UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Unleashing the Power of Communication: Exploring the Dynamics of Slack Channel Networks and Project Success

---

Author:
Eyuel MUSE WOLDESEMBET

Supervisor:
Dr. Albert DIAZ GUILERA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2023

Contents

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Unleashing the Power of Communication: Exploring the Dynamics of Slack Channel Networks and Project Success

by Eyuel MUSE WOLDESEMBET

Communication is at the center of any human organisation. In this thesis I analyse the communication patterns of employees from a company called Superside by extracting data from Slack, their communication platform, and analysing it as a network. I find that the distribution of users among channels is optimal when looked at from the perspective of robustness. Additionally I analyse the participation of users in conversations and find that the network they form is a scale-free network, which makes the company weaker to targeted attacks (loosing central employees). In that same line I perform an experiment in order to find out the network related metric to look at when performing layoffs, which turns out to be Betweeness Centrality. Finally I dig into performance and I find indication that projects assigned to overworked employees tend to be delivered late, moreover, I show that the Project Manager's centrality in the projects plays a significant role on the project being delivered late or on time.

Acknowledgements

This project would not have been possible without the person who actually brought me into the organization (Superside) Albert Franzi Cos, who also configured the pipeline that brought Slack data to the database. And the person inside Superside who made sure I had acces to all the data I needed, Jing Kjeldsen. And most importantly, it would not have happened without the knowledge on Complex Network imparted by Albert Diaz Guilera. I want also to take the opportunity to thank the coordinator of this master's Jordi Vitra, these months have been the greatest treasure of my formative years.

To all of you, Thank you.

Chapter 1

Introduction

Organization. It is defined as "a group of people with a particular purpose, such as a business or a governmental body". And the fact that we can create organizations is arguably what sets us apart from most animals. Borrowing from the book Harari, 2014, let me post a question to you. If you were to do an experiment in which you leave a human being and a chimpanzee in a desert island and come back 6 months later to monitor their states, who do you think might have done better in that environment? What about in the case in which you bring 100 humans and 100 chimpanzees? What about a 1000?

Well, as you might have guessed already, as the sample size increases, humans have the greater chances of survival. And this is simply because of our ability to communicate. Because we can communicate effectively with one another, we can cooperate without having to know one another very intimately, we can create believes and frameworks to trust one another to a scale no other species can. In this thesis I hope to bring you in a journey were we will try to uncover how communication and cooperation happens in an organisation through analyzing data from the different Slack channels of the organization.

The code used to perform all the analysis and experiments can be found in this GitHub repository

1.1 The Organization: Superside

Superside, as defined in Superside, n.d., is a design at scale company that offers a wide range of design services to businesses of all sizes. They specialize in providing on-demand graphic design, UI/UX design, illustration, branding, and presentation design services. The company was founded in 2015 and is headquartered in Palo Alto, California.

Superside's primary focus is on helping businesses with their design needs in a scalable and efficient manner. They employ a global network of talented designers who work remotely, allowing them to offer 24/7 service and quick turnaround times for design projects. Their platform connects clients with designers, streamlining the design process and ensuring consistent quality across projects.

One notable aspect of Superside's service is their subscription-based model. Which provides businesses with a dedicated design team that can handle ongoing design tasks. This approach is particularly beneficial for companies that require a high volume of design work but may not have the resources or need to hire an in-house design team.

Superside differentiates itself from a traditional creative agencies by the fact that all the communication happening with a given client takes place in a proprietary platfrom. Through the platform the client has instant access to all the information they need about their project, updates and the Superside team that is handling it.

1.1.1 Organization Structure

Superside is mainly organised into three teams that we can easily identify functionally. Tech, Data and Product (TPD), Operations, and Creative Ops. The TPD team builds and maintains the different features of the platform, the one the final customer interacts with. This team is mainly composed by Software Engineers, Data and Product Managers.

The Creative Ops team is the one in charge of actually delivering the projects given by final customers, this team is composed by Project Managers (they are the point of contact of the customer), Creatives and Creative Managers.

The last team is Operations, they are in charge on making sure everything runs smoothly. They assign creatives to sub-teams, and make sure the creative sub-teams have the resources they need, they also work on the incentive structure for the creatives and set evaluation metrics for the Creative Ops. team. The Data we will be analysing through this thesis will be that of the Creative Ops. team.

1.2 Slack

Slack is a cloud-based communication tool that allows teams to collaborate and communicate effectively. It was launched in 2013 and has become a popular tool for businesses of all sizes. Slack is designed to replace traditional email as the primary form of communication within a organization.

One of the key features of Slack is its ability to create channels for different topics, projects, or teams. This allows team members to easily find and join relevant conversations, and helps to keep communication organized.

Slack's user-friendly interface and extensive customization options make it a popular choice for remote teams and companies with employees in multiple locations.

The main advantage of Slack is that it contains the whole organization. And as explained by Stray and Moe, 2020, Slack makes it easy for employees from any departments to come together, in a way, makes the workplace more of a small world where the distance in terms of nodes between any two pairs of employees from any department, is small. We will see later on what this means in more detail, for now keep the buzz word (small world).

Through out the thesis we will be talking about three main components of Slack:

- Users: These are individuals inside the network, each Superside employee is a user in slack.
- Channels: These are the compartmentalised hubs of communication. They are created by users, once a given user creates a channel, she can invite others to join. Channels can be public or private and can contain one user or more.
- Messages/Threads: Inside a channel any user included in it can send a message or start a thread. A thread is a message with 0 or more replies. A message is a thread with 0 replies.

For instance in Figure 1.1, Tom, Alice, Bob and Linda are users, members of the Marketing channel.

Tom has sent a message (Thread 2) that has not been replied by anyone. Tom, Linda and Bob participate in Thread 1 and Alice and Bob participate in Thread 3.

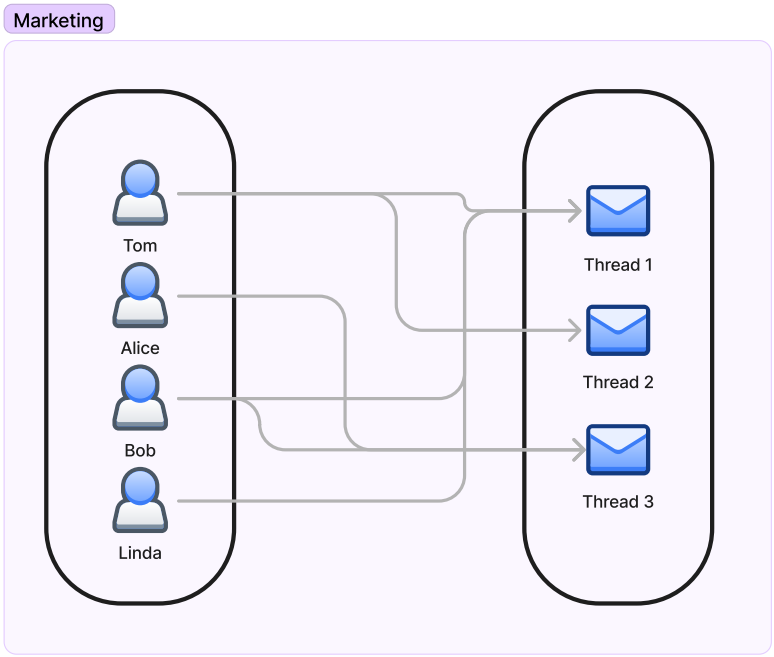

FIGURE 1.1: Example of a Slack Channel, Users and Threads

1.3 The Data

The source of the data we will be analysing is mainly Slack. The tool provides us with API endpoints in order to extract any type of data stored in the messaging app, we will be using only four of these endpoints. Users, Conversations, Conversation Members and Conversation History.

Additionally, for the last experiment, we add some project related data. More specifically, we link each project channel to the actual project to add an extra project attribute. Note that all data is anatomized. This means that we will be working mainly with IDs. No emails, No names.

1.3.1 Users

As explained in Slack, n.d.(d), from this endpoint, you can retrieve information about a user, such as their name, email address, and profile picture.

We will use the following fields:

- **User ID**: Alphanumeric identification of each Slack member.
- **Team ID**: Alphanumeric identification of the teams the user belongs to.
- **Name**: Name of the user
- **Deleted At**: Date when the user was deleted from the platform, null if the user is still active.
- **Is Bot**: A Boolean field indicating whether the user is a person or a bot?

1.3.2 Conversations

The Conversations endpoint, as defined in Slack, n.d.(a) is for working with public channels, private channels, and direct messages. One can retrieve information about a conversation, such as its name and topic, and one can create and archive conversations. One can also retrieve a list of all the conversations in a workspace (Superside's Slack account).

We will use the following fields:

- **Channel ID**: Alphanumeric identification of each Slack channel.
- **Is Archived**: A Boolean field indicating whether the channel has been deleted.
- **Is Private**: A Boolean field indicating whether the channel is private.

1.3.3 Conversation Members

The Conversation endpoint, as defined in Slack, n.d.(c) in is for working with the members of a conversation. One can retrieve a list of all the members of a conversation, add or remove members from a conversation, and retrieve information about a specific member of a conversation.

We will use the following fields:

- **Channel ID**: Alphanumeric identification of each Slack channel.
- **User ID**: Alphanumeric identification of each Slack member that is part of the conversation.

1.3.4 Conversation History

The Conversation endpoint, as defined in Slack, n.d.(b) is for retrieving the messages in a conversation. One can retrieve a list of messages in a conversation, filter messages by date range or user, and retrieve information about a specific message.

We will use the following fields:

- **Channel ID**: Alphanumeric identification of each Slack channel.
- **Message ID**: Alphanumeric identification of each message sent to the Slack channel.
- **User ID**: Alphanumeric identification of each Slack member that sent the message.
- **Reply User**: List of users who replied to the message in a thread.

Chapter 2

Complex Networks

2.1 What are Complex Networks?

In order to define what complex networks are, I would like to start by forming a general understanding of Complex Systems. Paraphrasing Albert-Laszlo Barabasi in Barabási, 2016, a system in which knowing the individual components of the system tells us little about the general behaviour of the system can be understood as a Complex System.

Imagine an alien landing on earth with an interest in understanding humanity, what would the fact of meeting you and understanding what and who you are tell the alien about humanity, our societal norms, our economical organizations...? Probably very little. None of those things are the mere result of just gathering a bunch of human beings together in a room. They are what in his book Creation Grand, 2000 calls Emergent Phenomena. They are the results of our interactions over time. These interactions have the ability to compound, to become something unpredictable, impossible to pin down to a word or a sentence, and they are ever changing, ever evolving as long as our interactions don't freeze. And even when the system is small enough, tiny changes in the initial settings of the elements or interactions can lead to wildly different results from two systems that are otherwise identical.

Well, Complex Networks are the way we represent the interaction patterns of Complex Systems as graphs. Each element in the system is what we call a Node, and each interaction is called an Edge.

There are several types of networks, but through out there thesis, we will be using only these types of networks: Undirected Weighted Networks, Bipartite Networks and Hypergraphs.

2.1.1 Undirected Weighted Networks

We have already seen that a network is composed by Nodes and Edges. As explained by Manríquez et al., 2021, an Undirected Weighted Network is a type of Network in which the edges are not directed, in some cases, like for example in a network of economical transactions where each transaction is an edge, we might be interested to encode the information of who gives something to whom. In this case for every interaction, we would have the giving node and the receiving node, this would be a Directed Network. In the case of Undirected Network, there is no giving node and receiving node, we are only interested in the fact that there has been an interaction between the two nodes, for instance, that both nodes have been part of the same message thread.

A network is weighted when we somehow weight each interaction. For instance, if node 1 has interacted with both node 2 and 3, but has done so much more with

node 3, we might want to encode that information. The way to do that in complex networks is by giving a weight to the edges.

The mathematical object that allows as to encode this information is a simple matrix called adjacency matrix. An adjacency matrix is a square matrix used to represent a finite network, the elements of the matrix indicate whether pairs of vertices are adjacent or not in the network.

2.1.2 Bipartate Network

A bipartite network is a type of network in which the nodes can be divided into two groups such that there are no connections between nodes within the same group, instead, all connections exist between nodes in different groups. This type of network is often used to model relationships between two different types of entities, such as users and products in e-commerce or authors and papers in bibliometrics.

In the case of Slack Channels for instance, we have channels that contain a set of users (type 1 node) and users (type 2 node). In this type of network, users don't connect directly to other users, they connect to the channels they belong to. Hence, we can understand that there are two layers to the network, the channels and the users. We can consider that the channels are the intereaction layer and the users the actual node layer.

This setup, however, as explained by Battiston et al., 2020 poses a challenge. In the case of pairwise interactions, which is what we find in the previous type of network, the interactions between nodes can be represented in an adjacency matrix. In the case of Bipartite networks, since there is no direct interaction between nodes of the same type, we have to consider the Unipartite networks obtained by projecting the bipartite on one of the two layers. Each interaction becomes a fully connected subgraph among the nodes belonging to the interaction (a Clique). In the case of channels and users, all users belonging to the same channel would become a Clique.

2.1.3 Hypergraphs

Sometimes, analysing pairwise interactions is not enough. We want to see what happens when nodes interact in groups of 3, 4 ... as well as 2. Hypergraphs full fill that need by encoding this information in the form of higher order interactions.

In the two previous types of networks, the mathematical object that allowed us to encode this information is a simple matrix called adjacency matrix. In the case of Hypergraphs, as explined by Battiston et al., 2020, we would have multiple matrices, one for each order of interaction (group size). And the mathematical object that helps us do that is a tensor. So, we can understand Hypergraphs as a generalization of Complex Networks.

Chapter 3

Distribution Of Users Among Channels

3.1 Problem Representation

In order to represent problem of the distribution of users among channels we will be using a Bipartite Network (Figure 3.1) because it is a very intuitive representation of the problem.

On the one hand we have channels (type 1 node) and on the other hand we have users (type 2 nodes). There is no direct interaction between one channel and another, and there is no direct interaction between users (note that we want to represent the distribution of users among channels, and not the interaction between them via messages).

3.1.1 Which Slack Channels and users do we consider?

As described in section 1.1.1 we have three types of teams in the organization. TPD, Creative OPs and Operations. Throughout the whole thesis, we will be only taking into account the users in the Creative Ops team and their interactions. The reason for this is that they represent the biggest team in the company, and they provide a very clear area of study, their goal is very well defined and we have actual metrics on how to measure the output of their collaboration.

As for the other two teams, they are actually support teams that try to make the life of the creatives easier, the nature of their goals is very volatile and there is no clear way of measuring the outputs of their collaboration.

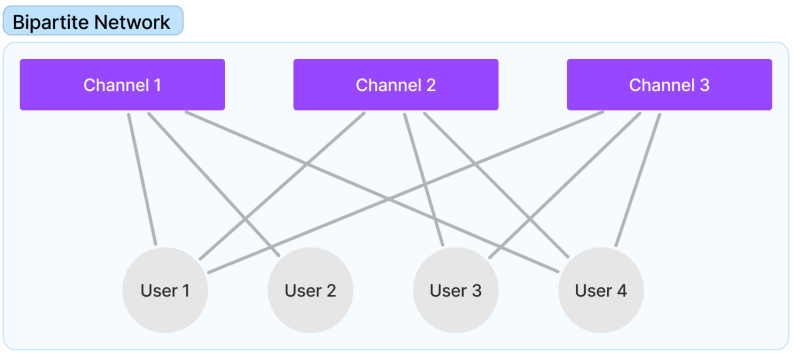

FIGURE 3.1: Example of a Bipartite Network

The channels we will consider in this problem are all private channels with more that 1 user in them from the Creative Ops team.

3.1.2 How do we measure distribution?

In complex networks "distribution" usually refers to *degree distribution*. The degree of a given node is simply the number of edges that are connected to it. Then the degree distribution of a network is the probability that a randomly chosen node has degree k.

The degree distribution of a network helps us understand the nature of the network, in the hopes that we can reproduce the complexity of the system it represents by building models that best resemble the real network.

We can differentiate between two types of networks based on their degree distribution, Random Networks and Scale-Free Networks.

Random Network

From the modelling perspective a network is a very simple object, it only consists in nodes (N) and links (L) or edges. However, the tricky part is knowing which pair of nodes to connect with an edge. In a Random Network we just place edges *randomly* between pairs of nodes.

More formally, a Random Network, as given by Barabási, 2016, is defined as a graph $G(N, p)$ where N is the number of nodes and p is the probability that any two pairs of nodes are connected.

If we look at the degree distribution of a Random Network, it will be a Binomial distribution. This means that the probability p_L that a Random Network has exactly L links is given by:

$$p_L = \binom{\frac{N(N-1)}{2}}{L} p^L (1-p)^{\frac{N(N-1)}{2}-L} \tag{3.1}$$

As 3.1 is a binomial distribution, the number of expected links is given by:

$$< L >= \sum_{L=0}^{\frac{N(N-1)}{2}} L_{p_L} = p\frac{N(N-1)}{2} \tag{3.2}$$

Scale-Free Networks

A Scale Free Network is a graph which degree distribution follows a power law (3.3). This means that if we represent it in a log-log scale, it follows a straight line. In other words, there are few nodes around which most of the interactions revolve. This is also known in economics as to 80/20 rule or the Pareto rule.

$$p_k \approx k^{-\gamma} \tag{3.3}$$

3.1.3 Our Network

As discussed above, our network is a Bipartite Network with two types of nodes, Channels and Users (see in Figure 3.1). In total it has 905 nodes, of which 382 are users and 523 are channels. And it has 5063 edges.

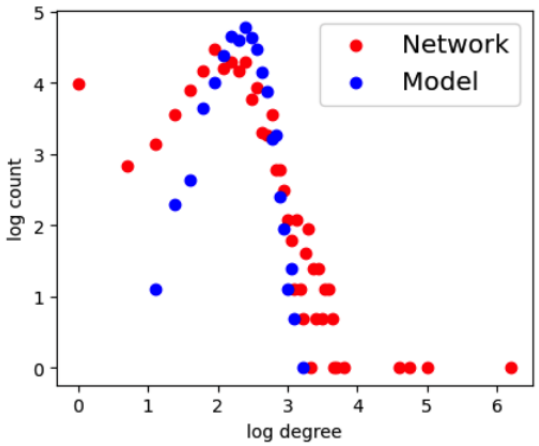

FIGURE 3.2: Comparison of Degree distribution. Random Model VS Our Network

Our Network is Random

Since we have the number of nodes and edges of our original network, using 3.2 we can obtain p_L (the probability that any two given pairs of nodes are connected) and create a model Random Network:

After simple algebraic transfomations to 3.2, we obtain that:

$$p_L = \frac{L}{\frac{N(N-1)}{2}}$$

$$p_L = \frac{5063}{\frac{905*904}{2}} = 0.0123$$

If we compare the log degree distribution of our network with the Random Network (Figure 3.2) with this precise value for p_L we can clearly see that except for some outliers our model network's distribution fits to that of our original network.

Properties of a Random Network

The biggest implication of our network being random is the fact that there are no set of users or channels that monopolise the network. There are users and channels with more edges than others, but the scale of the difference among them is contained. This means that there no big hubs around which most of the interactions in the network happens.

Is this a good thing or a bad thing?

Well, the answer to this question revolves around the concept of Robustness. Which is the ability of the network to resist failures of nodes and edges. In theory, the fact that there are no big hubs in the network means that it is more vulnerable to random failures and more robust to targeted attacks as compared to scale free networks which do have big hubs.

This is due to the fact that the failure of what could be considered a big node carries more damage than that of a small node, but the difference is not huge, hence,

failures of big nodes have no major effects, and are closer to that of failures of random nodes. In order to illustrate this, I have run two experiments on attacks to the network.

3.2 Attacks to the network

Losing Slack channels can be a serious disruption for an organization. As explained by Stray and Moe, 2020 Slack is often used as a central hub for communication, file sharing, and project management. If channels are lost, it can be difficult to recover important conversations or documents. Additionally, team members may be left without a clear understanding of what tasks are due, what has been completed, and what still needs to be done. This can lead to wasted time, missed deadlines, and decreased productivity.

In other words Slack does a great job on unifying team members by creating a network in which most members of the organization if not all, are connected. But more than one *connected component* can exist in a network. A connected component of a network is a subgraph in which every pair of nodes is connected by a path, and which is connected to no additional nodes outside of the subgraph. Another way to think about it is that a connected component is a group of nodes that are all reachable from one another by following edges in the network.

3.2.1 The largest connected component

Let $G = (N, L)$ be a graph, where N is the set of nodes and L is the set of edges. Then the largest connected component of G is a subset C of N such that:

1. For every pair of nodes n1, n2 in C, there exists a path from n1 to n2 in G.
2. C is maximal, meaning that there is no larger subset of N that is also connected.

So, building on the above, it would be fair to say the goal of having Slack in an organization is to have the biggest connected component possible so that most users are part of that network and ca reach any other user in it. But what would happen to the largest connected component if a bad actor intrudes our network and starts removing channels?

3.2.2 Sequential random removal of channels

In the following experiment, we will be sequentially removing slack channels from the network and analysing what happens to the largest component. The metric we will be using to measure the effect is the degree to which 3.4 decreases each time we remove an additional node.

$$p_{LC} = \frac{\text{Nodes in largest component}}{\text{Total Nodes in the network}} \tag{3.4}$$

If we look at Figure 3.3 we can see the effect of the node removals from 0 to 50. The damage to the largest component is generally not big, and the greatest damage comes when the intruder removes the 28_{th} channel.

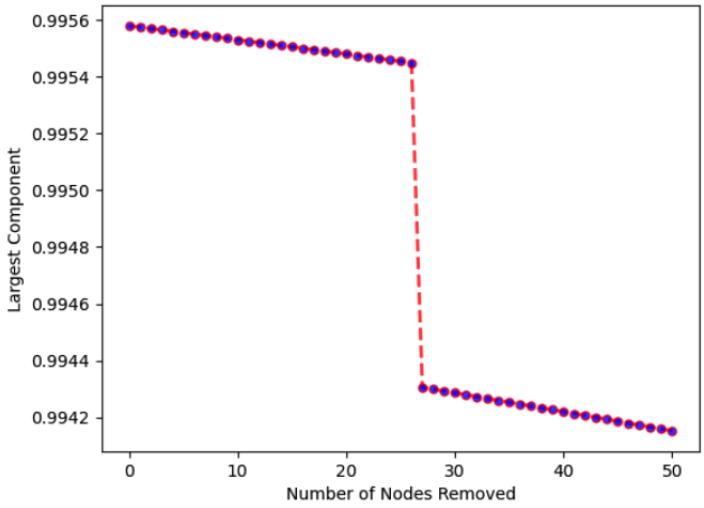

FIGURE 3.3: Sequential random removal of channels

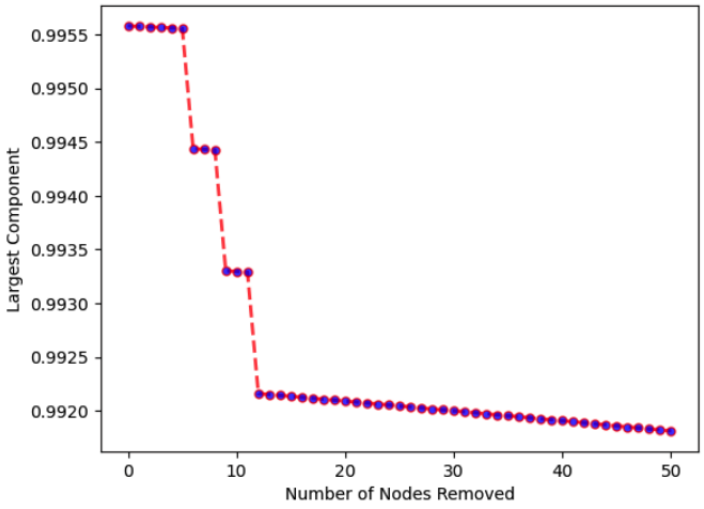

FIGURE 3.4: Sequential targeted removal of channels

3.2.3 Sequential targeted removal of nodes

Now, let's assume that our intruder decides to target the most popular slack channels first. Using the same setting as in the previous experiment and the same metric 3.4.

In the case of targeted attacks (Figure 3.4), the effects are noted much earlier, but after the 12_{th} node removal the effect of the additional removals is minimal.

Note that the effects both targeted and random attack is similar overall. This is due to what we discussed in section 3.1.3, because of the lack of hubs random networks are more robust to targeted attacks and more vulnerable to random attacks compared to scale free networks. We will have a look at scale free networks in the next chapter.

Chapter 4

Communication In the Network

4.1 Problem Representation

In the previous section we analysed the distribution of users among channels. In this section we will dive into their interactions.

We will be looking into the message threads that happen in the same set of channels we used in the previous section. So, for every message we will have a list of users that have participated in the conversation, and since we are interested in who interacted with who and not so much about who sent a message to who, we will be representing this data as an un-directed network (See Figure 4.1).

So, for every pair of nodes or users, there will be an edge if they have been part of the same conversation, and we will weight the strength of the bond by the amount of times they have been part of the same conversation.

4.1.1 Our network is a scale-free network

This network has 423 nodes and 2.289 links. And when compared to a Barabàsi Albert (BA) model, it looks like a scale free network (see Figure 4.2).

A BA model is a way to build scale free networks. It revolves around the idea of preferential attachment, which means that when a new node joins the network, there is higher probability that it will connect to other nodes in the network that have a high degree. The probability (p) that a new node will connect with node i is given by 4.1. Where k_i is the number of nodes connected to the node i and the denominator is total number of nodes in the network.

$$p_i = \frac{k_i}{\sum_{j=1}^{j} k_j} \tag{4.1}$$

Properties of a Scale Free Network

- Hubs: The main difference between random networks and scale free networks is that in scale free networks we find large hubs around central node where most of the interaction happens.
- Small Worlds: The side effect of having large hubs is that the distance or the steps necessary to connect any two pair of nodes is rather small.

Is this a good thing or a bad thing?

Well, it makes the communication of the organization highly dependent on few individuals. In fact, in the following experiments we will see how randomly loosing users affects our network compared to a targeted attack and compare the results to

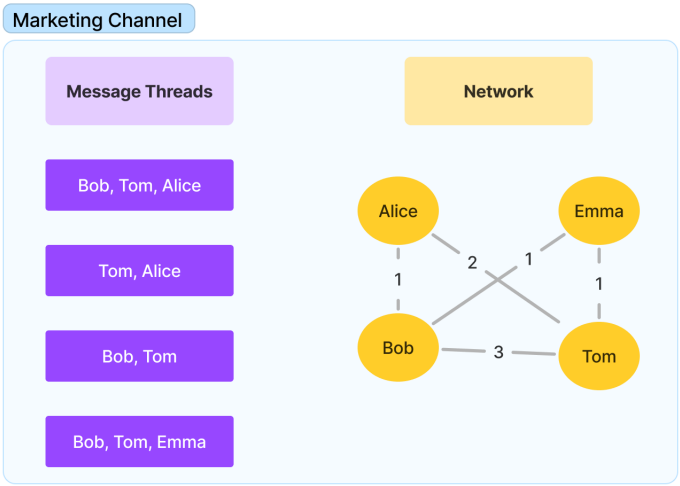

FIGURE 4.1: Example of an Un-directed Weighted Network

those of the network in chapter 3. We will conclude this chapter with the biggest experiment so far, "What to take into account when executing layoffs?".

4.2 Losing Users (Employees)

In any human company, turnover of employees is common theme and it could be due to several reasons. It could be due to downsizing, burnout, the employee not being happy in the company, the company not being happy with the employee etc.

For the following experiments, lets assume that there is a competitor in the market that is expanding aggressively and luring out employees with high wages. We will first look at what happens to communication inside our organization if the competitor takes employees randomly from us, next we will see what happens if the competitor has inside information about who the key employees are for flow of information inside our company and does a targeted attack.

4.2.1 Sequential random removal of users

For this experiment, we sequentially and randomly remove 50 users from the network and measure the impact of doing so at each step using 3.4.

The largest component goes from 99.5% to 97.5%. And the effects of the removals start being noticeable from the very beginning (see Figure 4.3).

4.2.2 Sequential targeted removal of users

Now let's look at what would happen if the competitor knew which employees to take from us to disrupt our communication.

This experiment is illustrated in Figure 4.4, by the removal of the 50th user, the largest component goes from 99.5% to 91.0%.

If we compare the ranges, we can see how the targeted removal has a much greater effect than the random removal, and compared to the network in chapter 3, the difference between the two ranges is much larger, this is due to the fact that the

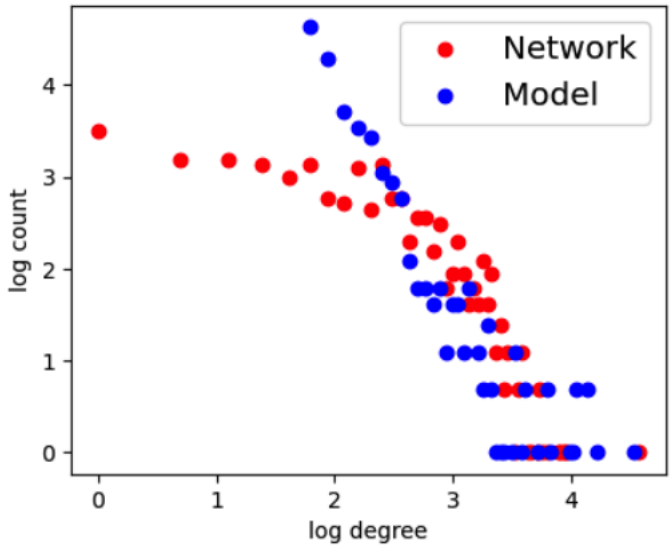

FIGURE 4.2: Log Degree distribution of our network compared to a model

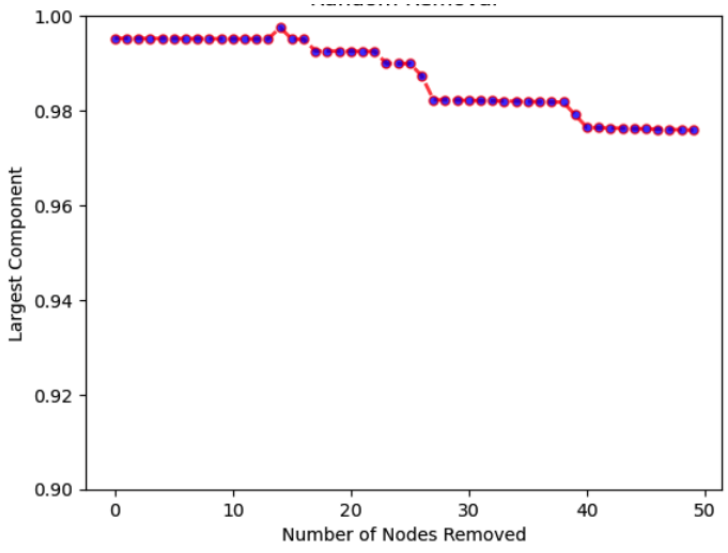

FIGURE 4.3: Sequential random removal of users

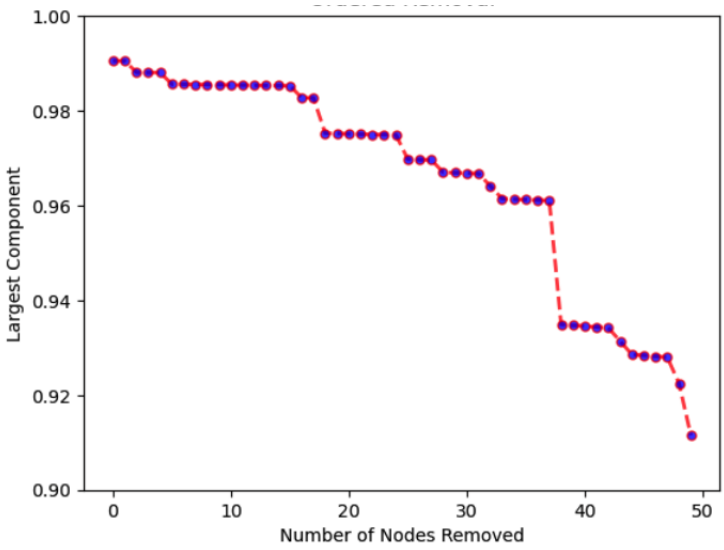

FIGURE 4.4: Sequential targeted removal of users

second network, as we have already stated, is a scale free network. The network definitely weaker to targeted attacks.

4.3 Layoffs. Letting go 10% of the workforce

As explained by Datta et al., 2010 layoffs can have significant impacts on companies, both financially and culturally. From a financial perspective, layoffs can be a way for companies to reduce costs during tough economic times or to restructure their business to focus on more profitable areas. However, layoffs can also have negative effects on employee morale and productivity, both for those who are laid off and for those who remain with the company.

For employees who are laid off, the experience can be traumatic and stressful. Losing a job can lead to feelings of anxiety, depression, and loss of self-esteem. It can also lead to financial strain and uncertainty about the future. Additionally, the process of being laid off can be a blow to an individual's sense of identity and purpose, particularly if they have been with the company for a long time.

For those who remain with the company, layoffs can also be stressful and demotivating. They may feel survivor guilt, wondering why they were not selected for layoff and whether they will be next. Additionally, they may feel increased pressure to perform and take on additional responsibilities, as the company tries to do more with fewer resources. This can lead to burnout and low morale, which can ultimately impact the company's bottom line.

There are several reasons why a company should not layoff personnel unless it is the only way for the company to survive. Moreover, judging from the experiment we saw in sections 4.2.1 and 4.2.2 it can have a severe effect on the effective communication among employees.

But given that the company is at a point where the only way to move forward is to let go some part of the workforce, besides the several other metrics to take into account such as performance and areas of the business the company decides to pull investment from, what metrics should be taken into account in order to maintain the

flow of communication as healthy as possibly? In the following experiment, we will try to find out exactly that.

4.3.1 The Experiment

The goal of the experiment will be to identify which are the variables (related to the network) the company should look at when doing layoffs in order to preserve the communication flow as intact as possible.

In order to define this experiment, the very first thing was to decide on the metric to measure the impact of layoffs in the network. The metric I decided for measuring the impact of the removed nodes in the network is, again 3.4. Remember the goal of Slack is to unify the employees inside the company, to have one big connected component, damaging that large connected component means damaging the communication inside our organisation.

The experiment consists on eliminating 10% of the nodes, measuring several of the network related metrics of those nodes such as Eigenvalue Centrally and taking their average, and finally, after eliminating the nodes from the network we measure the 3.4 of the network. That becomes one observation.

We perform the experiment 1000 times eliminating 10% of the nodes randomly from the network (with substitution) and recording the average node metrics and 3.4 after the elimination of the nodes.

Node related metrics

In each iteration we measure several metrics of the nodes we eliminate and we average them. Those metrics and their definitions as explained by Bloch, Jackson, and Tebaldi, 2023 are:

- Degree Centrality: The degree centrality of a node n in a network is the number of edges connected to that node.

$$C_D(n) = \frac{\text{number of edges connected to } v}{\text{total number of possible edges for } n} \tag{4.2}$$

- Eigenvalue Centrality: The eigenvalue centrality of a node in a network is a measure of its influence in the network. It can be calculated using the eigenvectors of the adjacency matrix of the network.

- Betweenness Centrality: The betweenness centrality of a node in a network is a measure of the number of shortest paths that pass through that node.

$$C_B(n) = \sum_{s \neq n \neq t} \frac{\sigma_{st}(n)}{\sigma_{st}} \tag{4.3}$$

  where σ_{st} is the total number of shortest paths between nodes s and t, and $\sigma_{st}(n)$ is the number of those paths that pass through node n.

- Clustering: The clustering coefficient of a node in a network is a measure of the degree to which neighbours of then node tend to cluster together (or how close they are to being a clique).

$$C(n) = \frac{2E(n)}{k_n(k_n - 1)} \tag{4.4}$$

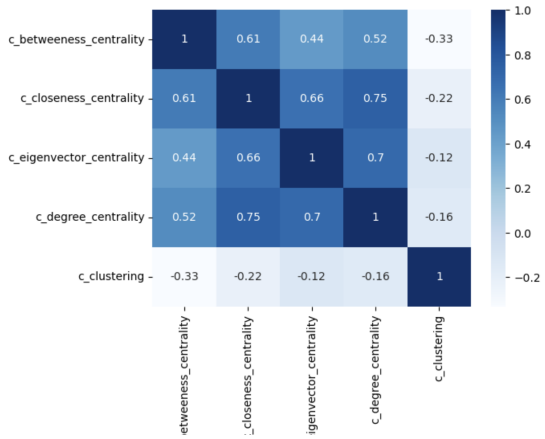

FIGURE 4.5: Pearson Correlation Coefficient

where $E(n)$ is the number of edges between the neighbors of node n, and k_n is the degree of node n.

- Closeness Centrality: The closeness centrality of a node in a network is a measure of how quickly that node can reach all other nodes in the network.

$$C_C(n1) = \frac{1}{\sum_{n1 \neq n2} d(n1, n2)} \tag{4.5}$$

where $d(n1, n2)$ is the shortest path length between nodes $n1$ and $n2$.

The Data Set

After performing the experiment 1000 times, we end up with 1000 observations. In each one of this observations we have the average of the measures described above for the removed nodes and the 3.4 of the network once we remove those nodes.

What is next?

Remember the goal of the experiment is to discover what variables to look at when letting go 10% of employees in order to maintain the communication as intact as possible.

In order to achieve that goal, based on the centrality metrics above for the nodes we remove, we will try to predict their effect, on the largest component.

We will do the prediction using three different algorithms. Linear Regression, Random Forest and finally a Multilayer Perceptron (MLP). Once we have trained our models we will study the feature importance of each variable for each model using Shapely values.

Why three different models?

There are two main reasons why I have chosen to run three different models on the same data.

The first one is to evaluate the data under different assumptions of the relationship between input variables and the target variable.

The second one is that the dataset is highly collinear (Figure 4.5). Which means that depending on the model, it will give a high importance to some input variables and a very low importance to others. Running three models will give us some diversity when trying to decide which one of the variables is most important.

4.3.2 Linear Regression, Random Forest and MLP

Linear Regression

Linear regression, as explained by Su, Yan, and Tsai, 2012 is a statistical method for modeling the relationship between a dependent variable and one or more independent variables. It assumes that the relationship between the variables is linear, meaning that changes in the independent variable(s) are associated with proportional changes in the dependent variable.

Random Forest

Random forest is an ensemble learning method that combines multiple decision trees to make predictions. It works by generating a large number of decision trees, each trained on a random subset of the data and a random subset of the features. The final prediction is made by aggregating the predictions of all the trees. More by Biau and Scornet, 2016.

As compared to the previous model, here we do not assume linearity.

MLP

MLP is a type of neural network that consists of multiple layers of nodes (neurons). Each neuron receives input from the previous layer and produces an output that is passed to the next layer. The output of the final layer is used to make a prediction. This model, as explained by Noriega, 2005 is an evolution of the perceptron algorigh that is able to solve non-linearly separable problems.

4.3.3 Shapely Values

Shapely values were introduced by Lloyd Shapely in 1953 in the context of cooperative game theory. They are used to distribute the total value generated by a group of players among the players in a fair way.

The core idea behind Shapley value based explanations of machine learning models is to use fair allocation results from cooperative game theory to allocate credit for a model's output $f(X)$ among its input features.

In order to compute the Shapely values of a given input feature X_i for a model's outputs $f(X)$, as explained by Merrick and Taly, 2020, we consider the following:

- Expected value of $f(X)$.
$$E[f(X)]$$
- Expected value of the X_i.
$$E[X_i]$$
- Conditional Expected value of $f(X)$.
$$E[f(X)|X_i = x_i]$$

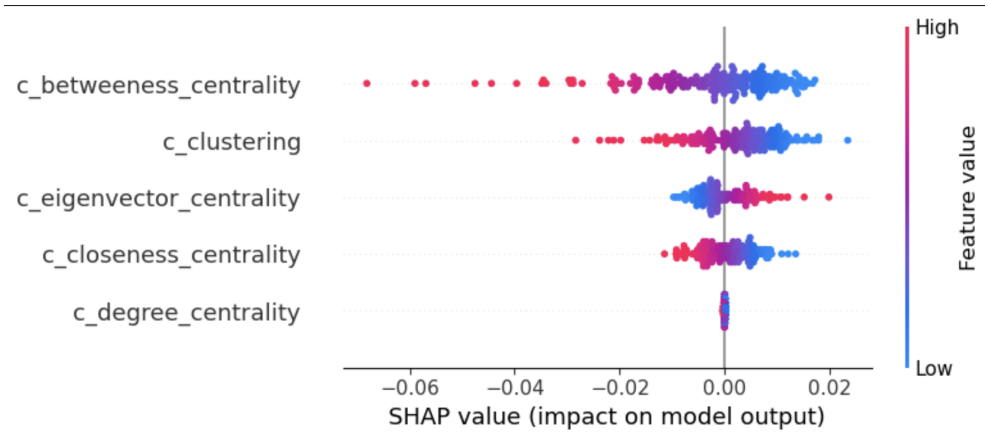

FIGURE 4.6: Shapely Values Linear Regression

In the simple expample when we only have X_i in our model, the Shapely value S_i of the variable can be interpreted as:

$$S_i = E[f(X)|X_i = x_i] - E[f(X)] \tag{4.6}$$

4.3.4 Results from applying Linear Regression

The best set of hyper parameters for this model was simply $Intercept = True$.

In Figure 4.6 we can see the Shapely values of the variables, and the most impactful metric to the outputs of the model is Betweeness Centrality. And, it has an inverse relationship with the model's output. This means that the higher the Betweeness Centrality of the removed nodes, the greater their negative impact on the predicted largest component.

The second measure with greatest impact (also in an inverse relationship) is the average clustering of the nodes. The more the neighbours of the nodes are connected on average, the greater the impact to the model's outputs.

The third metric with the greatest impact is eigenvector centrality. To be honest I was surprised to see that the relationship is not inverse as with the previous two metrics. This needs further study as to why it is happening.

As we can see the degree centrality is the one with the lowest impact. But this is due to collinearity among metrics. Given the rest, its addition to the model provides very little information gain for the model's outputs.

4.3.5 Results from applying Random Forest

In the case of the Random Forest, the best hyper parameters are given in AppendixA.

The variables with the highest impact (Figure 4.7) seem to be Betweeness Centrality and Degree Centrality. And the importance of the others is rather small.

This validates what we saw with the Linear Regression. In the previous case, due to the correlation that the rest of variables had with Degree Centrality, the impact of the average node degree was small. In this case, the same can be seen but in reverse order. Given that we take into account the Degree Centrality, the importance of the rest is small.

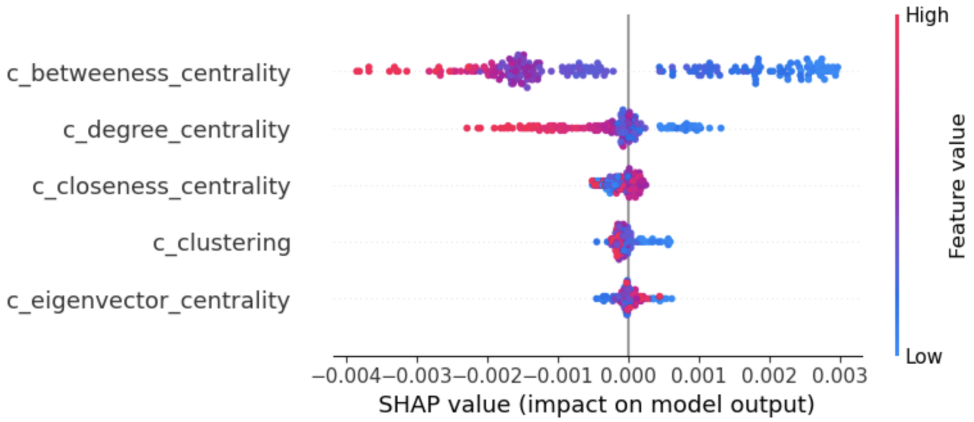

FIGURE 4.7: Shapely Values Random Forest

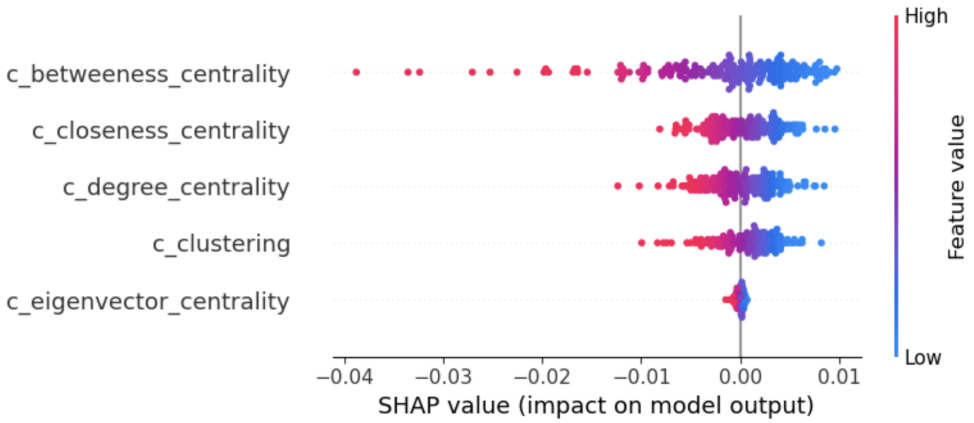

FIGURE 4.8: Shapely Values MLP

4.3.6 Results from applying MLP

In the case of the MLP, the best hyper parameters are given in AppendixA.

For MLP (Figure 4.8), Betweeness Centrality seems to be the most important metric again. And it also has an inverse relationship with the largest component. The greater the Betweeness Centrality of the removed nodes, the greater the impact to the model's outputs.

In this case, the variable the model discards due to collinearity, seams to be eigenvector centrality. Given the rest of variables are taken into account, the importance of the eigenvector centrality is small.

Observations

This results of this experiment indicates us that Betweeness Centrality is the metric we should look at in the case of layoffs if our objective is to maintain the communication flow inside the organization as intact as possible.

It is true that there is high collinearity among the variables, but looking at Figure 4.5 we can see the Betweeness Centrality is the variable that is the least correlated with the others after Clustering Coefficient.

The interpretation of this results is also very intuitive if we look at the definition of Betweeness Centrality (4.3). The higher the value of this metric the more probable the node is acting as a link between clusters of highly connected nodes. And so, eliminating nodes with high Betweeness Centrality, has a greater impact in how connected our network is.

Chapter 5

What about performance?

5.1 A Superside Project

As described in the Introduction, Superside is a company that delivers high quality Creative Projects to it's clients, it does so by means of having a pool of great Creatives and Project Managers that coordinate the Creatives and customer request.

All communication between the customer and the Project Managers happens through the proprietary platform that Superside has. The communication between Project Managers and Creatives happens in Slack channels. Every customer has a team of one or more Vreatives and one project manager assign to them. Depending on the type of project the customer requests, the team will look slightly different.

Note that a given Creative or Project Manager can have more than one customer assign to them.

The workflow looks as follows:

1. The Customer submits a project.
2. A team of one Project Manager and one or more Creatives is assigned to the project.
3. All collaborators of the project are added to a Slack channel specific to the project.
4. The Project Manager coordinates the communication with the client and transmits the customer's wants to the Creatives (communicating through the project Slack channel).

5.2 How do we measure performance?

The are two ways we can go on deciding how to measure performance. One is to assign some performance metric to each member of the Creative Ops team based on the projects they participate in. The second option is to measure the performance of a project as a whole.

I have decided to go with the second option. The reason being that attributing performance to each individual contributor of a project is too complex and it remains unsolved at the organization level. However, at the project level, there is one clear dimension that defines its performance in an unbiased and clear way. Was the project delivered on time or late?

Moreover, the fact that a project is delivered on time or not will highly depend on how efficiently the team members communicate between them and the customer. This intuitively brings the problem of performance closer to our complex networks world.

Now that we have decided our performance metric we will look at it from two different angles. Distribution of the team members among projects (a generic view), and communication dynamics inside each one of the project channels.

5.3 Distribution of the team members among projects

In this section we will try to study how projects are distributed among the members of the Creative Ops team. This, is a company-wide problem that the operations team is currently struggling with, some members get assigned to a lot of projects while others don't enjoy the same luck. However, this is not the problem we will try to tackle in this section.

Here, I just want to focus on performance. I want to discover if there is some underlying variable that drives the fact that some projects are delivered on time and others late.

As stated by Bruggen, 2015, the literature is mixed in terms of determining the effects of workload in the performance of employees. My hypothesis is that the workload, has an influence on the timely delivery of a project.

5.3.1 Problem Representation

In Chapter 2 we made use of Bipartite Network. In that case we had two different types of nodes, Channels and Users. In that network there were no edges between two nodes of the same type.

It could be tempting to use that same type of network to represent the current problem. However, in this case, I do want to encode information about the interactions between users (whether they have participated in the same project or not). Moreover, I also want to encode information about the size of the project they collaborated in (in this case, size means the amount of members participating in the project).

In section 2.1.3 we defined what Hypergraphs are. These are a type network that allow for the encoding of the information above, interactions between nodes of the same type, and the order of the interaction (In our case, this is the size of the project).

How does a Hypergraph actually work?

In order to aid our intuition, let's use the example by Battiston et al., 2020. Consider set of nodes $V = [a, b, c, d, e]$ and a family of interactions $I = \{i_0, ..., i_n\}$ such that $I = \{[a, b, c], [a, d], [d, c], [c, e]\}$. Each interaction in I represents a group interaction, and each one of them has a dimension $k - 1$ where k is the length of the group, we subtract one because for a given node, the interaction with itself does not count.

Given the above description of a system (which perfectly fits our problem), we define a Hypergraph as $H = (V, I)$ where V is the set of nodes in the system and I is the set of interactions among them, also called Hyperedges. Each Hyperedge $i \in I$ is a subset of V

Our Hypergraphs

For this study, we take all closed projects with 4 or more members and divide them into two groups, projects delivered on time, and projects delivered out of time. Then we build two separate hypergraphs for each category of project. We obtain that for

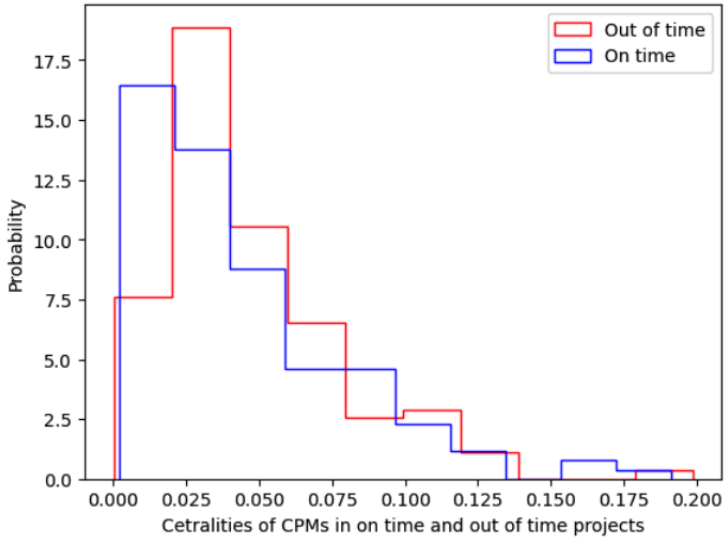

FIGURE 5.1: Histogram of cetralities of CPMs in on time and out of time projects

projects delivered on time, there are 639 nodes and 673 hyperedges. For projects delivered out of time, 641 nodes and 3680 hyperedges.

Out of all of the nodes in both categories of projects, 138 are Project Managers (CPMs). This means that all CPMs have at least one project delivered on time and at least one delivered out of time.

5.3.2 Centrality Measures (Clique Motif Eigenvector Centrality)

The centrality measure I will be using in this chapter is the Egenvector Centrality, because I would like the measure I use, to encode, not only information of the specific node, but also, information about the participation of the neighbouring nodes. And Eigenvector Centrality fits right in.

In simple networks, in order to obtain the Eigenvector Centrality, we obtain the Eigenvectors of the adjacency matrix. How does it work for Hyprgraphs?

Clique Motif Eigenvector Centrality

As defined by Benson, 2019, given a strongly connected Hypergraph H, the clique motif eigenvector centralities are given by the eigenvector 5.1, where $||c||_1 = 1$ and W_{ij} is the number of hyperedges containing the the nodes i, j and λ_1 is the largest eigenvector of W.

$$Wc = \lambda_1 c \tag{5.1}$$

5.3.3 Observations

The purpose of it all was to discover if workload has influence on a project being delivered in a timely manner. In order to illustrate that, we will compare the distribution of the CPM eigenvector centralities for projects delivered on time VS those delivered late.

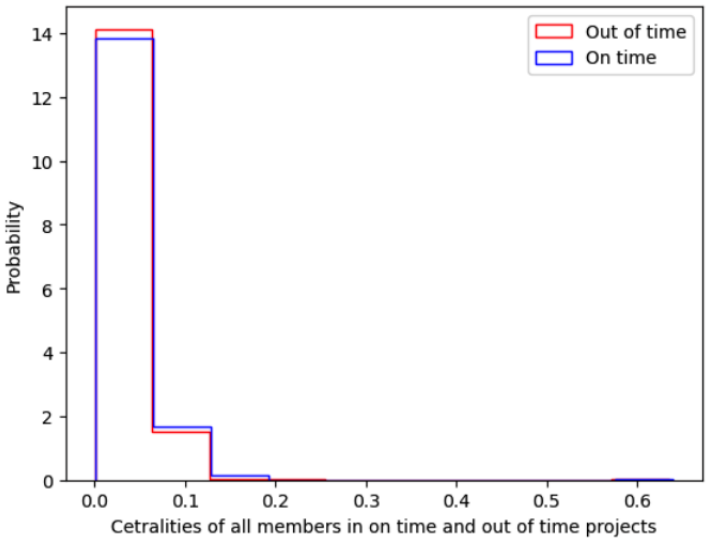

FIGURE 5.2: Histogram of cetralities of all members in on time and out of time projects

As we can see in Figure 5.1, the distribution of CPM eigenvector centralities for projects delivered out of time is slightly more skewed to the right when compared to project delivered on time. However, when performing a Kolmagorov Smirnov test (considering each distribution as a sample), we get a p-value of 0.069. And so, under 95% confidence level, we would reject the hypothesis that the distributions are different.

Nonetheless, the results are still significant if we consider the 90% confidence level. As I already mentioned in section 5.3.2, I am choosing the Eigenvector centrality as a measure because it encodes information about the importance of the neighbours of the nodes. Since CPM centralities are higher in the Hypergraph of projects delivered late, this could be an indication that their neighbours have more importance in the network. In this context, importance is acquired by participating in more projects, ergo, this is an indication that for the projects delivered late, the CPMs are collaborating more with creatives that are assigned to more projects. The more projects the creatives are assigned to, the more chances they are overworked.

This is more evident when looking at the same distribution without filtering only for CPMs (Figure 5.2). The distribution of centralities is almost identical, and when performing a Kolmagorov Smirnov test, we get a p-value of 0.39, way higher that for the previous case. All this indicates that for projects that are delivered late, CPMs are collaborating with creatives that are more important in the network, and so, possibly overworked.

5.4 Centrality Of CPMs in Project Channels

In this section, we will be taking a different a approach. We will consider the participation of the CPMs in each of the the projects.

My hypothesis is that the participation of the CPM in the project has an influence in the project outcomes.

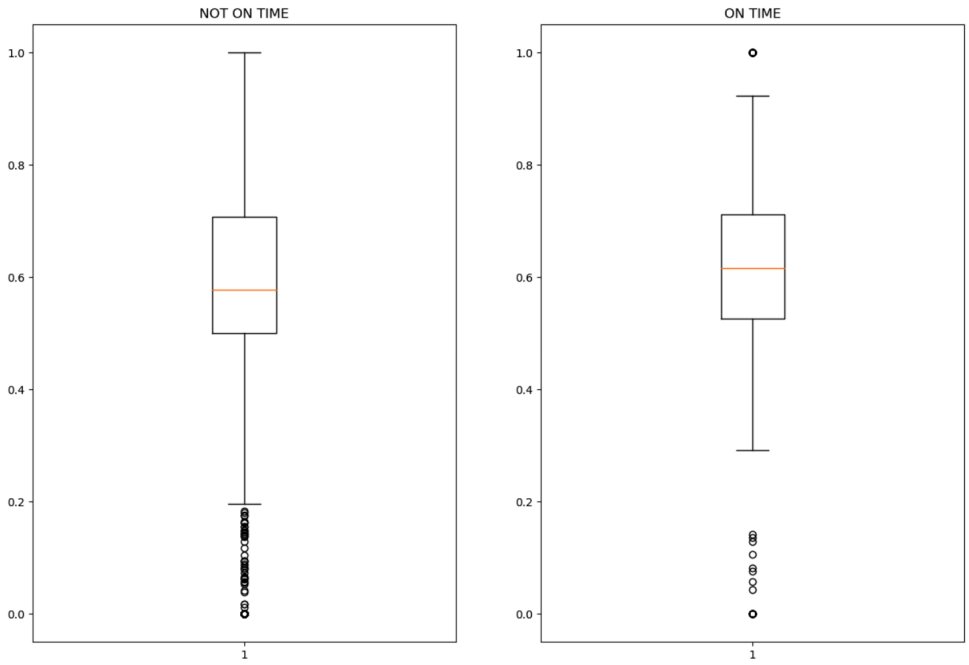

FIGURE 5.3: Boxplot of cetralities of CPMs individual projects

5.4.1 Our Dataset

In our dataset there are 2.711 projects with 5 members in average, out of which, one is a CPM. This means that we can build quite a big dataset of networks.

For this part of our analysis we will be taking the eigenvector centrality of the CPMs of each project. And we will compare the distribution of those centralities for projects delivered on time (353 in total) to those delivered out of time (2358 in total).

5.4.2 Observations

As we can see in Figure 5.3 for projects not delivered in time, there is a high number of projects where the centrality of the CPM is in the lower tail. When comparing the averages, we get that for those projects delivered on time, the centrality of the CPM is 0.61 VS 0.57 for those delivered out of time.

Is the difference significant

Well, as with the previous section I performed a Kolmogorof Smirnov test (taking the centralities of CPMs in projects deliverd on time and late as samples) and I got a P-Value of 0.027, which means that the difference is significant at the 95% confidence interval.

In order to make sure that the difference is not due to the size of the project (in terms of members) I also performed the test for the difference in the distribution of number of users for those projects delivered on time and those delivered late. The P-Value is 0.99 which means that the result is insignificant at the 95% confidence interval.

Chapter 6

Conclusions

I started this thesis making a claim about the importance of communication, and how it is the thing that sets us apart from any other species by allowing to create believes and frameworks that enable as to cooperate at scale. The end goal of it was to see how communication and cooperation happens inside an organization (Superside) and the method to do so was to analyse the communication patters of the employees of the company by extracting data from their communication platform (Slack) and looking at it from the perspective of Complex Networks.

The first question I tried to answer was whether the distributions of users among channels is optimal or not. I built a Bipartite Network and compared it to a model of a Random Network with the same parameters. Then, I attempted to answer the question from the perspective of robustness by performing two experiments of attacks to the network by removing channels. First randomly and then in a targeted manner, and I saw that there was not much difference between the targeted attack and the random attack. This happened because our network is random and there are no subset of nodes that monopolises the network. So, from the perspective of robustness, this is optimum.

Then I analysed the actual communication happening in the network by building an undirected network in which the nodes represented employees and the links indicated whether two pairs of nodes had interacted in any messages. This network, I concluded, was a scale-free network, and compared to the previous network, it was more sensible to targeted attacks, indicating that the company should shield its most central employees if it wanted to maintain communication as intact as possible.

Then I performed the biggest experiment of the thesis in order to answer to the question, "what metric should we look at when performing layoffs?". I built a dataset of 1000 observations where for each observation, I removed 10% of the nodes in the network and then measured their average centrality metrics, after removing them I measured the percentage of nodes in the largest components. Once I had the data set, I run different ML models trying to predict the largest component based on the centrality measures of the removed nodes. Then, using Shapely Values, I concluded that the most import metric is Betweeness centrality.

Finally, I had a look at performance, I tried to look if the centrality of the CPMs both in terms of their participation in projects and their participation in the communication of each individual projects affect the delivery of the project (on time or late). I got and indication that CPMs collaborating in projects with creatives that are involved in a lot of projects, or in other words, possibly overworked, these projects tend to be delivered late. And I also saw that the centrality of the CPM in each individual project had a significant impact on the project outcome (being delivered late or on time).

Chapter 7

Future Work

In section 4.3, I raised the caveat that the data was highly collinear. I tried to mitigate that fact by using three different models, Linear Regression, Random Forest and MLP. I chose them because they make different assumptions about the underlying data. I would like to see this expanded to different models and also using different subsets of the features I used.

In section 5.3 I introduced the Hypergraph in order to study the distribution of employees among projects. However, when computing the eigenvector centrality I used the Clique Motif matrix, and so, the underlying adjacency matrix, in the end became the same as for an underacted weighted matrix. I would like to see this experiment performed using different projections of the tensor other than Clique Motif.

Appendix A

Model Parameters

TABLE A.1: Best Hyper Parameters Random Forest

Criterion	Max Depth	Mean Samples Leaf	Mean Samples Split	N Estimators
absolute error	5	0.01	0.01	100

TABLE A.2: Best Hyper Parameters MLP

Activation	**Learning Rate**	**Max Iter**	**Random State**
identity	constant	2000	42

Bibliography

Barabási, Albert-László (2016). *Network Sience*. Cambridge University Press. URL: http://networksciencebook.com/chapter/0.

Battiston, Federico et al. (2020). "Networks beyond pairwise interactions: structure and dynamics". In: *Physics Reports* 874, pp. 1–92. URL: https://www.sciencedirect.com/science/article/pii/S0370157320302489.

Benson, Austin R (2019). "Three hypergraph eigenvector centralities". In: *SIAM Journal on Mathematics of Data Science* 1.2, pp. 293–312. URL: https://arxiv.org/pdf/1807.09644.pdf.

Biau, Gérard and Erwan Scornet (2016). "A random forest guided tour". In: *Test* 25, pp. 197–227. URL: https://arxiv.org/pdf/1511.05741.pdf.

Bloch, Francis, Matthew O Jackson, and Pietro Tebaldi (2023). "Centrality measures in networks". In: *Social Choice and Welfare*, pp. 1–41. URL: https://arxiv.org/pdf/1608.05845.

Bruggen, Alexander (2015). "An empirical investigation of the relationship between workload and performance". In: *Management Decision*.

Datta, Deepak K et al. (2010). "Causes and effects of employee downsizing: A review and synthesis". In: *Journal of management* 36.1, pp. 281–348. URL: https://www.researchgate.net/profile/James-Guthrie-7/publication/211384719_Causes_and_Effects_of_Employee_Downsizing_A_Review_and_Synthesis/links/0deec5277b354d9dd8000000/Causes-and-Effects-of-Employee-Downsizing-A-Review-and-Synthesis.pdf.

Grand, Steve (2000). *Creation, Life And How to Make It*. Harvard University Press.

Harari, Yuval Noah (2014). *Sapiens: A Brief History of Humankind*. Dvir Publishing House Ltd.

Manríquez, Ronald et al. (2021). "A generalization of the importance of vertices for an undirected weighted graph". In: *Symmetry* 13.5, p. 902. URL: https://research.usq.edu.au/item/q5683/eeg-sleep-stages-identification-based-on-weighted-undirected-complex-networks.

Merrick, Luke and Ankur Taly (2020). "The explanation game: Explaining machine learning models using shapley values". In: *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*. Springer, pp. 17–38. URL: https://arxiv.org/pdf/1909.08128.pdf.

Noriega, Leonardo (2005). "Multilayer perceptron tutorial". In: *School of Computing. Staffordshire University* 4, p. 5. URL: https://citeseerx.ist.psu.edu/documentrepid=rep1&type=pdf&doi=4c8339b893423f1e14e34cc1543faee4e5ee4244.

Slack (n.d.[a]). *Conversations*. URL: https://api.slack.com/docs/conversations-api.

— (n.d.[b]). *conversations.history*. URL: https://api.slack.com/methods/conversations.history.

— (n.d.[c]). *conversations.members*. URL: https://api.slack.com/methods/conversations.members.

— (n.d.[d]). *users.list*. URL: https://api.slack.com/methods/users.list.

Stray, Viktoria and Nils Brede Moe (2020). "Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack". In: *Journal of Systems and Software* 170, p. 110717. URL: https://www.sciencedirect.com/science/article/pii/S0164121220301564.

Su, Xiaogang, Xin Yan, and Chih-Ling Tsai (2012). "Linear regression". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.3, pp. 275–294. URL: https://www.cs.columbia.edu/~djhsu/ML/mlnotes/linreg.pdf.

Superside (n.d.). *superside.website*. URL: https://www.superside.com/why-us.