Dissertation on

"Social Network Analysis"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE19CS390B – Capstone Project Phase - 2

Submitted by:

Dhananjaykumar M P	**PES1UG19CS140**
Tejashwini Hosamani	**PES1UG19CS539**
Vaibhav S	**PES1UG19CS555**
Vrunda Patil	**PES1UG19CS583**

Under the guidance of

Dr. S Natarajan
Professor
PES University

August - December 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

‘Social Network Analysis’

is a bonafide work carried out by

Dhananjaykumar M P	**PES1UG19CS140**
Tejashwini Hosamani	**PES1UG19CS539**
Vaibhav S	**PES1UG19CS555**
Vrunda Patil	**PES1UG19CS583**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August - December 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature Dr. S Natarajan Professor	Signature Dr. Shylaja S.S. Chairperson	Signature Dr. B.K. Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Social Network Analysis "** has been carried out by us under the guidance of Dr. S Natarajan and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August - December 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG19CS140	**Dhananjaykumar M P**	_____
PES1UG19CS539	**Tejashwini Hosamani**	_____
PES1UG19CS555	**Vaibhav S**	_____
PES1UG19CS583	**Vrunda Patil**	_____

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. S Natarajan, Professor, Department of Computer Science and Engineering, PES University, for his/her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B - Capstone Project Phase – 2.

We are grateful to the project coordinator, Prof. Mahesh H.B., for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Shylaja S.S. Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M.R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J., Vice-Chancellor, PES University for providing us various opportunities and enlightenment every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

Our personal social networks are big and cluttered, and currently there is no good way to organize them. Social networking sites allow users to manually categories their friends into social circles (e.g., 'circles' on Google+, and 'lists' on Facebook and Twitter), however they are laborious to construct and must be updated whenever a user's network grows. Given a single user with her personal social network, our goal is to identify her circles, each of which is a subset of her friends. Circles are user specific as each user organizes her personal network of friends independently of all other users to whom she is not connected. This means that we can formulate the problem of circle detection as a clustering problem on the ego-network, the network of friendships between her friends.

Social network-based applications like Facebook, Twitter, and Instagram have been used by people of all age groups and backgrounds for the last few years. It is a rich platform for sharing knowledge amongst users online. This information is shared as feelings, opinions, interests, events, or comments in large volumes and varied forms of data. Many multidisciplinary researchers have conducted studies to find out the commercial values of social media data. The reason behind this interest in research is an affluence to access data from the web, process it, and pull-out useful information from the web. Researchers have worked upon and explored the topics like information spreading, relationship analysis in groups for some or other applications. This review paper conducts a survey on community detection problem in social networks, its analysis, and a study of research done on related areas.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

One of the most common forms of communication, connection, and cooperation is through online social networks (OSN). A recent study found a link between OSN and real-life connections. Data on billions of people's interactions is stored in online social networks. Open access to such data poses major privacy dangers, especially now that the European General Data Protection Regulation has been implemented. We favour synthetic datasets because they avoid the security and privacy concerns that come with real-world data [5].

A social network is a group of people who are linked by certain relationships (family, friends, common interests, etc.). Social graph analysis can reveal information about interpersonal relationships. It is possible to retrieve data about users' connections and interactions with other users on a specific OSN by using various APIs. This allows analysts to assess people's connectivity based on their OSN interactions. Knowing how connected people can be useful in a variety of situations. Synthetic datasets are a viable and required alternative when empirical data is not publicly available or is insufficient [8].

Expanding social graphs is a far more appealing option than building binary social graphs. Because this study is based on Facebook, the interaction parameters will be modelled after this OSN. Even though it appears that those traits should be more highly associated, having more close friends on Facebook does not improve the total amount of messages sent. We look at the possibilities of creating a synthetic enlarged social graph in this research. We offer the results of a rigorous investigation of Facebook's most popular OSN. permits future researchers to deal with expanding social graphs without needing access to actual life-like data [2].

A Social network is defined as a network of relationships, where the nodes consist of people or actors. Online social networks like Facebook, Twitter, LinkedIn, Myspace etc have gained popularity in very short time. Ego-networks are social networks made up of an ego along with all the social ties he has

with other people. It has been shown that neighbourhoods around egos can exhibit different patterns. There are several ways to learn representations for nodes in the social graph using deep learning techniques. For instance, Deep Walk and node2vec both use a mapping of nodes to a low-dimensional space of features which preserves the flexible notion of nodes' neighbourhoods. An ego-network is an unsupervised framework that learns representations for pieces of data [3].

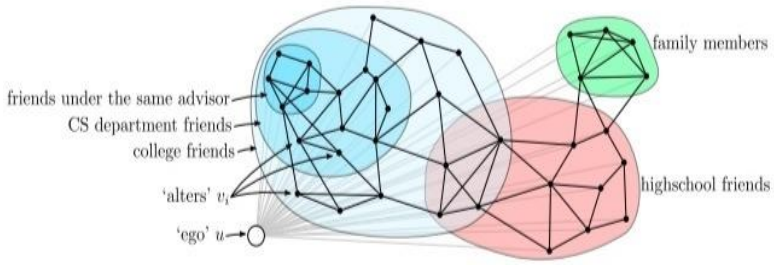

Fig 1.1: An ego-network with labelled circles.

CHAPTER 2

PROBLEM STATEMENT

Given an instance of set of nodes (users) in a social network graph, the aim is to find the influencing (important) users and to predict the likelihood of a future association (edge) between two nodes, knowing that there is no association between the nodes in the current state of the graph.

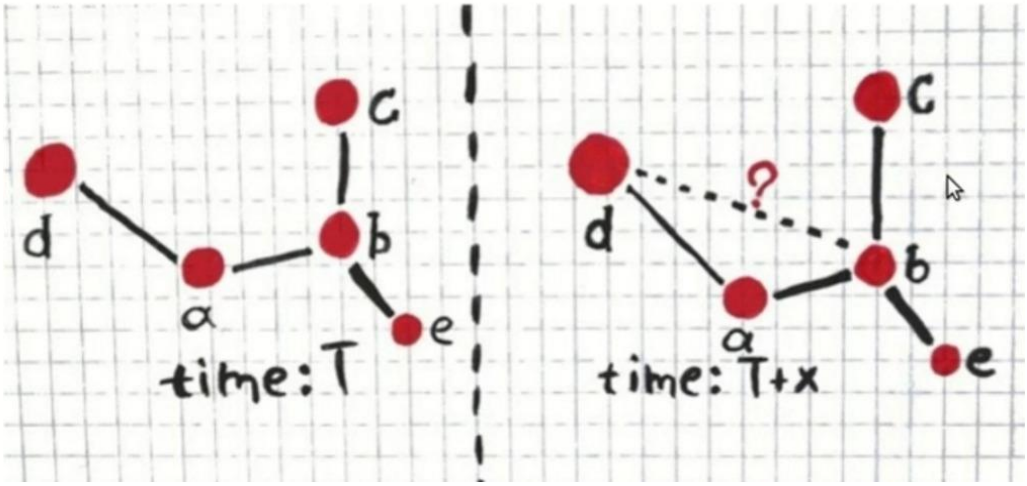

Fig 2.1 Edges at time T and T+x between the nodes.

The edges described in the problem statement could be of any form: friendship, collaboration, following or mutual interests. Here, we specifically study and build our model over Facebook's social network, with the following areas of motivation:

- General application of friend's recommendation to a particular user.
- Predicting hidden links in a social network group formed by terrorists along with identification of their leaders/ key influencers.
- Targeted marketing of products: Marketing through highly influential individuals and also identifying plausible customers.
- Suggesting promising interactions or collaborations that have not yet been identified within an organization. In Bioinformatics, link prediction can be used to find interactions between proteins.

CHAPTER 3

LITERATURE SURVEY

Literature survey provides understanding of existing literature, developments in the current field and helps in maintaining knowledge and relevancy in the field of interest. In this chapter, we present the current knowledge of the area and review substantial findings that help shape, inform and reform our study.

3.1 Exploring Local Structure and Centrality Measures in Egocentric Networks

M. Jia, et al., [1] Traversing Local Structure and Centrality Measures in Ego Networks
Typical centrality measures assess the importance of a node based on the distances to other nodes, shortest paths passing through it, or the eigen structure of the adjacency matrix. Local structure measures, on the other hand, capture network topological features by measuring how a motif are constructed from a substructure. Here, we are trying to know the suitability of several centrality measures and local structure measures in egocentric networks and investigate the relationships among them. Among all types of social networks, egocentric social networks are sometimes of particular interest when we care more about the immediate environment around each individual than the entire world.

Centrality measures are originally proposed to quantify the importance or the influence of nodes in a large network and are also capable of assessing the similarities or differences among ego nodes from their own networks. Some measures of the ego node, such as the closeness centrality and the eccentricity index, are uninformative when applied to egocentric networks.
Another set of regularly used approaches focuses on measuring the formation of local structures and they are of four types, but we use two of them for our capstone.

Clustering Coefficient

As one of the most popular metrics of network structure, the clustering coefficient assesses the extent to which the neighbours of a given node connect to each other. From a structural perspective, it measures the formation of triangles from open triads.

Closure Coefficient

The recently proposed closure coefficient provides a new perspective on network clustering by placing the focal node at the endpoint of the open triad. It is defined as the fraction of open triads, where the focal node sat at the endpoint, that form triangles. Compared to the classic clustering coefficient which is defined within the 1-hop neighbourhood, the closure coefficient covers 2-hops distance.

Degree Centrality

Through calculating the number of nodes directly connected to a node, degree centrality is a simple and straightforward way to assess the importance or influence of the node.

Closeness Centrality

The idea of closeness centrality is to measure the significance of a node through its distances to every other node in the network. The closer the node is to other nodes, the more important it is.

Betweenness Centrality

Another popular centrality measure based on shortest paths is the betweenness centrality, which measures the extent to which a node lies on the shortest paths between other nodes.

Eigenvector Centrality

Eigenvector centrality is one of the most popular spectral centrality measures, based on the idea that a node's importance depends on the importance of its neighbours.

After analysing a dataset of chronic pain patients (this was done to know the extent of pain and the

relation of one node(patient) with the pain category based on the local measure and centrality measure, though our idea is to analyse “social network”, this experiment gave us the idea of knowing how to implement all these measures and get the result so that we can have a conclusive analysis.

3.2 Exploring Information Diffusion in Ego-centric Networks

A. Kumar, et al., [2] Discussed Information Diffusion in Ego Networks

Social media analysis is a qualitative and quantitative analysis of social network, which uses graph theory and studies social structures for recommendations. Social network being a large network comprises of huge data of rumours, violence, coded data, transparent data. The most influential node will be having a group of users sharing similarities and therefore, similar information can be spread around amongst the users by the most influential person.

Steps Involved:

- Data Extraction
- Generating Network
- Centrality Measures
  i. Degree Centrality
  ii. Betweenness Centrality
  iii. Eigenvector Centrality
- Diffusion Models
  i. SIS Model
  ii. SIR Model
  iii. Independent cascading i.e., Independent Cascades (IC) model is a diffusion model
- Community detection

In this paper they have presented an overview of the Twitter ego network and studied many techniques and algorithms by analysing a real Twitter data set containing a root user (ego) and generating its network of its friends and to analyse its complete network we extracted friends of friends.

3.3 Classification approach to link prediction

Amin Rezaeipanah, et al., [3] Approaches to Classification of Link prediction
A social network can be termed in the form of nodes and edges, in which nodes play the role of users and the edges show social relationships between users. Discovered how can predict multiplex tie strength through the geographic location and social activities of users and apply it to the classic networks problem of link prediction

Variety of problem-solving methods involved:

- Link prediction in social networks
- Link prediction in multiplex social networks
- Sign prediction problem

Extracted features:

- Normalized reputation optimism (NRO) feature
- Common tweets (CT) feature
- Weighted CN (WCN) feature etc.

Dataset considerations:

- User's profiles on both networks have been manually checked for same users before the evaluation took place.
- In the multiple investigated networks, 45,000 users are considered from Twitter and Foursquare networks.

The paper concludes about the study which was conducted on two Twitter and Foursquare networks in

a directed and undirected form. It has focused on a particular representation of the link prediction problem regarding predicting of establishing links in future, this can be similarly used to study the missing links that require network structure at different times.

3.4 Benefits of social network analysis

F. Long, et al., [4] Discussing the assets of social network analysis
Author proposed, Semi-supervised Fusion framework for Multiple Network (SFMN), using gradient boosting decision tree algorithm (GBDT) to fuse the information of multi-source networks into a single network.

The main contributions are:
Firstly, semantic network construction method is used for specific datasets, which eliminates data noise by entity extraction and disambiguation. Multi-layer network structure is used to model coupled multi-source data and extract edge features. This edge features, proposed SFMN framework is used to actualize network fusion.

Networks fusion framework mainly includes:
1) entity relationship extraction
2) entity disambiguation and network alignment
3) network fusion.

Hence Authors concluded that the proposed SFMN fusion framework strengthens social networks analysis. This fusion analysis results through a community detection task which indicates the fused network can reflect the real social network structure of entities more accurately than a single network.

3.5 Examination of Facebook interaction

L. Humski, et al., [5] Diagnosis of interactions in Facebook network analysis

One of the most common forms of communication, connection, and cooperation is through online social networks (OSN). A recent study found a link between OSN and real-life connections. Data on billions of people's interactions is stored in online social networks. Open access to such data poses major privacy dangers, especially now that the European General Data Protection Regulation has been implemented. We favour synthetic datasets because they avoid the security and privacy concerns that come with real-world data.

A social network is a group of people who are linked by certain relationships (family, friends, common interests, etc.). Social graph analysis can reveal information about interpersonal relationships. It is possible to retrieve data about users' connections and interactions with other users on a specific OSN by using various APIs. This allows analysts to assess people's connectivity based on their OSN interactions. Knowing how connected people can be useful in a variety of situations. Synthetic datasets are a viable and required alternative when empirical data is not publicly available or is insufficient.

Data analysis of population distributions proved to be a difficult undertaking, but our goal was to develop theoretical distributions that fit the empirical distributions the best. The pre-tail component of most theoretical distributions is well approximated, while the tail is substantially looser. A combination of a power law distribution for the tail part and one of the previously stated theoretical distributions with their best fit parameters for each characteristic yields the best approximation theoretical distribution. We wanted to see if all Facebook users use the same method or if some people prefer to explore other posts, hit likes, and leave comments.

Expanding social graphs is a far more appealing option than building binary social graphs. Because this study is based on Facebook, the interaction parameters will be modelled after this OSN. Even though it appears that those traits should be more highly associated, having more close friends on Facebook does not improve the total amount of messages sent. We look at the possibilities of creating

a synthetic enlarged social graph in this research. We offer the results of a rigorous investigation of Facebook's most popular OSN. Future academics will be able to generate enlarged social graphs without needing access to genuine life-like data.

3.6 Effects of ego networks and communities on self-disclosure

Y. D. Kwon, et al., [6] Issues related to ego networks and communities on self-disclosure
In this paper, we conduct a quantitative study at different granularities of networks (ego networks and user communities) to understand users' self-disclosing behaviours better. We describe users into three types (open, closed, and moderate) based on the Communication Privacy Management theory.

Online Social Networks (OSNs) play an essential role as social platforms for millions of users who seek benefits. Understanding what factors affect and how much those factors contribute to users' self-disclosure is very important for both users and business intelligence agents. The present study uses a large-scale dataset consisting of 462 million edges among 30 million users.

We are inspired by the Communication Privacy Management theory (CPM) to characterize our Google+ users into three types which are open, closed, and moderate. Open users disclose all their personal information while closed users do not. Moderate users tend to have more dense ego networks. In this paper, we investigate users' self-disclosure in the contexts of communities with the intuition that a community can influence its users' behaviours. We show that users located in acritical position (i.e., bridge position) within a community tend to disclose more personal information.

Many works have focused on studying the network properties of communities in OSNs to understand user behaviour. No work has related such network properties to users' self-disclosing behaviours. We examine the relationship between self-disclosure and two levels of networks: ego networks and user communities.

We use a data-driven approach on a large-scale dataset with more than 70% of the entire user base to study users' privacy concerns. We are inspired by a well-known online privacy theory called CPM to categorize Google+ users into three groups and track their privacy-related features.
In this paper, we studied the self-disclosure of users based on their networks and communities. Users are more likely to disclose personal information when they can utilize positional advantages by playing bridging roles from their networks. There are several directions worthwhile investigating as future work.

3.7 Global and local feature learning for Ego-Network

Michael Granitzer, et al., [7] Gaining knowledge for Ego Networks
In an ego network, an individual(ego) organizes its friends (alters) in different groups (social circles). This social network can be efficiently analyzed after learning representations of the ego and it alters in a low-dimensional, real vector space**.**

With the exponential expansion of social networks, collecting node attributes and finding distinct neighborhood patterns for the entire network has become increasingly impractical. As a result, splitting up the network into smaller sub-networks is one effective way to characterize certain elements of large networks, and this can be done by considering particular node or subgraph level locality statistics provided on local portions of a network, known as neighborhood.

Ego-networks are social networks made up of an ego as well as all of his social links with other people (called alters), which are grouped into social circles. Ego-networks are a major area of research in anthropology because they may be used to characterize various fundamental aspects of social connections, such as prototypes of interactions between alters, archetypal neighborhood trends around egos, and so on. Finding vector representations that express the local neighborhood structure of egos is therefore particularly useful for ego-network analysis, and they can be used to detect and forecast social circles.

Deep Walk uses deep learning techniques to learn global representations for nodes in the social graph. There are several ways to have vector representations for nodes based on global attributes. By optimizing a neighbourhood preserving probability objective, the Skip-gram model is used to learn feature representations for nodes. Node2vec learns a node-to-low-dimensional-space-of-features mapping that keeps the flexible idea of node neighbourhoods. The unsupervised framework Paragraph Vector learns continuous distributed vector representations for text fragments.

Global and Local Feature Learning, we apply the techniques which have been used to model sentences and paragraphs of natural languages to model community structure in networks. Therefore, we capture information on the global and local network topology.
Learning global representation for each node we use Deep Walk vector representation method to structure the model. Learning local representation for each ego we use Paragraph Vector representation method to structure the model

Circle Prediction

It's a way of getting to know the people in your social circles. Users of online social networks must organize their personal social networks to cope with the information overload generated by their friends. Instead of doing it manually, which is laborious, error-prone, and unadaptable to changes, users must automatically organize their friends into social circles when they are added to the network.

So, fundamentally, we know how to provide a technique for ego-network analysis based on the concept of local network neighbourhoods and learning latent social representations for egos using new advances in language modelling.

3.8 Steps for mining social networks to find Ego networks

A. Madani, et al., [8] Discussing procedure for mining social networks to fine Ego networks
To better identify social network, a method that mainly utilizes features extracted from network topology and node profile is taken. MacAuley and Leskovec firstly studied the problem and then formulated it as social circles identifying problem. The most dominant method in this field is "Maximization Likelihood Like". And later he published "Enhanced Link Clustering".

This includes two main steps:

1. Feature extraction - It is done at two levels: Network topology and Node profile. Network topological feature calculation includes, Weighted Degree, Centrality measures, PageRank, Component ID, Clustering Co-efficient, Number of Triangles, Modularity Class.
2. Classification – It makes use of appropriate decision tree.

Evaluation measures analysed using a contingency table, using which we can calculate Accuracy, Precision, Recall and F1-Score. As a result, the proposed method has a higher accuracy rate as compared to the other two methods. Throughout our experiment we realized that the node profile feature does not have a significant contribution to the overall accuracy of detected circles. So, the main intention is to apply this method in marketing applications.

3.9 Some examples of Facebook social network exploration

Nadeem, et al., [9] Describing examples of Facebook social network exploration
This is mainly related to comparative analysis of four social network analysis tools- Networkx, Gephi, Pajek, IGraph based on platform, execution time, Graph types, algorithms complexity, input file format and graph features. Analyzing a social network is like the analysis of a graph because social networks form the topology of a graph.

Analysis tasks of social networks includes following:

1)Discovering the structure of social network

2)Finding various attribute values for the network- Ex. radius, diameter, centrality, betweenness, shortest paths, density etc.

3)Finding communities in the social network.

4)Visualizing the whole or part of the social network.

SOCIAL NETWORK SOFTWARE TOOLS COMPARISONS:

A. Comparison Based on Platform

B. Comparison Based on Network Types

C. Comparison Based on Graph Layout

D. Comparison Based on Algorithm Time Complexity, Input File Formats and Graph Features

Hence, IGraph is concluded as the better software based on execution time which is minimum for IGraph than Networkx. IGraph also provides most of graph features and handle large & complex network. Importing required libraries are more useful for tasks involving millions of nodes and for operations such as the union, difference between set of nodes and the clustering.

3.10 Example of ego network structure exploration

Valerio Arnaboldi, et al., [10] Describing examples of ego network structure exploration

The goal of this research is to find out if Dunbar's circles exist in OSN ego networks. We study a Facebook data set encompassing over 23 million social interactions for this purpose. The frequency of contact is extracted from the data set's ego networks. Then, on the distributions of the frequency of contact of the various ego networks, we use various clustering approaches (specifically, partitioning clustering and density-based clustering). As a result, we examine the data for the presence of a probable circular structure.

The features of OSN ego networks are found to be very comparable to those of offline ego networks. In real contexts, the average number of circles in the structure of virtual ego networks is equal to 4, and the average scaling factor between the concentric circles of the social structure is close to 3. Furthermore, the circle sizes, i.e., the number of social relationships of each category, are strikingly similar to those found in offline social networks. The average size of OSN ego networks is strikingly similar to Dunbar's number, which represents the average size of ego networks in offline social networks.

3.11 Knowing how to find social circles

Julian McAuley, et al., [11] Discussing how social circles are formed
Link prediction among composite and mutual links is a difficult problem that can be solved using the Support Vector Machine model. In the link prediction problem, using SVM improves accuracy. The positive or negative nature of a social network link is determined by the attitudes or beliefs of both entities making the link. The link prediction problem is concerned with the sign of linkages between composite entities in a network. The authors address this issue, which is solved via machine learning. The temporality of a feature is determined by the behaviour of nodes (people) in a network structure. For making interactions among the nodes, each node on the network has assigned a unique time value.

Scalability issues for link prediction have been presented by social networking sites like Facebook and Twitter. Acer looked at large-scale link prediction, using higher-order tensor models based on matrix factorization to predict interactions in vast social networks. The composite and mutual link prediction problems were considered by the authors. The behaviour of nodes on the network allows collaborative filtering to be used to generate learning features. The Logistic Regression, High Entropy Model, and Random Model were proposed by Rowe and Stankovic.

Algorithms for machine learning, such as, the support vector machine (SVM) is a supervised learning tool for classification and regression. Its learning algorithm creates a model that predicts whether a positive connection belongs in one of two categories.

CHAPTER 4

PROJECT REQUIREMENTS SPECIFICATION

4.1 Operating Environment

We will make use of Jupyter notebook with all necessary libraries required for our analysis.

4.2 Design Constraints, Assumptions and Dependencies

4.2.1 Design Constraints:

1. **Maintainability:** The software and the algorithms are very well defined, and the papers published give a true idea of the working but the main challenge for us is to mold the data set into readable form and remove the unnecessary information.
2. **Heterogeneity**: The data on the web (also in businesses and governments) is heterogeneous, unstructured, and often incomplete.
3. **Scalability**: Social Networks face serious scalability challenges due to their rapid growth and popularity. Scaling up is in general a non-trivial endeavor
4. **Missing Data**: The Analysis of Social Networks data is often suffered with non-responses and missing data. The most common way to deal with missing values is to replace them by some reasonable estimates using known or model- based cross-dependencies over the network in analysis.

4.2.2 Assumptions:

1. Assuming unsupervised dataset with the edges which are undirected.
2. Nodes and their edges are viewed as independent units.
3. Relational ties between nodes are the channels for the transfer of resources.
4. We propose a method that mainly utilizes features extracted from network topology and node profile to better identify social networks.
5. We conduct a quantitative study at different granularities of networks to understand users'

self-disclosing behaviours better.

4.2.3 Risks:

1. We are not able to predict the likelihood of the future association between the nodes.
2. 2. When there are multiple entries of nodes there will be multiple suggestion of them same entity.
3. A software risk analysis looks at code violations that present a threat to the stability, security, Performance of the code.
4. A malfunction within the electronic circuits or electromechanical components (disks, tapes) of a Computer system.

4.2.4 Dependencies:

1. Understanding what factors affect and how much those factors contribute to users 'self-disclosure is very important (e.g., for both users and business intelligence agents).
2. Depends on association between the nodes.
3. Depends on similarity features between the nodes.

4.3 Functional Requirements

4.3.1 Validity tests on inputs:

When we pass the data set, we need to be able to predict the proper association between two nodes.

4.3.2 Sequence of Operation:

1. **Measures for Centrality:** As our part of analysis, we used the following 4 centrality measures:

   a) **Degree of nodes**: Core idea: To find the nodes that have highest number of immediate neighbours (degree).

   b) **Closeness Centrality**: Core idea: A central node is one that is close, on average, to other nodes.

   c) **Betweenness Centrality**: Core Idea: A central actor is one that acts as a bridge, broker, or gatekeeper.

d) **Eigenvector centrality**: Core Idea: A central actor is connected to other central actors.

2. **Link prediction:** We will be using the Support Vector Machine Classification Algorithm, as it classifies nodes as connected or not connected using all the metrics.

4.3.3 Error Handling and Recover

Link disclosure between two individuals in a social network could be a privacy breach. To limit link disclosure, previous works modelled a social network as an undirected graph and randomized a link over the entire domain of links, which leads to considerable structural distortion to the graph.

4.3.4 Consequences of Parameters

1. The Vertical scale is too big or too small, or skips numbers, or doesn't start at zero. The graph isn't labelled properly. Data is left out.
2. No more than twelve attributes can be displayed on a graph.

4.3.5 Relationship of outputs to inputs

Graph with association between the nodes.

4.4 Non – Functional Requirements

1. **Performance requirements**: Achieve good response time with flexibility and by using the most advanced python engine.
2. **Safety requirements**: Using secured data with the license policy and the copyright approved.
3. **Security requirements**: Access to the analysed information only to higher prioritized people and the data analysts. Working on the project with a private server and most of the operations done with encryption.
4. **Manageability:** The measure of and set of features that support the ease, speed, and competence with which a system can be discovered, configured, modified, deployed, controlled, and supervised.
5. **Utility:** Network utilities are basic software tools designed for analysing and configuring various aspects of computer networks.

CHAPTER 5

SYSTEM DESIGN

Our first step was to know the problem statement and discover all the necessary components associated with our problem statement. Next step would be data gathering and pre-processing, here we collect the data and ensure everything is great when we are using the data to get some outcomes during the analysis phase. Later we are going to have a ML model that computes centrality measures and some other similarity measures which helps us to get some useful insights over the data we have provided. And after getting the results from the model we can extract a few useful insights and make precise predictions that allow us to know the actual link and association between the nodes.

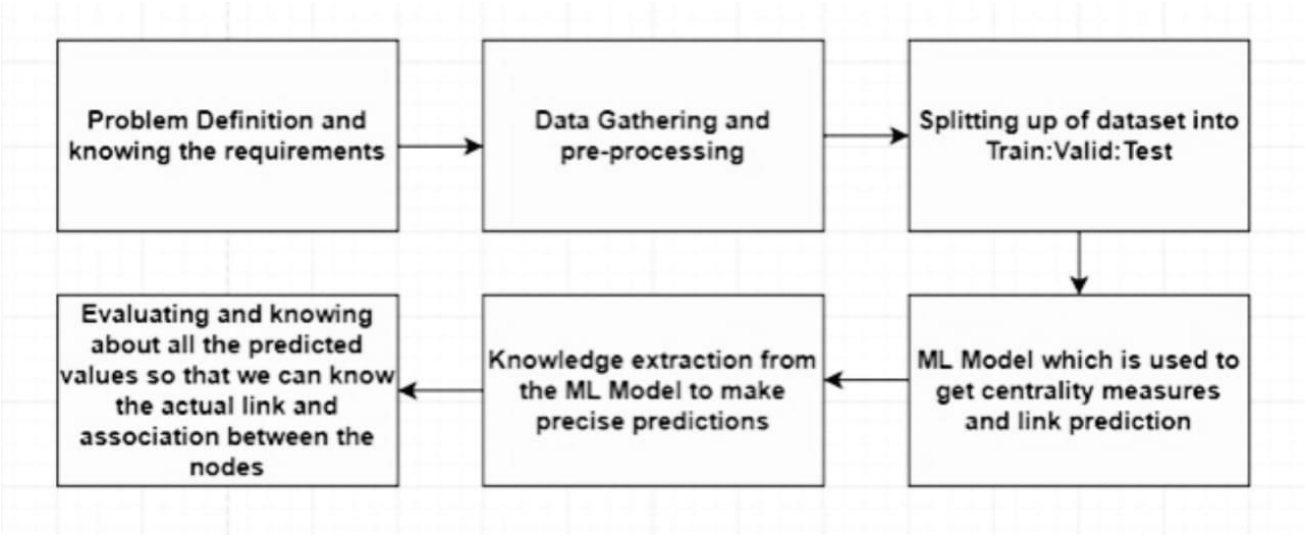

Fig 5.1 System design

CHAPTER 6

PROPOSED METHODOLOGY

Input of the model

- Nodes and edges of unsupervised data.

Methodologies/ Approaches

- Measures of Centrality
- Link prediction

  The idea is to assign the connection weight score to pairs of nodes, based on the input graph.
- The approaches adopted so far can be classified into:

1. Methods based on node neighbourhoods: Several approaches are based on the idea that two nodes x and y are more likely to form a link in the future if their sets of neighbours have large overlap. Example:

   •Common neighbours

   •Jaccard's coefficient

   •Preferential attachment
2. Methods based on the ensemble of all paths: Several methods refine the notion of shortest-path distance by implicitly considering the ensemble of all paths between two nodes.
3. Since we had multiple features associated with each node in an ego network, we performed our experiment based on the similarity of features between the two nodes.
4. Machine Learning models like Support Vector Machine could classify the set of nodes into two:

- Connection
- No connection Output to display □Graph with association between the nodes.

Output to display

- Graph with association between the nodes.
- Accuracy of different ML Models.

CHAPTER 7

IMPLEMENTATION AND PSEUDOCODE

1-Dataset search:(load dataset and Label the vertex, edges)

Firstly, we load the dataset which will displays all nodes and edges. A function is defined which inserts node if it is not present. Also, it will print the length of the vertices.

Dataset search

```
def load_dataset(fileName,g):
    fileNums=[0]
    for i,eachNum in enumerate(fileNums):
        print(eachNum)
        fileName="Dataset/facebook/edges/"+str(eachNum)+".edges"
        print('fileName=',fileName)
        f=open(fileName)
        line=f.readline()
        while(line!=''):
            c=(line.split())
            g=addVertex(g,c[0])
            g=addVertex(g,c[1])
            print('Adding ',c[0],'-->',c[1])
            g.add_edge(c[0],c[1])
            line=f.readline()
    g.simplify()
    return
```

2-Centrality Measures:

Function is defined to measure Eigenvector, Closeness and Betweenness centrality with maximum value in given range. List of all nodes is printed which has edge between them. Graph at time T and T+x is plotted using scatter plot which is best for 3D visualisation.

Centrality measures

```
def calculate_eigen(g):
    eigen=g.evcent(directed=False)
    for i in range(1,7):
        maxVal=max(eigen)
        print(i,'==node',g.vs[eigen.index(maxVal)]['name'],' with score of ',maxVal)
        eigen.remove(maxVal)
    eigen=g.evcent(directed=False)
    return
```

```
def calculate_closeness(g):
    close=g.closeness(g.vs)
    for i in range(1,6):
        maxVal=max(close)
        print(i,'==node',g.vs[close.index(maxVal)]['name'],' with score of ',maxVal)
        close.remove(maxVal)
    close=g.closeness(g.vs)
    return close
```

```
def calculate_between(g):
    between=g.betweenness(g.vs)
    for i in range(1,6):
        maxVal=max(between)
        print(i,'==node',g.vs[between.index(maxVal)]['name'],' with score of ',maxVal)
        between.remove(maxVal)
    between=g.betweenness(g.vs)
    return between
```

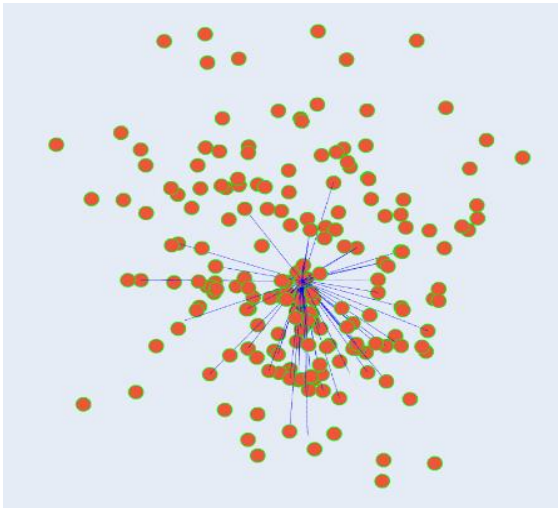

Fig 7.1 Graph at time T

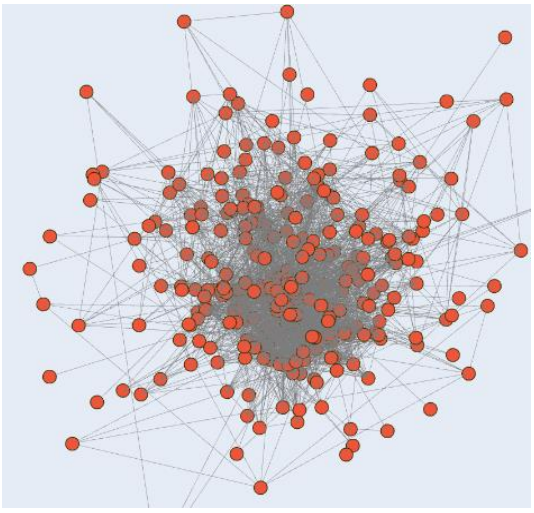

Fig 7.2 Graph at time T+x

3-Generate Train:Validation:Test Dataset:

After loading dataset, dataset is divided into positive and negative Train:Validation:Test in the ratio 2:1:1 for link prediction.

Generation of positive (train:validation:test) dataset

```
import random

def generate_datasets(g,num,train_filename,valid_filename,test_filename):
    load_dataset('appu',g)
    f=open(train_filename,'a+');
    global train_num
    train_num=int(len(g.es)*0.5)
    print('train length=',train_num)
    global test_num
    test_num=int(len(g.es)*0.25)
    global valid_num
    valid_num=int(len(g.es)*0.15)
    print('valid num=',valid_num)
    for i in range(train_num):
        edgeSet=g.es;
        r=random.randint(0,len(edgeSet)-1);
        t=edgeSet[r].tuple
        g.delete_edges(t);
        print('len of es=',len(edgeSet))
        write_tuple_to_file(f,retrieve_edge_name_tuple(g,t))
    f.close()
    f=open(test_filename,'a+');
    for i in range(test_num):
        edgeSet=g.es;
        r=random.randint(0,len(edgeSet)-1);
        print('r=',r)
        t=edgeSet[r].tuple
        g.delete_edges(t);
        print('len of es=',len(edgeSet))
        write_tuple_to_file(f,retrieve_edge_name_tuple(g,t))
    f.close()
    f=open(valid_filename,'a+');
    for i in range(valid_num):
        edgeSet=g.es;
        if(len(g.es)==0):
            break
        else:
            print('len of es=',len(edgeSet))
            r=random.randint(0,len(edgeSet)-1);
            print('r=',r)
            t=edgeSet[r].tuple
            g.delete_edges(t);
            write_tuple_to_file(f,retrieve_edge_name_tuple(g,t))
            if(len(g.es)==0):
                f.close()
                break
```

Generation of negative (train:validation:test) dataset

```
import random
def generate_negative_examples(pool,trainfilename,trainnum,validfilename,validnum,testfilename,testnum):
    f=open(trainfilename,'a+')
    for i in range(0,trainnum):
        r=random.randint(0,len(pool)-1);
        t=pool[r];
        pool.remove(t);
        f.write(str(t[0])+' '+str(t[1])+'\n');
    f.close()
    f=open(validfilename,'a+')
    for i in range(0,validnum):
        r=random.randint(0,len(pool)-1);
        t=pool[r];
        pool.remove(t);
        f.write(str(t[0])+' '+str(t[1])+'\n');
    f.close()
    f=open(testfilename,'a+')
    for i in range(0,testnum):
        r=random.randint(0,len(pool)-1);
        t=pool[r];
        pool.remove(t);
        f.write(str(t[0])+' '+str(t[1])+'\n');
    f.close()
```

4-Link Prediction:

Link prediction is the main part of our project, using different ML Models like SVM, KNN, Random Forest and Neural Network we have calculated accuracy, precision, and F1-score to show which is the best model for predicting link between nodes after time T+x. At last we came up with the conclusion that SVM is better compared to others in finding association between the nodes after time T+x with 90% accuracy.

SVM Model Algorithm

```
from seaborn import load_dataset
import pandas as pd
from sklearn import svm

aaa=svm.SVC(kernel='rbf', C=100).fit(X=train_X[:-1], y=train_Y[:-1])
from sklearn import model_selection
from sklearn.metrics import *

accuracy_score(testy, aaa.predict(testx))
```

KNN Model Algorithm

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 35)

md=knn.fit(X=train_X[:-1], y=train_Y[:-1])

accuracy_score(testy, md.predict(testx))
```

Random Forest Model Algorithm

```
#Import Random Forest Model
from sklearn.ensemble import RandomForestClassifier

#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=10)

#Train the model using the training sets y_pred=clf.predict(X_test)
rf=clf.fit(X=train_X[:-1], y=train_Y[:-1])

accuracy_score(testy, rf.predict(testx))
```

Neural Network Model Algorithm

```
#import neural network model
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix


training_a=train_X[:-1]
training_b=train_Y[:-1]
myscaler = StandardScaler()
myscaler.fit(training_a)
training_a = myscaler.transform(training_a)
testing_a = myscaler.transform(valid_X)
m1 = MLPClassifier(hidden_layer_sizes=(100, 120, 50, 50), activation='relu', solver='adam', max_iter=2000)
m1.fit(training_a, training_b.ravel())
# predicted_values = m1.predict(testing_a)
# print(confusion_matrix(valid_Y,predicted_values))
# print(classification_report(valid_Y,predicted_values))
accuracy_score(testy, m1.predict(testx))
```

Xgboost Model Algorithm

```
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
model.fit(X=train_X[:-1], y=train_Y[:-1])
y_pred = model.predict(testx)

accuracy_score(testy, model.predict(testx))
```

Ensemble Model Algorithm

```
#model1
ens1=model.predict_proba(testx)

#model2

ens2=aaa.predict_proba(testx)
```

```
all=(ens1+ens2)/2
fullout=np.argmax(all,axis=1)
ensemble=accuracy_score(fullout, testy)
print(ensemble)
```

CHAPTER 8

RESULTS AND DISCUSSION

- In the first phase of the project literature survey gave us better ideas related on how to solve our problem statement by keeping it in an organised manner with Requirement specification and High-level design.
- In this phase initially we implemented our first module by starting with the code i.e., calculating centrality measures using which we have plotted a graph at time T and T+x.
- Then we divided our dataset into train:valid:test for link prediction using different ML Models like SVM, KNN, Random Forest, Neural network, xgboost and Ensemble Models.
- After comparing results of all ML Models, we are concluding that Ensemble Model is better than other ML Models with more than 90% accuracy.

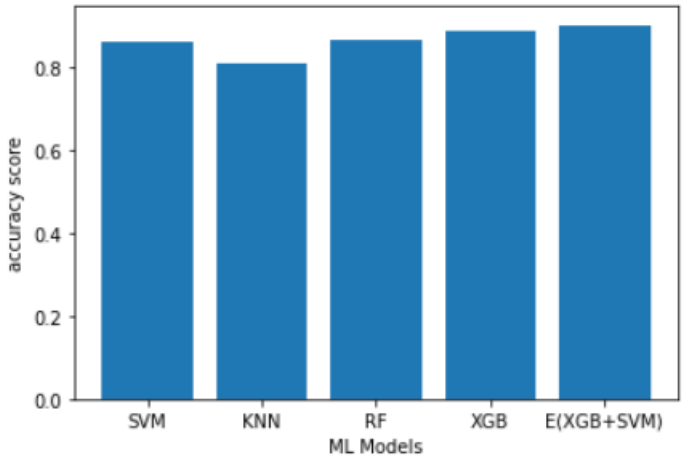

Fig 8.1 Accuracy score graph

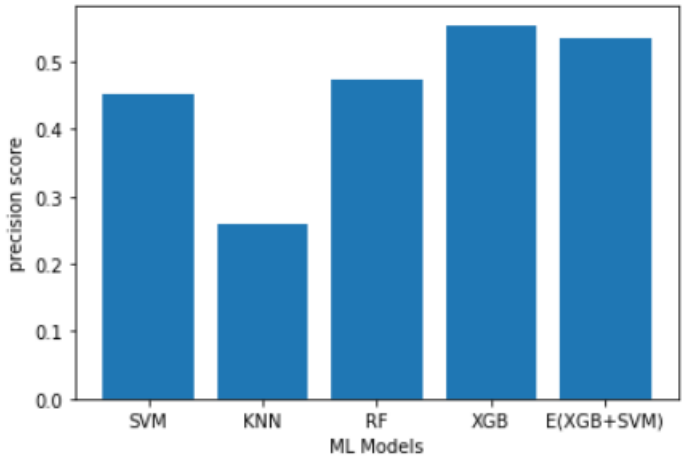

Fig 8.2 Precision score graph

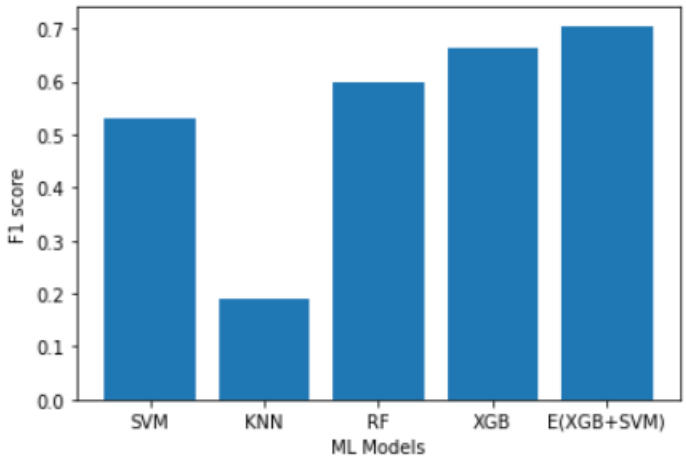

Fig 8.3 F1 score graph

CHAPTER 9

CONCLUSION AND FUTURE WORK

- The Project taught us new concepts which included centrality measures, different ML models and the ways to visualise the ego network using graphs.
- The implementation part mainly involved comparison of different ML models which results us in concluding that ensemble models (Combinations) give a better outcome.
- We represented our results with the help of graph showing the association which is easier to visualise.
- We organised our project into different modules Problem Definition, Dataset collection, Centrality measure, Link prediction using different ML Models, Link prediction using different ML Models.
- Future work is to implement this idea in most social media platform to incur a better possibility of link prediction.
- This paper publication is also one of the major aspects of our future work.

REFERENCES/BIBLIOGRAPHY

[1] M. Jia, M. V. Alboom, L. Goubert, P. Bracke, B. Gabrys and K. Musial, “Analysing Egocentric Networks via Local Structure and Centrality Measures: A Study on Chronic Pain Patients”, ICOIN 2022.

[2] A. Kumar, D. Chhabra, B. Mendiratta and A. Sinha, Analyzing Information Diffusion in Ego-centric Twitter Social Network, ICSC 2020.

[3] Amin Rezaeipanah, Gholamreza Ahmadi, Samaneh Sechin Matoori, A classification approach to link prediction in multiplex online ego networks Social Network Analysis and Mining, (Springer-Verlag GMBH Austria, part of Springer Nature) 2020.

[4] F. Long, N. Ning, C. Song and B. Wu, Strengthening Social Networks Analysis, ASONAM 2019.

[5] L. Humski, D. Pintar and M. Vranic, Analysis of Facebook Interaction as Basis for Synthetic Expanded Social Graph Generation, WCSE 2019.

[6] Y. D. Kwon, R. H. Mogavi, E. Ul Haq, Y. Kwon, X. Ma and P. Hui, Effects of Ego Networks and Communities on Self-Disclosure in an Online Social Network, ASONAM 2019.

[7] Michael Granitzer, Kontantiz Ziegier, Global and Local Feature Learning for Ego-Network Analysis Fatemeh Salehi Rizi, Germany, DEXA 2017.

[8] A. Madani and M. Marjan, Mining social networks to discover ego sub-networks, MEC ICBDSC 2016.

[9] Nadeem Akhtar, Analysis of Facebook Social Network, CICN 2013.

[10] Valerio Arnaboldi, Marco Conti, Analysis of Ego Networks Structure in Online Social Network, International Conference on Privacy, Security, Risk and Trust and 2012, International Conference on Social Computing 2013.

[11] Julian McAuley, Jury Leskovec, Learning to Discover Social Circles in Ego Networks, NIPS 2012.

[12] SNAP: Stanford Network Analysis Project (https://snap.stanford.edu/data/ego-Facebook.html).

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

1. Degree of nodes: The degree of a node is the number of connections that it has to other nodes in the network.

2. Closeness Centrality: Closeness centrality indicates how close a node is to all other nodes in the network.

3. Betweenness Centrality: Betweenness centrality is a way of detecting the amount of influence a node has over the flow of information in a graph.

4. Eigenvector centrality: Eigenvector Centrality is an algorithm that measures the transitive influence of nodes.

5. Social network: Social network is a social structure composed of people connected by specific relations

6. Social Graph: Social graph is composed of nodes and edges (or ties), where nodes represent people and edges represent relations between them.

7. Binary Graph: A binary social graph is a special case of social graph where only the existence of an edge is known, without type or weight assigned to it.

8. Expanded Social Graph: Expanded social graph is a term we will use for a graph which has multiple weighted edges between two nodes, each describing different types of relations or interactions between observed nodes.

9. Ego Networks: An ego network is defined as a portion of a social network formed of a given individual, termed ego, and the other persons with whom she has a social relationship, termed alters.

10. Social circle: A social circle is a group of socially interconnected people. A social circle may be viewed from the perspective of an individual who is the locus of a particular group of socially interconnected people and from the perspective of the group as a cohesive unit.

11. Alters: Ego networks consist of a focal node ("ego") and the nodes to whom ego is directly connected to these are called alters.

12. Ties: Ties are the relationships between the actors.