is possible that $k = k'$.

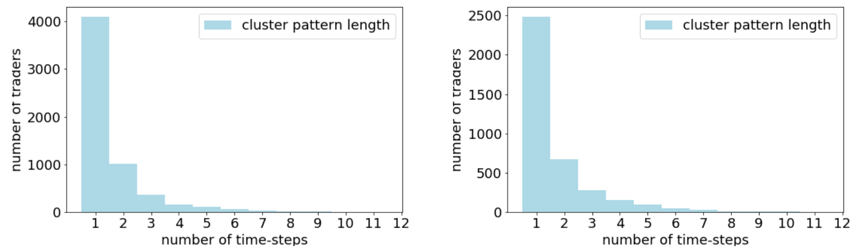
To see how the behaviour of the traders evolves over time, cluster transition matrix is

Table 4.1. *Clusters ordered by size are shortly described in terms of fairness and goodness separately for Bitcoin OTC and Bitcoin Alpha datasets. Column 'Number' shows how many traders there are in the cluster. 'Description' shortly outlines the type of traders in the clusters so that 'ungood' and 'unfair' refer to having the labels 'very low', 'low' or 'medium low' for goodness and fairness respectively. 'Good' and 'fair' refer to having the label 'medium high', 'high' or 'very high'. Medium labels are left unnoted.*

Dataset	Cluster	Number	Description
Bitcoin OTC	1	4399	-
	8	2001	Unfair
	3	906	Good
	9	702	Ungood
	2	367	Ungood
	5	315	Unfair
	4	313	Unfair and ungood
	7	150	Unfair and good
	0	139	Unfair and ungood
	6	84	-
Bitcoin Alpha	0	3266	-
	4	1236	Unfair
	8	680	Good
	7	504	Ungood
	2	174	Ungood
	6	146	Unfair
	9	141	Ungood
	5	125	-
	3	124	Unfair and good
	1	43	Unfair and ungood

formed by taking each trader's sequence of clusters and recording the transitions. Each trader n is present on $\tau(n)$ of the 12 time-steps. For each trader there is therefore a cluster pattern length ranging from 1 to 12, that illustrates how the behaviour of the trader evolves over time. If the last time-step a trader is seen in the data is not the last time-step of the data, the trader is interpreted to disappear from the network. Disappearing from the peer review network can be considered to indicate disappearing from the trading platform. All the same, it could be that a trader continues to trade Bitcoins without receiving or giving any further ratings. For that matter, it is noted that leaving the peer review network only means not being part of the *with whom to trade* -discussion anymore. Disappearing is denoted as if it would be a cluster represented by the number -1. The number of transitions are normalized to illustrate probabilities to move from one cluster to another.

Figure 4.6 shows the cluster pattern lengths over all traders. It is clear that the most of the traders are present only once, meaning that all the ratings they have received and given have happened during one half a year time period. There are significantly less traders that are seen in more than one of the aggregated networks. Consequently, it is expected that disappearing from the network dominates the cluster transitions. Transition matrices are also formed by not taking disappearance into account to illustrate better how



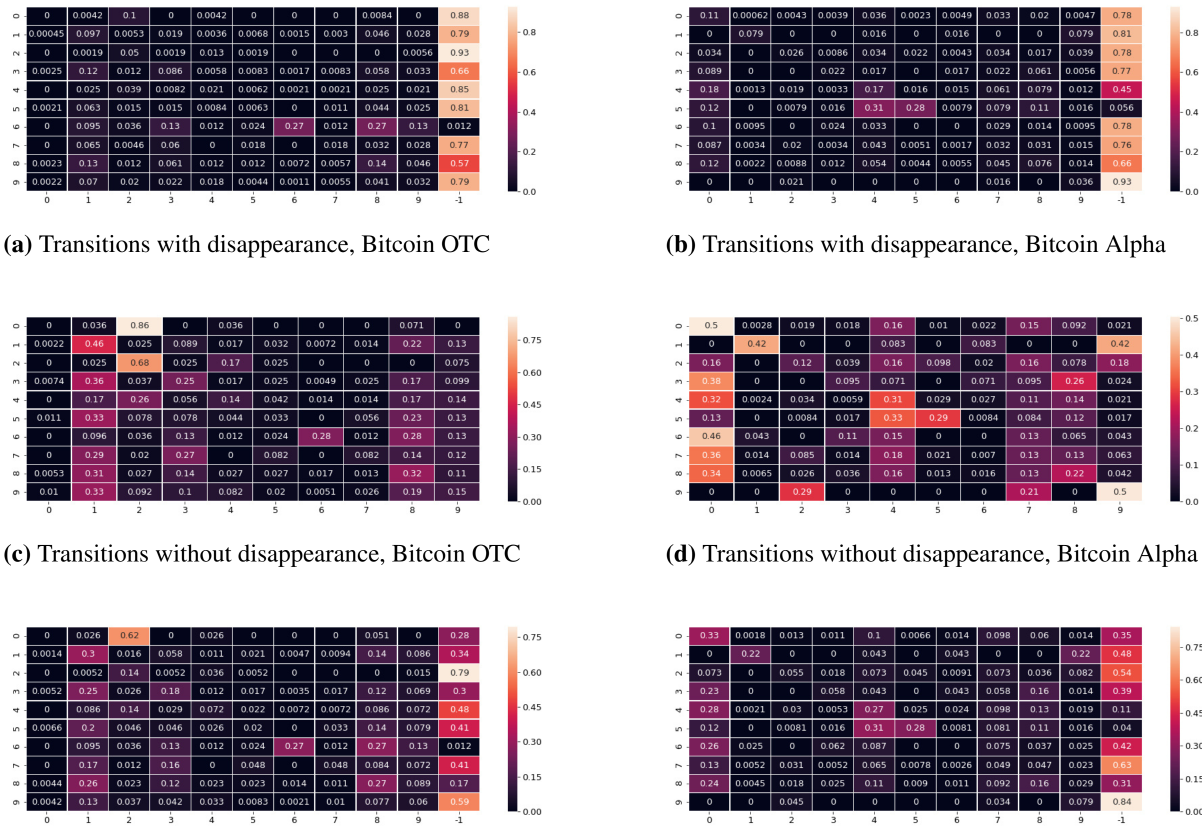
(a) Bitcoin OTC **(b)** Bitcoin Alpha

Figure 4.6. *A histogram of the cluster pattern lengths is shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The cluster pattern length of a trader is the number of time-steps the trader is present in a peer review network. The value range for cluster pattern length is from 1 to 12. Y-axis shows the number of traders, and x-axis shows the cluster pattern length.*

the traders change between the behavioural clusters. In addition, transition matrices are formed by including only traders who are present on more than one of the time-steps. This is to see how trader behaviour changes over time. Disappearance is included in these transition matrices, because it is interesting to see the clusters with the highest and the lowest disappearance probability among traders who are present on at least 2 of the time-steps.

Fig. 4.7 presents the cluster transition results. As mentioned above and clearly seen in 4.7(a) and (b), disappearing from the network dominates the transitions in both datasets. Interestingly, clusters $0, 2, 4$ in Bitcoin OTC (see Fig. 4.7(a)) have the highest probabilities for disappearing. These clusters include traders with *low* or *very low* reputation and goodness. In Bitcoin Alpha, the highest disappearance probability is seen in clusters 9 and 1 both of which include traders with *very low* goodness. Worth mentioning is also that cluster 0 in Bitcoin OTC and cluster 1 in Bitcoin Alpha have the label *very low* for fairness too. The clusters from which there are significantly less disappearance, cluster 6 in Bitcoin OTC and 5 in Bitcoin Alpha, are the clusters with *very high* attention and activity with all else feature labels being *medium*. Also, the clusters of the second least disappearance probability, cluster 8 in Bitcoin OTC and cluster 4 in Bitcoin Alpha, include active and noticed traders.

In Figure 4.7 (c)-(d), disappearance is not taken into account. In the datasets, comparably notable proportion of transitions from the cluster of very unfair and ungood traders, cluster 0 in Bitcoin OTC and cluster 1 in Bitcoin Alpha, are to the cluster of *very low* goodness and *medium* fairness, cluster 2 in Bitcoin OTC and cluster 9 in Bitcoin Alpha. This gives the impression that unfair behaviour is punished by being distrusted by other traders. It also indicates that very unfair traders can become more fair afterwards. Figures 4.7(e)-(f) show the transition matrices of the traders present on at least two of the time-steps. The figures highlight even further the fact that active and noticed traders with medium reputation remain in the network while ungood and unfair traders disappear or remain distrusted.



(a) Transitions with disappearance, Bitcoin OTC

(b) Transitions with disappearance, Bitcoin Alpha

(c) Transitions without disappearance, Bitcoin OTC

(d) Transitions without disappearance, Bitcoin Alpha

(e) Traders present more than once, Bitcoin OTC

(f) Traders present more than once, Bitcoin Alpha

Figure 4.7. *Transition proportions in Bitcoin OTC (left) and Bitcoin Alpha (right). Each cell in the heat-maps presents the proportion of transitions from the cluster on the y axis that ended up in the cluster on the x axis. Rows sum up to 1, and can be interpreted as cluster transition probabilities. In (a),(b),(e) and (f), -1 column represents disappearing from the peer review network. In (c) and (d) disappearance is not taken into account. In (e) and (f), transitions are recorded for traders present on at least two of the 12 time-steps. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

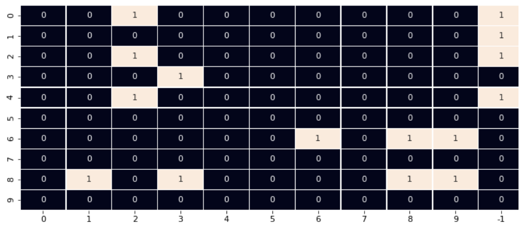
Interestingly, clusters of good and reputable traders who are not unfair, namely cluster 3 in Bitcoin OTC and cluster 8 in Bitcoin Alpha, have almost exactly the same disappearance probabilities in Fig. 4.7(a)-(b) and 4.7(e)-(f).

Due to normalization it is not shown in Fig. 4.7 how the number of transitions varies between the clusters. Therefore, in order to find exceptional transition probabilities, over-representation and under-representation of transitions from one cluster to another are tested using hyper-geometric test. The null-hypothesis in this case is that the observed number of cluster transitions is a result of random sampling from the cluster distribution. Consider there are τ observations of switching from cluster A to cluster B, size of which are denoted by $|A|$ and $|B|$ respectively. In case of over-representation, hyper-geometric distribution is used to get the probability that there are at least τ transitions from A to B when $|A|$ elements are sampled from the overall transition distribution. For under-representation, hyper-geometric distribution gives the probability that there are at most τ transitions. Significance level $\alpha = 0.01$ is used. As there are 10 clusters and 11 possible outputs when taking disappearance into account, Bonferroni corrected significance level $\hat{\alpha}$ is:

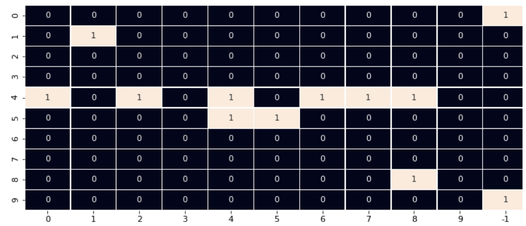
$$\hat{\alpha} = \frac{0.01}{10 \times 11} \approx 9.09 \times 10^{-5}.$$

The results of the hyper-geometric tests are shown in Figure 4.8, where the rejection of the null-hypothesis is denoted by 1. The last columns in the matrices in Fig. 4.8 show from which clusters there are unusually many or unusually few disappearances. In both datasets, traders in the main cluster are likely to disappear. Active and noticed traders with medium reputation disappear from the network exceptionally rarely. In Bitcoin OTC, traders in clusters 0 and 4 disappear exceptionally often. As seen in Table 4.1, these are the only clusters in Bitcoin OTC that contain both unfair and ungood traders. Also, cluster 2 that is the only cluster with 'very low' reputation and goodness shows over-representation in disappearance in Bitcoin OTC. For traders who are reputable, good and not unfair it is unlikely to disappear in Bitcoin OTC. The results indicate that certain type of adverse behaviour leads to disappearance while well behaving traders stay in the network. In Bitcoin Alpha, there are exceptionally many disappearances from cluster 9 that is the only cluster in Bitcoin Alpha that contains inactive traders. While traders in cluster 9 are also ungood and disreputable this is the case in other clusters too that do not show any exceptional cluster transitions.

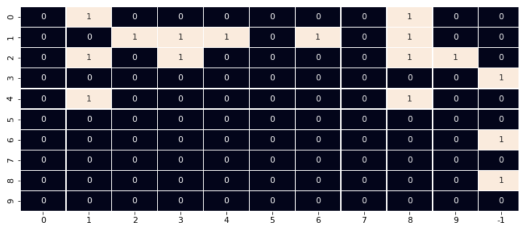
Looking into the diagonals in Fig.4.8(a)-(b), there are similarities between the datasets. The diagonals represent staying in the same cluster. In the datasets, reputable and good traders who are not unfair (cluster 3 in Bitcoin OTC, 8 in Bitcoin Alpha) are likely to remain as they are. For clusters of active and noticed traders ($6,8$ in Bitcoin OTC, $4,5$ in Bitcoin Alpha) it is also common to remain in the same cluster. Considering cluster 2 in Bitcoin OTC, it seems that very disreputable and ungood traders remain distrusted. In



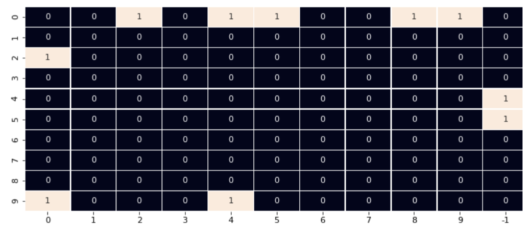
(a) Over-representation, Bitcoin OTC



(b) Over-representation, Bitcoin Alpha



(c) Under-representation, Bitcoin OTC



(d) Under-representation, Bitcoin Alpha

Figure 4.8. *Exceptional cluster transitions are denoted by 1. Significance level $\alpha = 0.01$ is used. Figures (a) and (b) show the over-representation results, i.e. the probability of the observed number of transitions from the cluster on the y axis to the cluster on the x axis is under Bonferroni corrected α in hyper-geometric tests. Figures (c) and (d) show the under-representation results. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

Bitcoin Alpha, staying in the cluster of very low fairness and goodness is over-represented, indicating that severe adverse behaviour is persistent. Based on the results, there are trader types for which behaviour is unusually persistent.

Considering the transitions from one cluster to another, it is observed that active and noticed traders with medium reputation switch to varying types of clusters. This type of traders can behave adversely afterwards but can also end up among the reputable and good ones. In both datasets, medium traders switch to other behavioural clusters exceptionally rarely. Considering the consequences of adverse behaviour it is observed that reputation is unlikely to be recovered. In Bitcoin OTC, switching from clusters 0 and 4 to cluster 2 is over-represented. This means that for unfair and ungood traders it is probable to lose reputation even further and to become more fair afterwards. In addition, these traders change to cluster 8 exceptionally rarely. Changing to cluster 8 would mean reputation recovery for these traders. Also, there are under-represented cluster transitions from the clusters of very low goodness but medium fairness in both datasets. These traders change exceptionally few times to the main cluster. It is also unusual that these traders would switch to the cluster of active and noticed traders with medium reputation. In Bitcoin OTC, it is unlikely that these traders become reputable and good afterwards. The results point out that it is unlikely that severely distrusted traders would recover their reputation.

4.3 Clusters of Unfairly Rated Traders

It is interesting to study how receiving unfair ratings from others affects a trader's own behaviour. The difference between a trader's goodness and reputation is used to measure how unfairly the trader is rated. Goodness can be thought as *fairness corrected reputation*. The reason for this is that the importance of the ratings received from unfair traders is decreased while the importance of the ratings received from fair traders is increased. If a trader has received ratings only from traders with maximum fairness, reputation is equal to goodness. This type of unfair treatment measure would then have the value zero. Unfair treatment is divided into two types: positive and negative unfair treatment. A trader whose reputation is greater than his/her goodness has received unfairly positive ratings. The same way, a trader whose goodness is greater than his/her reputation has received unfairly negative ratings.

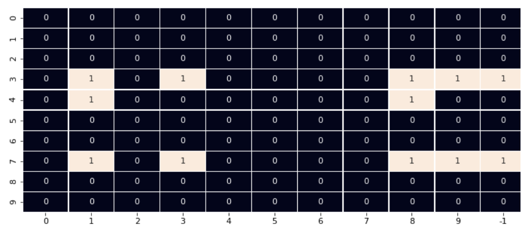
In each aggregated network, the difference between the reputation and goodness of each trader is used to extract occasions where a trader has received comparably unfair ratings. In more detail, using quantile values, traders among the highest in

$$\max(0, r(n) - g(n)),$$

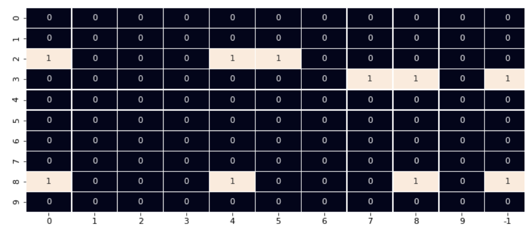
where $r(\cdot)$ is reputation and $g(\cdot)$ is goodness, are considered to be comparably unfairly positively rated. Similarly, traders among the highest in $\max(0, g(n) - r(n))$ are taken as comparably unfairly negatively rated.

Transition matrices of positive and negative unfair treatment cases are formed for both datasets. From the transition matrices, the over-representations of cluster transitions are calculated in order to extract the most anomalous results. The number of certain transitions among unfairly rated traders is compared to the overall transition counts to see if there are exceptionally many such transitions. More concretely, consider there are M transitions from cluster A to B among the unfairly rated traders. Denote the number of such transitions among all the traders by K. Hyper-geometric distribution is used to get the probability of observing at least M such transitions given the total number of cluster transitions among the unfairly rated traders and the total number of cluster transitions among all the traders. It is denoted by 1, if the Bonferroni corrected value of significance level $\alpha = 0.01$ is greater than the probability. In other words, rejected null-hypotheses are denoted by 1 while those that are not rejected are denoted by 0. The source clusters of the cluster transitions represent the behaviour of the unfairly rated traders on the same time-step they are unfairly rated. Destination clusters show how these traders behave afterwards. The cluster transitions in this case are used to interpret reactions to unfair treatment.

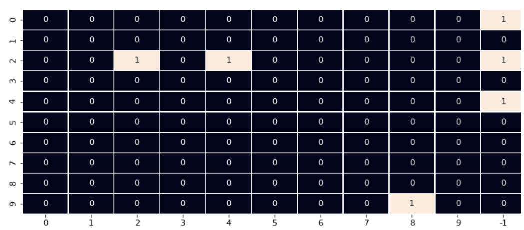
As mentioned above, unfair treatment is divided into positive and negative unfair treatment. The above described tests are done separately for the two cases. Over-representation results are shown in Figure 4.9 Comparing Fig. 4.9 to the cluster center labels in Fig.4.3 and the cluster descriptions in Tab.4.1, it is seen that exceptionally many of the cluster transitions



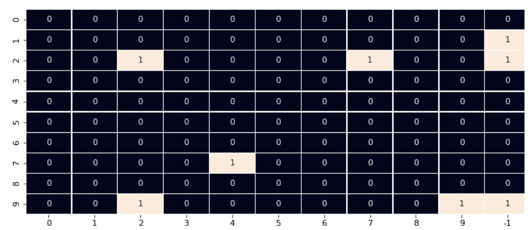
(a) Unfairly positively rated, Bitcoin OTC



(b) Unfairly positively rated, Bitcoin Alpha



(c) Unfairly negatively rated, Bitcoin OTC



(d) Unfairly negatively rated, Bitcoin Alpha

Figure 4.9. *Over-representations of the cluster transitions of unfairly rated traders are denoted by 1. 'Unfairly positively rated' refers to having better reputation than goodness, while 'unfairly negatively rated' refers to the opposite. The quantile value used in taking the most unfairly rated traders is* 0.95. *In the hyper-geometric tests, the Bonferroni corrected value of significance level* $\alpha = 0.01$ *is used. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

among unfairly positively rated are from the clusters of reputable and good traders. That is, clusters 3 and 7 in Bitcoin OTC and clusters 3 and 8 in Bitcoin Alpha are the source clusters of $10/12$ and $7/10$ respectively of the over-represented cluster transitions. In both datasets, the remaining over-represented cluster transitions are from the clusters of ungood traders. Based on the over-represented cluster transitions, unfairly positively rated traders behave in various ways afterwards. Some of them retain or improve their reputation but some of them lose their reputation.

Among the unfairly negatively rated traders, all the over-represented cluster transitions are from clusters of ungood traders as shown in Fig. 4.9(c)-(d). The cluster transitions are mostly to the clusters of ungood and disreputable traders in both datasets. Disappearing from the network after being disreputable and ungood is unusually common among the unfairly negatively rated traders. Remaining with very low reputation is also over-represented. The results indicate that receiving unfairly negative ratings can partly explain remaining disreputable and disappearing from the network. In both datasets, one of the over-represented cluster transitions is from being ungood to being active, noticed and unfair. This indicates that a reaction to unfair negative treatment is to actively give unfair ratings to others. Also, reputation and goodness are improved in these cluster transitions (from 9 to 8 in Bitcoin OTC, and from 7 to 4 in Bitcoin Alpha). The results show that some of the unfairly negatively rated traders are able to recover their reputation while retaliating unfairness by becoming unfair themselves.

The clusters of reputable and good traders dominate the over-representation results of the unfairly positively rated traders. The same way, clusters of ungood and disreputable traders include all the unfairly negatively rated cases. While the results might seem intuitive, it is not due to construction. As shown in Fig. 4.9(a)-(b), a trader in a cluster of ungood and disreputable traders might still be unfairly positively rated. Similarly, it could happen that a reputable and good trader would be unfairly negatively rated. It is interesting that this is not the case in either of the datasets. According to the similarities between the datasets, some of the reputable traders have achieved their comparably high reputation from unfair raters. Low reputation can also be due to unfair ratings. However, goodness is quite in line with reputation in all the clusters. Based on the cluster centers, there are no cases where a comparably good trader would have a comparably low reputation. Thus, reputation is not a notably misleading and incorrect way to judge with whom to trade. All the same, the results indicate that trader behaviour is affected by unfair negative treatment and support the assumption that a trader can retaliate unfairness.

5. BITCOIN TRADER COMMUNITIES AND CENTRALITY

In this chapter topological quantities of Bitcoin peer review networks are studied with respect to the behavioural clusters introduced in the previous chapter. This is to answer the following research questions:

> Are there communities of adversely behaving traders in Bitcoin peer review networks?
> How does centrality relate to traders' behaviour in Bitcoin peer review networks?

Edge weights are not taken into account in community detection and centrality. In Louvain community detection networks are interpreted undirected. It is studied how network structure related matters relate to the behavioral clusters. Therefore, the analysis combines structural and content-based analysis.

5.1 Cluster Proportions in Communities

After clustering the traders based on their behaviour, it is interesting to see *who trades with whom*. An edge between two traders can be thought to indicate that these traders traded or tried to form a deal with each other. Dividing traders into communities enables exploring what type of traders have been interacting with each other. To divide traders into communities, Louvain community detection algorithm is run with various values for *random state* parameter. Random state defines the seed for the random number generator, and the final result is sensitive to it. Figure 5.1 shows the number of communities per

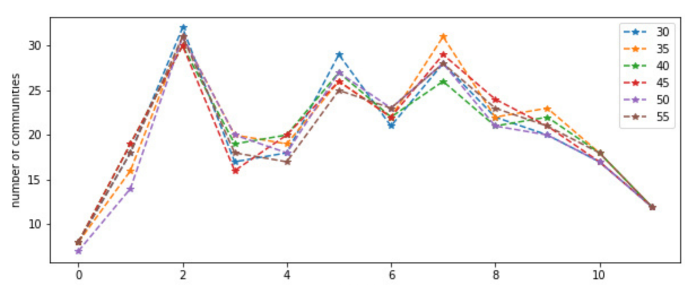
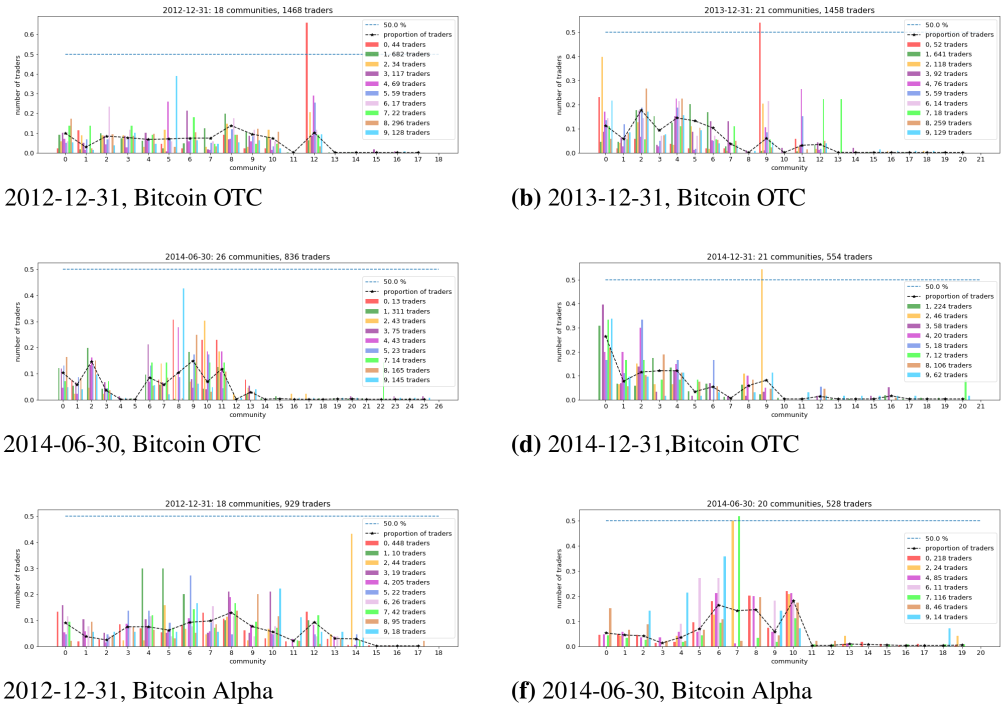


Figure 5.1. *The number of communities on each of the 12 time-steps using Bitcoin OTC dataset. The curves refer to the different values of 'random state' parameter in community detection algorithm. The values of the random state parameter are shown in the legend.*



(a) 2012-12-31, Bitcoin OTC

(b) 2013-12-31, Bitcoin OTC

(c) 2014-06-30, Bitcoin OTC

(d) 2014-12-31,Bitcoin OTC

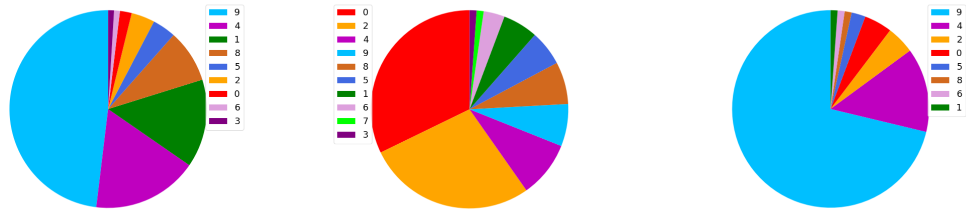
(e) 2012-12-31, Bitcoin Alpha

(f) 2014-06-30, Bitcoin Alpha

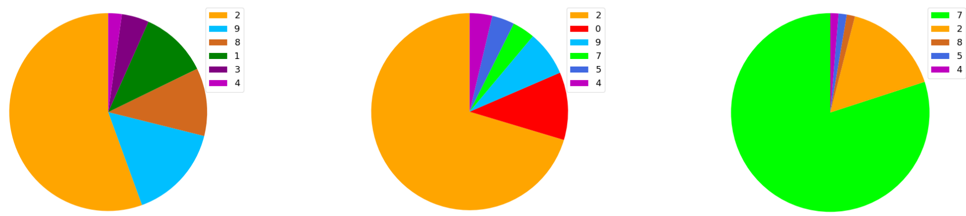
Figure 5.2. *The figures show how traders in each of the behavioural clusters are divided into communities. The timestamps of the aggregated networks are shown in the sub-figure titles. The number of communities and the number of traders are also shown in the titles. Communities are denoted by integers on the x axis. The proportion of traders is shown on the y axis. The black dashed line shows the 50% limit. The legends of the sub-figure legends show how the colors map to the clusters. The number of traders in each cluster on the time-step is also shown in the legend.*

each 12 time-steps for Bitcoin OTC. Based on the figure, the number of communities differs depending on the value of the random state parameter. All the same, the results are interpreted not to alter drastically. The existence of communities of adversely behaving traders is considered not to be sensitive to minor changes. Therefore, this algorithm is used for community detection with arbitrarily chosen random state value. For robustness check, community detection is run with various random state values to assure the below presented results are not only due to a particular parameter value. In other words, the communities of adversely behaving traders found from the data exist also with other random state values despite the number of communities varies with respect to that parameter.

Per each aggregated network, it is calculated how the traders in each behavioural cluster are divided into communities. Consider an aggregated network and the traders in it that belong to a cluster C_k. If the traders are mainly in one community, it indicates that this type of traders have been rating each other. In this way it is explored if communities of adversely behaving traders are found in the networks. Figure 5.2 shows how the clusters are divided into communities on time-steps when interesting communities appear in the network. A curve showing 50% limit is drawn in the sub-figures, to make it easier to



(a) OTC, 2012-12-31, 104 traders **(b)** OTC, 2013-12-31, 87 traders **(c)** OTC, 2014-06-30, 87 traders



(d) OTC, 2014-12-31, 45 traders **(e)** Alpha, 2012-12-31, 27 traders **(f)** Alpha, 2014-06-30, 75 traders

Figure 5.3. *Pie charts show how the traders are divided into the behavioural clusters in a certain community. The chosen communities are the ones where adverse behaviour dominates. In all of the pie charts, disreputable, ungood or unfair traders cover the majority of the traders in the community. The color bars next to each sub-figure show how the colors map to the clusters. The sub-figure captions present the dataset, the time-stamp and the number of traders in the community. 'OTC' refers to Bitcoin OTC and 'Alpha' refers to Bitcoin Alpha.*

observe the clusters whose traders belong to a small number of communities. In Fig. 5.2, the clusters are relatively evenly divided into communities. It seems that in general traders do not form peer review communities of similarly behaving traders. Yet there are clusters whose traders are split only into few communities.

Communities that are chosen for further examination are required to be statistically significantly dominated by adversely behaving traders. The communities discussed below are such that the majority and unusually many of the traders belong to a cluster of adverse behaviour. Only the communities of at least 20 traders are studied in order to focus on the main events. The statistical significance of the cluster proportions in the communities is tested using hyper-geometric test. Hyper-geometric distribution is used to get the probability of observing at least the number of traders in a certain cluster within a community, when the total number of traders in that community is randomly sampled among all the traders in the aggregated network. The Bonferroni corrected value of significance level $\alpha = 0.01$ is used. All the presented results are statistically significant. Figure 5.3 show how the traders in the chosen communities are divided into the behavioural clusters. Per each of the aggregated networks presented in the sub-figures in Fig. 5.2, there is a pie chart representation in Fig. 5.3 of one of the communities in the network. The pie charts present the communities that most clearly indicate that communities of adversely behaving traders exist.

An interesting observation present in the datasets is that when the proportion of -10 ratings increases, a community of adversely behaving traders appears. In Figure 3.1, the proportion of -10 ratings increases significantly during the second half of 2012 (the 5th time-step) in comparison to the previous time-steps. On the 5th time-step, a community of ungood and disreputable traders appears in both datasets as shown in Figure 5.2(a) and 5.3(a) for Bitcoin OTC and 5.2(e) and 5.3(e) for Bitcoin Alpha. A year later, the proportion of -10 ratings increases up-to nearly 20% of all the ratings in Bitcoin OTC according to Fig. 3.1(a). Simultaneously, a community of unfair and ungood (cluster 0) and disreputable and ungood traders (cluster 2) appears as shown in Fig. 5.3(b). While it is expected that disreputable traders exist on a time-step when there is a burst of -10 ratings it is not as self-evident that unfair ones would appear. It is also intriguing that the majority of these traders have been rating each other.

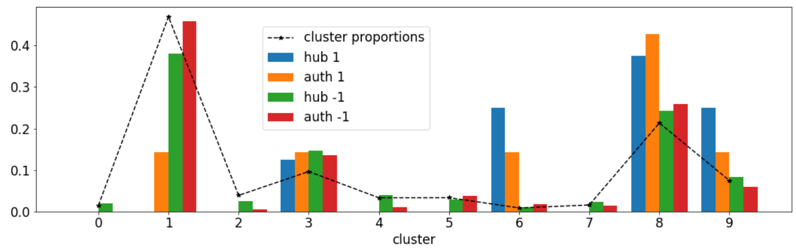
In both datasets, during the year 2014 the proportion of slightly negative ratings increases compared to all of the previous time-steps as shown in Fig.3.1. At the same time, communities of disreputable and ungood traders appear in both trading platforms. Worth noticing is that these traders have medium fairness. In other words, the majority of the traders clustered as ungood and not unfair are gathered in the same community in both datasets. To sum up, there are time-steps when communities of adverse behaviour emerge both in Bitcoin OTC and in Bitcoin Alpha. Based on the communities with exceptionally many adversely behaving traders, it can be concluded that communities of distrust and unfairness are present in Bitcoin peer review networks. The results indicate that traders can react to unfair and negative feedback by giving back similar feedback. Retaliating unfairness and reacting to negative feedback by adverse behaviour is in line with previous studies on human behaviour [4, 5, 7].

5.2 Behaviour of the Most and the Least Central Traders

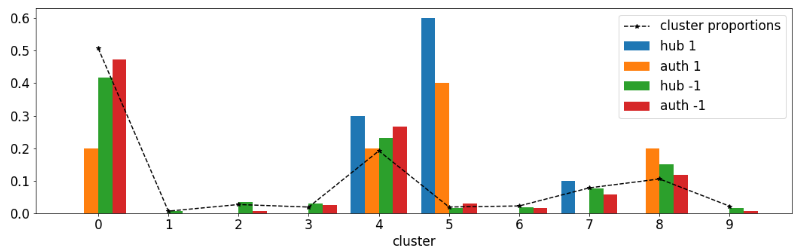
To answer the research question:

> How does centrality relate to traders' behaviour in Bitcoin peer review networks?,

HITS algorithm is run separately per each aggregated network. Each trader in an aggregated network is given a hub and a authority value. The values are scaled to range from -1 to 1. Cluster proportions among the highest and the lowest hub and authority values are calculated to see how the most and the least central traders behave. First, it is studied how the previous behaviour of the traders relates to centrality. In this way it is possible to analyse if a certain type of behaviour leads to being among the most important traders. Similarly, it is studied if a certain type of behaviour is related to becoming the least central



(a) Bitcoin OTC



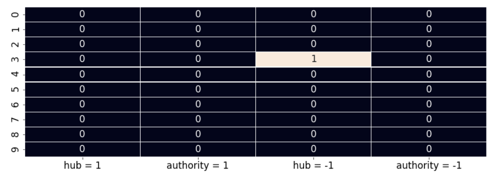
(b) Bitcoin Alpha

Figure 5.4. *Distributions of the previous clusters of traders with the minimum and the maximum hub and authority values are shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The hub and authority values are scaled to range from -1 to 1. The black dashed line shows the cluster proportions over all the traders. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

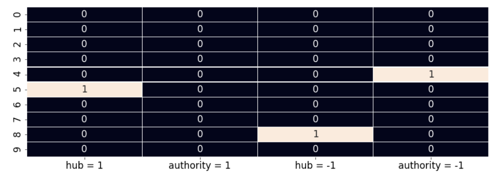
in the network. It is also studied how the current behaviour of the traders relates to the maximum and the minimum hub and authority values.

Centrality in this context is a measure of how much a trader contributes to a peer review network. The authorities in a peer review network are traders who have received many ratings especially from traders that have rated other authorities too. The hubs in a peer review network are traders that have actively rated others especially the authorities in the network. Traders with the maximum hub and authority values can be considered the key contributors in the discussion of *with whom to trade*. The impact of the traders with the minimum hub and authority value on the discussion can be considered negligible.

Per each aggregated network starting from the second time-step, traders with the maximum and the minimum hub and authority values are extracted. Among these traders, only those that exist in the network before are included. It is then calculated how these traders are divided into the behavioural clusters. The number of traders in each cluster is summed up over the time-steps to get the overall cluster proportions of such traders. These cluster proportions are shown in Fig. 5.4(a) for Bitcoin OTC and Fig. 5.4(b) for Bitcoin Alpha. Hyper-geometric test is used to extract exceptional cluster proportions to focus on statistically significant results. The over-representation of traders in a certain cluster is tested separately for each of the four cases: the maximum hub value, the maximum authority value, the minimum hub value, the minimum authority value. These results



(a) Bitcoin OTC



(b) Bitcoin Alpha

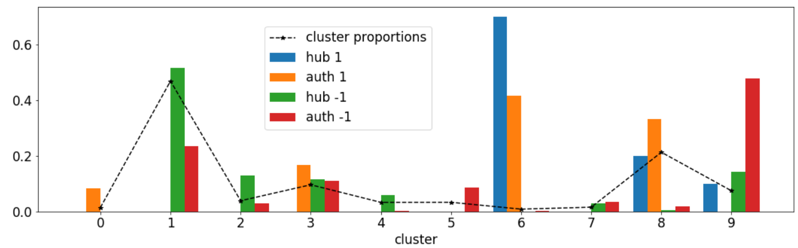
Figure 5.5. *The over-represented previous clusters of the traders with the maximum and the minimum hub and authority values are shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The y axis shows the clusters and x axis the different cases. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

are shown in the heat-maps in Figure 5.5. The Bonferroni corrected significance level of $\alpha = 0.01$ is used in the hyper-geometric test. Similar figures are made for the current cluster distributions. This is to see how the most and the least central traders behave in peer review networks. Figure 5.6 shows the current cluster proportions of the traders with the maximum and the minimum hub and authority values. Over-representation is again tested by hyper-geometric test. The results are shown in figure 5.7.

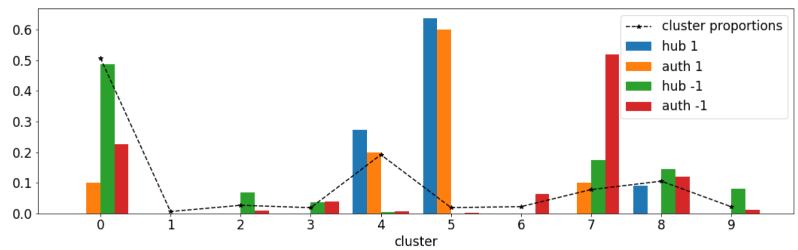
As seen in Fig. 5.4 the most of the traders with the maximum hub and authority values have previously been in the clusters of active and noticed traders. These clusters are 4 and 5 in Bitcoin Alpha and 6 and 8 in Bitcoin OTC. According to Fig. 5.5(a), clusters 6 and 8 are not over-represented among the most central traders in Bitcoin OTC. In Bitcoin Alpha, there are exceptionally many traders with the highest possible hub value that have been in cluster 5. Cluster 5 contains highly active and noticed traders. Interestingly, cluster 4 is over-represented among the lowest in authority although it also contains active and noticed traders. The traders in cluster 5 are more active and noticed but also more fair than the traders in cluster 4. This indicates that unfair behaviour is punished by losing authority. Being active, noticed and not unfair is the way to become one of the main hubs in Bitcoin Alpha. The same cannot be said about the traders in Bitcoin OTC.

An interesting result present in both datasets is that exceptionally many traders have ended up with the lowest possible hub value after being reputable, good and not unfair (cluster 3 in Bitcoin OTC and 8 in Bitcoin Alpha). Considering the goodness and fairness, these clusters can be stated to include the most well behaving traders. It is quite counter-intuitive that such traders do not gain importance afterwards. The results indicate that the most well behaving traders have the most negligible impact on the peer review network and have a lower need to rate others than the rest.

Analysing the current clusters, Fig. 5.6 indicates that the most of the hubs and the authorities are in the clusters of active and noticed traders. Also, medium traders seem to be the least central. According to the over-representations shown in Fig. 5.7, highly active and noticed traders that are not unfair are over-represented among the traders with the maximum hub and authority values.

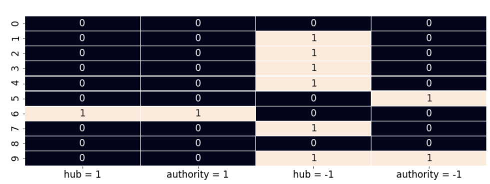


(a) Bitcoin OTC

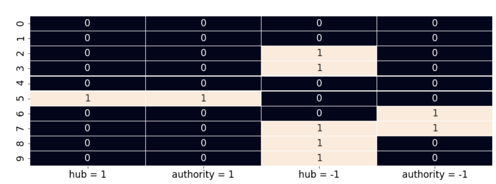


(b) Bitcoin Alpha

Figure 5.6. *Distributions of the current clusters of traders with the minimum and the maximum hub and authority values are shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The hub and authority values are scaled to range from -1 to 1. The black dashed line shows the cluster proportions over all the traders. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*



(a) Bitcoin OTC



(b) Bitcoin Alpha

Figure 5.7. *The over-represented current clusters of the traders with the maximum and the minimum hub and authority values are shown for (a) Bitcoin OTC and (b) Bitcoin Alpha. The y axis shows the clusters and x axis the different cases. Note that Bitcoin OTC figures cannot be compared to Bitcoin Alpha figures because the clusters are not the same.*

Based on how HITS algorithm works, it is expected that active traders have high hub values and traders who gain attention have high authority values. However, active and noticed traders who are unfair are not over-represented among the hubs and the authorities. This is the case in both datasets. In both datasets, there are many over-represented clusters among traders with the minimum hub value. Hence, traders with the minimum hub value can behave in various ways. Interestingly, the two over-represented clusters among traders with the minimum authority value are similar between the datasets. Traders who are either unfair or ungood with medium activity and attention are the least authorities in the network.

To sum up, the most well behaving traders become the least contributors in the peer review networks exceptionally often. The current behaviour of traders with the minimum hub value varies significantly. Expectedly, highly active and noticed traders are over-represented among the hubs and the authorities. Yet, it seems not to be enough to be active and noticed to end up among the most central traders. Traders who do not behave adversely are unusually often the hubs and the authorities in the networks. Being unfair leads exceptionally many times to having the minimum authority value in Bitcoin Alpha. The results can be interpreted so that fairness pays back in terms of becoming central while behaving well shows the opposite.

6. CONCLUSION

In this chapter, the research questions are shortly answered by summarizing the results presented in the previous chapters. Limitations and future prospects of this work are discussed. The last section discusses the contribution of this work.

6.1 Answers to the Research Questions

The first research question: *what kind of behavioural clusters are formed from Bitcoin traders*, is answered by clustering the traders into 10 clusters based on their behaviour. The results show clear similarities between the datasets. Roughly half of the traders belong to a behavioural cluster where all the cluster center labels are *medium*. Based on the labeled cluster centers, the clusters are distinct in both datasets. In both datasets, there is a cluster containing the most well behaving traders and a cluster containing the most adversely behaving traders in terms of goodness and fairness.

The second research question: *how does a trader's behaviour in a Bitcoin peer review network change over time*, is answered by extracting statistically significant cluster transitions. The results show some similarities between the datasets. The medium traders disappear form the network unusually often and switch to many other behavioural clusters unusually seldom. Active and noticed traders with medium reputation behave in various ways but disappear from the network unusually seldom. There are trader types for which the behaviour is unusually persistent. The most well behaving traders stay in the same cluster exceptionally often. For active and noticed traders with medium reputation it is common to remain that way. In Bitcoin Alpha, the most adversely behaving traders are likely to stay as they are. In Bitcoin OTC, highly disreputable traders remain the same way. Moreover, certain types of adverse behaviour seem to be punished. It is unlikely that severely distrusted traders would recover their reputation. In Bitcoin OTC, traders who are both ungood and unfair are prone to end up among the lowest in reputation or disappear from the network. In Bitcoin Alpha, ungood and inactive ones are likely to disappear from the network.

The third research question: *what is the impact of receiving unfair ratings on a trader's behaviour in a Bitcoin peer review network*, is answered by studying the over-represented cluster transitions among unfairly rated traders. The results indicate that some of the most reputable traders have received their comparably high reputation from unfairly positive ratings. Also, some of the disreputable traders have received unfairly positive ratings. There is no clear pattern of how the behaviour of unfairly positively rated traders evolve over time. Some of the unfairly positively rated traders retain or improve their reputation but some of them lose their reputation. Unfair negative treatment can be interpreted to partly explain disappearance from the network after being distrusted by other traders. Interestingly, one of the over-representations in both datasets shows that unfairly negatively rated traders can become more unfair themselves. Although thorough explanation for the change in behaviour cannot be stated based on the cluster transitions, the results indicate that some of the unfairly negatively rated traders start to rate others more actively and unfairly, while receiving more ratings and improving their own reputation.

The fourth research question: *are there communities of adversely behaving traders in Bitcoin peer review networks*, is answered by finding communities where adverse behaviour statistically significantly dominates. The results show that on time-steps where the proportion of negative ratings increases compared to the previous time-steps, communities of adverse behaviour are found. In other words, there are groups of unfair and ungood traders who have mainly been rating each others. This together with the result of unfairly negatively rated traders becoming unfair themselves gives the impression that traders react to negative feedback and unfair negative ratings by giving similar ratings to others.

The last research question: *how does centrality relate to traders' behaviour in Bitcoin peer review networks*, is answered by calculating the cluster proportions among traders with the maximum hub value, the maximum authority value, the minimum hub value and the minimum authority value. Both the previous and the current clusters of such traders are analysed. Surprisingly, in both datasets the most well behaving traders end up having the minimum hub value exceptionally often. Expectedly, being highly active and noticed is a way to become and remain a key contributor in the peer review networks. Yet, it is not enough to be active and noticed. The most central traders are also medium fair while active, noticed but unfair traders end up with the minimum authority value unusually often in Bitcoin Alpha.

6.2 Limitations and Future Prospects

The results presented in this work are based on the trust rating data from two online Bitcoin trading platforms. Due to excluding other types of data, the results are considered preliminary. Data from other Bitcoin trading platforms and Bitcoin transaction data could be added to further analyse Bitcoin trader behaviour. Also, the research topic could be extended to include data about other types of peer preview networks. In the future, it would be interesting to add more data to study if the results show similarities with the results presented in this work.

Considering the mathematical methods chosen in this work, it is outside the scope of this work to compare different methods. Although the methods are deliberately decided, it limits the generality of the results that the analysis is not done by comparing the methods. Analysing trader behaviour based on the behavioural clusters is considered a reasonable approach to find the main trader types and the clearest patterns in trader behaviour. It is noted, however, that clustering simplifies the data. The cluster centers do not fully represent the traders in the clusters, and the results are considered indicative. For future research it would be interesting to compare for example different clustering algorithms. Future research could also include testing various models to explore how trader behaviour changes over time. Moreover, cluster transitions and feature relations could be analyzed even further as the scope of this work left many behavioural questions outside.

In accordance with the previous studies, the traders are given the maximum fairness value in case there are no given trust ratings. It can be argued that such a choice is not the most reasonable. For example, traders with no given ratings could be given the average of the fairness values over the traders who have been rating others. Alternatively, clustering could be modified to exclude fairness of traders with no given ratings. Also, giving the reputation and goodness value zero to traders with no received ratings can be questioned. Traders with zero reputation are considered comparably distrusted in Bitcoin OTC, which can be argued to be a false interpretation of traders with no received ratings. However, it would also be misleading to give positive or negative reputation and goodness for such traders because there is a clear meaning for the signed rating values. Giving the value 0 is considered a natural choice to represent that a trader's trustworthiness is unclear. In that way, reputation is in line with how goodness is designed. The impact of the choice on clustering result could be removed by excluding the reputation and goodness of the traders with no received ratings. The choices made in this work are not considered to notably affect the results. Yet, alternative ways to interpret the cases of no given and no received ratings could be studied further.

6.3 Contribution

This work continues the state-of-the-art research on online social networks by analysing trader behaviour in Bitcoin trader peer review networks. Previous studies on the same datasets concentrate on more general topics: link prediction and fraudulent user detection. This work builds on top of the existing research by taking a specific view and a deeper look into Bitcoin OTC and Bitcoin Alpha trust rating datasets. To the best of my knowledge this is the first research that models Bitcoin trader behaviour using peer review network data. For this reason the results are interesting as cryptocurrencies and online interactions between strangers become more popular every day. This work also contributes to the study on interactions between humans by providing results on how humans rate each other in peer review networks and how they react to unfairness. To assess unfair treatment, the features developed in previous studies are used to derive a measure for how unfairly a trader is rated in a peer review network. This work builds on top of the existing research and provides results on the functionality of peer review networks.

In this work, methods and quantities related to the study of complex networks are combined with techniques of exploratory data science. The study of complex networks is applied in deriving the features from the signed, weighted edges. Clustering the feature vectors in turn is a typical approach in exploratory data science. Also, network structure related matters, communities and centrality, are studied in relation to the behavioural clusters. This work contributes to the research on online social networks by applying mathematical methods from multiple scientific fields. In its specificity, this work contributes to Bitcoin trading by providing observations of Bitcoin trader behaviour in peer review networks. In its generality, the results contribute to the research on social networks and analysis of human behaviour. The same approach can easily be applied in the analysis of other peer review networks and other types of social networks.

REFERENCES

[1] Bitcoin otc trust weighted signed network. https://snap.stanford.edu/data/soc-sign-bitcoin-otc.html. Accessed: 2019-08-09.

[2] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems, 2000.

[3] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, 2003.

[4] Michael Luca and Georgios Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12):3412–3427, 2016.

[5] Daniel Houser, Stefan Vetter, and Joachim Winter. Fairness and cheating. *European Economic Review*, 56(8):1645–1655, 2012.

[6] Eamonn Ferguson, John Maltby, Peter A. Bibby, and Claire Lawrence. Fast to forgive, slow to retaliate: Intuitive responses in the ultimatum game depend on the degree of unfairness. *PLoS ONE*, 9(5):e96344, 2014.

[7] Nives DellaValle and Matteo Ploner. Reacting to unfairness: Group identity and dishonest behavior. *Games*, 8(3):1–18, 2017.

[8] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[9] Charu C. Aggarwal. *Social Network Data Analytics*. Springer Science + Business Media, 1. aufl. edition, 2011.

[10] Cristian Bisconti, Angelo Corallo, Marco De Maggio, Francesca Grippa, and Salvatore Totaro. Quantum modeling of social networks. In Miltiadis D. Lytras, Ernesto Damiani, John M. Carroll, Robert D. Tennyson, David Avison, Ambjörn Naeve, Adrian Dale, Paul Lefrere, Felix Tan, Janice Sipior, and Gottfried Vossen, editors, *Visioning and Engineering the Knowledge Society. A Web Science Perspective*, pages 70–77, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[11] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org online article*, 2008. available in: https://bitcoin.org/bitcoin.pdf.

[12] Malcolm Campbell-Verduyn and Marcel Goguen. Blockchains, trust and action nets: extending the pathologies of financial globalization. *Global Networks*, 2018.

[13] Fran Casino, Thomas K Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telematics and Informatics*, 2018.

[14] Quinn DuPont. Experiments in algorithmic governance: A history and ethnography of "the dao," a failed decentralized autonomous organization. In *Bitcoin and Beyond*, pages 157–177. Routledge, 2017.

[15] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 2017.

[16] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 221–230. IEEE, 2016.

[17] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 333–341. ACM, 2018.

[18] Bitcoin alpha trust weighted signed network. https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html. Accessed: 2019-08-09.

[19] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. . Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175–308, 2006.

[20] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[21] Manisha Pujari and Rushed Kanawati. Supervised rank aggregation approach for link prediction in complex networks. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1189–1196. ACM, 2012.

[22] Mohammad A. Hasan and Mohammed J. Zaki. *A Survey of Link Prediction in Social Networks*, pages 243–275. Springer US, Boston, MA, 1 edition, 2011.

[23] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[24] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. *oddball: Spotting Anomalies in Weighted Graphs*, volume 6119, chapter 2, pages 410–421. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[25] Caleb C Noble and Diane J Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2003.

[26] Bryan Hooi, Neil Shah, Alex Beutel, Stephan Günnemann, Leman Akoglu, Mohit Kumar, Disha Makhija, and Christos Faloutsos. Birdnest: Bayesian inference for ratings-fraud detection. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 495–503. SIAM, 2016.

[27] Daniel A Sprague and Thomas House. Evidence for complex contagion models of social contagion from observational data. *PloS one*, 12(7):e0180802, 2017.

[28] Prasanna Gai and Sujit Kapadia. Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2120):2401–2423, 2010.

[29] Chris J Kuhlman, VS Anil Kumar, Madhav V Marathe, SS Ravi, and Daniel J Rosenkrantz. Finding critical nodes for inhibiting diffusion of complex contagions in social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 111–127. Springer, 2010.

[30] N. Altman and M. Krzywinski. Clustering. *NATURE METHODS*, 14(6):545–546, 2017.

[31] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003.

[32] Grigorios Tzortzis and Aristidis Likas. The minmax k-means clustering algorithm. *Pattern Recognition*, 47(7):2505–2516, 2014.

[33] P. Berkhin. *A Survey of Clustering Data Mining Techniques*, pages 25–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[34] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[35] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[36] Mauro Faccin, Piotr Migdał, Tomi H Johnson, Ville Bergholm, and Jacob D Biamonte. Community detection in quantum complex networks. *Physical Review X*, 4(4):041012, 2014.

[37] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[38] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3):42, 2016.

[39] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[40] Giuseppe Davide Paparo and MA Martin-Delgado. Google in a quantum network. *Scientific reports*, 2:444, 2012.

[41] Leonardo Ermann, Klaus M. Frahm, and Dima L. Shepelyansky. Google matrix of bitcoin network. *The European Physical Journal B*, 91(6):1–13, 2018;2017;.

[42] Xiujuan Lei, Siguo Wang, and Fang-Xiang Wu. Identification of essential proteins based on improved hits algorithm. *Genes*, 10(2):177, 2019.

[43] Yajun Du, Xiuxia Tian, Wenjun Liu, Min Wang, Wen Song, Yongquan Fan, and Xiaoming Wang. A novel page ranking algorithm based on triadic closure and hyperlink-induced topic search. *Intelligent Data Analysis*, 19(5):1131–1149, 2015.

[44] Networkx hits implementation. https://networkx.github.io/documentation/networkx-1.9.1/_modules/networkx/algorithms/link_analysis/hits_alg.html#hits. Accessed: 2019-09-08.

[45] Amy N. Langville and Carl D. Meyer. A survey of eigenvector methods for web information retrieval. *SIAM Review*, 47(1):135–161, 2005.

[46] John A Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[47] Michele Tumminello, Salvatore Miccichè, Fabrizio Lillo, Jyrki Piilo, and Rosario N. Mantegna. Statistically validated networks in bipartite complex systems. *PLoS ONE*, 6(3):e17994–e17994, 2011;2010;.

[48] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[49] Python implementation for fairness-goodness algorithm. https://cs.stanford.edu/~srijan/wsn/. Accessed: 2019-04-13.

[50] Joseph Lee Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[51] Galit Shmueli. To explain or to predict? *Statistical Science*, 25(3):289–310, 2010;2011;.

[52] Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan. Predicting trust and distrust in social networks. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 418–424. IEEE, 2011.

[53] Michael J. Brusco and Douglas Steinley. A comparison of heuristic procedures for minimum within-cluster sums of squares partitioning. *Psychometrika*, 72(4):583–600, 2007.