



FAKULTÄT FÜR
INFORMATIK

Digital Engineering

# IEEE Visual Analytics Science and Technology (VAST) Challenge 2020; Mini-Challenge 1

Digital Engineering Project Report

Seyedbehnam Beladi
Atrayee Neog
Xiongjun Wang

September, 2020

Supervisor: Prof. Dr.-Ing. Bernhard Preim
Dr.-Ing. Monique Meuschke
Uli Niemann

# Abstract

The main objective of this report is to solve the Mini Challenge 1 introduced by the *VAST Challenge 2020* committee. The main purpose of these challenge is to advance the field of *visual analytics*. This mini challenge aims at identifying and visualizing the activities of a white hacker group who had created a worldwide outage in the internet. Provided data contains network graphs. The objective is to use a template sub-graph given by the committee as ground truth to find the required hacker network. We applied various *visual analytics* tools to build and compare various sub-graphs with the template sub-graph using *graph analysis*. We also developed algorithms and strategies in addition to these tools to search and detect the potential hacker groups. Finally, based on our analysis, the most likely group was selected as the responsible group. This report summarizes our approach and visualization techniques followed by the gained insights in solving the five tasks required to complete the Mini Challenge 1.

# Contents

# List of Acronyms

**VAST** IEEE Visual Analytics Science and Technology

**CGCS** Center for Global Cyber Strategy

**PC** Parallel coordinates

# List of Figures

# List of Tables

# 1 Introduction

The IEEE Visual Analytics Science and Technology (VAST) Challenge is the basis of this project and here we present an overview to the challenge as well as the structure and the tasks that were given by the VAST committee to solve.
We first start with the objective of the VAST Challenge and then we take a look at the Mini-Challenge 1 and what is the overall problem statement. Mini-Challenge 1 presents a set of questions regarding graph-based datasets and specifically a large data structure that represents a vary large graph. Finally, we will examine all the tasks in more details and present the questions.

## 1.1 VAST Challenge

In this section, we will give a brief description of the VAST challenge as well as the overall structure of it. Here is a description from the VAST website:
"The goal of the annual IEEE Visual Analytics Science and Technology (VAST) Challenge is to advance the field of visual analytics through competition. The VAST Challenge is designed to help researchers understand how their software would be used in a variety of analytic tasks and encourage innovation in data transformations and interactive visualizations. VAST Challenge problems provide researchers with realistic tasks and data sets for evaluating their software."
It is worth noting that there are three Mini-Challenges that form the challenge and each Mini-Challenge is submitted separately.

## 1.2 Task Definitions

In this section, we will describe and explain the tasks given by the VAST Mini-Challenge 1 in details. In other words, we are going to define the problem statement

and try to clarify it. We start by giving an overview of the overall challenge and Mini-Challenge 1, and then we explain all the tasks that are included.

### 1.2.1 Overview

In response to an increase in malicious cyber-attacks, numerous "white hat" hacker organizations have taken it upon themselves to fight back to protect the global internet. One white hat group, who has so far stayed anonymous, accidentally launched a cyber event that took down the global internet.
The world's experts need to get in touch with the group so the effects on the internet can be neutralized and services restored. The only hope for a solution is a cyber think-tank – Center for Global Cyber Strategy, or CGCS – that may hold the key to identify the group that caused the malfunction.
The CGCS maintains offline databases of anonymized data donated by the white hat community for research purpose. CGCS also has an ongoing project to explore the motivation, structure and infrastructure of white-hat hacker groups (one of which is responsible for the current situation).

### 1.2.2 Mini-Challenge 1 Overview

CGCS research has resulted in the creation of profiles of typical white hat groups. One such profile has been identified by CGCS sociopsychologists as most likely to resemble the structure of the group involved in this accidental shutdown. The task is to examine CGCS records and identify those groups who most closely resemble the identified profile. [20]

### 1.2.3 Mini-Challenge 1 Tasks

1. Using visual analytics, compare the template sub-graph with the potential match provided. Show where the two graphs agree and disagree. Use your tool to answer the following questions:

   (a) Compare the five candidate sub-graphs to the provided template. Show where the two graphs agree and disagree. Which sub-graph matches the template the best?

(b) Which key parts of the best match help discriminate it from the other potential matches? [10]
Additional points: six datasets are given. Five of these datasets represent a sub-graph of the hypothetical network of hackers. A dataset is also given as the template of a potential hacker group. Regarding the key parts that is mentioned, since they are not defined so we must discover, define and point out these key parts.

2. CGCS has a set of "seed" IDs that may be members of other potential networks that could have been involved. Take a look at the very large graph. Can you determine if those IDs lead to other networks that matches the template? [20]
Additional points: The seeds are one row of data that correspond to a node in the large graph which is a big dataset containing the whole network of hacker groups.

3. Optional: Take a look at the very large graph. Can you find other sub-graphs that match the template provided? [20]
Additional points: The main task here would be discovering how to distinguish and define a sub-graph and also find a way to extract meaningful sub-graphs from the large graph.

4. Based on your answers to the question above, identify the group of people that you think is responsible for the outage. What is your rationale?[20]
Additional points: This question requires a comparison between all the given graphs and finding the most similar sub-graph to the template sub-graph.

5. What was the greatest challenge you had when working with the large graph data? How did you overcome that difficulty? What could make it easier to work with this kind of data? [20]
Additional points: Since the large dataset is huge in size, it cannot be loaded simply with usual methods and this question asks about the method that is used to handle the dataset.

# 2 Background Knowledge

In this section, we attempt to give a brief explanation of the various methods and concepts that we have used to solve the five tasks mentioned earlier in subsection 1.2. We first give an overview about the key concepts such as visual analytics, graph theory and parallel coordinates. Then we explain in details the similarity measures that we used to evaluate the graphs as well as the meaning and use case of the Wasserstein distance metric.

## 2.1 Visual Analytics

The field of visual analytics is a collaboration between the fields of data analytics and visualization. It incorporates various data analysis techniques along with interactive visualization methodologies to increase human cognitive abilities such that data driven decisions can be made to solve complex problems without much expertise.

Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets [11].

The use of visual representations and interactions to accelerate rapid insight into complex data is what distinguishes visual analytics software from other types of analytical tools. Visual representations translate data into a visible form that highlights important features, including commonalities and anomalies. These visual representations make it easy for users to perceive salient aspects of their data quickly[8].

As mentioned in subsection 3, the data provided for this task is very complicated and intertwined. As such, it was necessary to solve each task based on conclusions from extensive data analysis as well as visualization based approaches.

## 2.2 Graph Theory

Graph theory is a section of mathematics that explores the relationship between items by depiction as graphs. The items are represented as nodes of a graph and the relationships between different items are depicted through connections between them which are known as edges.

Graphs can be directed or un-directed based on the if the edges depict vectors or scalars. Also, different weights can be assigned to the edges to emphasize on the relative importance of a particular relationship. The graphs can represent various types of networks including social media, communication, client-server relationships, internet etc.

The data given to us represent different network groups which we then analyze using the various concepts of graph theory.

## 2.3 Visualisation Techniques

In this section we elaborate on the various visualisation techniques that were used for our analysis and explain what is the best use case for each.

### 2.3.1 Network (Graph) Visualisation

As per the visualisation principles stated in [7], the two main purposes of network visualizations are exploration of data and communication of findings. A network diagram should therefore be designed to display the information relevant for an analytic perspective. As a consequence, there cannot be a single best way of representing social networks graphically, which in turn creates lots of opportunities for visualization and algorithm design.

This is why it was important for us in our analysis to not only plot the data using graph visualisation tools and libraries but also to depict the data in the graphs in a way so as to display all relevant information for us to draw conclusions from. This was achieved by varying the node sizes and edge widths along with the usage of color to depict various aspects of the network.

### 2.3.2 Parallel Coordinates

A commonly used information visualization technique is parallel coordinates1,2 which is used for visualizing multivariate data. In parallel coordinates the axes are placed parallel and equally separated. Each axis corresponds to a variable and each data item, having values for all variables, is represented as a series of line segments intersecting the axes at the corresponding values [10].

As mentioned in [10], parallel coordinates could be used to effectively and efficiently perceive large amount of data. Therefore, we used this visualization to have an overview of all the measures for each of the graphs. This was done to compare as well as to select measures that were showing more evident patterns for further investigations.

### 2.3.3 Scatterplots

Scatterplots have been proven successful and useful diagramming techniques in descriptive statistics and information visualization. They take discrete data points with two data dimensions as input, and produce a 2-D plot of those data points by drawing respective dots on a diagram with two orthogonal axes representing the two data dimensions. Scatterplots are effective in displaying relationship in the data, such as correlation or other patterns [2].

We used interactive scatterplots multiple times throughout this work. We used this technique for many reasons. It is easier to perceive since it is a common visualization technique. We were also able to use shape, size and color to display more dimensions and discover our desired patterns.

### 2.3.4 Arc Diagrams

An arc diagram is a special kind of network graph. It is constituted by nodes that represent entities and by links that show relationships between entities. In arc diagrams, nodes are displayed along a single axis and links are represented with arcs [1].

We used the concept of arc diagrams on top of some of the scatterplots to convey the relationship between these nodes. In that way we were able to comprehend the complex dataset more efficiently. It also gave us the ability to add another dimension

without creating a huge clutter in the graphs.

## 2.4 Similarity Measures

In this section, we give a brief overview of the various similarity measures that are used in general and that we have used in our project, to quantitatively compare between two network graphs. We referred to [17] in order to understand what each measure highlights in the different graphs and how we can use each to solve the tasks in hand.

### 2.4.1 Degree Centrality

*Degree centrality* is defined as the number of edges connected to a node. It gives a measure of how "busy" a node is in the network and how important a node is in the entire network in terms of connections. In case of a directed graph, the degree can be categorised as "In Degree" and "Out Degree".

As our task includes finding nodes which are potential hackers in a network, it is evident that nodes with a high degree (or in other words with a high number of connections) will have a higher probability in being one of the hackers.

### 2.4.2 Betweenness Centrality

Betweenness is a centrality measure based on shortest paths, widely used in complex network analysis [3]. In other words, *Betweenness centrality* measures the number of shortest paths that cross a node in a graph. It represents the nodes which are central for the information flow within a network. The nodes with a high betweenness have a very high influence on the network and are information hubs.

In our analysis, this information can help highlight the nodes which are overall very influential in the network and might have started the entire outage being at the center of the network.

### 2.4.3 Closeness Centrality

The *Closeness centrality* of a node in a directed graph measures the average length of the shortest paths between the particular node and all other nodes of the graph.

This measure helps determine clusters in the graph and nodes which are central to the clusters. Closeness centrality is an important concept in social network analysis. In a graph representing a social network, closeness centrality measures how close a vertex is to all other vertices in the graph [13].

In terms of finding the hacker networks, the measure helps identify the nodes which are the centers of very close knit groups and who might have had an influence on how the other nodes in the cluster behaved.

### 2.4.4 Eigen Vector Centrality

*Eigen Vector centrality* directly calculates the relative influence of a node in the graph wrt. other nodes. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. That means it takes into account the influence of other nodes unlike the measures described previously.

This might be very central to finding the suspicious hacker group, as the measure highlights the highly influential group of interconnected nodes which act as a cluster, instead of highlighting individual nodes with very high number of connections or nodes which are closest in terms of distance. Eigenvector centrality is designed to be distinctively different from mere degree centrality when there are some high degree positions connected to many low degree others or some low degree positions are connected to a few high degree others [6].

### 2.4.5 K-Nearest Neighbour

The *K-Nearest Neighbour* as the name suggests, calculates a user defined (k) number of neighbours for a particular node to emphasize on how many nodes, the node n question can influence. The neighbours are determined in a metric space by calculating a user defined distance measure for example Euclidean distance.

For our analysis, the KNN emphasizes on nodes which have very high neighbours and hence have a higher influence on the network.

### 2.4.6 Out-Degree Centrality

As defined in subsection 2.4.1, for directed graphs, the number of connections can be divided into two types. The *Out Degree* measure of a node is the number of edges that originate in the particular node or in other words is directed outwards from the node.

As shall be discussed in section 3, the data given for the analysis has people, items, places and journals as nodes of the network graph. As such, in order to find potential hacker groups, it is important to lay more emphasis on the activities of people rather than, on what items were most bought/sold or which places were most frequently visited. As items, places and publication nodes have only edges coming into them, the Out Degree centrality hence, plays an important role in highlighting the persons who are most active in the network.

### 2.4.7 Page Rank Centrality

The *Page Rank centrality* is a normalized version of Eigen Vector centrality. The measure for a node is calculated exactly in the same way as 2.4.4, but the relative importance assigned to the nodes are scaled by considering the directionality of the edges, the shape and size of the nodes and also the distance between the nodes.

In our analysis, as the various network graphs to be compared and analyzed differ in their respective shapes and sizes, the Page Rank helps nullify the difference by scaling the scores accordingly.

## 2.5 Wasserstein Distance Metric

The *Wasserstein Distance Metric* is quantitative a measure to find the relative similarity and dissimilarity between two distributions or two graphs. It outputs a number which refers to the effective "cost" or effort that needs to be put in, in order to transform one graph to the other. The higher the value, the more effort required to transform one graph to another, and hence the two graphs being more dissimilar. So in other words, two very similar graphs will have a very low Wasserstein metric value.

The similarity measures described in subsection 2.4 result in distribution curves for the given sub-graphs to be compared. The *Wasserstein metric* as described in [18] is used to find the similarity between the various distributions of these similarity measures to compare between the sub-graphs and to identify the potential hacker group from them.

As you can see in Figure 1, the left image represents a standard way to compute distance between different distributions, while on the right is the Optimal Transport way which looks at a the best mapping function T which transports the blue function on the red one. T is the quantity of energy required to transport one distribution to another. The smaller T is, the closest f and g are despite being potentially very different coordinate-wise.

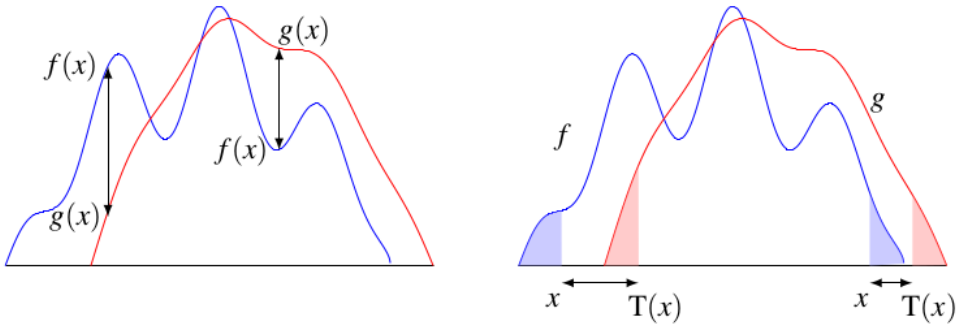



Figure 1: An intuitive understanding of Wasserstein distance metric [16].

# 3 Data Pre-processing

Before explaining the approach we took for solving the task, it is very important to understand the data first. Viewing raw data also often identifies problems in the data set, such as missing data, or outliers that may be the result of errors in computation or input. Depending on the type of data and the visualization techniques to be applied, however, some forms of pre-processing might be necessary[21]. In order to understand some of the visualizations and analysis that are presented in the following sections, it is crucial to have an overview of the datasets.
In this section we will explain the datasets, their characteristics and attributes and how they are connected to each other. We will also inspect the different aspects of the data, how they are categorized and their values.

## 3.1 Datasets

There are a total of 14 .csv files were given which are as follows:

1. CGCS-Template.csv : The template is a profile of activities that CGCS has built to represent suspicious activity associated with the hack.

2. Candidate sub-graphs: These graphs are portions of the large graph that have been extracted. Five sub-graphs are provided for comparison to the template. They are:
   - Q1-Graph1.csv, Q1-Graph2.csv, Q1-Graph3.csv, Q1-Graph4.csv, Q1-Graph5.csv

3. Seed Graphs: Each seed graph has a single record from which you will attempt to build a graph that matches the template. Three seed graphs are supplied as starting points for your search in task 2:
   - Q2-Seed1.csv, Q2-Seed2.csv, Q2-Seed3.csv

4. The BIG graph: All records collected by CGCS are contained in a single file (CGCS-GraphData.csv).

5. The three files with a description of the other datasets.

## 3.2 Data Overview

The CGCS has collected data from the community of cyber researchers over the last year. There are 123,892,863 records. They have compiled the data into a graph where each person is identified by an anonymous ID number. There are different channels of data, all of which are represented as a transaction between two nodes. The graph data is presented in a compact edge list format.([20])
Each of the records is a row in the large dataset. It represents an edge in the network. All graph files contain the following columns:

- Source: an integer Id of the source of the communication (could have different meanings based on the eType column).
- eType (edge type): a number between 0 and 6 (inclusive).
- Target: an integer Id of the source of the communication (could have different meanings based on the eType column).
- Time: Time is in seconds from 12:00 AM Jan. 1, 2025, time span related to the cyber event are exactly one year.

The source and target represent two nodes in the network that are connected to each other. These could be a direct connection between two people, a person going on a trip, a person buying an item, etc.
The data can be classified into 6 different channels. Each channel represents a different kind of transaction between two nodes. These are the channels: **Communication** (Includes two channels: Email, Phone), **Procurement**, **Co-Authorship**, **Demographic**, **Travel**,

In Table 1, you can see and overview of the channels, their possible values and a brief description.
Many of the channels also include: Weight: float values with different meaning based on the channel (such as length of travel, price of an item, etc.). SourceLocation:

Table 1: Overview of the different channels in the dataset.

| *Channel Names* | Communication (phone and email) | Procurement | Co-authorship | Demographic | Travel |
|---|---|---|---|---|---|
| *eType* | 0 & 1 | 2 & 3 | 4 | 5 | 6 |
| *Represents* | Direct connections between two persons | Buying and selling an item | publication of scientific or technical articles | spending habits of a person | Connecting people by location |
| *Location* | Some | no | no | no | yes |
| *Weight* | Always 1 | Value of the item | 1 divided by the number of authors | Money spent | Length of trip (days) |
| *Source* | person | person | person (author) | person / category | person |
| *Target* | Person | item | publication | person / category | location |
| *Notable points* | - | For each sell row exists: a buy row | Date might not be relevant | 29 distinct categories | Some weights are negative |

integer values between 0 and 5 representing countries[1]. TargetLocation: integer values between 0 and 5 representing countries. SourceLatitude: latitude locations within the country. SourceLongitude: longitude locations within the country. TargetLatitude: latitude locations within the country.

## 3.3 Dataset Details

There are five different node types in each dataset and seven different edge types. Node type 1, which represents persons, serves as a unifying type throughout the data set. It enforces the direction of the edge from itself. Type 1 nodes are the only nodes with a spatial location assigned. Node types are as follows:

[1] fictional countries

1. Person (used in all channels)
2. Product category (for the procurement channel, eType = 3)
3. Document (from the co-authorship channel, eType = 4)
4. Financial category (from financial demographics channel, eType = 5)
5. Country (from the travel channel, eType = 6)

As mentioned, there are some files that give extra information about the datasets. One of these files provides the node types which represent the following:
*0: Email / 1: Phone / 2: Sell (procurement) / 3: Buy (procurement) / 4: Author-of / 5: Financial (income or expenditure, depending on direction) / 6: Travels-to.*

## 3.4 Implementation of Data Pre-processing

There were few challenges with data processing. One of the major problems that we faced was related to the size of the large graph dataset, which was more than what a normal computer could process easily. To tackle this issue, we took many approaches and some of them were more effective than the others.
One of the approaches was to use libraries(such as ff [15]) that load the data in the hard drive instead of the RAM which is more typical. Using online platforms like Google Colab[2][5] was also another way of handling the dataset as well as the processing power needed to perform tasks.
The other issue was that the data had a lot of missing and repeated values specially the location data and we had to impute a lot of the these columns and rows. Based on what early analysis of the data showed, imputing this data would have not substantially affect the outcome of the analysis and removing the repeated rows and dependent columns had also reduced the required computational power.
One other issue worth mentioning was the intertwined and complicated nature of

[2]Google Colaboratory more commonly referred to as "Google Colab" or just simply "Colab" is a research project for prototyping machine learning models on powerful hardware options such as GPUs and TPUs. It provides a serverless Jupyter notebook environment for interactive development. Google Colab is free to use like other G Suite products.

the data which made it really difficult to perform analysis and visualize the data. Due to graph structure of the data, the powerful python package called "NetworkX[3]"[4] was used. NetworkX is a Python package for modeling, analyzing, and visualizing networks. It provides classes to represent several types of networks and implementations of many of the algorithms used in network science. NetworkX is relatively easy to install and use, and has much of the functionality built-in, so it is ideal for learning network science and performing analyses on small or medium sized networks[14].

The basic features of NetworkX are contained in several Python classes that represent different types of networks. In particular, this Graph, DiGraph, MultiGraph, and MultiDiGraph. These classes can be used to represent, analyze, and visualize most networks[14].

[3]NetworkX version: 2.4 — October 17, 2019
[4]https://networkx.org

# 4 Methodology

This section elaborates the methods used in the project to solve the five tasks given in subsection 1.1. The methods can be divided into two types namely:

1. Graph Comparison: this can be further subdivided into two types,
    - Quantitative Approach
    - Visualization-based Approach
2. Graph Building

## 4.1 Graph Comparison

Let's start with a "black box" view of a complex network. Let's pretend we are at a distance and instead of nodes, edges, and their attributes, we see a fuzzy grayish cloud. What can we tell about that cloud? Not much: only its size and density[22].

### 4.1.1 Quantitative Approach

The quantitative approach refers to the various data analytics methods that were used from graph theory like the similarity measures described in subsection 2.4 and the Wasserstein metric described in subsection 2.5. This approach directly aims at deriving insights by comparing quantitative measures between two graphs. The measures calculated for the same are described below.

**Similarity Measures:** The seven similarity measures as described in 2.4 were calculated for all candidate sub-graphs and the template which resulted in seven distribution curves for each of the sub-graph. The first approach was to calculate the mean, minimum value, maximum value and standard deviation of these distributions and compare the values of the template with the sub-graphs. However, comparing only certain parameters of the distributions did not give much insights from where

we could derive data driven conclusions on the similarity or dissimilarity between two graphs. Hence, it was necessary to compare each point of the distribution to get a more solid understanding on how the different measures compare between the template and the sub-graphs.

As a result, visualizations were created to compare the different distributions [4.1.2] and also the Wasserstein metric was calculated to have a quantitative comparison between them.

**Wasserstein Metric:** The distribution curves derived as above had to be compared from point to point for the template and the various sub-graphs. Hence, the Wasserstein distance metric as described in subsection 2.5 was used to compare the various distribution curves for the seven similarity measures for the sub-graphs and the template.

Wasserstein-based test is a function of the **waddR**[5] [19] library in R. The test results in a Wasserstein distance value which ,if low, depicts that for that particular similarity measure, the graphs are similar. It also outputs a p-value which determines how confident the test is for two graphs being similar or dissimilar based on the particular similarity measure and also states whether the graphs differ by shape, size or location of the nodes. The higher the p-value for the comparison (threshold being 0.5), the more statistically significant or confident the test is about the reported similarity.

### 4.1.2 Visualization-based Approach

We define visualization as the communication of information using graphical representations. Pictures have been used as a mechanism for communication since before the formalization of written language. A single picture can contain a wealth of information, and can be processed much more quickly than a comparable page of words[21].

The quantitative approach helped us obtain information about various aspects of the network data. It also gave us a better understanding based on the seven similarity measures that were calculated. However, this approach could not provide

[5]version 1.2.0 - August 2020

us with a certain answer to task 1, 2 or 4.

In order to visualize the data in a more meaningful way, we decided to use subsets of data rather than the whole dataset. We chose these subsets based their corresponding channel. These channels could be identified by the nodes e-Types. The motivation behind this was to observe the similarity of different networks based on each of these channels. If the person nodes in the channels of a network show the same behaviour as the template's channels, we could then conclude that they are similar.

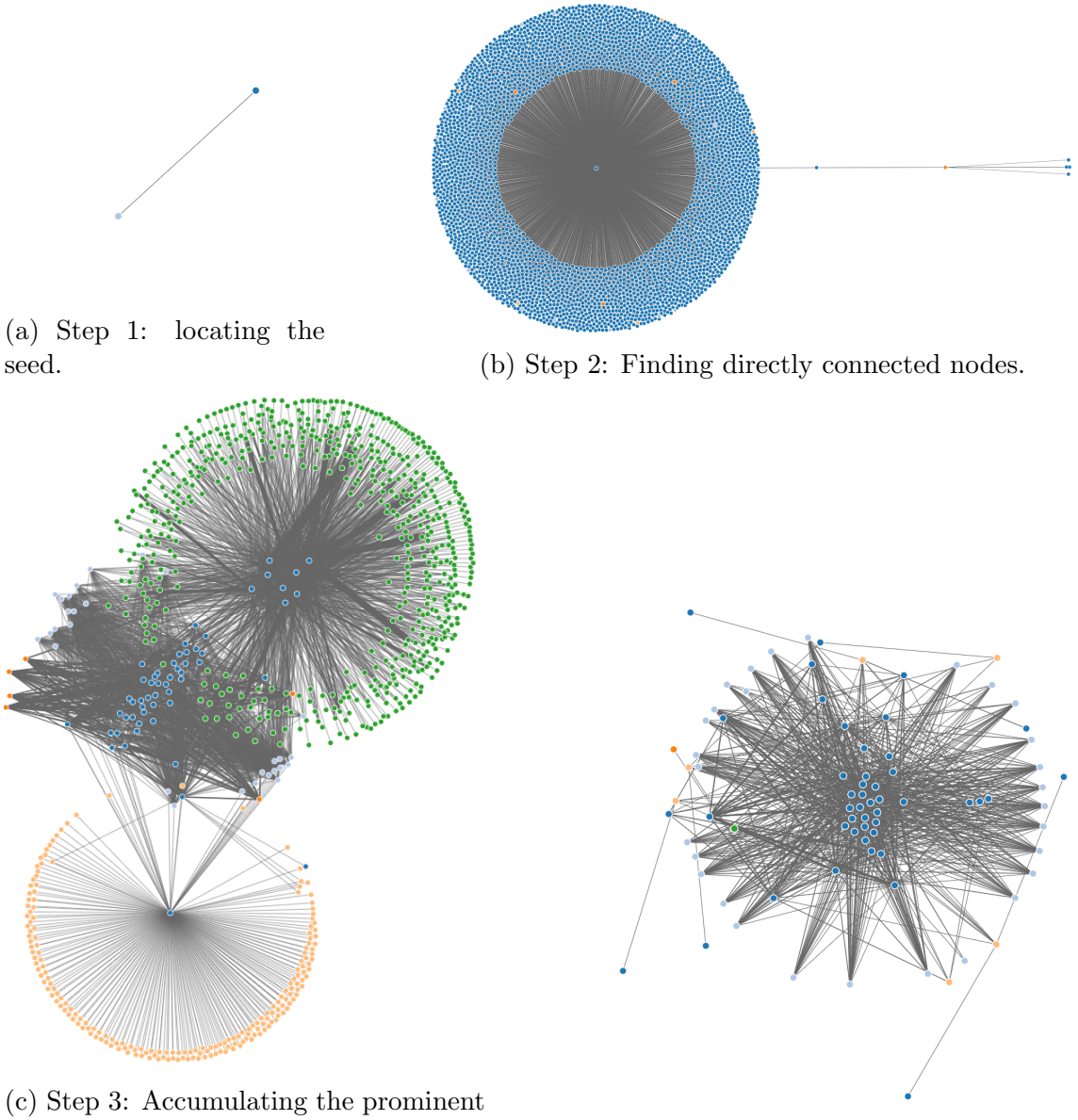
Since each of these channels had a different structure, we had to use different visualization techniques to find their underlying patterns. *Interactive visualizations* such as *Scatter plots* and *arc diagrams*, *interactive dashboards*, *Parallel Coordinates* and softwares like *Gephi* were used to analyze the channels in more details. Interactive abilities such as zooming, highlighting, selection and filtering were implemented in most of the visualization. Interacting with the data gave us the ability to find interesting patterns in the datasets much faster and easier.

Beside only visualizing the channels, we created interactive dashboards were one could compare the sub-graphs by simply selecting them. This was particularly useful when we wanted to compare different sub-graphs with the template sub-graph. We were able to switch between all the sub-graphs quickly, interact with the datasets of both and detect similarities or differences. These comparisons then helped us make a clear decision for tasks 1, 2 and 4 and helped us derive data driven results.

## 4.2 Graph Building

In this section, we will explain the process of finding a graph from the seeds briefly. There are four major steps that we took to find the graphs. The first step was to locate the edge in the big dataset. This was considered an extra step because some of the seeds were repeated more than once and this situation needed to be handled by removing the repeated values. Figure 2a is an example of this step.
In the second step we found all the nodes connected directly to the person IDs in the seeds. This was important because this could indicate whether it could lead to an eligible network or not. It also helped us identify which channels these nodes

belong to. An example of this step can be seen in Figure 2b.



(a) Step 1: locating the seed.

(b) Step 2: Finding directly connected nodes.

(c) Step 3: Accumulating the prominent nodes.

(d) Step 4: Finalizing the network.

Figure 2: Graph building process steps.

In the third step, we separated the nodes corresponding to people based on channels. Then we analyzed them based on channels and only the most prominent

nodes were kept. For instance, for the Communication channel, nodes with less than or equal to the minimum Closeness centrality were considered less prominent and removed. Finally, we added every channel data for the remaining person nodes and created an extended graph. As you can see in Figure 2c, a complicated network has been created in this step.
And finally, in the fourth step, based on existing patterns in the templates channels, we then reduced the extended graph in each channel to best resemble the template. There were nodes which were irrelevant by the decided criteria such as degree and also the nodes and edges which did not contribute to creating similar patterns to the template. We removed these nodes from the the network and a network such as Figure 2d was then generated.

# 5 Discussion and Results

This section reports the results for the five tasks of the challenge. We shall also discuss what techniques were finally helpful in drawing these results and which techniques did not work out.

## 5.1 Task 1

Task 1 part a) required us to compare five candidate sub-graphs with the template and determine which sub-graph resembles the template the most. It was also required to show where the various sub-graphs agree or disagree to the template.

For this purpose, we used the methods for graph comparison as described in subsection 4.1.1 and 4.1.2. The various similarity measures were calculated. Then their corresponding Wasserstein based test [4.1.1] with the Wasserstein value, the p-value and the pie-charts can be seen in the Figure 3. Pie-charts are depicting which among the three factors of shape, size and location of the nodes is responsible for the corresponding two graphs being dissimilar. We needed a percentage comparison between the three factors to depict which among the three was predominantly responsible for the two sub-graphs being dissimilar. For an overview, pie-charts was the best option as the actual numbers were not interesting to us. It can be seen that the sub-graphs 1, 2 and 3 are equivalent in their similarity to the original in almost all respects and sub-graphs 4 and 5 are the most dissimilar. This also answers the question as to in which aspects the various sub-graphs agree and disagree.


Figure 3: A comparison of sub-graphs for each of the seven distribution measures with density curves for each measure, Wasserstein metric (W), p-value (p) and the relative contribution of location, size and shape of nodes as pie charts.

Typical scientific data is represented on a grid with appropriate interpolation or approximation schemes,defined on a continuous domain. The visualization of such data in parallel coordinates may reveal patterns latently contained in the data and thus can improve the understanding of multidimensional relations [9].

In order to have an overview of all the measurement and compare the different sub-graphs to the template, we decided to create PC's for each of the sub-graphs. Not only it helped us identify the similar sub-graphs, it also made it clear for us which measurements could distinguish these differences more. We then used this information in further analysis.

Based on the results from the PC, the high p-values from the Wasserstein based test and the reasons stated in 2.4.6, 2.4.2 and 2.4.4, three measures were selected

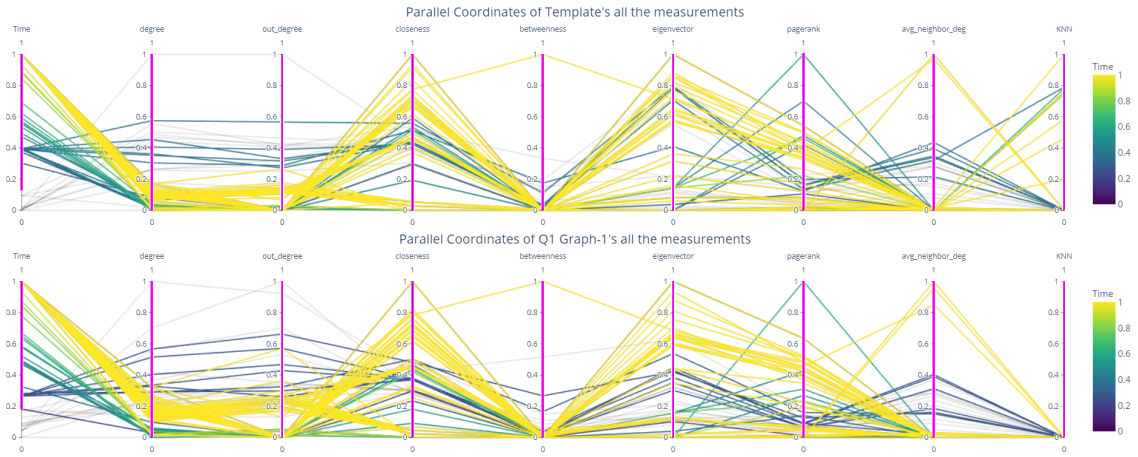



Figure 4: A parallel coordinates illustrating the template sub-graph and a graph 1 sub-graph. This is an example of the measures for both being similar.

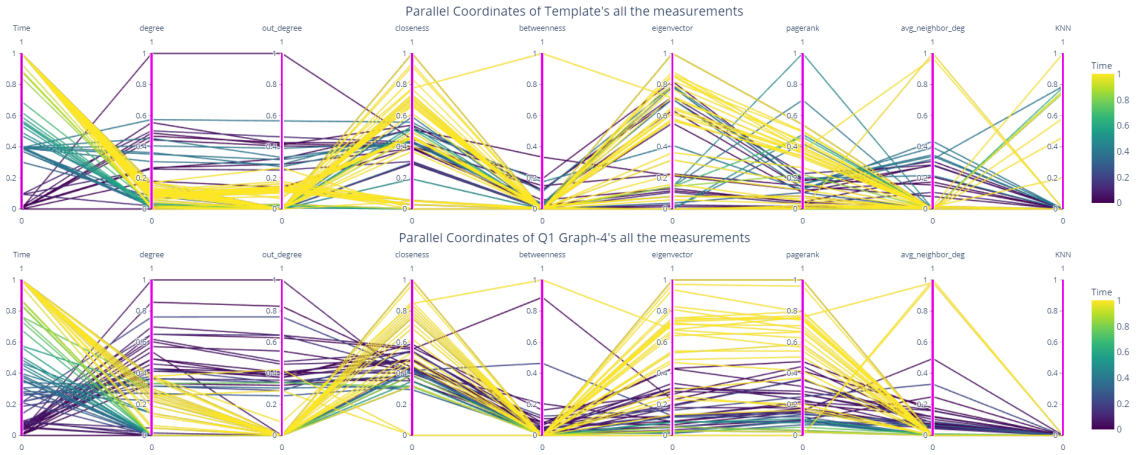



Figure 5: A parallel coordinates illustrating the template sub-graph and graph 4 sub-graph. This is an example of difference between measures.

to further investigate similarities and dissimilarities between the sub-graphs, namely Out Degree, Betweenness and Eigen Vector centrality. In Figure 6, Figure 7 and Figure 8 sub-graph networks were plotted with node sizes proportional to the magnitude of these measures. This although re-established the conclusion of sub-graphs 1 and 2 being very similar to the template and sub-graphs 4 and 5 being the most dissimilar, it showed that sub-graph 3 was dissimilar to the template for these mea-

sures.

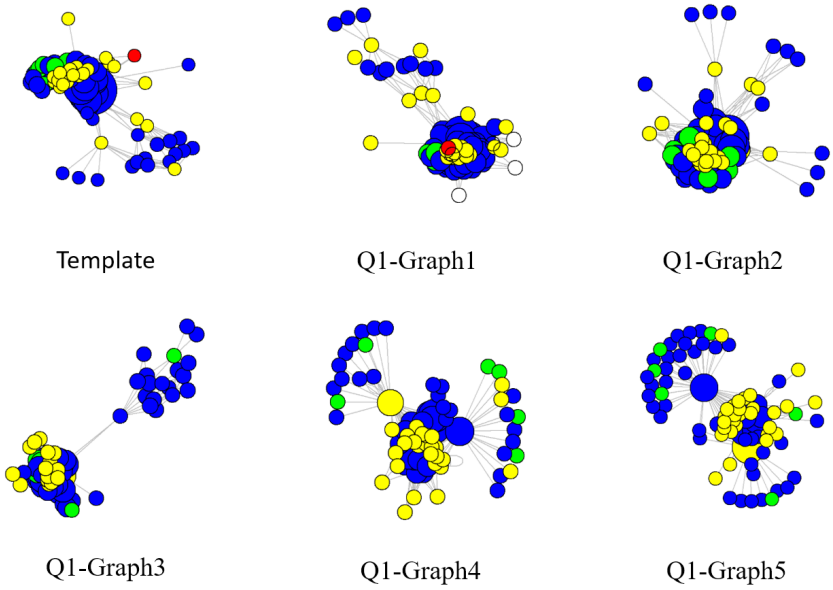



Figure 6: Comparison with network graph with node sizes proportional to Out Degree centrality and node color representing the node type.

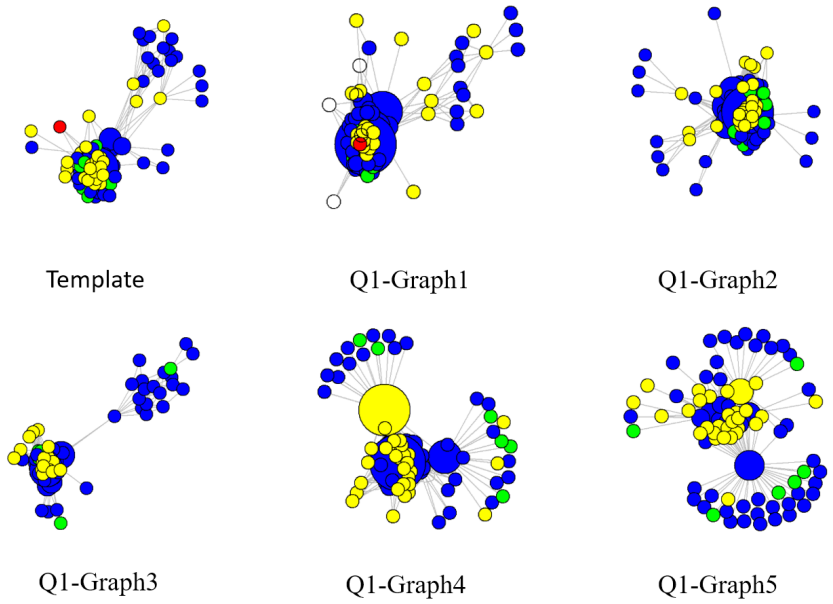



Figure 7: Comparison with network graph with node sizes proportional to Betweenness centrality and node color representing the node type.

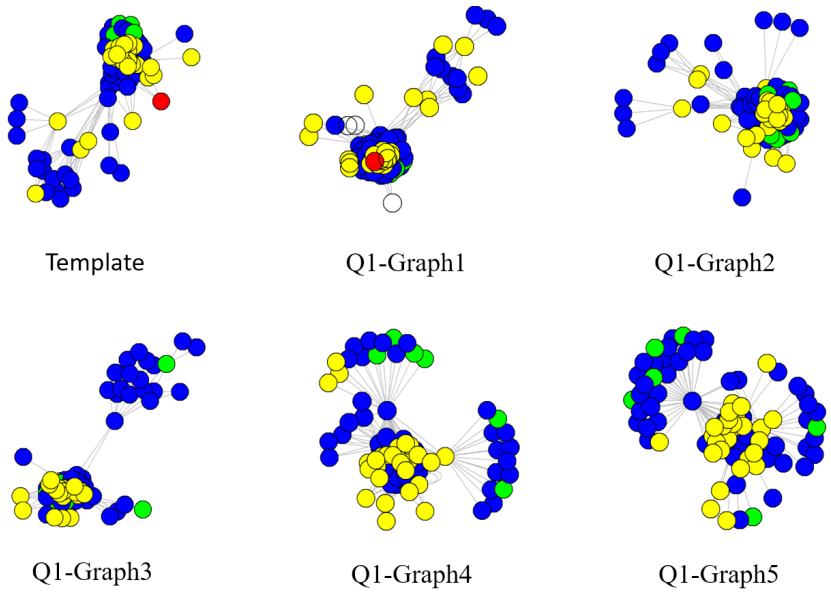



Figure 8: Comparison with network graph with node sizes proportional to Eigen Vector centrality and node color representing the node type.

In order to answer the question which sub-graph among 1 and 2 was the closest to the template, an interactive dashboard was created to visually compare the density curves from Out Degree, Betweenness and Eigen Vector for the sub-graphs with the template.

As it can be seen in Figures 9 and 10, the density curves of sub-graph 2 resembles the template the most.

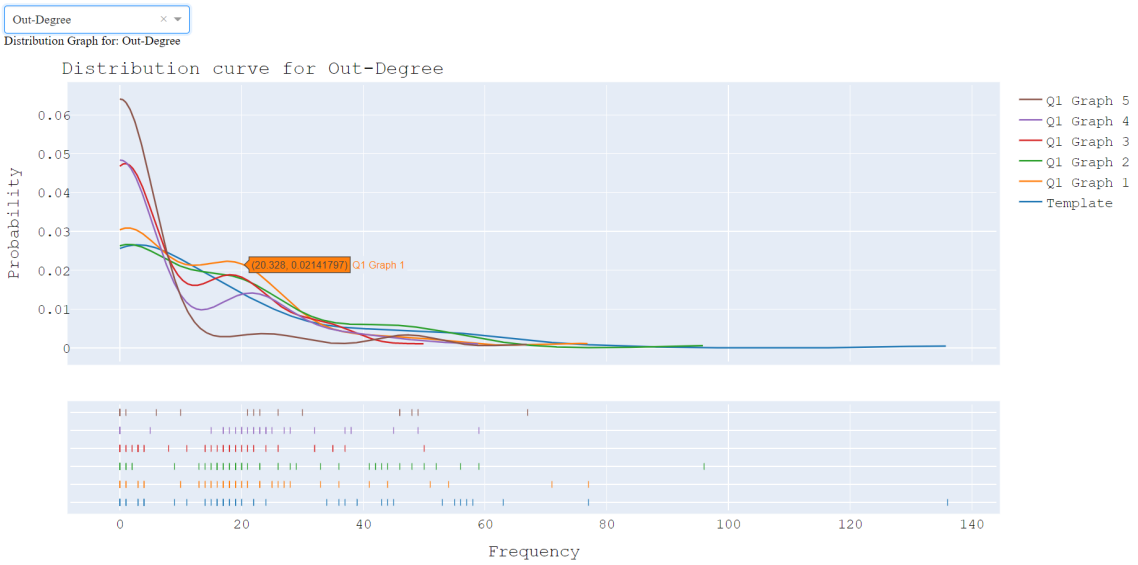



Figure 9: Comparison of distribution curves for Out Degree.

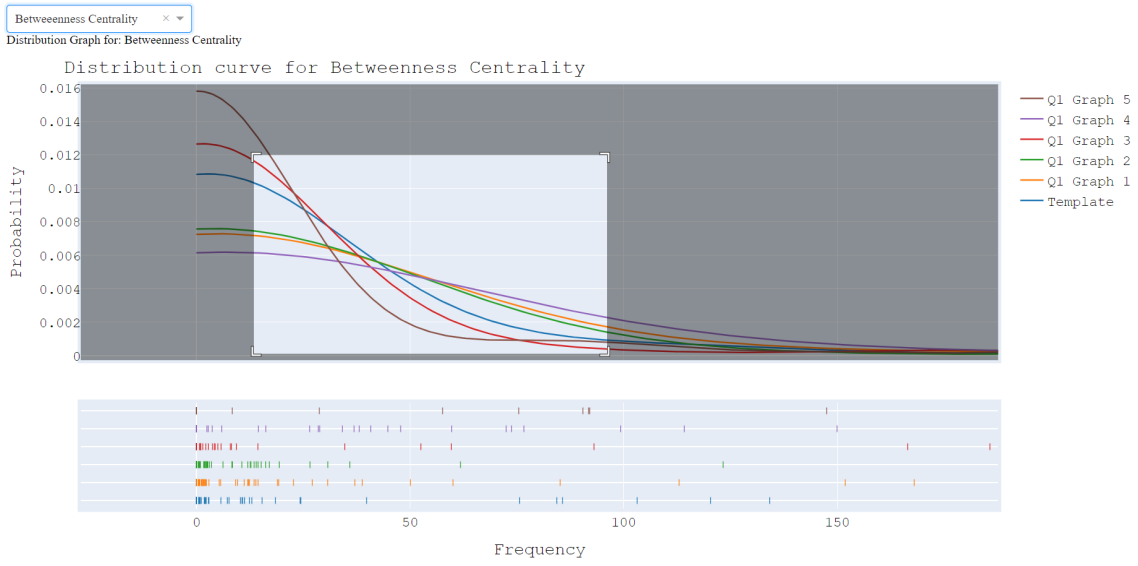



Figure 10: Comparison of distribution curves for Betweenness (zoomed in).

**Gephi** is an open source software for graph and network analysis. It uses a 3D render engine to display large networks in real-time and to speed up the exploration. A flexible and multi-task architecture brings new possibilities to work with complex data sets and produce valuable visual results [4].

When you explore an unfamiliar complex network for the first time, it often helps to perform a quick visual check of its structure before engaging in expensive code writing. Sometimes, you can semi-automate even the network construction itself [22].

The main purpose of using the Gephi[6] software was to include the time dimension and see the changes to the graph dynamically. We could also see the different measurements for graphs throughout time and compare them. There was a lot of flexibility for filtering, coloring, changing the size as well as changing the presentation order of the networks. These options facilitated our analysis to a huge extend.

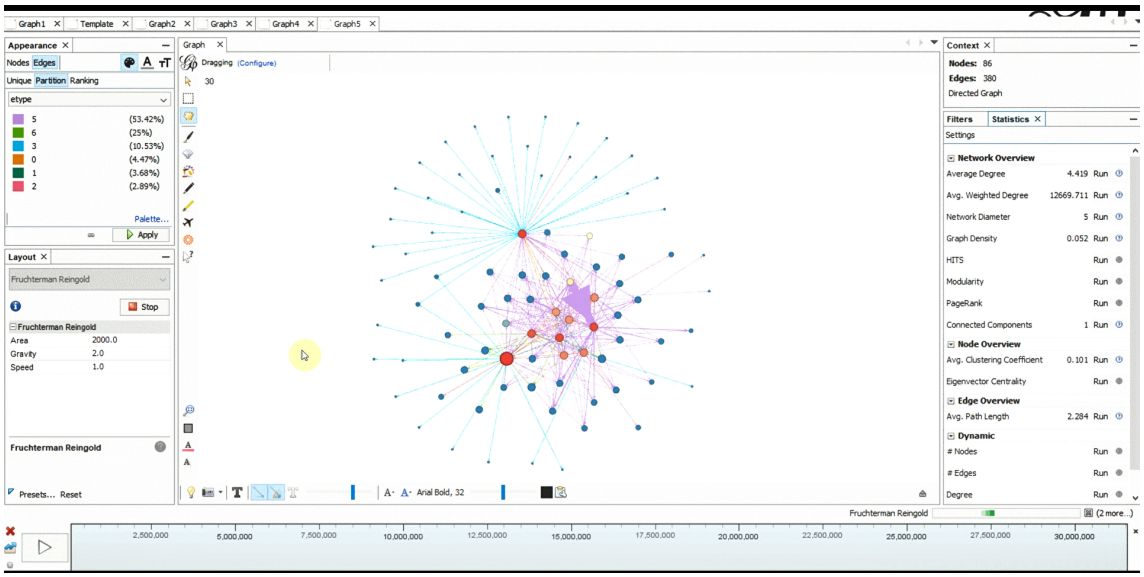



Figure 11: Visualizing and comparing the network graphs, network analysis and changes based on time in Gephi software.

[6]Gephi version: 0.9.2 201709241107

For part b of Task 1, it was required to do an in-depth analysis of the different sub-graphs to determine in which areas they resembled the template and to what extent. This is where the channel-based analysis was done for all the sub-graphs as mentioned in 4.1.2.

Starting with the travel channel (eType 6), there were certain clusters which emerged for template, sub-graphs 1, 2 and to an extent in 3 when the data was filtered on the basis of Target Location. As it can be seen in Figure 12 people travelled from the same Source Locations to the same Target Locations around the same time. These clusters were missing for sub-graphs 4 and 5, see Figure 13.

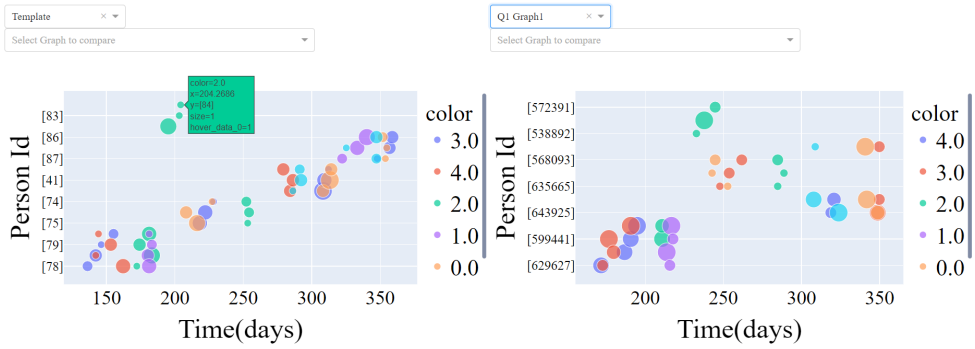



Figure 12: Clusters forming for travel channel of sub-graph 1 and template; x:Time, y:Person IDs, color:Source Location, filter: Target Location.

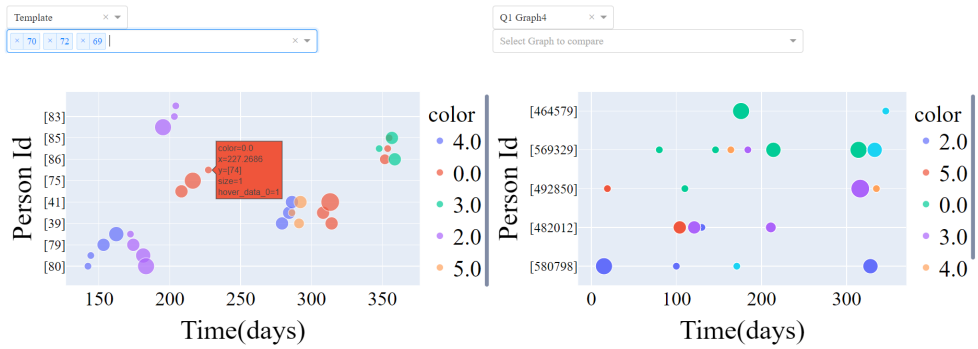



Figure 13: No clusters forming for travel channel of sub-graph 4 and template; x:Time, y:Person IDs, color:Source Location, filter: Target Location.

For the demographic channel which included incomes and expenses, sub-graphs 1,2 and 3, as seen in Figure 14 were resembled the patterns of the template the most

while sub-graphs 4 and 5 did not show any such common patterns as in Figure 15.

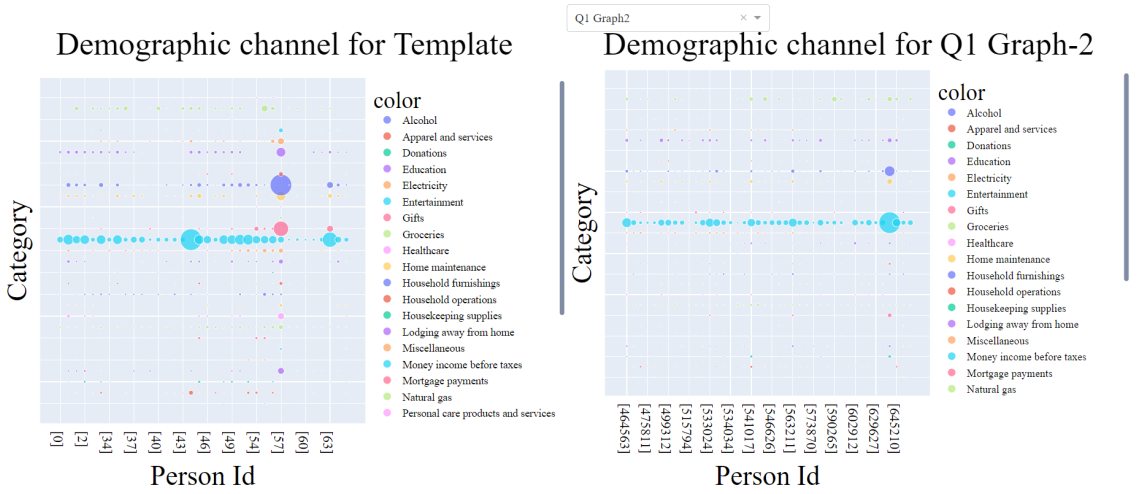



Figure 14: Similar patterns seen for template and sub-graph 2 for demographic channel; x:Person IDs, y:Category IDs, color:Categories.

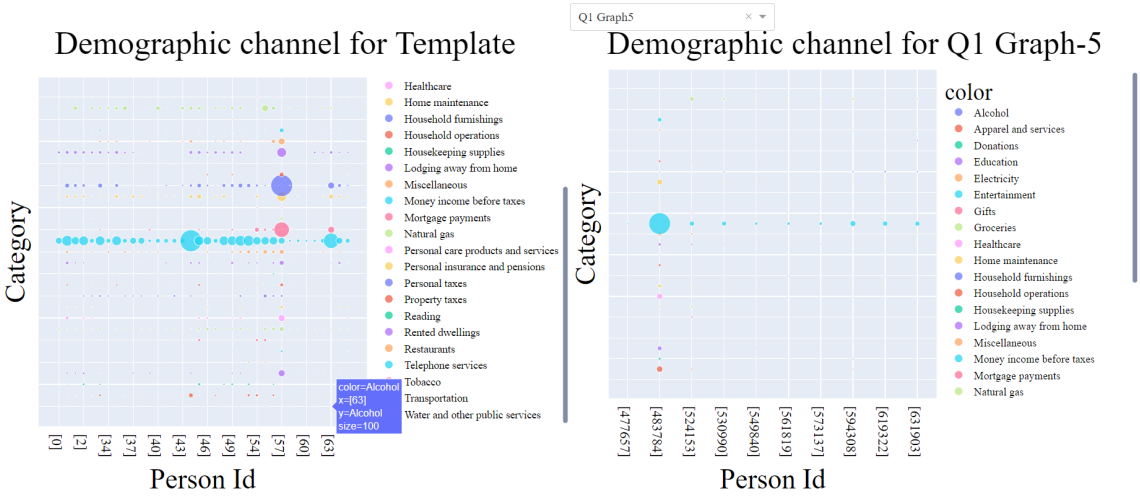



Figure 15: No patterns seen for sub-graph 5 with the template for demographic channel; x:Person IDs, y:Category IDs, color:Categories.

The procurement channel for template, sub-graphs 1, 2 and 3 showed that multiple transactions had been made by two persons for one particular item at different points in time as can be seen in Figure 16. For sub-graphs 4 and 5, there were no multiple transactions between two persons as seen in Figure 17.

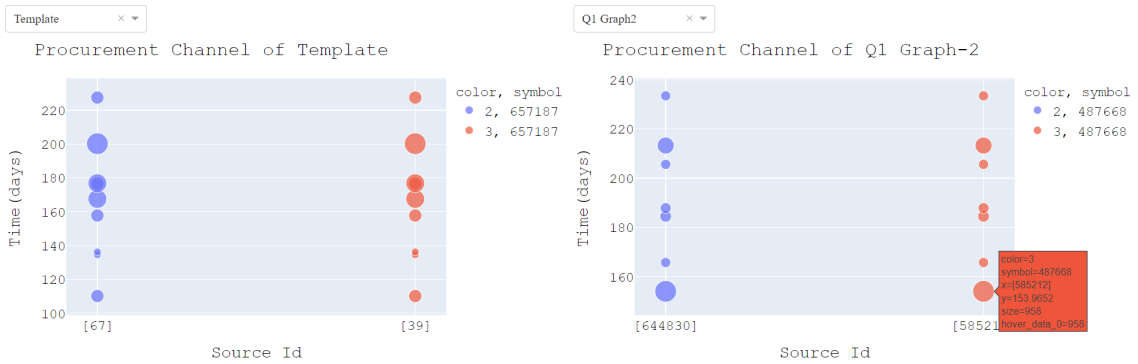



Figure 16: Similar patterns seen for template and sub-graph 2 for procurement channel; x:Person IDs, y:Time, color:Seller/Buyer.

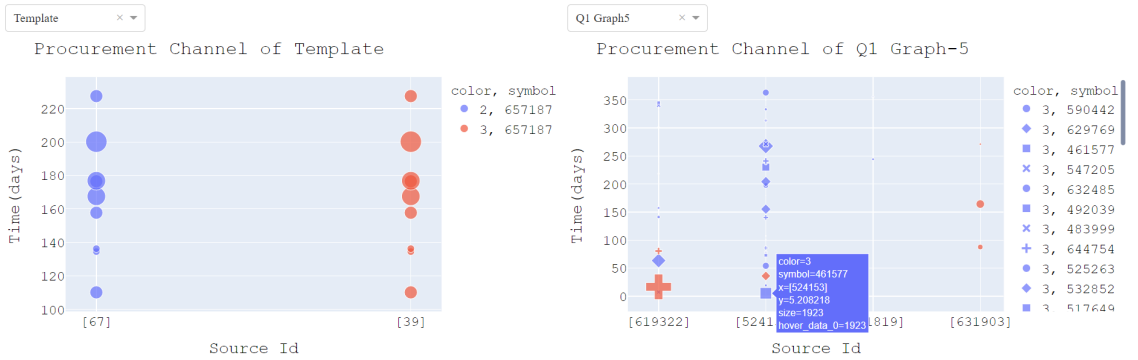



Figure 17: No patterns seen for sub-graph 5 with the template for Procurement channel; x:Person IDs, y:Time, color:Seller/Buyer.

The co-authorship channel did not have enough data to provide much insights into the sub-graphs and hence was not included in the analysis.

We visualized communication channel using a modified version of an arc diagram. Arc diagrams are an established method to visualize relations between nodes in a simple path graph, and are laid out in one dimension [12]. Our goal was to create a comprehensive way of showing the connection of nodes over the time. One can clearly see how people in the network communicated with each other by phone or via email over the time.

The communication channel as seen in Figures 18 and 19 indicated the same conclusion. It showed that sub-graphs 1 and 2 resemble the template the most, while sub-graphs 3, 4 and 5 had no such similarity as seen in Figure 25. It was

seen that for this channel, sub-graph 2 showed slightly more resemblance in terms of frequency and pattern of communication than sub-graph 1.

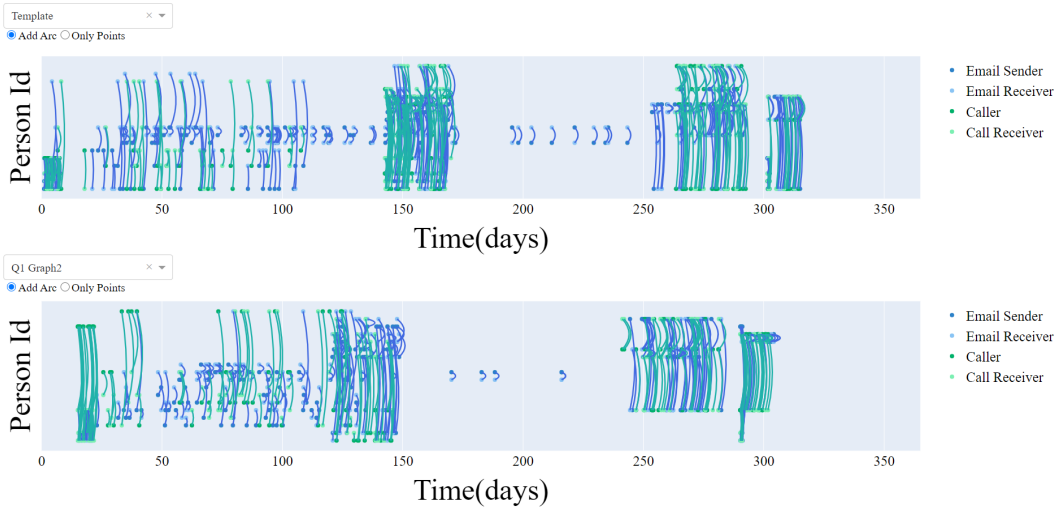



Figure 18: Arc visualization depicting similar patterns for sub-graph 2 with the template for communication channel; x:Time, y:Person IDs, arcs: Email/Call, color:Sender/Receiver for email/call.

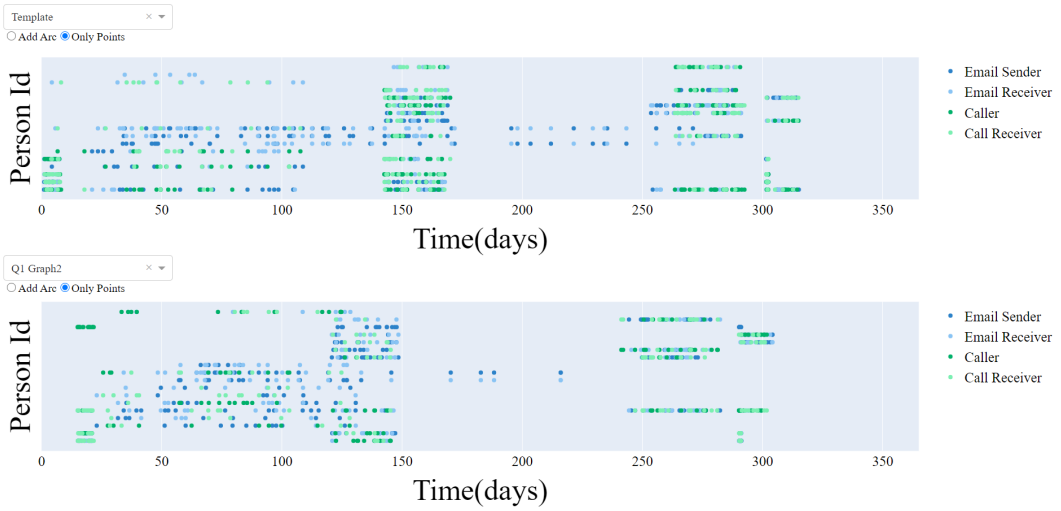



Figure 19: Point visualization depicting similar frequencies of communication for sub-graph 2 with the template for communication channel; x:Time, y:Person IDs, points: Email/Call, color:Sender/Receiver for email/call.

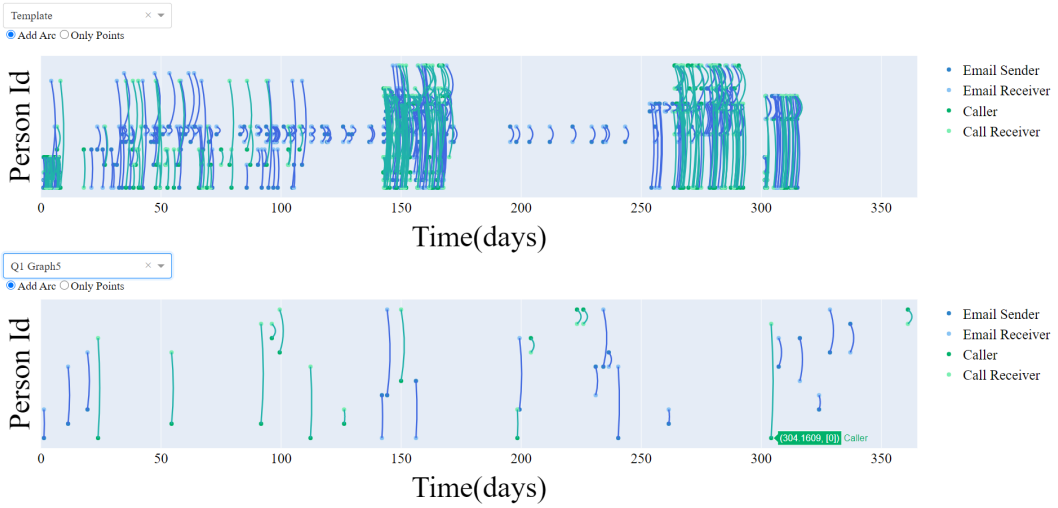



Figure 20: Arc visualization of sub-graph 5 shows no patterns common with template for communication channel; x:Time, y:Person IDs, points: Email/Call, color:Sender/Receiver for email/call.

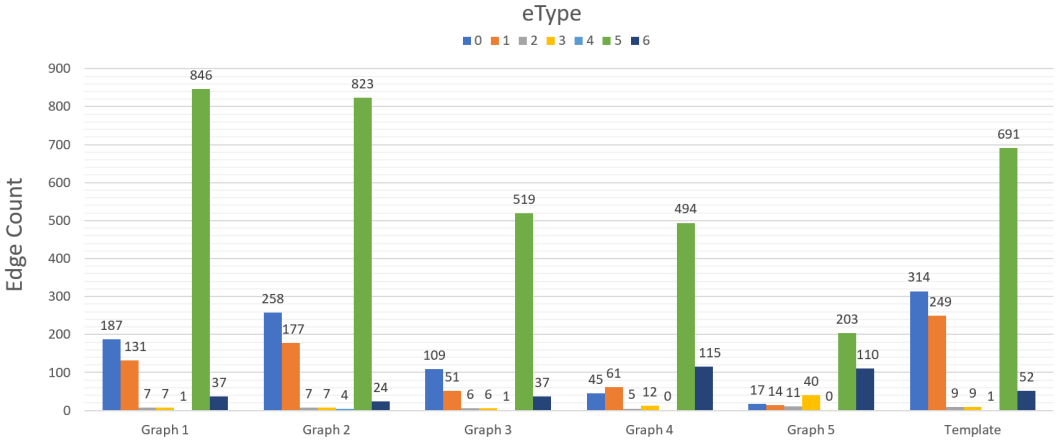



Figure 21: Barplot depicting the distribution of the data per channel.

As can be seen from Figure 21 the communication channel predominates all other channels and this is the reason that sub-graph 2 is concluded to be the most similar to the template among all sub-graphs.

## 5.2 Task 2

In this task, we were supposed to create the networks from three given seeds. As mentioned in section 4.2, an algorithm was created with four major steps. This procedure was applied to all three seeds.

Seed 2 did not go beyond the third stage because almost all the retrieved nodes had only co-authorship data. Therefore, it was concluded that this node was not leading to a network.

Seed 1 and Seed 3 on the other hand lead to two networks and the initial analysis showed that they were in fact similar to the template to some extend. The process of building these graphs were visualized to give further insight into the process and improve it. Further analysis on Seed 1 and Seed 3 was then done on all the channels of these two seeds similar to Task 1b.

These are the conclusions that we got based on this analysis:
*Procurement Channel*: seed 1 and seed 3 sub-graphs have a maximum of two transactions for the same item which is lower than the template, but there are transactions nonetheless.
*Travel Channel*: Similar to template and sub-graph 2, seed 1 and seed 3 sub-graphs include nodes which have travelled from one Source Location (depicted by color) to the same Target Location at close points in time.
*Demographic Channel*: seed 1 and seed 3 sub-graphs show similar income and expense patterns as the template and sub-graph 2.
*Communication Channel*: seed 1 and seed 3 sub-graphs frequencies of communication between sources are compared to template.

We conducted more extensive analysis and comparison in Task 4.

## 5.3 Task 3

In the previous task we were given 3 Seed nodes and we build the graphs from this starting point. However in this task we were asked to find potential hacker group networks by looking into the large graph. These two tasks might seem very different but we took an approach similar to what we did in the Task 2. In this task we tried to identify patterns in large graph that are similar to the template sub-graph.

We first examined each of the channels. we wanted to see which channels have patterns that could potentially be used in breaking down the large graph. In other words, we were looking for channels that had meaningful accessible patterns. Communication channel of the large graph was complex containing nodes with thousands of connections, making it impossible to visualize or analyze. There were no particular patterns in the single data point for the co-authorship channel and based on our findings in the Task 2, not all of the demographic data of a person was considered in the subgraph. Therefore, we selected procurement and travel channels to look for patterns.

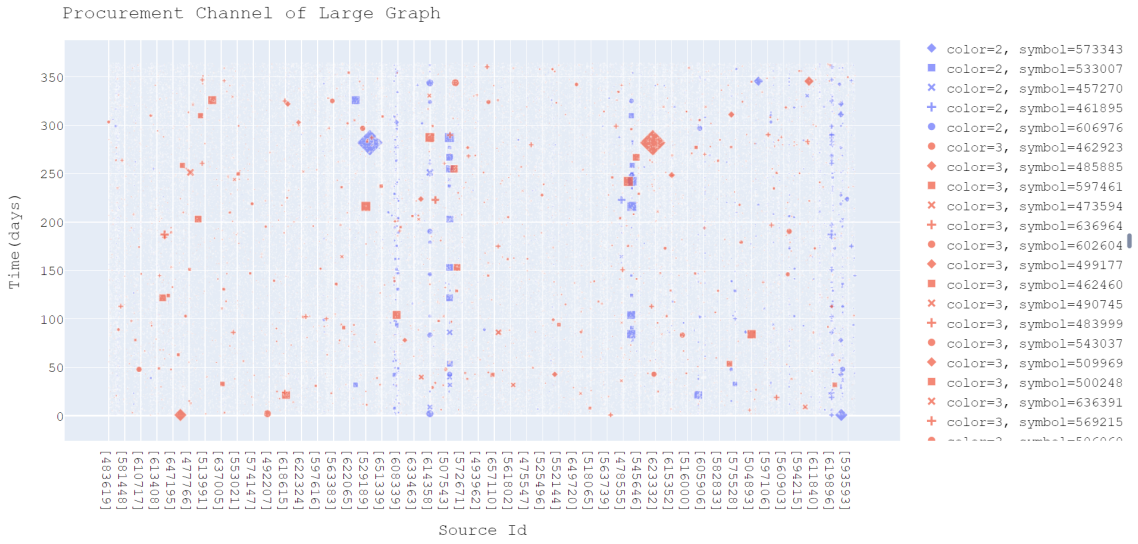



Figure 22: Procurement channel of large graph was visualized and analyzed to find patterns. Color: eType, symbol: items.

Our goal here was to identify groups of node which showed the same patterns as the template sub-graph in both channels. We would then cross-check these groups

and find a number of seeds. Finally we would use the same method in Task 2 to create networks if possible. It is worth noting that, we also visualized all the large graph channels except the communication. This visualizations gave us an overview of the channels and created a better perception of the large graph.

Figure 22 shows an interactive visualization of the the sellers and buyers in the procurement channel of the large graph. From the procurement channel of the template sub-graph we concluded that buyers and sellers who had seven or more transactions over the year, could potentially be a part of the same hacker group. We developed an algorithm which helped us first remove the nodes with very few transactions and then we were able to find a list of nodes that had many transactions. Further filtering and visualizing these nodes enabled us to find three pairs of nodes. This meant that we had three procurement channels for potential hacker groups.

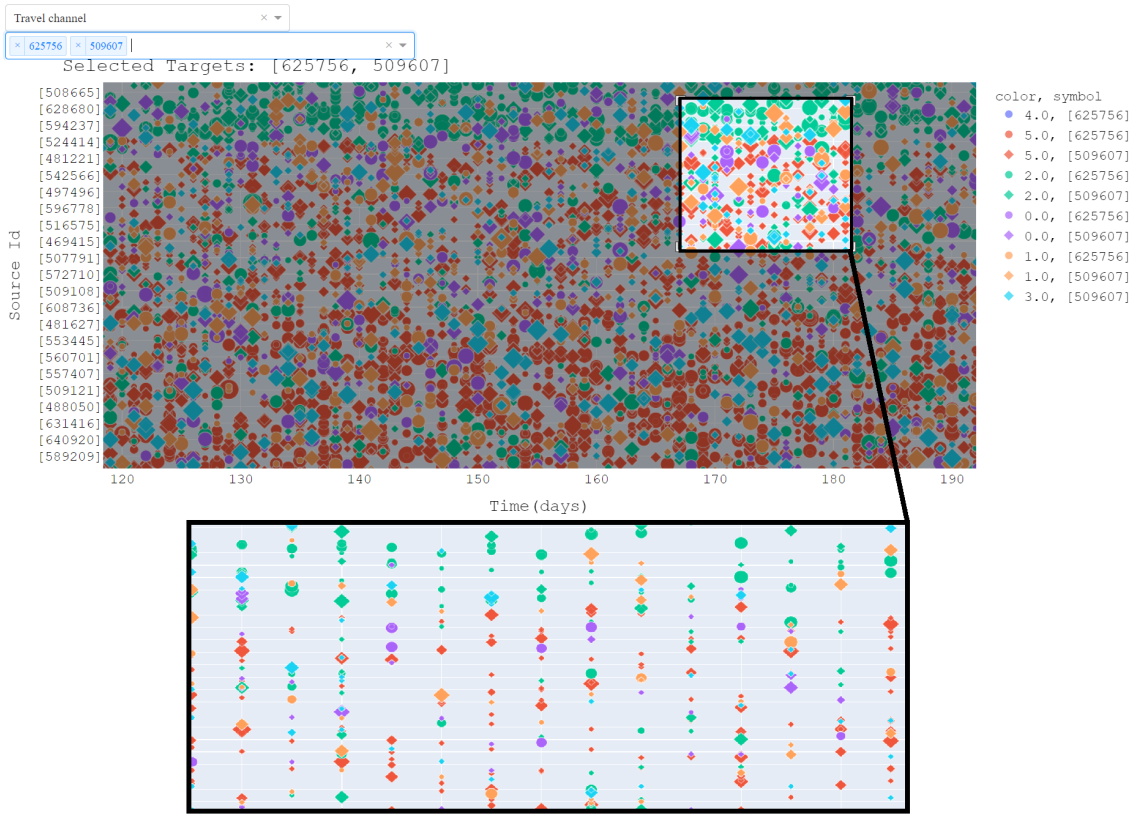



Figure 23: The travel channel for the seeds derived from the procurement channel were extracted and the Sources were put into a list.

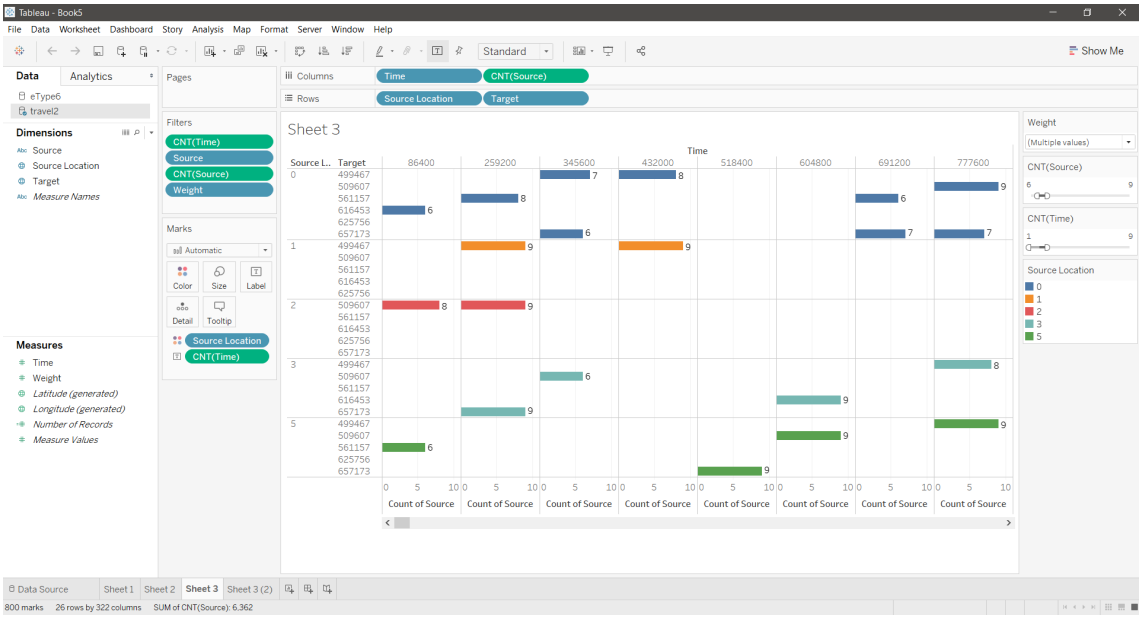



Figure 24: Tableau Desktop was used to extract the people who travelled together-from the same source location to the same target location at around the same time-from the large graph. Lists of Source Ids for each cluster was then obtained.

We couldn't approach the travel channel in the same way, because as can be seen in Figure 23, the data was much more complicated. Variety of shapes (representing the Target Id) and colors (representing the Country) in Figure 23 indicates many values in each dimension. To tackle the high dimensionality issue, we used Tableau Desktop[7]. We applied interactive capabilities of this tool to clean, analyze and filter the data and found over fifty meaningful clusters from thousands, see 24. We did this by considering the patterns we observed in the template sub-graph. Patterns based on departure and arrival time, duration of the travel, etc.

Then we located the travel channels of the node pairs extracted from the procurement channel and cross-checked the channels with our travel clusters. We discovered sets of nodes with the proper travel channels. Finally, We took the same approach as Task 2 with a small difference (we didn't need to expand the procurement and travel channels in the extended networks). At the end, we discovered two potential hacker group networks.

[7]Tableau Desktop version 2019.4.7, Tableau Software Inc., Mountain View, CA, USA

## 5.4 Task 4

Task 4 required us to highlight and conclude as to which group we think were the hackers to have caused the outage. As such, it was required to compare between the template and the sub-graphs from each task that resembled the template the most.

For this purpose, all experiments from subsection 5.1 were repeated with the template, sub-graph 2 from Task 1, sub-graph 1 and 3 from Task 2 and sub-graphs 1 and 2 from Task 3. The first was the Wasserstein based test whose results can be seen in Figure 25. The next step was to plot the network graphs for Out Degree, Betweenness and Eigen Vector centrality as seen in Figures 26, 27 and 28.

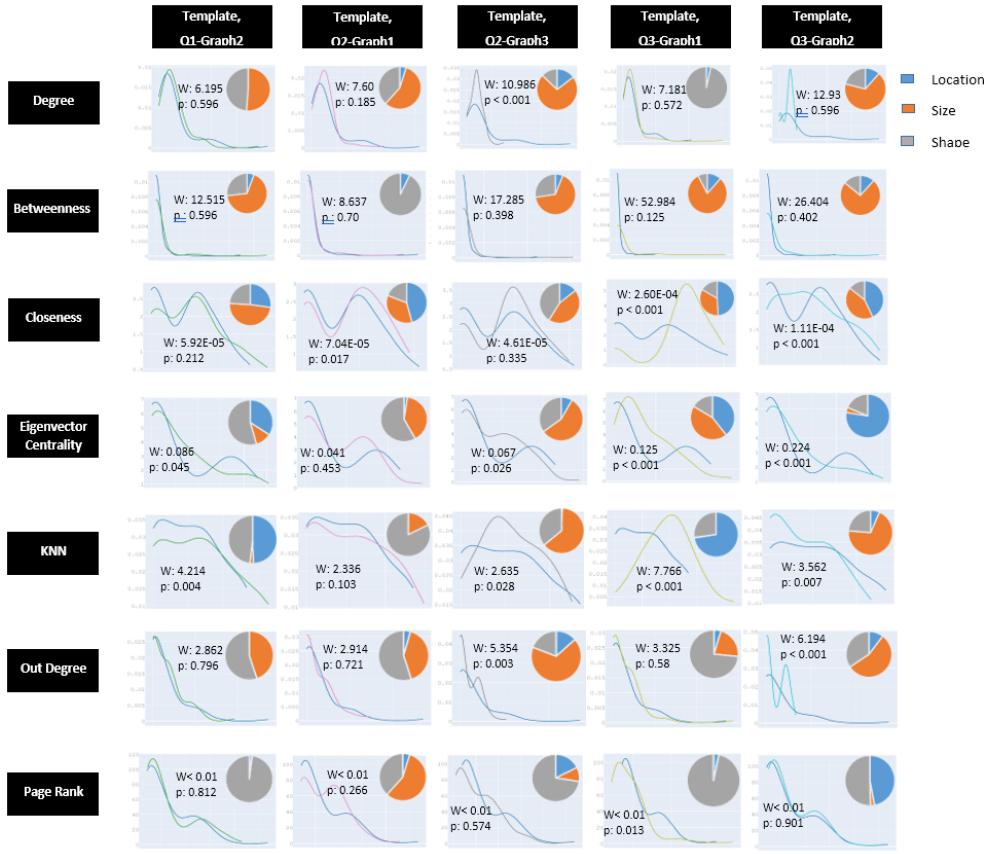



Figure 25: Wasserstein-based test for the most similar sub-graphs of Tasks 1, 2 and 3.

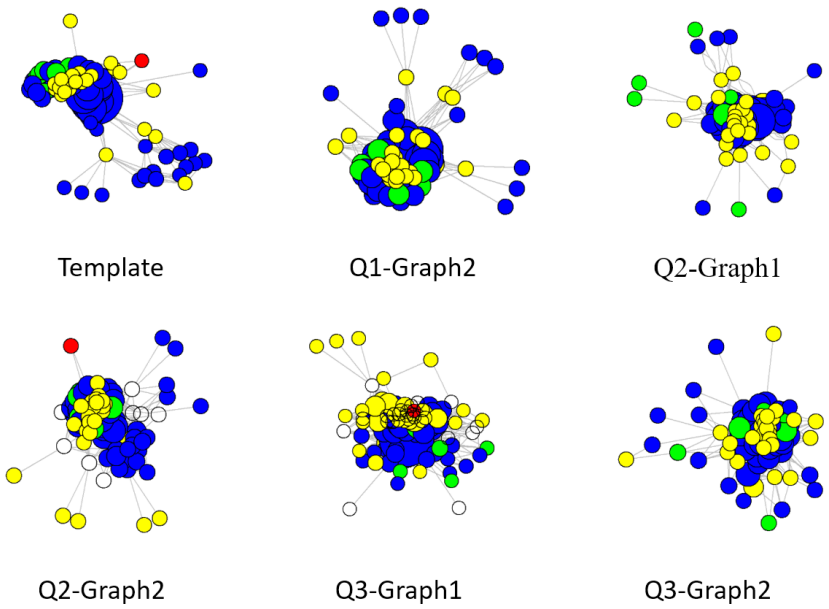



Figure 26: Network graphs with node size proportional to Out Degree and node color representing the node type.

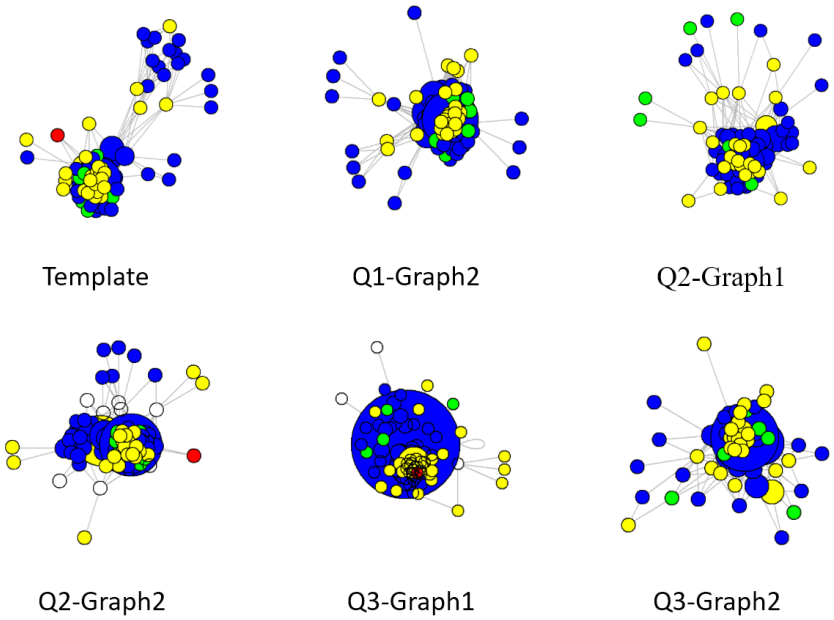



Figure 27: Network graphs with node size proportional to Betweenness and node color representing the node type.

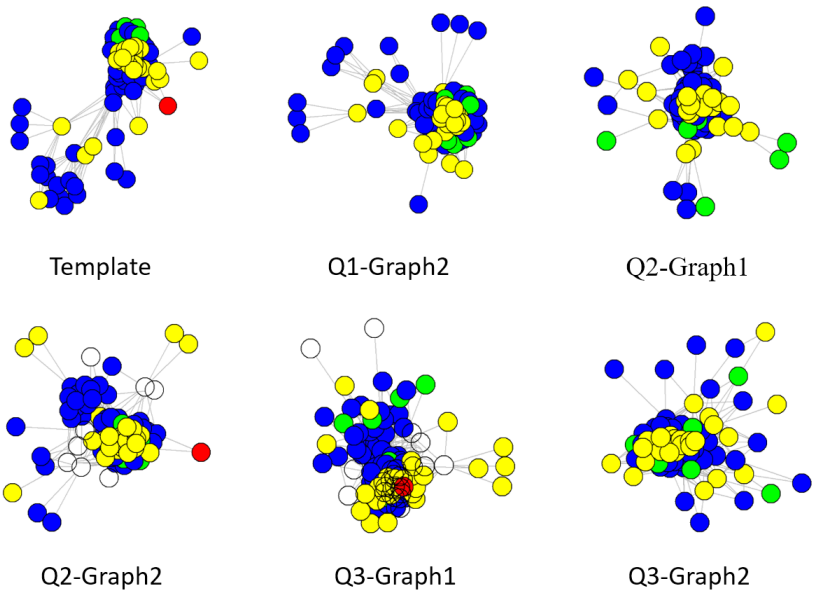



Figure 28: Network graphs with node size proportional to Eigen vector and node color representing the node type.

The interactive dashboard was used to compare the density curves for the sub-graphs and template, as seen in Figure 29.

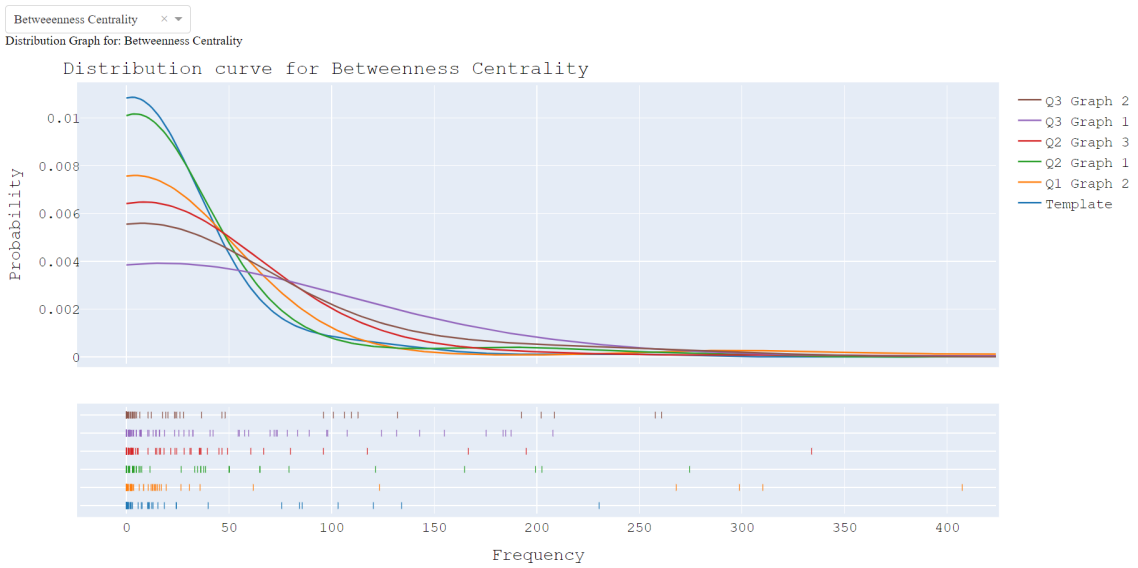



Figure 29: Comparison of distribution curves for Betweenness.

From all the analysis, it could be concluded that sub-graph 1 from Task 2, that

is the sub-graph built from Seed 1 is the most similar to the template.

Figure 30 below shows the ***required hacker group that created the outage.***

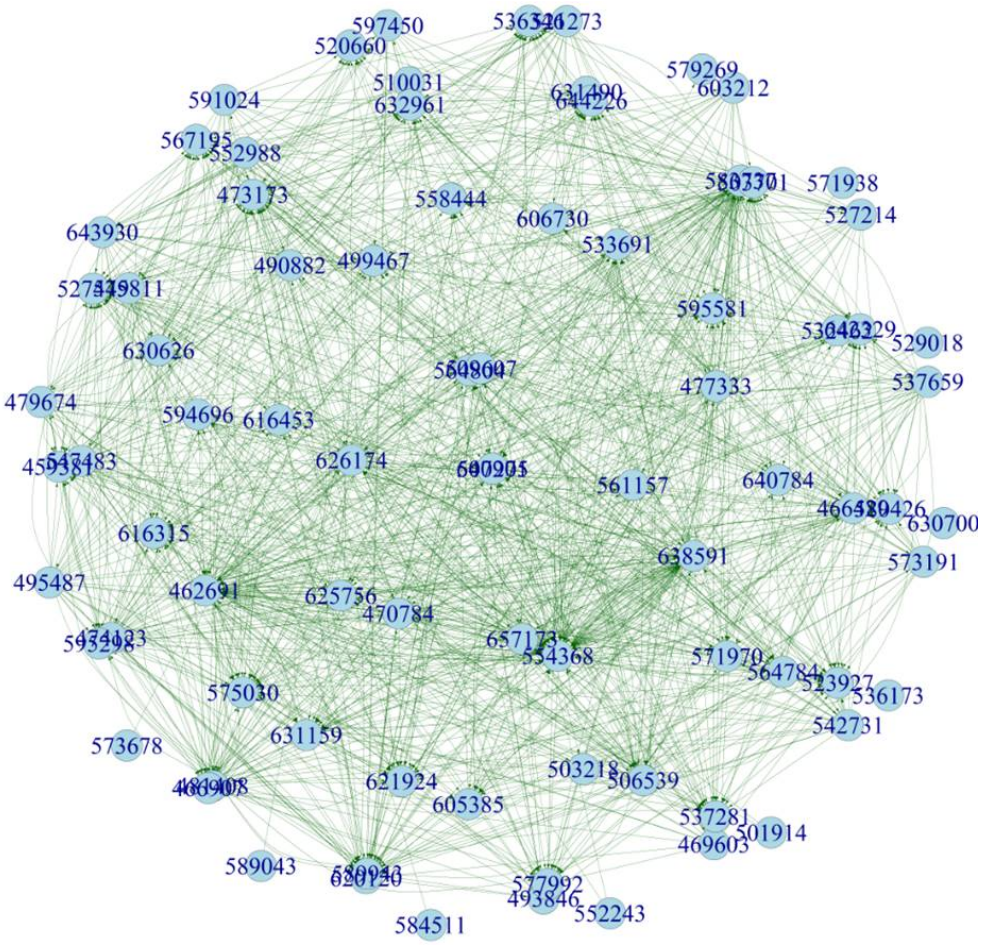



Figure 30: The potential hacker network to have created the outage.

## 5.5 Task 5

Task 5 required us to state the greatest challenges that were faced during solving these tasks and the solutions that we found for those.

The difficulties faced with the large graph data were as follows:

1. Loading the entire dataset: Not many libraries are recognized which are able to load this huge amount of data efficiently.

2. Memory: The dataset was too large to store in the RAM and needed to be saved in the hard-drive which costed computational time.

3. Computational Complexity: The dataset was too large and complex and required a large amount of computational time.

4. Complexity in Interpretation: The dataset was too large and complex to visualize or interpret in one go and needed to be divided into parts for better interpretability. Also, defining the whole dataset as a graph dataframe and finding sub-graphs or any other graph related characteristics was computationally impossible.

The solutions found by us to deal with these difficulties are:

1. R libraries like "fread" and "ff package" were able to load the large dataset quickly.

2. Using online tools like "Colab" helped optimize the use of RAM and gave us more computational power.

3. Dividing the dataset into Channels and working on the patterns, helped segment the data for a better interpretation.

The usage of larger RAMs and the libraries or tools which can do the computations in the GPU would have made these computations also much faster and easier.

# 6 Conclusion

The entire Mini-Challenge with all it's tasks was a very innovative way to test the various concepts of graph theory, visual analytics and data analysis. Each task required a different set of theoretical, intellectual as well as coding skills for it to be completed. Having completed each of the tasks, including the optional Task 3, it is our belief that we have acquired an in-depth knowledge about various methodologies to handle complex, interconnected data and a number of visualization tools and libraries to make our visualizations interactive, descriptive and sophisticated.

Although our conclusion that sub-graphs 1 and 2 were closest to the template was correct, it was sub-graph 1 which was the most similar to the template, contrary to our findings of sub-graph 2 being the closest. We could have reached the correct conclusion by giving more importance to the visualisation based techniques rather than the similarity measures. The Wasserstein metric and the similarity measures were better suited for probabilistic graph comparisons whereas our graphs were deterministic. Instead, a more detailed interaction with the graphs through visualization could have led us towards the correct conclusion effectively.

In hindsight, we have not only learnt how to work with very large and complex data structures, but we have also learnt the process of exploratory analysis of these complex, interconnected network datasets and to derive conclusions based on a data driven and visualisation based approach.

# References

[1] *ARC DIAGRAM*. URL: https://www.data-to-viz.com/graph/arc.html.

[2] Sven Bachthaler and Daniel Weiskopf. "Continuous scatterplots". In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1428–1435.

[3] David A Bader et al. "Approximating betweenness centrality". In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer. 2007, pp. 124–137.

[4] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. "Gephi: an open source software for exploring and manipulating networks." In: *Icwsm* 8.2009 (2009), pp. 361–362.

[5] Ekaba Bisong. "Google Colaboratory". In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 2019, pp. 59–64.

[6] Phillip Bonacich. "Some unique properties of eigenvector centrality". In: *Social networks* 29.4 (2007), pp. 555–564.

[7] Ulrik Brandes, Linton C. Freeman, and Dorothea Wagner. "Social networks". In: *Handbook of graph drawing and visualization*. Ed. by Roberto Tamassia. London: Chapman  Hall, 2010, pp. 805–839. ISBN: 978-1-58488-412-5.

[8] Brian Fisher. "Illuminating the Path: An RD Agenda for Visual Analytics". In: Jan. 2005, pp. 69–104. ISBN: 0769523234.

[9] Julian Heinrich and Daniel Weiskopf. "Continuous parallel coordinates". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1531–1538.

[10] Jimmy Johansson et al. "Perceiving patterns in parallel coordinates: determining thresholds for identification of relationships". In: *Information Visualization* 7.2 (2008), pp. 152–162.

[11] Daniel Keim et al. "Visual Analytics: Definition, Process, and Challenges". In: (Mar. 2008). DOI: 10.1007/978-3-540-70956-5_7.

[12] Till Nagel and Erik Duval. "A visual survey of arc diagrams". In: *IEEE Visualization*. 2013.

[13] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. "Ranking of closeness centrality for large-scale social networks". In: *International workshop on frontiers in algorithmics*. Springer. 2008, pp. 186–195.

[14] Edward L Platt. *Network Science with Python and NetworkX Quick Start Guide: Explore and Visualize Network Data Effectively*. Packt Publishing Ltd, 2019.

[15] *R library ff: Memory-Efficient Storage of Large Data on Disk and Fast Access Functions*. URL: https://cran.r-project.org/web/packages/ff/index.html.

[16] Filippo Santambrogio. "Optimal Transport for Applied Mathematicians. Calculus of Variations, PDEs and Modeling". In: (2015). URL: https://www.math.u-psud.fr/~filippo/OTAM-cvgmt.pdf.

[17] Mohammed Zuhair Al-Taie and Seifedine Kadry. *Python for graph and network analysis*. Springer, 2017.

[18] Matteo Togninalli et al. "Wasserstein Weisfeiler–Lehman Graph Kernels". In: *Advances in Neural Information Processing Systems 32 (NeurIPS)*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 6436–6446.

[19] *Two Sample Testing Based on The 2-Wasserstein Distance*. URL: https://www.bioconductor.org/packages/devel/bioc/vignettes/waddR/inst/doc/wasserstein_test.html.

[20] *VAST Challenge data description*. URL: https://vast-challenge.github.io/2020/MC1.html.

[21] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[22] Dmitry Zinoviev. *Complex network analysis in Python: Recognize-construct-visualize-analyze-interpret*. Pragmatic Bookshelf, 2018.