# TEMPORAL NETWORKS: AN ECONOMIC CASE STUDY

UNIVERSITA DEGLI STUDI DI MILANO

´

FACOLTA DI SCIENZE POLITICHE,

ECONOMICHE E SOCIALI

Laurea Magistrale in Data Science and Economics

Author: Peesapati Venkata Sai Vamsi,

Matriculation Number: 933987

Supervisor: Prof. Gaito Sabrina Tiziana, Dept. Of Computer Science, Management and Quantitative Methods

Co-Supervisor: Prof. Silvia Salini, Dept. Of Economics, Management and Quantitative Methods

Academic Year 2020-2021

# Declaration

I declare that this dissertation entitled **Temporal Networks: An Economic Case Study** is a result of my research and any outside information used has been properly cited on the reference page. This report, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

# Abstract

Many studies have discovered patterns in static graphs, identifying properties in a single snapshot of a large network, or in a very small number of snapshots; these include in- and out-degree distributions, small-world phenomena, and others. However, due to a lack of information on the network's evolution over extended periods of time, has been difficult to obtain observations concerning the patterns over time. Understanding, predicting, and optimizing the behavior of dynamical systems is aided by the network structure, which describes how the graph is wired. Edges depict sequences of instantaneous or nearly instantaneous connections in networks of communication via e-mail, text messaging, or phone calls, for example. The interactions between members on a Bitcoin trading platform in our example. Here we study dataset from Snap Stanford and identify the properties in single snapshot or number of snapshots. The small world and scale free phenomenon of a static network are addressed. First, I investigate the densification process, that explains the density of each snapshot over time. Second, I Investigate if the top central nodes in each snapshot are varied over time. The network as single snapshot is considered for static networks and to analyze the densification process and central node change 7 snapshots of the data was considered.

# Acknowledgment

I owe my gratitude to my Prof. Gaito Sabrina Tiziana, my thesis supervisor, and co-supervisor, Prof. Silvia Salini, for their assistance and guidance in completing this research. I owe my gratitude to my parents and friends who have aided me in every aspect of my life. Their assistance enabled me to successfully accomplish this research.

## CONTENTS

## Table of Figures

# Chapter 1

## Background

### 1.1 Temporal Networks

Network analysis is a methodology for retrieving topological information from social, technological, and other complex systems in the real world. In this paper, we present a brief overview of complex networks and their extensions, such as temporal networks. The complex network theory is based on the empirical examination of real-world networks. Complex networks, ranging from scientific to biological networks, allow us to analyze real-world phenomena. For example, people are connected in society through relationships or the Internet, which is built by routers and computer connections. These networks are frequently referred to as complex networks since their behavior cannot be predicted from their components, yet understanding their topological description allows us to examine and manage them. Understanding the propagation of viruses across transportation networks is one example of how these structures might be used to anticipate pandemics.

The concept of complex networks is expanded to incorporate temporal networks. A temporal network, also known as a time-varying network, is made up of links that are only active at specific instants, [Holme, 13]. Each connection contains information such as when it is active and other properties such as weight. Because each link provides a contact opportunity and the temporal ordering of interactions is incorporated, temporal networks are particularly effective for spreading activities such as the transfer of information and illness. Communication networks with temporary or instantaneous linkages, such as phone calls or e-mails, are known as temporal networks.

On both networks data are disseminated, and the second network is where certain computer viruses spread. Networks of physical closeness that encode who meets whom and when can be represented using temporal networks. Physical contact can spread some diseases, such as airborne infections. Epidemic modelling has been improved by using real-world data from time-resolved physical proximity networks. Since the activity of neurons is time linked, neural networks and brain networks can be classified as temporal networks. Individual linkages in temporal networks are characterized by periodic activity. Certain network evolution models, on the other hand, may have an overall time dependency on the network size. A simple method to acquire an overview of complex systems, from the Internet to metabolism, from the proteome to the network of sexual connections, is to depict the system as a graph. A graph is a mathematical structure made up of a collection of vertices (the system's components) and edges (the pairs of vertices that interact with one another).

The advantage of modelling the system as a graph is that we can conclude much about the dynamical system's behavior without requiring a need to study the actual dynamics. We can calculate how much one portion of the network impacts another, how effectively the network is optimized concerning the dynamical system, and which vertices perform similar functions in the system's operation.

### 1.1.1 Temporal networks definition

According to [Holme, 15], A temporal network with *N* nodes and *E* edges is defined by a set of nodes '*i'* connected by edges 'j' at times '*t'*, i.e. (*i*, *j*, *t*). Since '*t'* is discrete, an edge activation at the time '*t'* means that the activation occurred in the interval $[t, t + \delta)$ where $\delta$ is the temporal resolution. Each of these intervals is called snapshot, or time step.

In Temporal networks, we consider an additional dimension 'time', where the times when edges are active are an explicit component of the representation. The time dimension has been projected out by aggregating the contacts between vertices to edges. Alternatively, the data has been segmented into neighboring time windows, where contacts are aggregated into edges, and the temporal development of the network structure in these windows are studied. This method does not account for all features of contact patterns' temporal structure. The edges connecting the vertices of temporal networks, for example, may not have to be transitive. If A is directly linked to B and B is directly connected to C in a static network, whether directed or not, then A is indirectly connected to C through a path across B. But, In temporal networks, however, if the edge (A, B) is active only at a later time than the edge (B, C), A and C are unconnected since nothing can propagate from A to C via B. As a result, the order in which events occur is critical. Another major distinction from static networks is that connectivity is temporal: even if there is a direct or indirect link from point A to point B now, there may be none a second later. As a result, unless the period (interval) of this connection is specified, the statement "A is linked to B" is invalid. Because of the aforementioned concerns, a temporal network can never be reduced to a static network without losing information or changing the meaning of the nodes.

In (a), On the edges, the timings of the encounters between vertices A–D are noted. Assume that the disease begins to spread at vertex A and continues to spread as soon as a contact is made. For four distinct times, the dashed lines and vertices depict this spreading process.

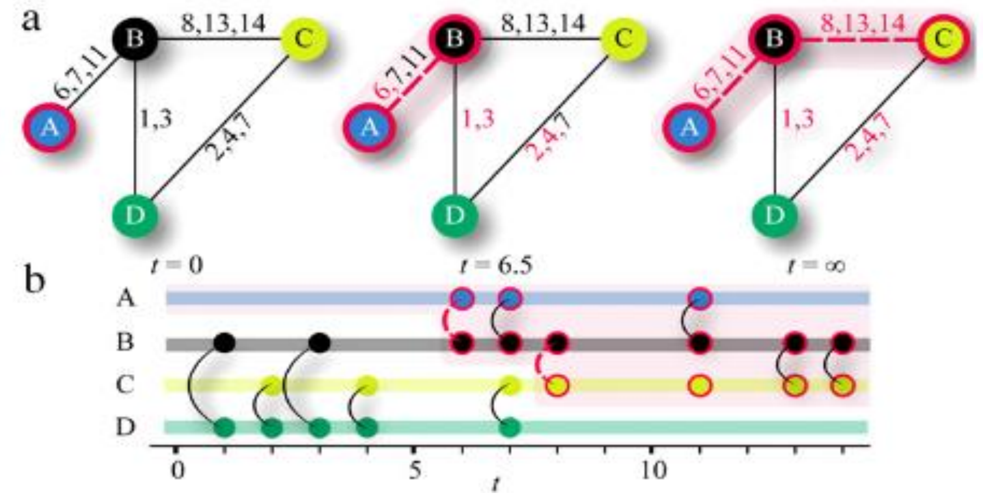


*Figure 1: Illustration of the reachability issue and the intransitivity of temporal networks.*

The infection will not spread beyond what is shown in the t = ∞ picture, i.e., D will not be infected. If the infection began at vertex D, however, the complete set of vertices would ultimately be infected. The effect that occurs from the temporal ordering of interactions cannot be captured by aggregating the edges into a single static graph.

(b) depicts the same event by emphasizing the time dimension.

The underlying static network and the dynamical system on the network are separated in traditional network modelling. In contrast to this image, temporal network techniques relocate information about when events occur from the dynamical system to the network, which is the underlying structure on which the dynamics occur.

Some systems can be described as temporal networks all over the place. When converting a temporal network structure to a static graph, information is always lost.

In certain circumstances, the loss of information is likely modest enough to compensate for the more difficult analysis and modelling required by the temporal graph technique. An instance of a contact sequence that is relatively well described by a weighted graph with the assumption that contact times are random and have a frequency proportionate to the edge weight can be observed in the figure below.

In the context of spreading dynamics, the application limitations of aggregated contact sequences. The figure displays a schematic contact sequence that, assuming a regular random contact process, might be pretty effectively described as an aggregated weighted graph, as shown to the right.

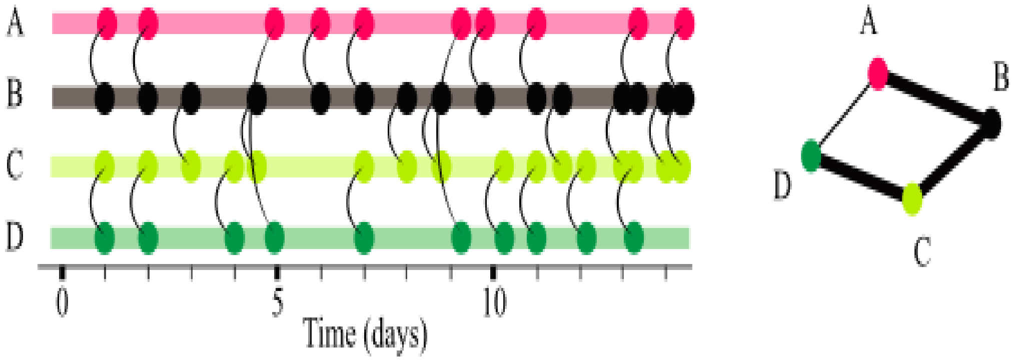


*Figure 2: Schematic contact sequence*

## 1.2 Type of Temporal Network

Interactions between species or other types of creatures are recorded in ecological networks. They could be trophic, indicating which species prey on which, or mutualistic, indicating how two species interact in a mutually beneficial relationship.

Ecological networks are dynamic in various ways. They can alter with the seasons, for example, when creatures go through different stages of their life cycles. Temporal networks could be effective for modifications in interaction patterns in reaction to environmental changes, annual and circadian cycles, and other factors.

## 1.3 Models

In general, network models can be applied in a variety of applications. They may be used to generate synthetic networks with desirable, controllable properties that may then be used in the dynamic process of computations, or they may be used to demonstrate how crucial network features develop. Randomized reference networks are a sort of model in which empirical networks are used as inputs and randomization techniques are used to remove some of the common linkages. In this scenario, machine learning approaches are used to generate numerical models from limited amounts of data.

### 1.3.1 Temporal Exponential Random Graphs

The exponential random graphs technique is often employed by social scientists. The significance and incidence of certain topological fundamentals and subgraphs, such as triangles and stars, are encoded in the constraints of such graphs, which are derived from experimental data. A comparable modelling method for temporal exponential random graphs has been proposed. The fundamental purpose is to define the bounds of a period variable exponential graph model that effectively allows interaction between two points defined by an observable temporal development of vertices states, reflecting a dynamical system on the graph.

### 1.3.2 Randomized Reference Models

One common way for determining the relevance, unexpectedness, or over-under representation of a topological property in empirical networks is to compare the characteristics to a randomized reference model. The most used reference model is the configuration model, in which the original network's linkages are randomly rewired pairs. In this reference model, the novel degree arrangement is kept, but the networks are otherwise completely random. Time-domain structure and correlations in temporal graphs can be removed using a similar method. The original occurrence orders are reconstructed in a predetermined sequence at random. However, there are other different sorts of temporal correlations, as well as many times balances, that are important for constructing a null model. Relatively, one can switch off specific types of associations by building adequate null models to better understand how they contribute to the empirical temporal network's observable time-domain properties. The temporal null model has also been used to investigate the effects of various types of associations on dynamical operations on temporal graphs.

### 1.3.3 Randomly Permuted Times

While keeping the network architecture and the number of connections between all pairs of vertices same, the interaction times can be permuted as a temporal supplement to the conformation model. This is theoretically much easier than utilizing the edge redoing method, as it only requires randomly changing the timestamps of all contacts or simply rearranging the order. No tests equivalent to the RE rule are necessary for contact sequences; however, non-overlap should be tested for interval graphs. Because this null model retains all network architecture and the number of contacts for each boundary, it might be used to explore the effects of complicated event sequences like burstiness.

### 1.3.4 Contact Network Models

[Volz, 11] developed a simplified methodology for incorporating changes in neighbors into static graphs. This model is similar to the RE randomization process, except it is used to simulate the change of disease relationships.  As with the RE approach, it works by picking two borders at random and switching them. Because the model is based on reworking an existing network, the architecture of the accumulation network and the interaction patterns between restrictions must be established using other models.

### 1.3.5 Models of Social Group Dynamics

[Zhao, 11] proposed a framework for modelling social networks in which edges represent transient social connections such as face-to-face interaction. Their method is based on fundamental equations that represent the predictable change in the number of individuals in a collection of a specific dimension and can detention explanations like the longer interactions with a group, the less likely it is to authority the group; more the isolated the mediator is, the less likely it is to interrelate with a cluster.

## 1.4 Analysis

### 1.4.1 Randomization

A prominent approach for analyzing the influence of temporal structure in systems is to use randomized approaches to assess the influence of a fundamental feature of the system. Consider the following example of a fixed network notion to better understand the rationale for randomness. Strong grouping and smaller route sizes are characteristics of tangible systems. developed 'arbitrary' analogues to the real network, with a comparable number of clusters and links as the experimental systems, but with linkages introduced randomly among nodes. Researchers discovered that experimental systems contain orders of magnitude more route size grouping than their randomized counterparts, which were both larger and smaller. In limited devices, the degree spread is typically kept. The purpose of randomness is the same in temporal networks, but the spectrum of possible randomized algorithms is much wider. The approach remains the same: we want to anticipate the impact of a particular temporal networking characteristic and then test the impact by randomizing the feature.

To understand the significance of diurnal rhythms, the frequency can be shuffled, replaced with irregular intervals drawn from a homogeneous dispersion, connections can be shuffled to demolish geometrical frameworks, time can be reversed to identify the relevance of causal patterns, and so on. The objective would be to simulate a significant operation on theoretical networks and compare the characteristics of that method to those of an identical test performed on groups of networks, which would be more randomized than the original system. As its underlying structures correlate to communication occurrences, randomizing the network as per the methodologies discussed previously often does not make sense. This usually leads to configurations of linkages that would never happen in actual communication systems.

As a result, developing random methods that preserve these basic patterns and understanding how well the basic patterns affect current work on network randomizations is a promising field for further study Most randomizations in stable systems obey degrees distribution or higher-level patterns, the fundamental topology characteristic in such systems, therefore a system for randomness which follows network classes will be equivalent.

### 1.4.2 Generative Models

The notion of leveraging basic patterns to generate a new synthesized network is tightly linked to randomness. Another essential strategy of studying complicated dynamical processes is to use modelling techniques that mimic certain aspects of the subject in investigation and its behavior. Authentic synthesized information is significant as it allows one to examine moving systems using synthesized time systems. Synthesized systems have accessibility to unlimited quantities of information and since we made it, we can analyze the network's time variations and develop aggregates of systems to explore variation in result provided a certain characteristic, while genuine systems often have only a specific example, as a result, a variety of methods for generating temporal systems have been examined. The 'graph sequence technique,' which picks nodes as per a dense probability dispersion then connects these to a specific amount of Neighbour's layer by layer, is arguably the easiest way. This concept has always been the topic of extensive mathematical effort and has already been developed in a variety of methods owing to its accessibility. Another easy method is to create a stable system using a single-stage generation process, such as the configuration framework, and then design linking activation sequences the above-mentioned study on simplicial combinations is also another way. An array of 2D random walkers may also be used to create systems with interconnections emerging when walkers are close to one another. Some intriguing ideas use local rules to 'grow' networking devices. One may conceive of generative models for communities as a model for systems, even though they concentrate on bigger structures. The reason for employing Hawkes's methods is to establish temporal correlations, which would be premised on the notion that there is still a positive connection among event timings in experimental observations. This methodology has also been used to anticipate, for instance, retweet patterns.

Because the framework may not contain the limits on dynamics given by the underlying structure studies conducted on artificial information could have minimal significance for everyday situations in the context of all of these current dynamic systems. The dynamic class structure, on the other hand, provides an entirely new technique of creating artificial time structures.

One may develop network simulations by merely building time-sequences of genuine basic elements for a specific system because the basic frameworks are a reflection of each channel's real-world generating process Quantitative approaches may be used to measure the accuracy of these systems.

### 1.4.3 Link Activity and Link Prediction

The patterns as to how connections are active/inactive, as well as movement linkages among groups of connections in a system, are two further variable system properties heavily influenced by system type. Such trends are often driven by long-duration encounters among groupings of persons in face-to-face networking, while back-and-forth movements are frequent in text messaging channels.

The time connection forecasting is tightly linked to the link activities. Its goal is to analyze connection frequency trends as well as machine learning techniques to forecast future instances of connections in the system depending on regional node/link attributes. Anomaly detection is a big issue in stable concepts, notably in software engineering, and it concentrates on forecasting the existence of connections that have already been purposely destroyed or deleted due to congestion of certain forms. In temporal networks, the goal is to forecast all or some interconnections in the future time interval, for instance.

According to our knowledge of the changes in link activities across categories, it is indeed evident that perhaps the basic patterns provide a framework for explaining why link forecasting characteristics might range dramatically across networks.

There is nothing else quite like it among both forecasting links in sequential, numerous systems, in which temporal bridge are groups as well as systems usually survive for days, as well as message conversation channels, where people could be involved in numerous long discussions as well as text titbits were also small. As a result, link forecasting strategies built on one category of networking may perform badly on channels from other categories, because characteristics may differ drastically based on the structure category. (Gelardi *et al*., 2021) Whenever link prediction is being utilized to predict numbers for absent data, these concerns become much more relevant.

### 1.4.4 Spreading Processes

There has been scientific proof that there are modest changes in distributing procedures throughout multiple areas but that attitudes, behaviors, and data propagate in varied contexts than illnesses if one examines outside pandemic spread.

Researchers term the phenomenon complex contagion when several elements of contact to an invention are necessary for spread to occur. Boundary frameworks, in which transmission risk rises as a factor of a per cent of affected neighbors, are an important paradigm for modelling complicated contagion dynamics in temporal systems They have been applied to temporal systems as well.

### 1.4.5 Communities

In network-based, communities that include clusters of vertices with such a significant number of inner links, there is indeed a striking parallel between grouping in artificial intelligence and population identification in secure communication.

As a result, generalizations to temporal systems provide a lot of variation in techniques. The easiest method for recognizing temporal groups is to divide the collection of time-stamped connections into series of seismic energy, group every level separately but instead compare the community from throughout levels to the temporal groups. Three-way mass spectrometer, period charts, as well as probabilistic frame model are some of the methods that may effectively group the complete pile of chronological levels. The temporal development of its essential components inside the reactive courses, as stated in the Inquiry, is presently underdetermined in the system. Likewise, it is not sure how and where to define groups among those dynamical classifications. Nevertheless, the main point that needs to be emphasized about community is that approaches for detecting groups in temporal systems will most probably change based mostly on the channel's functional category.

# Chapter 2

## Dataset Description and Analysis

### 2.1 Dataset Overview

A crypto-trading platform is a platform for exchanging digital currencies.

A traditional market is one in which participants trade with each other without the aid of a central exchange or broker.

A decentralized market in which participants trade stocks, commodities, and currencies directly between peers using bitcoin, rather than through a central exchange or broker, is known as an Over-The-Counter market. In an OTC market, a deal can be completed between two members without the knowledge of other participants.

### 2.2 Dataset Description

The Bitcoin OTC trust network is the dataset employed from [Snap] in the research. ‘Snap’ is a Stanford Network Analysis Platform, which contains large network datasets. These datasets can be retrieved and analyzed using some techniques. Bitcoin OTC is a crypto network infrastructure, as well as a platform for buying and trading bitcoin. In a negotiation, a transaction is directed between two parties. The first individual sells a bitcoin, while the second person purchases one.

Every platform has its own security measures in place to protect users from scams or fraudulent activities that might compromise their privacy or cause financial harm. Defining a trust connection between the two parties is critical to avoiding fraudulent acts.

Due to anonymity, there is risk, which has led to the emergence of several exchanges where Bitcoin users rate the level of trust they have in other users. [Snap] created this dataset from an exchange - Bitcoin-OTC. These exchanges allow users to rate others on a scale of -10 to +10 (excluding 0). According to OTC’s guidelines, a rating of -10 should be given to fraudsters while at the other end of the spectrum, +10 means “you trust the person as you trust yourself”. The other rating values have intermediate meanings. Therefore, this exchange explicitly yields WSNs. By separating each user into a ‘rater' with all of its outgoing edges and a ‘product' with all of its incoming edges, the network is rendered bipartite. Members of the platform assess each other on a scale of -10 to +10, with -10 representing total distrust and 10 representing total trust.

The aim of the thesis can be explored using a dataset in csv format.

The format of the data is as follows:

*SOURCE:* Rater.

*TARGET:* Ratee.

*RATING:* Rate, source’s rating for target ranging from -10 to +10.

*TIME:* The time of rating measure as seconds since epoch. The data is ordered according to the time.

Source, Target, and Rating being of the 'integer' type, and Time being of the 'float' type since it is measured in seconds since the epoch.

## 2.3 Objective

The objective of this thesis is to use the Bitcoin OTC Trust Network to develop and analyze the initial network and compute the temporal metrics. The main purpose of this research is to answer the below mentioned research questions.

## 2.4 Aim

The aim of this project is to analyze the research questions that are related to the network. We analyze the network topological structure and answer the queries regarding the process of densification, we check if the network is small-world, scale free and analyze if the central nodes change with time.

## 2.5 Research Questions

**Question 1:** Is the network a Scale Free Network?

**Question 2:** Is the network a Small World Network?

**Question 3:** What is the Densification process of the network?

**Question 4:** Does the Central nodes change?

## 2.6 Dataset Exploration

Python is a clear and powerful object-oriented programming language, compared to Perl, Ruby, Scheme, or Java.

Python's features use an elegant syntax, making the programs that are written easier. It is an easy-to-use programming language and, is ideal for prototype development and other ad-hoc programming tasks. It has large standard library that supports many programming tasks.

Python Libraries are a set of useful functions that eliminate the need for writing codes from scratch. Python libraries play a vital role in developing machine learning, data science, data visualization, image and data manipulation applications and more. The libraries that are considered in this research are

*Pandas* is a sophisticated and versatile open-source data analysis and manipulation tool built on the Python programming language that is used to examine the data.

*NumPy* is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Network X is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Network X is a library for graph representation in Python. Developers can use it to create, manipulate, and visualize graphs, as well as for non-visual graph data science analysis.

*Matplotlib* is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

*Seaborn* is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

*Power law* is a Python package for analysis of heavy-tailed distributions. This software package provides easy commands for basic fitting and statistical analysis of distributions. Notably, it also seeks to support a variety of user needs by being exhaustive in the options available to the user.

*tqdm* is a Python library that allows you to output a smart progress bar by wrapping around any iterable.

In this paper, python programming language was enhanced to analyze the network, as python consists of numerous built-in libraries to analyze complex networks. The networkX library in python, helps us to analyze and explore the structure of the network.

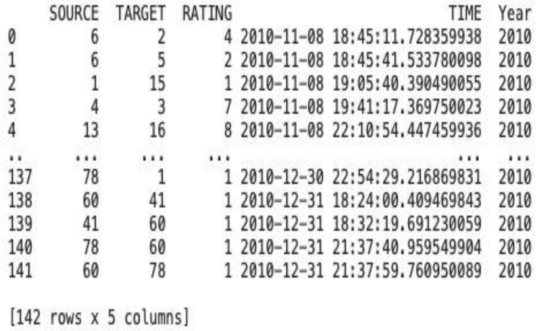
The analysis was carried out using python networkx library. Initially, as the dataset is extracted from [Snap], the time dimension has been measured in seconds per epochs, this measure is converted into human readable format to analyze the timeline of the data.

Therefore, the source completes the execution of trade and rates the target, based on the trustworthiness. The time at which the source rates the target are considered. Time ordering at which the ratings happen are analyzed, in our case, the time is ordered according to continuous timeline. Hence, I have created a network, by initially creating an empty graph and adding nodes to the existing graph.

As mentioned earlier, the time dimension in the temporal networks is divided into particular snapshots. These snapshots are the time-snaps, where, they contain data that is framed on some timeframe.

In our case, this data considers ratings from 08th November 2010 to 25th January 2016. The snapshots have been framed on the basis of 'year'; therefore, 7 snapshots have been considered based on each year ('2010','2011','2012','2013','2014','2015','2016').

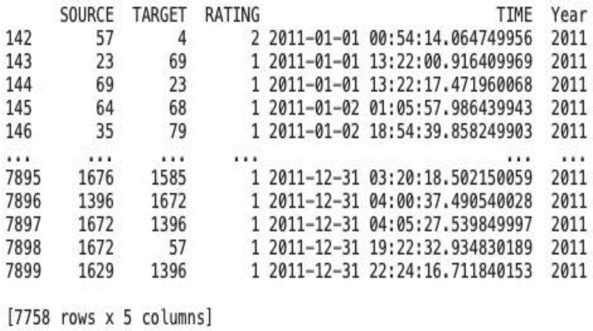
Each snapshot contains the information of source rating the target from the mentioned years. The snapshots description is seen below.



```
     SOURCE  TARGET  RATING                          TIME  Year
0         6       2       4 2010-11-08 18:45:11.728359938  2010
1         6       5       2 2010-11-08 18:45:41.533780098  2010
2         1      15       1 2010-11-08 19:05:40.390490055  2010
3         4       3       7 2010-11-08 19:41:17.369750023  2010
4        13      16       8 2010-11-08 22:10:54.447459936  2010
..      ...     ...     ...                           ...   ...
137      78       1       1 2010-12-30 22:54:29.216869831  2010
138      60      41       1 2010-12-31 18:24:00.409469843  2010
139      41      60       1 2010-12-31 18:32:19.691230059  2010
140      78      60       1 2010-12-31 21:37:40.959549904  2010
141      60      78       1 2010-12-31 21:37:59.760950089  2010

[142 rows x 5 columns]
```

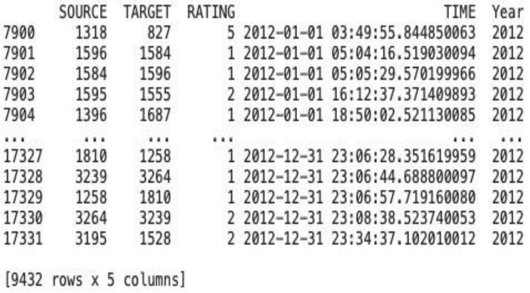
Table 1(a). Snapshot 1, where the number of nodes = 55 and number of edges = 142.



```
      SOURCE  TARGET  RATING                          TIME  Year
142       57       4       2 2011-01-01 00:54:14.064749956  2011
143       23      69       1 2011-01-01 13:22:00.916409969  2011
144       69      23       1 2011-01-01 13:22:17.471960068  2011
145       64      68       1 2011-01-02 01:05:57.986439943  2011
146       35      79       1 2011-01-02 18:54:39.858249903  2011
...      ...     ...     ...                           ...   ...
7895    1676    1585       1 2011-12-31 03:20:18.502150059  2011
7896    1396    1672       1 2011-12-31 04:00:37.490540028  2011
7897    1672    1396       1 2011-12-31 04:05:27.539849997  2011
7898    1672      57       1 2011-12-31 19:22:32.934830189  2011
7899    1629    1396       1 2011-12-31 22:24:16.711840153  2011

[7758 rows x 5 columns]
```

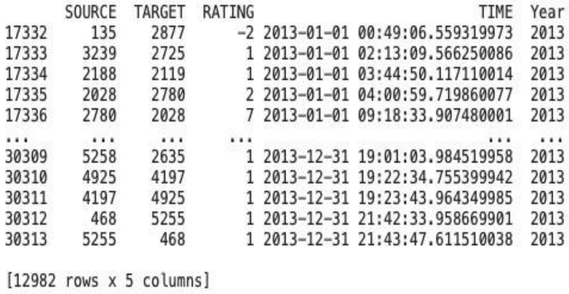
Table 1(b). Snapshot 2, where the number of nodes = 1625 and number of edges = 7758.



```
       SOURCE  TARGET  RATING                          TIME  Year
7900     1318     827       5 2012-01-01 03:49:55.844850063  2012
7901     1596    1584       1 2012-01-01 05:04:16.519030094  2012
7902     1584    1596       1 2012-01-01 05:05:29.570199966  2012
7903     1595    1555       2 2012-01-01 16:12:37.371409893  2012
7904     1396    1687       1 2012-01-01 18:50:02.521130085  2012
...       ...     ...     ...                           ...   ...
17327    1810    1258       1 2012-12-31 23:06:28.351619959  2012
17328    3239    3264       1 2012-12-31 23:06:44.688800097  2012
17329    1258    1810       1 2012-12-31 23:06:57.719160080  2012
17330    3264    3239       2 2012-12-31 23:08:38.523740053  2012
17331    3195    1528       2 2012-12-31 23:34:37.102010012  2012

[9432 rows x 5 columns]
```

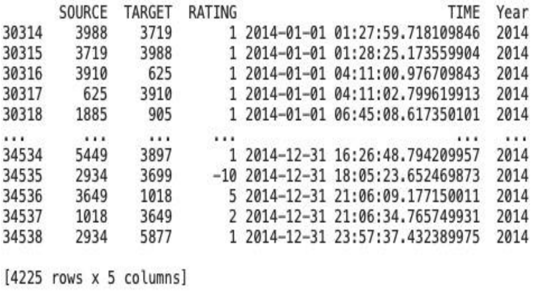
Table 1(c). Snapshot 3, where the number of nodes. = 1926 and number of edges = 9432.



```
       SOURCE  TARGET  RATING                          TIME  Year
17332     135    2877      -2 2013-01-01 00:49:06.559319973  2013
17333    3239    2725       1 2013-01-01 02:13:09.566250086  2013
17334    2188    2119       1 2013-01-01 03:44:50.117110014  2013
17335    2028    2780       2 2013-01-01 04:00:59.719860077  2013
17336    2780    2028       7 2013-01-01 09:18:33.907480001  2013
...       ...     ...     ...                           ...   ...
30309    5258    2635       1 2013-12-31 19:01:03.984519958  2013
30310    4925    4197       1 2013-12-31 19:22:34.755399942  2013
30311    4197    4925       1 2013-12-31 19:23:43.964349985  2013
30312     468    5255       1 2013-12-31 21:42:33.958669901  2013
30313    5255     468       1 2013-12-31 21:43:47.611510038  2013

[12982 rows x 5 columns]
```

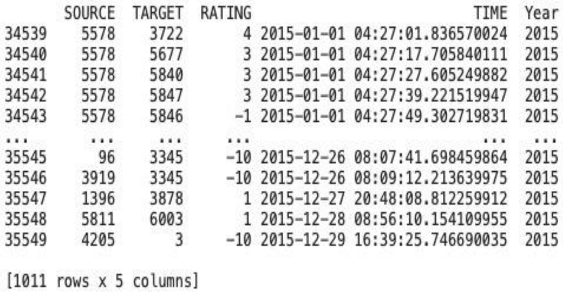
Table 1(d). Snapshot 4, where the number of nodes =. 2682 and number of edges = 12982.



```
       SOURCE  TARGET  RATING                          TIME  Year
30314    3988    3719       1 2014-01-01 01:27:59.718109846  2014
30315    3719    3988       1 2014-01-01 01:28:25.173559904  2014
30316    3910     625       1 2014-01-01 04:11:00.976709843  2014
30317     625    3910       1 2014-01-01 04:11:02.799619913  2014
30318    1885     905       1 2014-01-01 06:45:08.617350101  2014
...       ...     ...     ...                           ...   ...
34534    5449    3897       1 2014-12-31 16:26:48.794209957  2014
34535    2934    3699     -10 2014-12-31 18:05:23.652469873  2014
34536    3649    1018       5 2014-12-31 21:06:09.177150011  2014
34537    1018    3649       2 2014-12-31 21:06:34.765749931  2014
34538    2934    5877       1 2014-12-31 23:57:37.432389975  2014

[4225 rows x 5 columns]
```

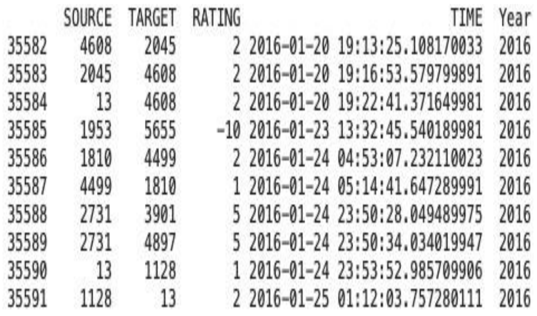
Table 1(e). Snapshot 5, where the number of nodes = 1145 and number of edges = 4225.



```
       SOURCE  TARGET  RATING                          TIME  Year
34539    5578    3722       4 2015-01-01 04:27:01.836570024  2015
34540    5578    5677       3 2015-01-01 04:27:17.705840111  2015
34541    5578    5840       3 2015-01-01 04:27:27.605249882  2015
34542    5578    5847       3 2015-01-01 04:27:39.221519947  2015
34543    5578    5846      -1 2015-01-01 04:27:49.302719831  2015
...       ...     ...     ...                           ...   ...
35545      96    3345     -10 2015-12-26 08:07:41.698459864  2015
35546    3919    3345     -10 2015-12-26 08:09:12.213639975  2015
35547    1396    3878       1 2015-12-27 20:48:08.812259912  2015
35548    5811    6003       1 2015-12-28 08:56:10.154109955  2015
35549    4205       3     -10 2015-12-29 16:39:25.746690035  2015

[1011 rows x 5 columns]
```

Table 1(f). Snapshot 6, where the number of nodes = 371 and number of edges = 1011.



```
       SOURCE  TARGET  RATING                          TIME  Year
35582    4608    2045       2 2016-01-20 19:13:25.108170033  2016
35583    2045    4608       2 2016-01-20 19:16:53.579799891  2016
35584      13    4608       2 2016-01-20 19:22:41.371649981  2016
35585    1953    5655     -10 2016-01-23 13:32:45.540189981  2016
35586    1810    4499       2 2016-01-24 04:53:07.232110023  2016
35587    4499    1810       1 2016-01-24 05:14:41.647289991  2016
35588    2731    3901       5 2016-01-24 23:50:28.049489975  2016
35589    2731    4897       5 2016-01-24 23:50:34.034019947  2016
35590      13    1128       1 2016-01-24 23:53:52.985709906  2016
35591    1128      13       2 2016-01-25 01:12:03.757280111  2016
```

*Figure 3:Tables of snapshots.*

Table 1(g). Snapshot 7, this snapshot displays the last 10 nodes. The number of nodes = 39 and number of edges = 42.

For brief analysis of the network, the figure below describes the edge rating frequency and distribution. It gives detailed insights on ratings given by the users.

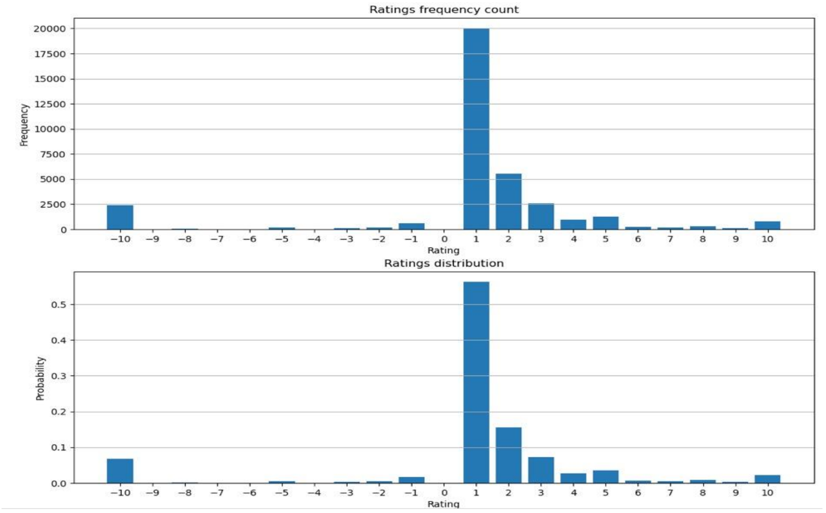
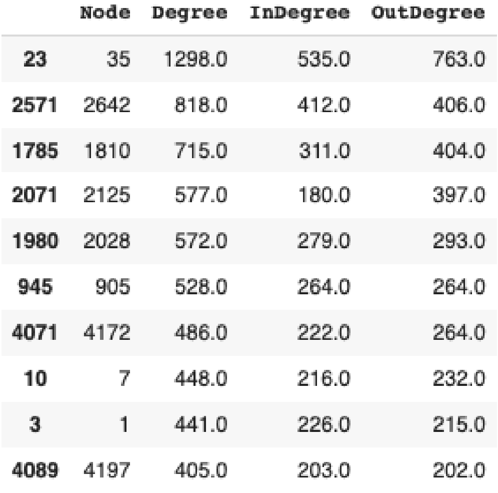


*Figure 4:Illustrates the frequency and probability distribution of rating*

The rating frequency is the count of users who has rated another user on a specific count ranging from +10 to –10. From the above figure, we conclude that the rating ‘1’, is the most frequently used rating by users.

The degree centrality measures the significance of a node by checking the number of edges associated to it. The degree centrality for a node v is the fraction of nodes it is connected to. The degree centrality values are normalized by dividing by the maximum possible degree in a simple graph n-1 where n is the number of nodes in G. In our case, the network is directed so in- and out-degrees ought to be considered. This parameter gives us a sense of the significance of a node. In-degree is the number of connections that point inward at a node. Out-degree is the number of connections that originate at a node and point outward to other nodes. Formally, the in-degree centrality for a node v is the fraction of nodes its incoming edges are connected to. The out-degree centrality for a node v is the fraction of nodes its outgoing edges are connected to. Python library 'networkX' helps us to compute the degree centrality measures.

The degree measures are computed for all the nodes and the top 10 nodes with the highest degree centrality are shown in the below table.



| | Node | Degree | InDegree | OutDegree |
|---|---|---|---|---|
| **23** | 35 | 1298.0 | 535.0 | 763.0 |
| **2571** | 2642 | 818.0 | 412.0 | 406.0 |
| **1785** | 1810 | 715.0 | 311.0 | 404.0 |
| **2071** | 2125 | 577.0 | 180.0 | 397.0 |
| **1980** | 2028 | 572.0 | 279.0 | 293.0 |
| **945** | 905 | 528.0 | 264.0 | 264.0 |
| **4071** | 4172 | 486.0 | 222.0 | 264.0 |
| **10** | 7 | 448.0 | 216.0 | 232.0 |
| **3** | 1 | 441.0 | 226.0 | 215.0 |
| **4089** | 4197 | 405.0 | 203.0 | 202.0 |

*Figure 5:Table showing the top 10 nodes with highest degree.*

Here, the node '35' has the highest degree among all the other nodes as it has 1298 total connections, where, 535 are in coming edges and 763 are out going edges.

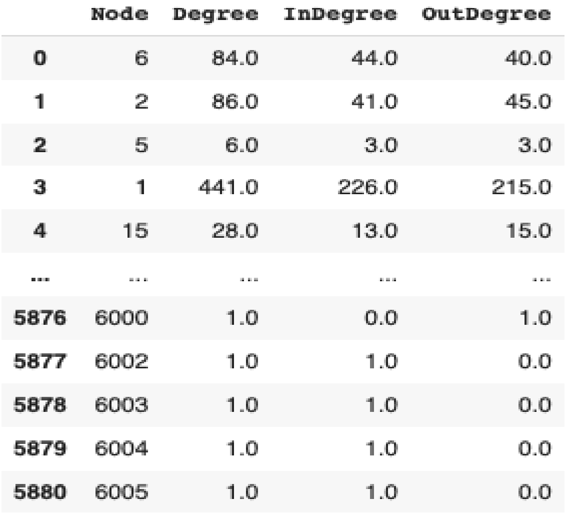
Another curious property of networks is the reality that there are a lower number of nodes with as it were out-links but a higher number of nodes with as it were in-links. Having as it were in-links proposes that clients are inactive. The larger part of these nodes is appraised adversely by dynamic clients.

### 2.6.1 Degree Distribution

*Degree*

In an undirected network, the degree of a node can be defined as the number of edges that linked to one node. Nodes have in-degree and out-degree, which are represented by din,dout in the case of directed edges. In our network, the degree represents the number of ratings of source.

*Observation: In any directed graph, the summation of in-degrees is equal to the summation of out-degree.* According to the below table, the number of degrees, id-degrees, out-degrees of first 5 and last 5 nodes have been shown. They explain the above observation in detail. I consider the degree of node ‘6’ is 84, according to the observation the sum of number of in-degree and number of out-degree of a node are equal to the degree. So, the number of in-degrees for node is ‘44’ and number of out-degree is ‘40’. Similarly, it follows the same for all the nodes.



| | Node | Degree | InDegree | OutDegree |
|---|---|---|---|---|
| **0** | 6 | 84.0 | 44.0 | 40.0 |
| **1** | 2 | 86.0 | 41.0 | 45.0 |
| **2** | 5 | 6.0 | 3.0 | 3.0 |
| **3** | 1 | 441.0 | 226.0 | 215.0 |
| **4** | 15 | 28.0 | 13.0 | 15.0 |
| **...** | ... | ... | ... | ... |
| **5876** | 6000 | 1.0 | 0.0 | 1.0 |
| **5877** | 6002 | 1.0 | 1.0 | 0.0 |
| **5878** | 6003 | 1.0 | 1.0 | 0.0 |
| **5879** | 6004 | 1.0 | 1.0 | 0.0 |
| **5880** | 6005 | 1.0 | 1.0 | 0.0 |

*Figure 6:Observations*

According to [Barbasi, 09], the distribution of node degree is an essential component of complex networks. The members of any distribution can be used to characterize it. These are the degrees of all nodes in the network in our example. The degree distribution $p_d$ (or $P(d)$, or $P(d_v = d)$) gives the probability that a randomly selected node *v* has degree *d*. Because $p_d$ is a probability distribution □$^{\infty}_{d=0}$ $p_d = 1$. In a graph with *n* nodes, $p_d$ is defined as $p_d = {}^{n_d}$ / n, where $n_d$ is the number of nodes with degree *d.*

For example, for a given node ‘u’, it is calculated as the outright contrast between the in-rating for node ‘u’(uin) and the out-rating for node ‘u’)(uout). The degree distribution was computed and plotted by using power law.

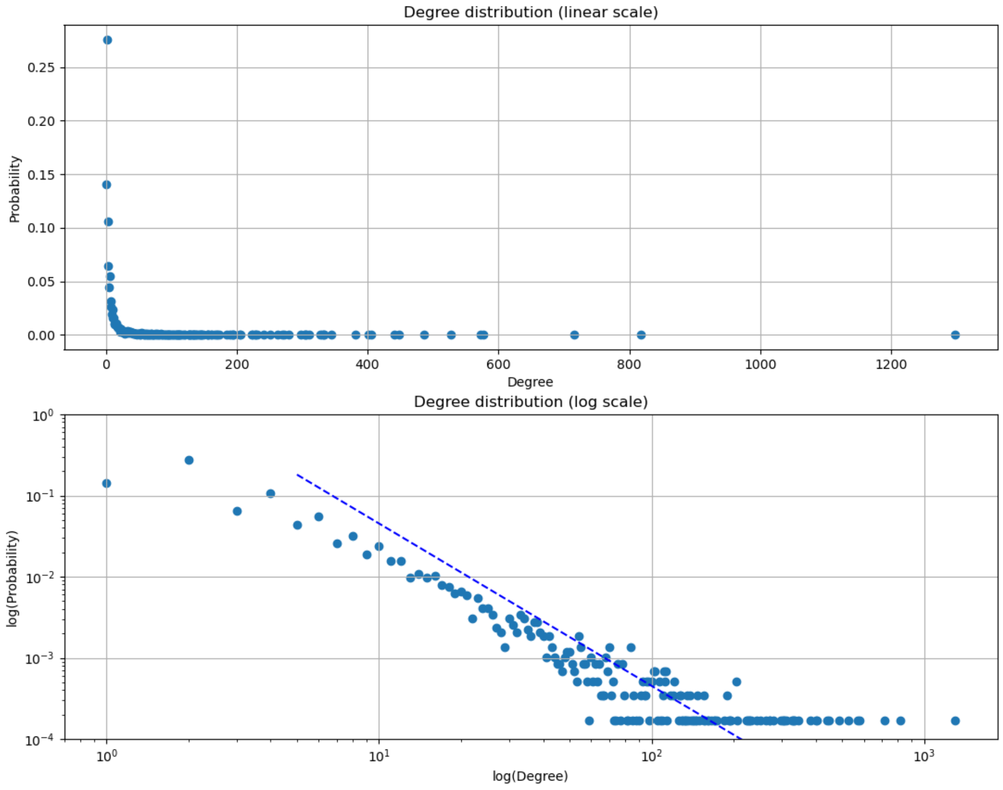


*Figure 7:Plot showing the degree distribution in a. linear and b. log scale*

Figure 7a shows the linear degree distribution of nodes and figure 7b shows the log representation of degree distribution. The x-axis represents the degree and y-axis represents the probability. The probability distribution can be computed by the frequency of degree of node ‘x’ by the total frequency.

### 2.6.2 In-degree distribution

The number of connections a node has to other nodes determines its degree, and the degree distribution is the probability distribution of these degrees throughout the whole network. The number of edges linking a vertex determines its degree. The number of head ends next to a vertex is referred to as the vertex's indegree. In-degree distribution in our case is the distribution of probability and degree of incoming nodes.

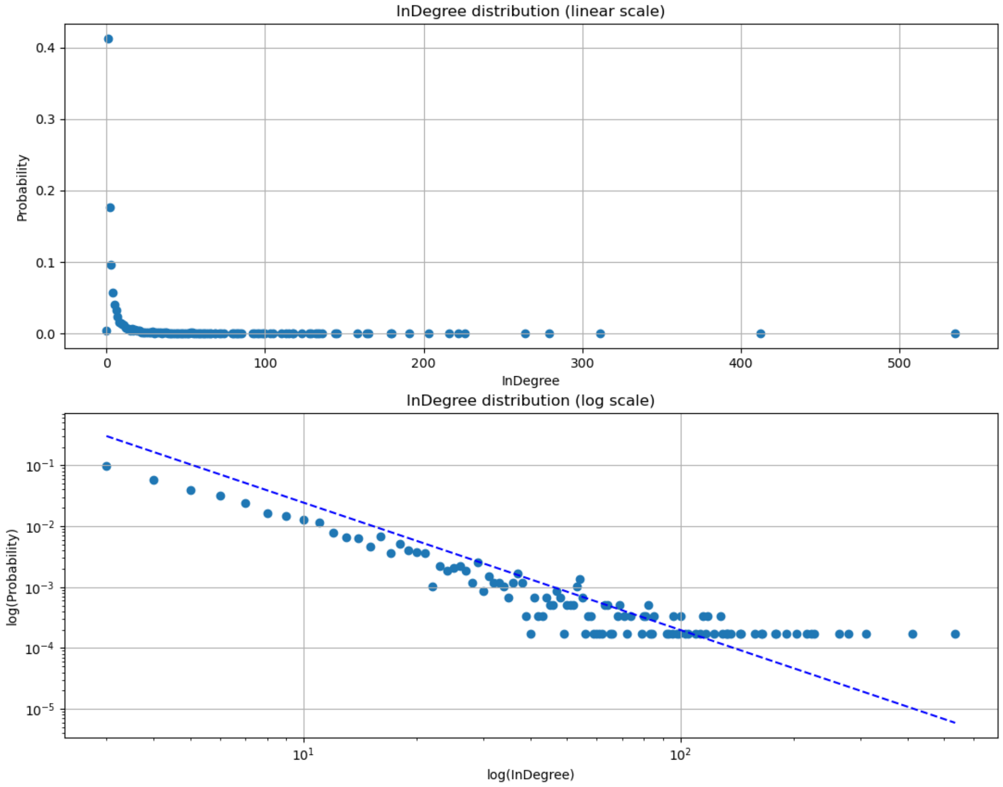


*Figure 8:Plot showing indegree distribution. Fig a. linear scale, Fig b. log scale.*

### 2.6.3 Out-degree distribution

Out-degree distribution is the distribution of probability and degree of outgoing nodes. The Out-degree of a vertex V written by deg+(v), is the number of edges with v as an initial vertex. To find the out-degree of a vertex we count the number of edges from the vertex.

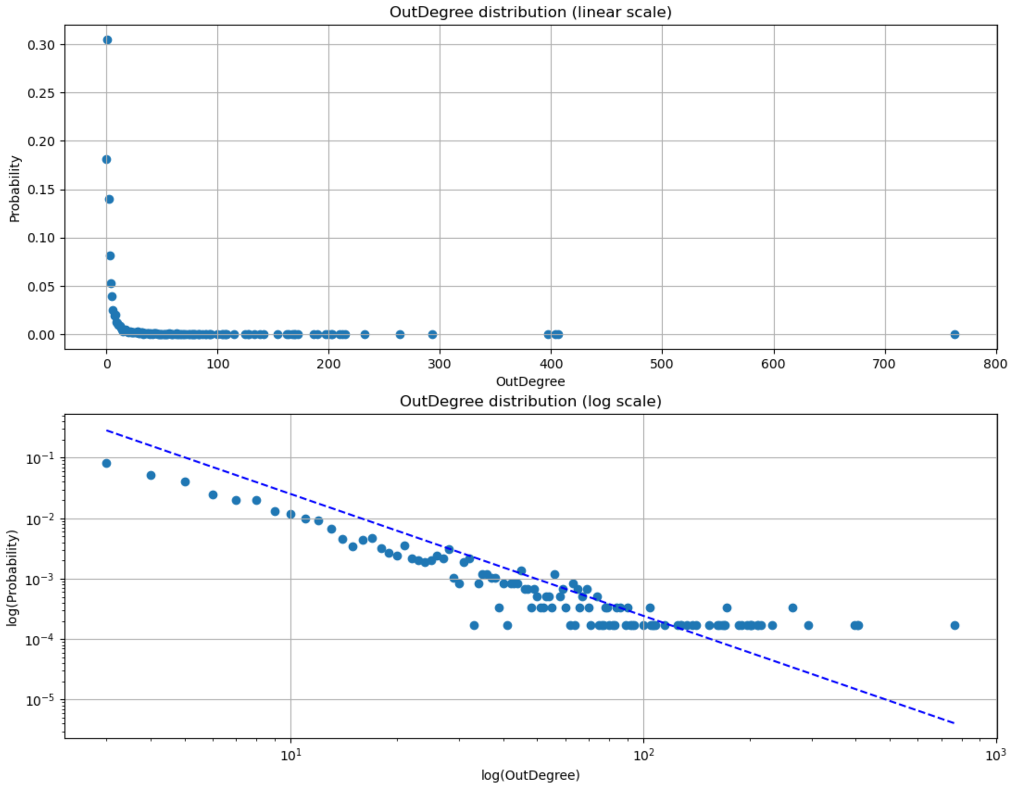


*Figure 9:Plot showing outdegree distribution. Fig a. linear scale, Fig b. log scale.*

The figure above shows the plot between probability and the out-degree measure. The plot represents the out-degree distribution of each node and figure ‘b’ shows the log representation of out-degree distribution.

### 2.6.4 Network Summary

The analysis of incoming links and outgoing links gives an additional information of the summary of the network. We can deduct from the analysis that there are 1067 total incoming links and 4814 outgoing links.

The plot below, depicts the distribution of ratings of incoming links and outgoing links. The information on the ratings tells rating pattern of each incoming link and outgoing link.

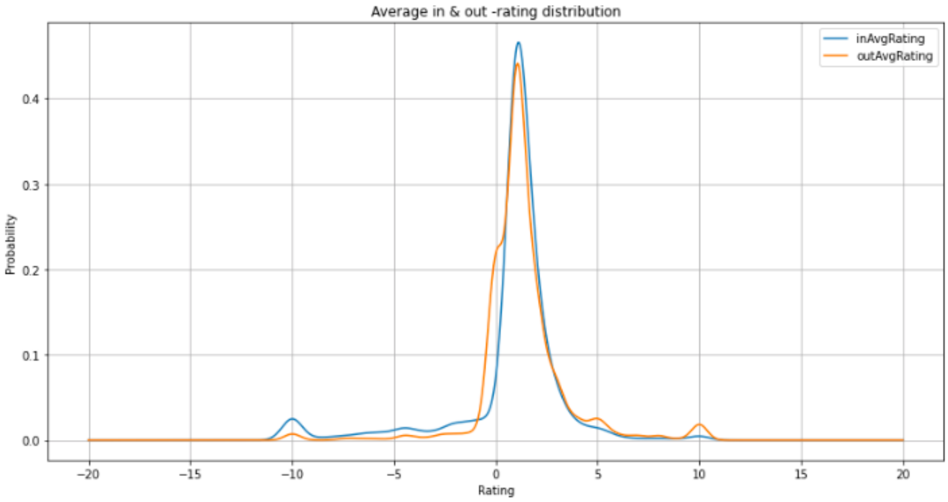


*Figure 10:Plot displaying the average in and out degrees.*

We can conclude that most of the user's rate others well and get positive review. The users who get negative rates and rate others adversely might demonstrate false conduct.

### 2.6.5 Connected Components

Connected components are defined as groups of vertices between which a path can always be discovered in static networks, and vertices are either connected or not in static networks. For directed or timed graphs, connectedness is not a symmetric relation, as previously stated. The attribute of connectedness in directed graphs may be separated into two parts: strong connectivity, where all pairings of vertices have a directed path, and weak connectivity, where all pairs of vertices have a path if the edges are regarded undirected. These two ideas may be applied to temporal networks as well. Nicosia et al., proposed two definitions: two vertices i and j of a temporal network are defined to be strongly connected if there is a directed, time-respecting path connecting i to j and vice versa, while they are weakly connected if there are undirected time-respecting paths from i to j and j to i, i.e. the directions of the contacts are not considered.

A node vi is connected to node vj (or vj is reachable from vi) if there exists a path from vi to vj. A graph is connected if there exists a path between any pair of nodes in it. In a directed graph, the graph is weakly connected if there exists a path between any pair of nodes, without following the edge directions. The graph is strongly connected if there exists a directed path between any pair of nodes. A component in an undirected graph is a subgraph such that, there exists a path between every pair of nodes inside the component. Connected component is a set of vertices in a graph that are linked to each other by paths.

According to our analysis, 80% of the nodes have a place to the same unequivocally associated component. Twenty-two components have an estimate between 2 and 6 components. All remaining SSCs comprise of as it were one node(singletons).

## 2.7 Additional Papers on this Dataset

## Paper 1

#### Summary

S. Kumar, F. Spezzano, V.S. Subrahmanian, and C. Faloutsos referenced this research at the IEEE International Conference on Data Mining (ICDM), 2016.

The authors of this work looked at the challenge of predicting the weights of edges in WSNs. They propose two node behavior measures, *GOODNESS,* goodness of a node intuitively captures how much this node is trusted by other nodes and *FAIRNESS*, fairness of a node captures how fair the node is in rating other nodes trust level, as well as axioms that these two ideas must satisfy. They provide these two notions definitions and show that they converge to a single solution in linear time.

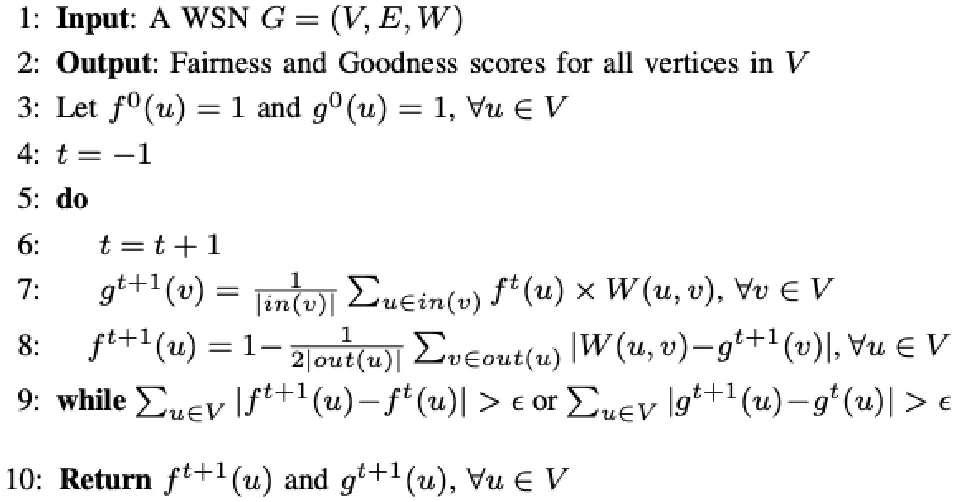
The fairness and goodness measures nearly always have the best prediction power when compared to many separate algorithms from both the signed and unsigned social network literature, according to the authors. They then propose a Fairness-Goodness Algorithm (FGA) that repeatedly computes these two scores at the same time, proving that the method will converge in linear time. They look at the BTC OTC Trust network and do a number of tests to see how accurate the procedures are. They show that when compared against several individual algorithms from both the signed and unsigned social network literature, the fairness and goodness metrics almost always have the best predictive power. They then use these as features in different multiple regression models and show that we can predict edge weights on Bitcoin Trust Network. Moreover, fairness and goodness metrics form the most significant feature for prediction in most cases. This paper is the first to show how to predict edge weights in weighted signed networks and contributes:

- *Effectiveness in Signed Edge Weight Prediction:* They show that fairness and goodness can be used to calculate unknown weights in WSNs with higher precision than previous techniques.
- *Convergence and uniqueness:* They show that fairness and goodness converge to a unique value in time linear to the size of the network.
- *Novel metrics:* They proposed two vertex-based metrics called fairness and goodness to assess the reliability of a node in rating others, and to assess how much the node is liked/trusted by other nodes, respectively.

## Proposed Algorithm

### Fairness and Goodness Algorithm

FGA algorithm to compute fairness and goodness scores for each vertex in the network. By default, all vertices' fairness and goodness scores are set to 1 and 1, respectively, by default (line 3). In line 7, the goodness score for each vertex is adjusted using the previous iteration's fairness scores. In line 8, the fairness scores are changed in the same iteration using the newly updated goodness scores. The goodness of all the vertices, for example, becomes their average in-degree after the first repetition. Both the goodness and fairness scores are recursive, and they are updated until they converge (line 9). When the difference between fairness and goodness scores for all vertices in consecutive iterations is smaller than an error bound, the algorithm converges. The final scores are the fairness and goodness scores from the previous iteration (line 10).



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

## Paper 2

### Summary

This paper was cited by Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V.S. Subrahmanian in Proceedings of 11th ACM International Conf. on Web Search and Data Mining (WSDM 2018).

In this paper, the authors present Rev2, a system to identify such fraudulent users. They propose three interdependent intrinsic quality metrics—fairness of a user, reliability of a rating and goodness of a product. The fairness and reliability quantify the trustworthiness of a user and rating, respectively, and goodness quantifies the quality of a product.

They propose six axioms to establish the interdependency between the scores, and then, formulate a mutually recursive definition that satisfies these axioms, and extend the formulation to address cold start problem and incorporate behavior properties. They develop the Rev2 algorithm to calculate these intrinsic scores for all users, ratings, and products by combining network and behavior properties and prove that this algorithm is guaranteed to converge and has linear time complexity. By conducting extensive experiments on BTC OTC rating datasets, they show that Rev2 outperforms nine existing algorithms in detecting fraudulent users This task is challenging due to the lack of training labels, disbalance in the percentage of fraudulent and non-fraudulent users, and camouflage by fraudulent users.

They model user-to-item ratings with timestamps as a directed bipartite graph. For instance, on an online marketplace such as Amazon, a user u rates a product p with a rating (u,p). We propose that each user has an (unknown) intrinsic level of fairness F (u), each product p has an (unknown) intrinsic goodness G(p), and each rating (u,p) has an (unknown) intrinsic reliability R(u,p). Intuitively, a fair user should give ratings that are close in score to the goodness of the product, and good products should get highly positive reliable ratings. Clearly, F(u),G(p),R(u,p) are all inter-related.

They define an axiom that establishes the relation between the intrinsic scores and their behavior properties, and propose a Bayesian technique to incorporate the behavior properties of users, ratings, and products into the formulation by penalizing for unusual behavior. Cold start treatment and behavioral properties together, and present the Rev2 formulation and an iterative algorithm to find the fairness, goodness and reliability scores of all entities together.

The authors presented the Rev2 algorithm to address the problem of identifying fraudulent users in rating networks. This paper has the following contributions:

- *Algorithm:* They defined three mutually-recursive metrics—fairness of users, goodness of products and reliability of ratings. They incorporated behavioral properties of users, ratings, and products in these metrics, and extended it to address cold start problem. They proposed the Rev2 algorithm to iteratively compute the values of these metrics.

- *Theoretical guarantees:* They proved that Rev2 algorithm has linear time complexity and is guaranteed to converge in a bounded number of iterations.
- *Effectiveness:* By conducting five experiments, they showed that Rev2 outperforms nine existing algorithms to predict fraudulent users.

## Proposed Algorithm

### The REV2 Algorithm

The authors present the Rev2 algorithm in Algorithm 1 to calculate the metrics' values for all users, products and ratings. The algorithm is iterative, so let F , G and R denote the fairness, goodness and reliability score at the end of iteration t . Given the rating network and non-negative integers α1,α2,β1,β2,γ1,γ2 and γ3, we first initialize all scores using their respective behavior scores ΠU , ΠR , and ΠP . When behavior scores are not present, then these scores are initialized to highest value 1. Then we iteratively update the scores using equations in REV2 formulation until convergence. Convergence occurs when all scores change minimally. ε is the acceptable error bound.

They set the values of α1,α2,β1,β2,γ1,γ2 and γ3 in unsupervised, the algorithm is run for several combinations of α1,α2,β1,β2,γ1,γ2 and γ3 as in- puts, and the final scores of a user across all these runs are averaged to get the final Rev2 score of the user. Formally, let C be the set of all parameter combinations {α1,α2,β1,β2,γ1,γ2,γ3}, and F(u|α1,α2,β1,β2,γ1,γ2,γ3) be the fairness score of user 'u' after running Algorithm withα1,α2,β1,β2,γ1,γ2,γ3 as inputs.

In a supervised scenario, it is indeed possible to learn the relative importance of parameters. They represent each user u as a feature vector of its fairness scores across several runs, i.e. F(u|α1,α2,β1,β2,γ1, γ2,γ3), ∀(α1,α2,β1,β2,γ1,γ2,γ3) ∊ C are the features for user u. Given a set of fraudulent and benign user labels, a random forest classifier is trained that learns the appropriate weights to be given to each score. The higher the weight, the more important the particular combination of parameter values is. The learned classifier's prediction labels are then used as the supervised Rev2 output.

# Chapter 3

## 3.1 Specific Analysis

Real-world networks, particularly social networks, have a distinct structure that sets them apart from randomly generated mathematical networks. Scale- free and small world phenomenon are two characteristics of a static network, these properties define a network, if it is random or static.

According to [Barabasi, 14], earlier, it was considered that real-world networks exhibited random network features. Erdos and Renyi defined random networks theoretically, stating that the probability of two nodes is constant. The poisson degree distribution is investigated in such networks (most nodes have degree closer to average). The poisson curve decreases exponentially, implying that the number of nodes linked will decrease.

[Barabasi, 14] has demonstrated that real-world networks do not display random network features since the degree distribution is a power law. When compared to the poisson distribution curve, the power law decays more slowly. The power law expects a few nodes to be very highly connected and nodes have smaller degree than the average. The characteristics of static networks are scale-free and small world phenomenon.

Is the network a Scale-free network?

According to [Barabasi, 14], scale-free networks are defined as the networks that has a power law degree distribution.

Power law distribution

The power law can be mathematically represented as p(k) = akpower−b. Let k be the degree of a node, p(k) denotes the fraction of node with degree k, here b is the power law exponent and a is the power-law intercept. By taking the logarithm from both sides for the mathematical representation of power law and obtain the log-log plot of a power law distribution which is a straight line.

If the features of a network are independent of its size, i.e., the number of nodes, it is said to be scale-free. This means that the basic structure of the network does not change as the network expands. This form of P(k) decays slowly as the degree increases, increasing the likelihood of finding a node with a very large degree. Networks with power-law distributions are called scale-free because power laws have the same functional form at all scales.

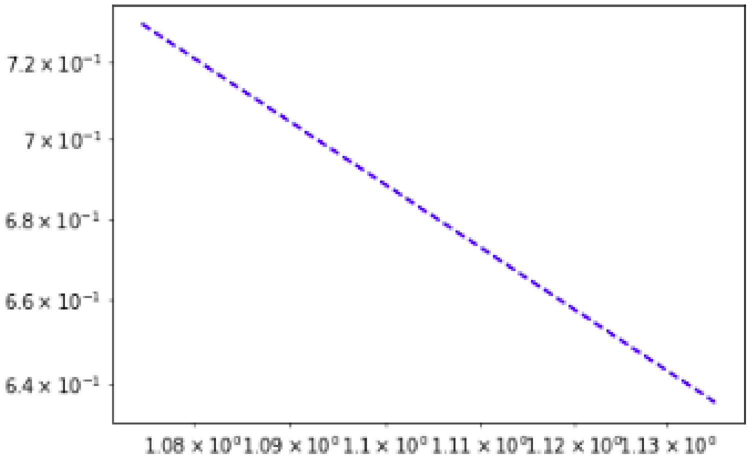


*Figure 11:Plot for Scale-free network*

In our case, the value is 2.53, that has been computed using the power law, the power law library was manually imported in python programming language.  The above plot also shows that the temporal network follows the power law strongly. The degree distribution follows the power law rule. Hence the network is scaling free network and the values of alpha calculated are also minimal.

Is the network a small world network?

According to [Zafarani, 14], in a random graph model the connections are formed at random. Random graphs can model average path lengths in real world network but not clustering coefficient. Small world phenomenon was proposed by Duncan and Steven, to address this problem. In real world networks, the nodes have limited or fixed number of nodes. Any two nodes of a network are connected via paths.  The average path length remains small.

In our research, the small world phenomenon is carried out using the built-in library in networkx. In networkx, the function is used to estimate the small world ness of a graph.

A small world network is characterised by a small average shortest path length and a large clustering coefficient. The small wordless is commonly measured with the coefficient omega or sigma. The coefficient compares the average clustering coefficient and shortest path length of a given graph. I have used the omega coefficient, which can be imported using the networkx algorithms, which returns the small world coefficient of a graph and ranges between-1 and 1. The values closer to 0 indicate the graph ‘g’ features small world phenomenon, whereas, the values close to -1 indicates that the graph g has lattice shape and value closer to 1 indicates that the graph g is a random graph. In our analysis, sigma is 0.96 at the 100th node. Because the Value is closer to 1. The network is a not a small world network, but a random graph. It will become even larger when additional nodes are added

## 3.2 Temporal analysis

### Densification Process

In many social and economic dynamical networks, the number of edges grows superlinearly in number of nodes is known as the "densification power law," in which the average degree grows with the number of nodes, i.e. "densification." A similar sort of scaling comes from population dynamics in temporal networks, when nodes join and exit the system in a series of network snapshots, keeping the likelihood of two nodes being the same low. In such systems, densification is "explosive," with the scaling exponent growing with N. The densification process of scaling may be isolated and recognized from data if a certain kind of scaling is seen. The relationship between the temporal evolution of the degree distribution and graph densification is investigated.

Densification is the process of increasing the network's density. According to our dataset, there are 7 snapshots, I have investigated the densities of each snapshot and monitored it, by considering how it changes over time. We analyze how the networks dense over time, that is, the number of edges grow super linealry in number of nodes. I have considered a densification power law plot and investigate if the network dense over time. Initially the computed density of each snapshot is 0.04781145, 0.00294982, 0.00173405, 0.00113831, 0.00104375,0.00102874, 0.00102926 respectively. Here, I have considered the densification plot for the above densities of snapshots. The densification plot refers to the densities of snapshots over time in linear scale.

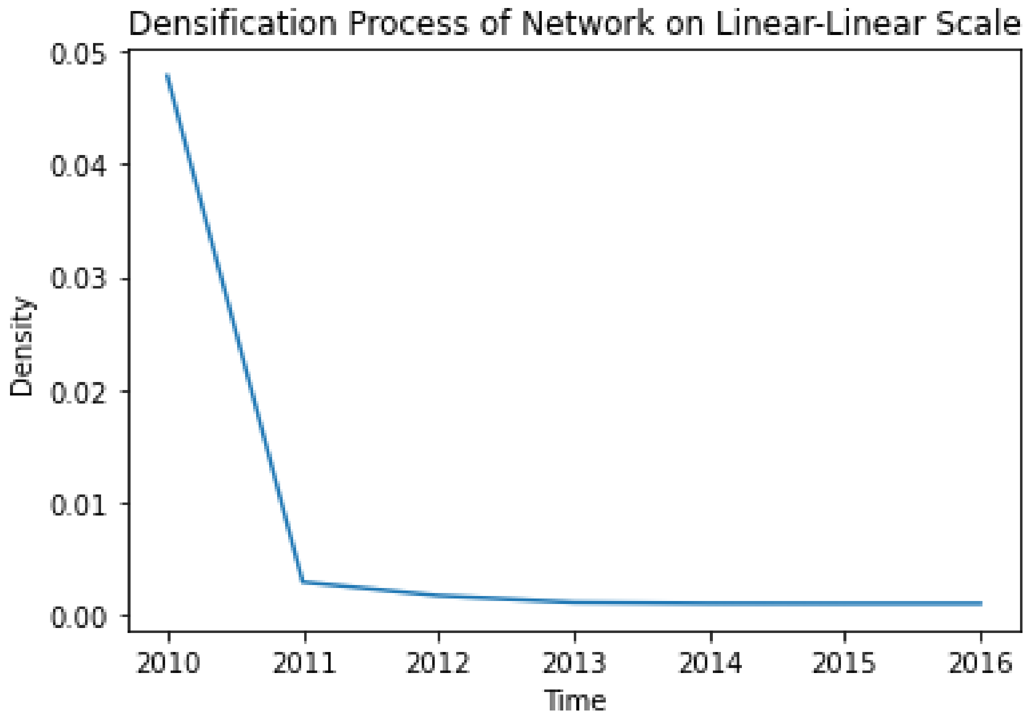


*Figure 12:Plot for densification process*

The graphs above illustrate the densification plot of number of nodes and number of edges in linear scale. Here, according to our plot, the slope is super linear, hence we can conclude that the average degree increases over time. Hence, the number of edges increases with number of nodes and satisfies the densification law.

Do the central nodes change over time?

The relevance of a node in a network is defined by its centrality. The other consideration that I have monitored is, how the central nodes vary over time. Dynamic networks' nodes vary over time, which may be investigated using a centrality metric. The centrality metric describes a node's relevance in the network. The comparable ideology of snapshot was addressed in our investigation. For each snapshot, the degree centrality was considered and the centrality measure of top 10 nodes for each snapshot is shown below.

Snapshot 1- *[6: 0.12962, 2: 0.1111, 5: 0.05555, 1: 0.3333, 15: 0.01851, 4: 0.16666, 3: 0.09259, 13: 0.129622, 16: 0.018517, 10: 0.14814].*

Snapshot 2- *[57: 0.008620, 4: 0.02093, 23: 0.0098, 69: 0.00307, 64: 0.04618, 68: 0.00307, 35: 0.08435, 79: 0.0006, 41: 0.03633, 1: 0.07450].*

Snapshot 3- [1318: 0.00363, 827: 0.00103, 1596: 0.00051, 1584: 0.012987, 1595: 0.00155, 1555: 0.0212, 1396: 0.05090, 1687: 0.00051, 1629: 0.01558, 1564: 0.0005].

Snapshot 4- *[135: 0.01380, 2877: 0.00149, 3239: 0.00335, 2725: 0.00895, 2188: 0.006713, 2119: 0.00037, 2028: 0.0387, 2780: 0.00298, 1810: 0.0656, 1967: 0.0055.]*

Snapshot 5- *[3988: 0.149475, 3719: 0.00524, 3910: 0.00524 625: 0.0034, 1885: 0.00174, 905: 0.0332, 4499: 0.0104, 5025: 0.00349, 4553: 0.01136, 5251: 0.012237].*

Snapshot 6- *[5578: 0.018918, 3722: 0.018918, 5677: 0.00270, 5840: 0.0054, 5847: 0.00270, 5846: 0.0027, 5876: 0.0027, 35: 0.1324, 5878: 0.0027, 5525: 0.02972].*

Snapshot 7- [3988: 0.14947, 3719: 0.00524 3910: 0.00524, 625: 0.00349, 1885: 0.001748, 905: 0.03321, 4499: 0.0104, 5025: 0.00349, 4553: 0.011363, 5251: 0.01223].

We achieve this with help of degree centrality measure. The degree centrality considers node with many connections and observe the change of node in each snapshot. According to [Zafarani, 14] the degree centrality of an undirected graph is represented as C(n)=d, where C(n) is the centrality for node 'n'. Whereas, in directed graph we can use the in-degree or out-degree or combination of both as centrality values. Mathematically, C(n)=d'in, C(n)=d'out or both ie., C(n)=d'in+d'out. The normalized degree centrality is used to compared the values of centrality in different networks. The normalization methods include normalizing by the maximum possible degree with number of nodes.

To determine if the central nodes change, I have considered the centrality measures of top 5 nodes of each snapshot and compare if these nodes vary. Top 5 central nodes of each snapshot are shown below.

Top 5 nodes of Snapshot 1- [21, 10, 4, 7, 1]

Top 5 nodes of Snapshot 2- [257, 832, 1, 35, 7]

Top 5 nodes of Snapshot 3- 2067, 905, 1810, 2028, 35]

Top 5 nodes of Snapshot 4- [3129, 4172, 2125, 35, 2642]

Top 5 nodes of Snapshot 5- [35, 1352, 3722, 2125, 3988]

Top 5 nodes of Snapshot 6- [1810, 4532, 2045, 1052, 35]

Top 5 nodes of Snapshot 7- [905, 2067, 1810, 2045, 2731]

From the above summary, I have observed that the central nodes changes over time for each snapshot considered. Hence, we can conclude that the central nodes change over time and satisfies the condition.

## Chapter 4

# Conclusion

In this study, I looked at the phenomena of temporal networks and analyzed the topological network structure of data from [Snap] The dataset was first investigated, and the metrics of the provided data were examined. The static characteristics of the network are analyzed by checking the scale-freeness and small-worldness phenomena with the help of python programming language and its libraries. The static networks are differentiated from temporal networks when an additional time dimension over the network is included. Therefore, to examine a static network if it is dynamic, I consider some metrics which are observed over time. A dynamic graph changes over time and static graph remains constant over time. In this thesis, the ‘time‘ dataframe was converted to human readable format and was divided into some timeframes. Each timeframe is represented as Snapshot, each snapshot contains data between 7 timeframes. The density over all these 7 snapshots was considered and examined over time i.e., densification process and have examined how the central nodes changes over time.

Each questions mentioned above are considered as research topics, which were tracked on the Bitcoin OTC Trust Network in the article. The research questions have been mentioned chapter 2. The analysis was performed on mentioned questions and therefore can be concluded that both the densification process and change of central nodes over time are obeyed.

# Future Aspects

Temporal network research has been a thriving sub-discipline of network science. Some of the initial issues have been addressed, while others have not. The emphasis has switched from research that simply applies static-network principles to temporal networks to research that creates new temporal-network-specific approaches throughout this time. The general research pathways, however, are comparable to those for static networks. One area of research that has a lot of room for advancement is temporal network visualization.

Another fundamental challenge for temporal networks is how to efficiently rescale or subsample a data source. Many techniques, especially those motivated by statistical physics, rely on consistent ways to modify the size of a network. This is an issue even for static networks: just generating subgraphs based on a random selection of nodes will almost surely modify the network's structure. Finally, I believe there is still a lot of work to be done in the field of temporal-network robustness and fragility, which has implications ranging from network security to public health and the effective planning of dependable public transit systems.  This is an area where it is possible to go beyond static-network analogies.

## References

[Snap] https://snap.stanford.edu/data/soc-sign-bitcoin-otc.html

[Holme, 15] Holme, P.: Modern temporal network theory: a colloquium. Eur. Phys. J. B 88, 234 (2015)
[Saramaki, 12]Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. 519, 97–125 (2012)
[Holme, 13] Holme, P.: Epidemiologically optimal static networks from temporal network data. PLoS Comput. Biol. 9, e1003142 (2013)
[Holme, 05] Holme, P.: Network reachability of real-world contact sequences. Phys. Rev. E 71, 046119 (2005)
[Holme, 03] Holme, P.: Network dynamics of ongoing social relationships. Europhys. Lett. 64, 427–433 (2003)

[Zafarani, 14] Reza Zafarani, Mohammad Ali Abbasi, Huan Liu: Social media mining: an introduciton.(2014)
[Dmitri, 19] Dmitri Goldenberg. 2019. Social Network Analysis: From Graph Theory to Applications with Python. In Proceedings of Israeli Python Conference 2019 (PyCon ’19). ACM, New York, NY, USA, 9 pages.

[Barabasi, 14] Albert-Laszlo Barabasi. Network Science The Scale Free Property(2014).

[Leskovec, 07] Leskovec, J., Kleinberg, J. & Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowl. Discov. from Data (TKDD)* 1, 2 (2007).

[Kobayashi, 20]Kobayashi, T. & Génois, M. Two types of densification scaling in the evolution of temporal networks. *Phys. Rev. E, in press. arXiv:2005.09445* (2020).

[Leskovec, 05] Leskovec, J., Kleinberg, J. & Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 177–187, ACM (2005).

[Bettencourt, 09] Bettencourt, L. M., Kaiser, D. I. & Kaur, J. Scientific discovery and topological transitions in collaboration networks. *J. Informetrics* 3, 210–221 (2009).

[Jeff, 14] Jeff Alstott, Ed Bullmore, Dietmar Plenz. (2014). powerlaw: a Python package for analysis of heavy-tailed distributions.

[Abello, 04] Abello, J. 2004. Hierarchical graph maps. Comput. Graph. 28, 3, 345--359.

[Telesford, 11] Telesford, Joyce, Hayasaka, Burdette, and Laurienti (2011). “The Ubiquity of Small-World Networks”. Brain Connectivity. 1 (0038): 367-75. PMC 3604768. PMID 22432451. doi:10.1089/brain.2011.0038.

[Scholtes, 16] Scholtes, I., Wider, N. & Garas, A. Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities. *Eur. Phys. J. B* 89, 1–15 (2016).

[Dekker, 05] Dekker, Anthony (2005). "Conceptual Distance in Social Network Analysis". *Journal of Social Structure*.

[Dangalchev, 20] Ch, Dangalchev (2020). "Additional Closeness and Networks Growth". *Fundamenta Informaticae*. **176** (1):

[Barrat, 13] Barrat, A., Cattuto, C.: Temporal networks of face-to-face human interactions. In: P. Holme, J. Saramäki (eds.) Temporal Networks, pp. 191–216. Springer, Berlin (2013)

[Rosvall, 10] Rosvall, M., Bergstrom, C.T.: Mapping change in large networks. PLoS One 5(1), e8694 (2010)

[Xing, 06] S. Hanneke, E.P. Xing, Discrete temporal models of social networks. Workshop on statistical network analysis, in: Proceedings of the 23rd International Conference on Machine Learning, ICML-SNA, 2006.

[Kolar, 10] M. Kolar, L. Song, A. Ahmed, E.P. Xing, Estimating time-varying networks, Ann. Appl. Stat. 4 (2010)

[Zhao, 11] K. Zhao, J. Stehlé, G. Bianconi, A. Barrat, Social network dynamics of face-to-face interactions, Phys. Rev. E 83 (2011) 056109.

[Stehle, 10] J. Stehlé, A. Barrat, G. Bianconi, Dynamical and bursty interactions in social networks, Phys. Rev. E 81 (2010) 035101.

[Pan, 11] H.-H. Jo, R.K. Pan, K. Kaski, Emergence of bursts and communities in evolving weighted networks, PLoS One 6 (2011).

[Saramaki, 07] J.M. Kumpula, J.P. Onnela, J. Saramäki, K. Kaski, J. Kertész, Emergence of communities in weighted networks, Phys. Rev. Lett. 99 (2007).
E. Volz, L.A. Meyers, Susceptible-infected-recovered epidemics in dynamic contact networks, Proc. Roy. Soc. B 274 (2007).
[Turova, 02] T.S. Turova, Dynamical random graphs with memory, Phys. Rev. E 65 (2002) 066102.
[Snijders, 10] T.A.B. Snijders, G.G. van de Bunt, C.E.G. Steglich, Introduction to stochastic actor-based models for network dynamics, Social Networks 32 (2010).

[Volz, 07] E. Volz, L.A. Meyers, Susceptible-infected-recovered epidemics in dynamic contact networks, Proc. Roy. Soc. B 274 (2007) 2925–2934.