Towards predicting Harmful Conspiracies through Phase Transitions in Complex Interaction Networks

A Computational Study of the 5G and COVID-19 Misinformation Event

Kaspara Skovli Gåsvær

Thesis submitted for the degree of
Master in Computational Science: Physics
60 credits

Department of Physics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Spring 2022

Towards predicting Harmful Conspiracies through Phase Transitions in Complex Interaction Networks

A Computational Study of the 5G and COVID-19 Misinformation Event

Kaspara Skovli Gåsvær

Towards predicting Harmful Conspiracies through Phase Transitions in Complex Interaction Networks

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

In this thesis, we study the spread of content related to a conspiracy theory with harmful consequences, a so-called Digital Wildfire (DW). We aim to identify drivers, in complex temporal interaction networks underlying Twitter user activity, to model these phenomena as phase transitions. Furthermore, we investigate the component of *The 5G and COVID-19 Misinformation Event* that progressed on Twitter in the first half of 2020. The 5G and COVID-19 Misinformation Event is the term adopted for all communications surrounding the alleged connection between the 5G-network and the COVID-19 pandemic and all the real-world implications and consequences that followed. The main goal of the thesis is to lay the foundation for the development of methods that enable us to predict misinformation with the potential of causing harmful consequences. To the best of our knowledge, this thesis is the first attempt at modeling DWs in online social networks (OSNs) as phase transitions.

The main finding of the thesis is the identification of characteristics in the dynamics of the communication underlying the DW showing similarities to phase transitions. Furthermore, we identify candidates for the driving forces of the observed transition, namely influential users. The results show a nearly perfect overlap between the vertex with the highest degree centrality and the largest cluster in our network, as well as a minor group of vertices ($< 4\%$ of the population) with high degree centrality, at times, being inbound to over half of the edges. These findings suggest that only a few influential users are crucial in driving the conversation on a large scale, i.e., drawing a significant amount of new users to the conversation.

Through three community detection algorithms, Leiden [1], Louvain [2], and Label Propagation [3], we can conclude the existence of more than one significantly large conversation cluster. Moreover, the largest conversation cluster at the beginning of the DW does not stay the largest over time. Thus, we find evidence for the DW extending from multiple significant origins. Furthermore, while tracking, we observe oscillations in the largest clusters, where two or more clusters go back and forth between being the largest. Towards the peak on Twitter, we observe an increase in the fraction of the vertices belonging to the top 10% largest clusters, indicating a centralization of the overall discourse. The sum of the observations pointed out in this paragraph indicates that DWs begin from multiple origins of misinformation narratives that more and more become unified towards the peak of the DW. This centralization process is an exciting candidate for an early indicator of misinformation spreading with the potential of becoming a DW.

Contents

1 Introduction **1**
- 1.1 Research Questions and Challenges . . . 4
- 1.2 Contributions . . . 5
- 1.3 Limitations . . . 6
- 1.4 Outline . . . 7

2 Background **11**
- 2.1 Types and Subtypes of Misinformation . . . 12
- 2.2 The 5G and COVID-19 Misinformation Event . . . 14
- 2.3 Networks . . . 16
- 2.4 Twitter: an Overview . . . 18
- 2.5 Centrality Measures . . . 20
- 2.6 Degree Distribution & Assortativity . . . 22
- 2.7 Community Detection in Networks . . . 24
  - 2.7.1 Modularity-based Algorithms . . . 25
  - 2.7.2 Label Propagation Algorithm . . . 29
- 2.8 Phase Transitions . . . 30

3 Approach **33**
- 3.1 The Data Base & Extracting Temporal Data . . . 34
- 3.2 Graph Building . . . 37
- 3.3 Extracting Slices . . . 38
  - 3.3.1 Defining a Slice . . . 38
  - 3.3.2 Accumulative Slices . . . 38
  - 3.3.3 Temporal Slices . . . 39
  - 3.3.4 Contact Slices . . . 40
- 3.4 On Vertex Activity . . . 42

4 Experimentation, Results & Analysis **43**
- 4.1 Parameter Configuration for Slices . . . 46
- 4.2 Preliminary Analysis . . . 48
  - 4.2.1 Data Exploration . . . 48
- 4.3 Cluster Detection . . . 52
  - 4.3.1 Vertices in the Largest Clusters . . . 53
- 4.4 Cluster Tracking . . . 59
  - 4.4.1 Tracking the Path of the Largest Cluster Across Slices 59
- 4.5 A Closer Look at Temporal Slices . . . 64

4.5.1 Average Nearest Neighbour Degree 69
4.6 Summarizing Discussion 76

5 Conclusion **79**
5.1 Future Work 82

List of Figures

2.1 Information & Subtypes 12
2.2 *5G Misinformation Event* Timeline 15
2.3 Types of graphs 17
2.4 Simple example graph 21
2.5 Louvain method 28

3.1 Building the dataset (part 1) 35
3.2 Building the dataset (part 2) 37
3.3 Illustration of accumulative slices 39
3.4 Illustration of temporal slices 39

4.1 The distribution of node activity in $G_{\downarrow}$ 47
4.2 The distribution of node activity in $G_{\downarrow}$ 47
4.3 Vertex and edge distribution for accumulative and temporal slices 49
4.4 Vertex and edge distribution for contact-based slices 50
4.5 Vertex and edge distribution for temporal slices (Δt = 1)(Timeline) 51
4.6 Cluster size distribution (Accumulative slices) 52
4.7 Cluster size distribution (Temporal slices) 53
4.8 Cluster size distribution (Contact slices) 54
4.9 Cluster size binned-distribution (Accumulative slices) 55
4.10 Cluster size binned-distribution (Temporal slices) 56
4.11 Cluster size binned-distribution (Contact slices) 56
4.12 Size of largest cluster (Accumulative slices) 57
4.13 Size of largest cluster (Temporal slices) 57
4.14 Size of largest cluster (Contact slices) 58
4.15 Size of largest cluster (Contact slices) 58
4.16 Path of largest cluster (Accumulative slices) 62
4.17 Path of largest cluster (Contact slices) 63
4.18 Vertex and edge distribution for temporal slices (Δt = 4h) (Timeline) 65
4.19 Vertex and edge distribution for temporal slices (Δt = 3d) (Timeline) 66
4.20 Activity of most active vertices ($\Delta t = 4$h) 67
4.23 Most active vertex vs. largest cluster ($\Delta t = 4$h) 67
4.21 Activity of most active vertices ($\Delta t = 1$d) 68
4.24 Most active vertex vs. largest cluster ($\Delta t = 1$d) 68

4.22 Activity of most active vertices ($\Delta t = 3$d) 69
4.25 Most active vertex vs. largest cluster ($\Delta t = 3$d) 69
4.26 Average nearest neighbour degree (all slices, $\Delta t = 4$h) 70
4.27 Average nearest neighbour degree (all slices, $\Delta t = 1$d) 71
4.28 Average nearest neighbour degree (all slices, $\Delta t = 3$d) 72
4.29 Average nearest neighbour degree (phases, $\Delta t = 4$h) 73
4.30 Average nearest neighbour degree (phases, $\Delta t = 1$d) 74
4.31 Average nearest neighbour degree (phases, $\Delta t = 3$d) 75

List of Tables

2.1 Types of Twitter statuses 18
2.2 Attributes to Twitter statuses 18

3.1 Timeline of the data collection. 34
3.2 Twitter keywords for query 36

List of Acronyms

DW Digital Wildfire

OSN online social network

ANND average nearest neighbour degree

Acknowledgments

Just as with the years spent at UiO during my bachelor's in physics, this master's degree has offered a lot of ups and downs, a steep learning curve, and a large pinch of self-realization. Overall, I have had the great pleasure of learning a ton of interesting stuff and developing as a physicist and scholar. However, what I am most grateful for is the multitude of kind, intelligent and *special* people I have met and gotten to know over the past five years. I would like to extend a big thank you to my supervisors, Morten, Johannes and Daniel, for providing excellent advice and help along the way. Especially Daniel, your dedication to this thesis and motivation have been priceless, so thank you. Another thank you goes out to all the people close to me, my family and friends; you know who you are. I would not have been able to finish this degree if it weren't for your love and support. You are all irreplaceable. Thank you to the reviewers, Prof.Pedro Lind and Prof.Torsten Bringmann, I'm looking forward to hearing your thoughts on the thesis. Hagen Echzell, thank you for the great suggestions and tips you provided during the final iterations. An honourable mention goes out to Battery and Epok for fueling my caffeine- and nicotine addiction, they saved me during long evenings at the office (sorry, but who has the time or energy to deal with bad habits during a master's degree?).

Now, on to less sentimental matters. Looking back, the biggest challenge I faced as a native Norwegian speaker when writing this thesis was having to tell myself a quintillion times that there is always a comma in front of "men" (if you know, you know), but not always in front of "but" (oops!..I did it again?). To summarise my overall experience, writing a master thesis goes like this:

1. You **think** you know what you are doing.

2. It begins to dawn on you that you do **not**, in fact, know what you are doing.

3. You start doing the things you just now figured out you **should** be doing.

4. You start to **understand** the things you are doing (Maximum levels of confidence and joy).

5. You realize how little you actually understood a few months ago, leading to **a lot** of re-writing sections.

6. You have 1 week left and much **more** than 1 week of stuff to do (Maximum levels of stress).

7. You **transcend** the laws of physics and manage to finish anyways.

The takeaway here obviously being *Think not, should understand a lot more. Transcend.* At least it was for me. So long, and thanks for all the fish!

Chapter 1

Introduction

Before the birth of the internet, people relied on newspapers and radio as their main sources of news and facts. Back then, information was mainly broadcast with a clear separation between source and consumer; the flow of information was linear and slow, and it was common practice to trust professionals when seeking counsel.

On January 1, 1983, the internet was born [4], a service that, in time, would revolutionize the way we transmit and receive information. In its early days, the internet was dominantly populated by users with backgrounds in the STEM-fields. Their discussions did not yet reach a group of consumers larger than those of the traditional broadcasting sources. The big change came in the late 1990s with the launch of the first OSNs which suddenly allowed access to people from all different backgrounds. Now, since anyone can post on the internet, a strict distinction between source and consumer is no longer evident. But without this distinction, can we still trust the sources?

The reliability of today's news agencies can be debated as it varies very much from country to country and agency to agency. However, they are generally held to a higher level of accountability than content posted on social media accounts, where the limit for what is allowed to be posted is usually only drawn by the freedom of speech. In the rare cases when content posted in OSNs is fact-checked by independent fact-checkers, it is only after posts have already reached potentially large groups of consumers.

However, the biggest problem is the extreme amount of data available online. In 2018, about 2.5 quintillion bytes of data was produced on a daily basis around the world [5], a number which has more than likely only grown since then. With this amount of information, it is impossible to manually fact-check; thus, most misinformation spreads unchecked, especially on social media. In short, the fact that (1) anyone, regardless of their qualifications, can post about anything online, (2) the resulting sheer amount of unchecked misinformation, and (3) the lack of accountability imposed on the providers of OSNs and their users, turns the sea of available information online into a maze; tricky to navigate even if aware of these challenges, and potentially dangerous if not.

Another significant change brought by the rise of the internet is the

increased speed at which information travels across national borders. A post on an OSN is instantly visible to a global audience, possibly resulting in severe real-world implications [6–9].

When the spread of misinformation leads to destructive consequences, who is to blame and what can we do about it? This has recently been a hot topic, with the European Union discussing making OSNs accountable for the consequences of content published on their platforms [10]. The problem is still that no matter how many people are hired to maintain the integrity of the content, they cannot keep up with the vast amount of information being published. Thus, the risk of DWs spreading online stays present. According to the World Economic Forum [11], a DW is a fast-spreading misinformation that leads to real-world harm. To tackle the pressing issue of DWs we need to discover ways of taking advantage of automated systems to understand how these phenomena occur and how to stop them before they lead to real-world implications.

Natural language processing is a class of automated systems used widely to classify suspicious content automatically, where an established strategy is to create manually labeled training sets. Even though this approach significantly reduces the required amount of manual labour, machine learning models lack an understanding of context. Thus, features like humour or irony may not be taken into account, leading to miss-classifications. This is especially true when considering social media posts, like tweets, that tend to be brief and thus provide little context.

Due to these shortcomings, there is considerable motivation to explore other, more general automatic detection methods. **In this thesis, we aim for a more generic approach exploiting not only the content but rather the underlying interactions within OSNs to gain knowledge about the properties and dynamics of the spread of misinformation with harmful consequences on a societal scale.** Specifically, we investigate the evolution of the temporal networks induced by the interactions between Twitter users during a misinformation event.

The particular temporal network we study originates from contacts between Twitter users connected to *the 5G and COVID-19 Misinformation Event* [12], a series of tweets claiming a link between the COVID-19 virus and 5G technology that lead to a DW. This DW reached its peak around April 2020, resulting in the destruction of 5G-related telecommunication equipment and the harassment of technicians. Only after these real-world consequences occurred did Twitter realize the threat connected to the DW and placed a ban on tweets promoting attacks on 5G-equipment (see Section 2.2). The 5G and COVID-19 Misinformation Event is a good example of how difficult it is to identify DWs before real-world consequences occur, and the magnitude of the consequences expresses how important it is that we find methods to identify them in their early stages.

We examine the temporal evolution of the interaction network, i.e., the spreading of misinformation related to the event. On Twitter, individual information cascades exist in the form of tweet threads. Realizing that some DWs consist of a multitude of information cascades, we investigate the evolution of the 5G and COVID-19 misinformation event by looking at the

entire set of related cascades simultaneously. We begin by using methods from Complex Network Theory such as community detection [1, 2, 13] to investigate the dynamics of the network/spread of the DW. Moreover, we look into the centrality and activity of the vertices to identify the impact both group- and individual- activity has on the temporal evolution of the network.

This thesis aims to study dynamics to describe and predict the evolution of complex temporal networks related to the spread of misinformation. We propose that this knowledge can be applied to tracking misinformation events so that they can be identified early on and stopped before they turn into DWs with real-world consequences. Specifically, aiding human moderators in identifying the groups or conversations that have the highest probabilities of causing DW in the future.

We found that the DW we look at in this thesis displays behaviours similar to phase transitions. Moreover, we observed tendencies for centralization of the conversations toward the peak of the DW and indications of influential users being viable candidates for drivers of the transition we observed.

The motivation behind looking at this problem from the view of a physicist lies in the advancements and contributions already made in the field of complex network theory by researchers with a background in physics. Albert-László Barabási [14–16], Mark Newman [13, 17–19], and Santo Fortunado [20] are great examples of physicists who have developed methods for analyzing complex networks or in other ways contributed to the progression of the field. Studying the temporal evolution of dynamical systems is strongly related to physics, so even if the network at hand is produced by social science data, the investigation into how it evolves is done through methods much closer to physics than social science. *Social Physics* [21] is a field that has emerged from problems like the one we have at hand. In short, it is the implementation of methods and models from physics and mathematics applied to problems in the world of social science. Interdisciplinary fields like this provide new insight into problems that can not be seen using methods from one field alone.

For the code developed during the course of this thesis, see the GitHub repository **KasparaGaasvaer/MasterThesis**.

1.1 Defining Research Questions and Challenges

This thesis aims to understand the spread of misinformation events with severe real-world consequences, the so-called DWs. Previous research has shown that investigating only the diffusion pattern of such kind of misinformation on an individual basis is not promising at all [22, 23]. Even more, it seems that we can only understand a DW when examining the entirety of information cascades associated to it [24]. Thus, the main research question of this thesis can be summarized as follows:

> **Given the interaction data of an entire digital wildfire from the online social network Twitter, can we explain its dynamics and temporal evolution on a societal scale by using complex and temporal networks?**

Given for this thesis is the data, more precisely, tweets, retweets, and replies, from the OSN Twitter related to the so-called *5G and COVID-19 misinformation event* (see Section 2.2), a DW that found its beginning in January 2021 and whose aftermath continues to this day [12].

From the main research question, we can derive the sub-research-questions and challenges, which we divide into the categories information extraction & pre-processing, modeling of social networks and temporal data, and analysis.

Information Extraction & Pre-processing

In this thesis, we work with a given dataset of billions of tweets containing keywords related to the COVID-19 pandemic. The first big challenge at hand is to handle this large amount of data in an efficient and organized manner. We are investigating the DW revolving around the connection between 5G network and COVID-19, and we expand on a framework of filtering the dataset for relevant data points connecting the two. From there the problem is to identify the data properties that are most essential to modeling communication-based user interaction on a large scale and make use of those to enrich our dataset through thread completion[1].

Modeling of Social Networks and Temporal Data

The next challenge is to derive graph representations from the tweets that can help us analyze the temporal evolution of the data and study the spread. A big question is; how do we bend and section our data to maximize the information available? Is there more to be gained by slicing based on time, events, or graph attributes, and how can we extract such series in general? From there, methods from complex network theory, such as community detection algorithms and vertex centrality scores, seem promising in describing the interactions in the graph. Thus, a challenge is to identify

[1] Thread completion in this context is to complete conversations in the filtered dataset through looking for data containing certain attributes in the original dataset and extracting parent-statuses from Twitter.

suitable methods from these categories, where the over all goal is to answer the question of whether or not it is possible to extract information about the DW from the evolution of communities or through the activity of central users.

Analysis

Finally, during the analysis of our results, the goal is to identify indicators of a DW going viral in the graph representation of the dataset. A challenge is to identify a natural way of dividing the DW into distinct phases, and if there exists a well-defined phase for the emergence of the DW, can we determine if the origins of a DW is a single source or multiple sources? Furthermore, we must identify means we can use to investigate the scale of which the activity of central users affects the total of active users contributing to the DW as well as how the dynamics of the communities within the graph affect the structure of the DW. A topic of interest here is looking for the existence of a critical cluster size where the growth velocity of the cluster substantially increases.

1.2 Contributions

This thesis proposes answers to the questions asked in the previous section. The answers and discussion of our results contribute to several fields ranging from Social Science to Complex Network Theory. In this section, we highlight the main contributions.

In the fields of Computational Social Science and Social Physics, we are the first ones that approach understanding the dynamics underlining a DW with the help of physics-based methods. In particular, we use community detection algorithms and centrality measures to identify possible drivers of the temporal evolution in interaction networks related to the spread of misinformation on a societal scale. As emphasized earlier, the real-world consequences of a DW are destructive, and identifying the drivers that lead to virality can allow for identification in the early stages and thus the opportunity to intervene. Furthermore, we introduce the idea of modeling a DW going viral as a phase transition in its underlying interactions.

Finally, we provide a set of candidates for drivers of the phase transition and the physical quantities that show unique behaviors around the transition. For our dataset, we show that a smaller partition of vertices is the main contributor to the temporal evolution of the network. We suggest a driver of the virality phase transition be the activity of central users, where the physical quantity where we observe a changed behavior is the increase in the total number of active nodes in the network. Through community detection, we find that the relative size of the 10% largest clusters grows as the total number of active vertices in the network grows. Moreover, we observed tendencies of centralization of the conversations towards the peak of the DW. However, we did not find sufficient indicators to attribute a substantial

change in the velocity of the growth of the largest cluster to some critical size of the largest cluster.

In the fields, Algorithms for Network Analysis and Complex Network Theory, we build large scale interaction networks based on a DW (see Section 2.2). To the best of our knowledge, this is not only the first data set of this kind, but this thesis also presents the first examination of interactions related to DWs on a societal scale.

Even though examining news spreading phenomena in interaction networks is not entirely new, we introduce a new method for temporal slicing interaction networks using the evolution of the size of the neighborhoods. More explicitly, we introduce the idea of collective memory in temporal interaction networks to track conversations of significant value across discrete temporal slices. In particular, this thesis presents the theory and results for a slicing method using a constant maximum neighbourhood size and introduces the theoretical background for performing such partitioning using a vertex-varying neighbourhood size.

Performing the entire set of community detection-related analyses with three different algorithms; Louvain [2], Leiden [1], and Label Propagation [3] gives us not only the opportunity to evaluate those for our specific use case but even identify Label Propagation as the best fit for similar problems. Given the relevance and increasing popularity of this topic, we assume this finding is valuable for future work, as it is a method for optimizing the communities found by unsupervised methods without the addition of external supervision.

Lastly, in the field of Data Science, we extend an existing framework for handling a large set of data containing information about user interaction on Twitter. We filter the given set of statuses to produce a smaller dataset containing only statuses relevant to the DW we are investigating, as well as enriching the filtered data through thread completion. We deemed the interactions through statuses as a more concrete indication of active interaction than the act of users following each other. For example, one user following another user that has posted a status connecting 5G to COVID-19 does not guarantee that the following user ever reads or interacts with the content in the post. This realization leads us to implement a method for modeling the interaction of users through an underlying network where the vertices are statuses, and the edges are what we define as a *contact* (see Section 3.1) between two statuses.

1.3 Limitations

The main limitation of the thesis is the uniqueness of the dataset. DWs are very hard to identify while they are developing and often not recognized before the real-world consequences are revealed, so catching a DW and gathering information about it while it is happening is difficult. To the best of our knowledge, this dataset is the first of its kind. The consequence is

that it is impossible to generalize or confirm our statements by applying our methods on different datasets. As the objective of this thesis is to provide a fundamental basis for further exploration into the modeling and predictions of DWs, generalizability is not our goal for the time being.

The main problem lies in the confirmation or rejection of our hypotheses. There is no way of knowing if the tendencies we are observing in our data are simply behaviours exclusive to this particular DW or if they point to universal behaviors of DWs in general. Another shortcoming is the possible incompleteness of the data. During the enrichment and thread completion process, it was not possible to find all "child"-statuses not containing the queried keywords. This means that not all our chains of conversations are necessarily complete.

Another limitation comes from the new method we developed for slicing the graph after contacts. We can not find other published research that has done a similar slicing after the neighbourhood size procedure, so it is impossible to compare our slices' characteristics to other work. This limits the understanding we have at this point about how to interpret the slices.

We only look into the largest clusters of the network and into the activity of the most active nodes, which is a small search area, so there is a possibility of other drivers than the ones we propose in this paper. This, as well as catching other DWs to expand the set of available data, will have to be the subject of future work on the problem.

We work with an undirected graph representation of the Twitter interaction network underlying the DW. This limits our ability to differentiate between sources and targets and forces us to use other attributes of the network and former experience with the dynamics of conversations on Twitter to make assumptions about the direction of the links between the vertices. The reason behind choosing to work with an undirected graph is that we always want to start exploring new ideas in the most general manner, adding levels of complexity in parallel with the gain of knowledge about the subject and slowly reaching more specific descriptions.

1.4 Outline

This thesis is further structured as follows.

Chapter II: Background and Related Work
We give a brief introduction to subjects like community detection algorithms, centrality measures, phase transitions and Twitter. After reading this chapter, the reader should have an understanding of

- What *The 5G and COVID-19 Misinformation Event* is.
- What separates misinformation from disinformation.
- What a Digital Wildfire is and why it can be so dangerous.
- What a network/graph is and what separates complex graphs like social networks from random networks.

- What Twitter is and what type of data we can acquire from the Twitter API.
- What community detection is, and how the methods selected for this thesis work.
- The difference between degree-, betweenness-, and closeness centrality.

In short, this chapter aims to provide the reader with the necessary information and knowledge needed to understand the dataset and the relevance of the research questions investigated in this paper.

Chapter III: Approach
In this chapter, we present original frameworks, the development of our methods and models, as well as the mathematical formulations of said methods. After reading this chapter, the reader should have an understanding of

- The data-extraction process resulting in the dataset.
- The definition of a contact.
- The development of the underlying graph.
- The extraction of the different types of graph-slices.
- The difference in the networks produced by the three slicing methods.
- The definition of vertex activity.

This chapter provides a walk-through of how we apply methods to investigate the problem and how the technique we developed for contact-based slicing works. Thus, providing the information necessary for reproducing the methods and for the reader to interpret the results.

Chapter IV: Experiments
In this chapter, we present the experiments done on the different sets of slices produced. This includes metrics of the communities, vertex activity measures, the graphs' temporal development, and the correlation between the evolution of the DW as a whole and the evolution of the communities. After reading this chapter, the reader should have an understanding of

- The evolution of the DW in terms of the number of users and contacts.
- The outcome of the three community detection algorithms applied on the graph in terms of cluster distribution and the evolution of the largest clusters.
- What we deem the most plausible drivers of the DW in light of our results.

- What we argue is the preferred slicing method and community detection algorithm for our dataset.

This chapter should clarify our results and how we interpret them in light of the underlying theory and our hypotheses.

Chapter V: Conclusion
In the final chapter, we conclude the thesis by summarizing the main findings in light of our hypotheses and research questions. Furthermore, we dedicate a section to plans and ideas for future work predicting misinformation with real-world consequences. After reading this chapter, the reader should understand

- What our main findings and contributions are.
- The answers to our research questions.
- Our ideas for future work on the subject.

Chapter 2

Background

The Background chapter covers the information necessary for a reader to understand the experiments conducted in this thesis and their results. We begin by defining terms related to misinformation spreading on an online social network (OSN) and move on to introducing the Digital Wildfire (DW) that our dataset contains, namely *The 5G and COVID-19 Misinformation Event*. Moreover, we briefly introduce networks in general as well as some of the subclasses of networks. From there, we provide an overview of Twitter, the OSN we harvested our data from.

After this, we focus on the measures and methods we use to extract information about the network. We cover some centrality measures and the average nearest neighbour degree (ANND) before explaining the concept of community detection. In this thesis, we apply three methods for community detection; Leiden, Louvain, and Label Propagation. Lastly, we provide a general introduction to phase transitions and percolation transitions.

2.1 Types and Subtypes of Misinformation

In the age of the online society, the flow of information to the masses is no longer restricted to news outlets and scholars. Online social networks (OSNs) provide a stage for everyone to post nearly anything they want. Looking at only Twitter, in 2020 there were, on average, 500 million tweets posted daily [25]. With numbers of this magnitude, it is obvious that keeping track of the content available to readers is futile. A user posting in an OSN is both consumer and source, most likely not trained in fact-checking and often only interested in sharing his or her thoughts. When compared to the stream of information in the pre-internet age, the amount of content spread on OSN daily combined with the lack of credibility of the sources has led to a massive increase of false and biased information reaching the public. While terms like *misinformation* and *disinformation* have been around for a while, recently, new terms like fake news or digital wildfires have been used to describe inaccurate information. This section aims to define the most widely used terms related to the spread of false information.

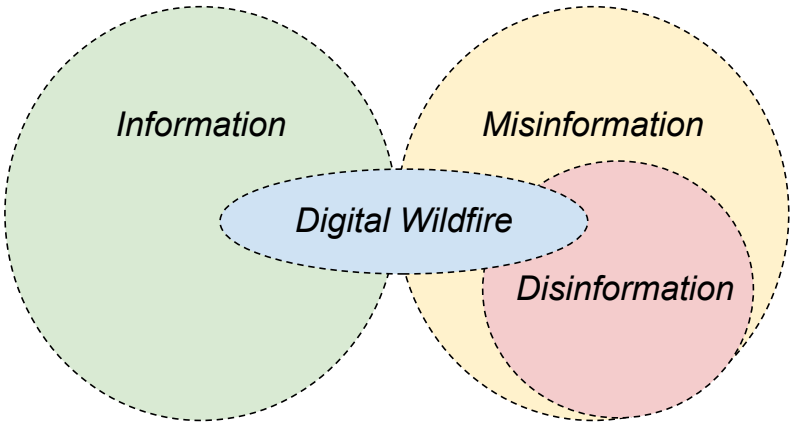

Figure 2.1: Illustration showing the relationship between information and the subtypes of misinformation. A digital wildfire can contain true information, misinformation, and disinformation.

The definitions of information and misinformation vary widely between sources and are often context-specific. We identified what we believe to be the most commonly accepted definitions of these terms by exploring sources such as Wikipedia and the Merriam-Webster dictionary, which are the ones we will be going by in this thesis. Figure 2.1 illustrates the relationship between information types according to our definitions.

Information

When referencing information, we go by the following definition

Definition 2.1.1 (Information)**.** The communication of true statements between sources.

Misinformation

We adopt the following definition of misinformation

Definition 2.1.2 (Misinformation)**.** Misinformation is false, inaccurate, or misleading information that is communicated regardless of an intention to deceive [26] .

The above definition relies on information being something true itself. This might not be the appropriate definition of information for all intents and purposes, but we go by the stated one for this paper.

Disinformation

Disinformation is closely tied to misinformation but differs from it by having the intention of deceiving [27]. We adopt the definition

Definition 2.1.3 (Disinformation)**.** Misinformation that is deliberately used to deceive.

Digital Wildfires

The term Digital Wildfire (DW) is used to describe the phenomena of rapid and uncontrolled spreading of online misinformation [11]. In this thesis, we define it more specifically as

Definition 2.1.4 (Digital Wildfires)**.** Social media events in which inaccurate, harmful or false content spreads rapidly and broadly, causing significant harm and real-world implications.

As displayed in Figure 2.1, a DW can contain all of the subtypes of information defined above. Some people believe in the inaccurate content they are spreading, while others spread with the intent of deceiving. A DW can also contain information that is not misinformation or disinformation. Imagine, for example, a situation where fact-checkers post clarifying facts as an attempt to stop the spread of a DW. The statement could potentially end up introducing a new audience to the DW. There is no guarantee that everyone in this audience believes in the fact-checkers, thus possibly leading to more people believing in the misinformation itself.

Fake News

There is no official definition of the term *fake news*, but usually, it is used to describe inaccurate information. In the article *The science of fake news* [28], David Lazer defines it as "Fabricated information that mimics news media content in form but not in organizational process or intent". The term fake news has long been up for high debate, especially since the presidency of Donald Trump. During this time, the term was thrown around very loosely and used to describe one man's opinions instead of objective facts. Therefore, we refrain from using the term "fake news" further in this thesis. Instead, we rely on more specific terms to define the types of information we encounter.

Conspiracy Theories

According to the Merriam-Webster dictionary (see Definition 2.1.5), *conspiracy theories* are centered around the idea that some people of power, usually the government or a large, influential organization, make up events or explanations about something to hide/cover up something else or derail the public from the truth.

Definition 2.1.5 (Conspiracy Theory)**.** A false theory that explains an event or set of circumstances as the result of a secret plot by usually powerful conspirators [29].

2.2 The 5G and COVID-19 Misinformation Event

As shown in previous research [30], soon after the COVID-19 outbreak in Wuhan, China, a series of tweets surfaced on Twitter containing insinuations of a possible link between COVID-19 and 5G wireless technology. The first tweet was posted in the early stage of the pandemic before the virus became an international problem. Initially, the conspiracy did not seem to gain much attraction. It took many weeks before real-world consequences occurred as a series of arson attacks on 5G towers in multiple countries, including the UK [31]. Such spreading of online misinformation leading to real-world implications is known as a Digital Wildfire (DW) (see Definition 2.1.4) and has been ranked as one of the top global risks in the 21 century by the World Economic Forum[11]. In this thesis, *The 5G and COVID-19 Misinformation Event* is used as an umbrella term for all communication surrounding the connection between the 5G-network and the COVID-19 pandemic and all real-world implications and consequences of the spreading of such information.

Before the Event

To pinpoint the start of the DW is difficult as there are multiple narratives, and we can only rely on the first tweet in our dataset mentioning both *5G* and *corona virus* as a definite starting point. However, there could be an earlier linking between 5G and COVID-19 on other online social network (OSN) or in real-life conversations.

When investigating the dataset, we came across an entire spectrum of conspiracy narratives claiming a causality between 5G radiation and the coronavirus. Even though these narratives seem to be as diverse as the individuals spreading them, they share the idea that the 5G technology is dangerous, can hurt people, and thus should not be implemented. In the following, we list a subsample of different conspiracies we came across

- *The 5G network, or more specifically the radiation, is weakening your immune system and so not directly making you sick but rather making you less equipped to deal with viral or bacterial infections.*

- *The radiation from 5G antennas is directly harmful, even deadly, and the government is using a made-up pandemic, COVID-19, as a cover-up.*
- *The vaccine for the COVID-19 virus is a microchip that enables the government to do mind control on people through 5G network devices.*

One can argue that even if it were possible to find the very first statement linking 5G and COVID-19, this would not necessarily mark the start of the DW. In fact, a later tweet might have been the true catalyst that turned the collection of these separate conspiracies into a DW. This line of thinking is speculative at best and thus not a white rabbit we continue following.

During & After the Event

Before the end of January 2020, 685 tweets and 1,081 retweets containing both keywords referencing COVID-19 and 5G had been posted on Twitter. Throughout January and February 2020, we observed a slow growth in daily tweets insinuating a connection between COVID-19 and 5G, as well as content related to the DW slowly beginning to gain some traction on other platforms such as YouTube. When the virus began to take a foothold in Europe in March, the global media coverage turned its focus to the pandemic, and the online spread of the DW picked up its pace. This resulted in four times as many tweets on the 5G-Corona conspiracy from late March to early April, which was the period right before arson attacks on 5G-related telecommunication equipment occurred and the harassment of technicians. The first series of attacks happened in the UK, the Netherlands and New Zealand during the weekend of April 3, 2020. Multiple more followed in the week after, and later some occurred in Canada as well. By July 2, 2020, there were reports of 273 cases of clashes between people who believed in some version of the conspiracy, as well as 121 reports on arson and other types of destruction [32], including the detainment of 8 telecommunication workers in Peru. In April of 2020, Twitter banned statuses and users promoting attacks on 5G infrastructure, and the spreading of content related to the connection seemed to halt. However, even as late as the first quarter of 2021, suspected cases of arson in Africa and Canada [33, 34] started to occur.

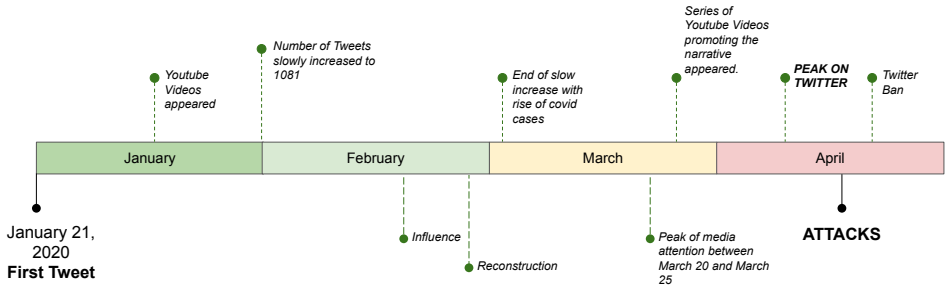

Figure 2.2: Most significant events that occurred during the course of the COVID-19 and 5G Misinformation event that transpired online in 2020.

2.3 Networks

A graph or network represents connections between a set of units, often denoted vertices, where the interrelations between the vertices are called edges. Formally, a *static* network G defined by a set of vertices V and edges E can be defined as in Definition 2.1.

$$G = (V, E) \text{ with } E \subseteq V \times V \tag{2.1}$$

In this thesis, we use networks mainly to model the contact between users within the OSN Twitter. Here, vertices represent users and have multiple attributes. The edges represent the connections, i.e, contact between two users through statuses and can like vertices have attributes such as being weighted or directed after type of interaction. The following are examples of different types of networks based on their edges where the vertices are twitter users [35]:

- **Weighted and Directed** - Edges points to/from users based on which one is following the other and are weighted after how many interactions (i.e, replies/quotes/retweets) there are between them.
- **Unweighted and Directed** - Edges points to/from users based on which one is following the other, but they are not weighted after number of interactions.
- **Weighted and Undirected** - Edges have no direction and only represents a link between users, i.e, there is some connection but it does not represent who is is following who, and are weighted after how many interactions (i.e, replies/quotes/retweets) there are between them.
- **Unweighted and Undirected** - Edges have no direction and only represents a link between users, i.e, there is some connection but it does not represent who is is following who, and they are not weighted after number of interactions.

All of the examples above are illustrated in Figure 2.3. The edges of a network are often encoded in an adjacency matrix E, where $E_{i,j} = e_{i,j} = 1$ means that there is an edge connecting vertex i to vertex j and $E_{i,j} = e_{i,j} = 0$ means there is not. For undirected graphs the adjacency matrix is symmetric.

Social Networks

Social networks are usually defined by their vertices representing individuals and their edges representing the connections in between them [36]. These networks are not simple networks with random traits and are therefore categorized as *complex networks*. Complex networks display characteristics not found in random networks and contain attributes such as community structures, being built up of a multitude of different subgraphs, dynamically evolving over time, and other non-random but not entirely systematic

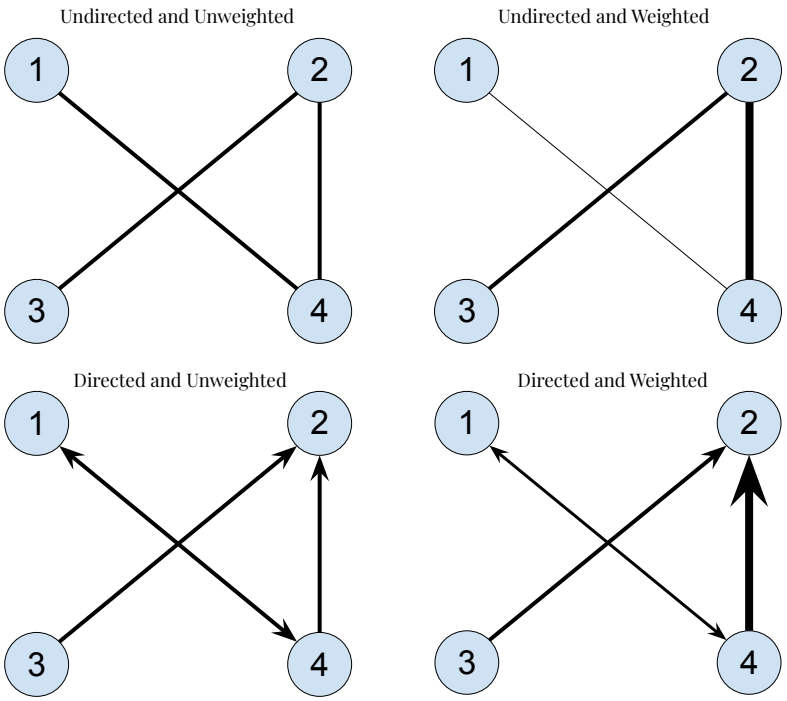

Figure 2.3: Illustration showing different types of graphs in terms of edge properties. The thickness of the edges represent weight and the arrowheads represent direction.

features [37, 38]. Examples of other network classes that fall under the category of complex networks are scale free [39] models and small-world [40] networks.

Temporal Networks

Temporal networks is a term used to describe a class of networks whose edges, in some way or another, vary with time [41]. Sometimes this means that you have edges that are always present in the network, but are turned on an off as a function of time. When they are off, information cannot travel from one vertex to another vertex across that edge. In networks that are fundamentally temporal themselves, like a social network where new users are continuously created and old users make new connections on a daily basis, new vertices and edges pop into existence over time. In other words, a temporal network can have static vertices and temporal edges, or both temporal vertices and temporal edges. Many types of networks fall under the category of temporal networks, ranging from biological networks, like a disease spreading in a population or the neural networks of the human brain, to technological networks, i.e., the Internet or computer clusters. Formally, we can define a temporal network with static vertices as

$$\mathcal{G} = (V_{\downarrow}, E_0, ..., E_T),$$

where E_i represents the subset of all edges turned on in the network at time i.

2.4 Twitter: an Overview

Twitter is an OSN that revolves around the sharing of status updates known as tweets. A tweet can contain up to 280 characters and may include images, videos and links, which essentially is a form of "micro-blogging". There are two privacy settings on Twitter, public and protected. If a user's privacy settings are set to public, their tweets can be viewed by anyone, even by people who do not have Twitter users themselves. If it is protected, only accepted followers can see content produced by the corresponding user. It is important to know that there are several ways of sharing information on Twitter (see Table 2.1).

Table 2.1: This table shows an overview of the most important types of Twitter statuses.

Tweet	A new post
Reply	A comment to an already existing status
Retweet	Re-sharing an already existing status
Quote	A retweet with either additional comments or modifications from the original

All of the types listed in table 2.1 are recognized as their own individual tweet objects and are collectively referenced as *statuses* in this paper. It is worth mentioning that users can also "like" another user's status, but for the remainder of this thesis, a *like* will not be regarded as a contact (see Section 3.1) between users. In more detail, a status contains several attributes [42] where the most important related to our research are shown in Table 2.2

Table 2.2: Examples of the most central attributes to Twitter statuses.

id	Unique integer identifier of the tweet
text	The text content of the tweet
user	The user in its entirety
retweeted_status	Representation of parent tweet

All reply-, retweet-, and quote-objects contain the **id** of the status they are referencing. This attribute makes it possible to find the parent status of a retweet, quote, etc. However, child statuses, i.e., statuses posted later than the status in question, are not attainable.

Accessing Twitter's Data

Due to Twitter's terms of service, we are not allowed to obtain data directly from Twitter's web interface. Instead, we use Twitter's Developer-API[1] for data access. The data collection related to the COVID-19 pandemic was done during the UMOD project[2] at the Simula Research Laboratory. Since the amount of data that can be collected with the help of Twitter's API is limited, at least when used conventionally, we applied and extended particular strategies initially developed during the UMOD project for our data collection [43–45]. This approach allowed data collection in sufficient amounts [22, 46].

When searching for tweets containing specific keywords, Twitter's Search-API only allows access to tweets not older than two weeks. By the very nature of a DW, this circumstance makes it difficult to obtain the required data as real-world harm might only be evident after the corresponding misinformation event has developed for more than two weeks. The simple solution to this problem is a proverbial search for a needle in a haystack, in which the haystack takes the form of a massive database that has to be built beforehand. In our case, this resulted in a massive amount of data related to COVID-19 in general, which "coincidentally" also contained data related to the COVID-19 & 5G DW (see Section 2.2).

Twitter's underlying Networks

Two main categories of networks are underlying Twitter's database. First, the network reflecting a user's decision to follow another user's content, and second, the network reflecting user interactions. In the following, we introduce both categories briefly.

Follower Network

The sum of all follower connections on Twitter makes up the Twitter follower network, in which vertices represent users and edges represent the act of users following each other. There can be several reasons for one user to start following another; they can, for example, be real-life friends, one can be a fan of another high-profile user, or they can have mutual interests which they would like to discuss. Another reason might be that Twitter suggests new users to follow through a recommender algorithm based on a user's interest, i.e., what users they are already following. Something which can influence the "validity" of the connections is users who follow as many other random users as possible to gain followers.

Interaction Networks

Interaction networks are built from the interactions between users, like retweets, quotes, and replies (see Table 2.1). Whereas in follower networks,

[1] At the time of data collection, Twitter had not released its API v2, so v1 was used. Twitter search API: https://bit.ly/3KOGqRS

[2] UMOD: Understanding and Monitoring Digital Wildfires.

the edges represent a user's intention to subscribe to the entirety of another user's content, in interaction networks, they represent interactions. The vertices still represent users. The act of interacting allows for the assumption that one user has in one way or another been affected by the other user's status through its content. This can not be assumed by a simple follow in the follower network, as it does not indicate what specific content reaches the other user. The confirmation of interaction through content related to the DWs is the reason why we, in this thesis, use interaction networks and not the follower networks. Looking at the follower network of each user that has contributed to the conversation about the 5G-COVID-19 connection can provide other types of information about the contributors, but that is beyond the scope of this thesis.

2.5 Centrality Measures

We use metrics to quantify the qualities of the networks created from the dataset, some of which are centrality measures. Centrality measures provide insight into the significance of individual vertices in terms of network placement, i.e., the network topology. The position of a vertex says a great deal about the amount of information that potentially flows from, to or through a vertex and is, therefore, an indication of the potential "power" a vertex holds in the network.

Degree Centrality

Degree centrality [47] is the most simple form of centrality measurement. It is simply the sum of all edges connected to a vertex and can be formulated mathematically as

$$\mathcal{C}_{\text{deg}}(i) = \sum_{i,j \in \mathcal{N}(i)} e_j, \tag{2.2}$$

where $\mathcal{N}(i)$ is the neighbourhood of vertex i (see Eq. 3.4 for formal definition of neighbourhood), or in terms of the adjacency matrix (see Section 2.3)

$$\mathcal{C}_{\text{deg}}(i) = \sum_{j=1}^{N} A_{i,j}, \tag{2.3}$$

where N is the number of vertices in the graph and $A_{i,j}$ is an element of the adjacency matrix A. Here, $A_{i,j}$ takes the value 1 if there exists an edge between vertices i and j and 0 if not. For directed networks, we can calculate the degree centrality for both inbound and outbound edges. Moreover, if the edges of the network are weighted, the degree centrality must also account for those. However, since we only consider undirected and unweighted networks for the remainder of this thesis, we present the degree centrality as the sum of all edges adjacent to a vertex.

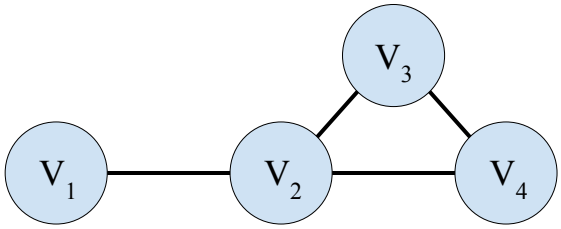

Figure 2.4: Simple example of an undirected/unweighted network with $N = 4$ vertices and $M = 4$ edges.

Betweenness Centrality

Another centrality measure considers how often a vertex lies on the shortest path between other vertices. For a vertex i on the shortest path between vertices j and k, any information traveling between j and k would first have to pass through i. This means that vertices with a high betweenness centrality, i.e., a vertex lying on many of the shortest paths between other pairs of vertices, will have a lot of information traveling through it. In many cases, this also means that removing such nodes will be highly disruptive to the flow of information in the network. There exist multiple variations of how betweenness centrality is defined. However, for this thesis, we will go off the definition proposed by Ulrik Brandes in the article *On variants of shortest-path betweenness centrality and their generic computation*[48]. We define the number of shortest paths (there can be more than one of equal length) between two vertices j and k as $\sigma(j,k)$ as well as a variable $\sigma(j,k|i)$ which is the number of occurrences where a vertex $i \neq j,k$ lies on a shortest path between them. If $j = k$ then the number of shortest paths between them are one, $\sigma(j,k) = 1$, but no other vertex can lie on that path so $\sigma(j,k|i) = 0$. We get the betweenness centrality of a vertex i by going over all other pairs of vertices and finding the number of times i lies on a shortest path between the pairs relative to the number of shortest paths between them.

$$\mathcal{C}_{\mathrm{B}}(i) = \sum_{j,k \in V} \frac{\sigma(j,k|i)}{\sigma(j,k)}. \tag{2.4}$$

Closeness Centrality

The closeness centrality of a vertex is the inverse of the sum of all distances to all other vertices in the network and can be calculated as

$$\mathcal{C}_{\mathrm{c}}(i) = \frac{1}{\sum_{j=1}^{N} d_{i,j}}. \tag{2.5}$$

We define distance $d_{i,j}$ as the shortest path between to vertices i, j in terms of number of edges. See Figure 2.4 for an example. Vertex v_2 has distances $d_{2,1} = d_{2,3} = d_{2,4} = 1$ which gives a closeness measure of $\mathcal{C}_{\mathrm{c}}(2) = \frac{1}{3}$. Note that there are two paths from v_2 to v_4, but the distance used in the closeness centrality measure is always the shortest.

In other definitions of closeness centrality, it is normalized by multiplying with $N-1$ to allow for comparing between graphs of different sizes. A high closeness score indicates that the vertex in question is closely related to the other vertices of the network, so the distance information would have to travel to another vertex is short. This could indicate that a vertex with a high $\mathcal{C}_{\text{c}}$ score can spread information more efficiently than one with a low $\mathcal{C}_{\text{c}}$ score.

2.6 Degree Distribution & Assortativity

When looking into how vertices connect in a network, we often find that vertices are more likely to connect to other vertices with similar attributes, something which is known as *network homophily* [49]. These attributes can be anything from age or gender to geographic location or if you are a Gryffindor or a Slytherin. Especially when concerned with social networks, where the vertices represent people interacting with each other, there can very well be factors not explicitly represented in the network structure that are part of the underlying reasons why the network is as it is. In other words, we look at the structure as a symptom we can use to work our way back to a potential cause.

Average Nearest Neighbour Degree

Average degree connectivity tells us something about the relationship between the vertices of the network who connect [50]. It can provide insight into how connected the neighbourhoods of vertices with certain degrees are. Do vertices with high degree centrality usually have neighbours with high of low degree centrality?

Let us begin by looking at a vertex i. The average neighbourhood size of the neighbours of vertex i can be calculated as

$$K(i) = \frac{1}{\mathcal{C}_{\text{deg}}(i)} \sum_{j \in \mathcal{N}(i)} \mathcal{C}_{\text{deg}}(j), \tag{2.6}$$

where $\mathcal{N}(i)$ is the neighbourhood of vertex i or by using the adjacency matrix as

$$K(i) = \frac{1}{\mathcal{C}_{\text{deg}}(i)} \sum_{j=1}^{N} A_{i,j} \mathcal{C}_{\text{deg}}(j). \tag{2.7}$$

This tells us something about how connected the vertices in the neighbourhood of vertex i are and is called the *average degree connectivity* of a node. What is more interesting is how well connected, on average, the neighbours of any vertex with a certain degree centrality $\mathcal{C}_{\text{deg}} = k$ are. Calculating $K(i)$ for all vertices in a network we can find the *average nearest neighbour degree (ANND)* [51] of all vertices with degree k as the mean of

$K(i)$ for all $\mathcal{C}_{\text{deg}}(i) = k$, that is to say we get a function $K(k)$

$$K(k) = \frac{1}{\left|\tilde{V}\right|} \sum_{i \in \tilde{V}} K(i), \quad \tilde{V} = \{v \,|\, \mathcal{C}_{\text{deg}}(v) = k\}. \tag{2.8}$$

This function is sometimes called the *degree correlation function* or *nearest neighbor degree* [50] and can be also be written in terms of probability

$$K(k) = \sum_{k'} k' P(k'|k), \tag{2.9}$$

where $P(k'|k)$ is the conditional probability that a vertex with degree $\mathcal{C}_{\text{deg}} = k$ has a vertex in its neighbourhood with degree $\mathcal{C}_{\text{deg}} = k'$ [50]. This function can be linearly fit to examine the correlation between degree centrality of a vertex and the degree centrality of its neighbours.

Assortativity

One attribute where network homophily does not necessarily apply is the degree of a vertex in relation to its average nearest neighbour degree. Assortativity is a measure defined the quantify a networks tendency of homophily [17]. A network is deemed assortative in terms of degree correlation if the ANND (see Eq. 2.8) grows with larger k, or in other words well connected vertices are more likely to be connected to other well connected vertices. For a disassortative network the average degree connectivity decreases with higher values of k, that is to say that well connected vertices are on average connected to sparsely connected vertices and visa versa. For neutral networks there are no correlation between the two [50]. This degree-degree correlation is the most common attribute to use when looking at the assortativity of a network, but one could in principle use any vertex attribute whom in turn could yield a very varying range of assortativity scores [52]. A study done on degree correlation in social networks report them as typically assortative in terms of degree correlation, while non-social networks show a tendency of being disassortative [53]. To determine whether a network is assortative or disassortative one can calculate the mean nearest neighbour degree for all vertices of degree k by looping through the network and calculating the corresponding ANND to each k, or as this is a measure of correlation between two variables, we can use the Pearson correlation coefficient. As calculating the corresponding ANND to each k provides us with a broader view of the networks tendencies than the single value given by the Pearson correlation coefficient, we will in this thesis be implementing the former.

2.7 Community Detection in Networks

Community detection might sound like something inherently social science, but it has shown to be of interest to natural sciences, including physics. Many different networks, ranging from natural to social, at some point divide into communities. In network theory, a community is usually characterized as a group of vertices sparsely connected to the rest of the network but densely connected to each other. However, there is no single agreed-upon definition in the field. In this thesis, we alternate between the terms community and cluster, as the latter is a more general expression for an arbitrary network partitioning. The very simplest form of a community is a "clique", where all pairs of vertices are directly connected through an edge so that all vertices in the community are directly related [3]. A popular method to identify and distinguish the clusters is by optimizing the modularity of the network. In short, modularity is a measure of how connected the clusters of a network are (see Definition 2.7.1), and by maximizing it, we can gain insight into the clusters of a network.

One of the downsides of traditional algorithms for modularity maximization is that they are computationally costly. One widely used method is *simulated annealing*, a method first introduced in 1983 by the computer scientist Scott Kirkpatrick [54]. This method is an adaptation of the Metropolis-Hastings algorithm for Monte Carlo sampling [55]. Simulated annealing becomes too slow for large-scale networks due to the computational cost. Thus, new methods are needed. In the paper *Modularity and community structure in networks* [19] the physicist M.E.J Newman describes a method that makes use of the eigenvectors of what he calls "the modularity matrix", which is the similarity matrix of the network minus the matrix where the elements are the expected number of edges between the vertices if the network was a random network. This approach reduces community detection to a spectral clustering problem instead, which for those familiar with linear algebra and eigenvalue problems, can be used to reduce the dimensionality of the problem before actually beginning with the clustering.

Another take on the problem comes from Liang Yang *et. al* in the paper *Modularity Based Community Detection with Deep Learning* [56] where the authors propose the use of a non-linear model in deep neural networks to build a new algorithm for cluster detection, optimized through the use of stochastic gradient descent.

What does all of this tell us about the relevance to the field of physics? Reading through this section, a physicist should have recognized many familiar terms. Monte Carlo simulations with Metropolis sampling is a widely used technique in thermodynamics and modeling of quantum mechanical systems and is often already introduced to undergraduate students of physics through the Ising Model [57]. Eigenvalue problems are common to many parts of physics, from finding the energy states of a system through the eigenvalues of the Hamiltonian in quantum mechanics to solving the partial differential equations used for buckling analyses of structures in classical physics. All methods and models mentioned above have been underlying factors in developing cluster detection methods, and physicists

often develop the methods themselves. For the eager reader, we recommend the paper *Community detection in graphs* [20] where physicist and professor in complex systems Santo Fortunato breaks down the concept of community detection in graphs "with a special focus on techniques designed by statistical physicists".

After reading this section, it should be clear that there exist many methods of community detection. In the following, we introduce the three different clustering methods implemented during this thesis, Louvain [2], Leiden [1], and Label Propagation [3]. The first two are modularity based, which is commonly agreed upon as the go-to measure for community detection, while Label Propagation is chosen as a safeguard.

2.7.1 Modularity-based Algorithms

When talking about cluster detection methods, an important term is *modularity*. In his paper *Modularity and community structures in networks* [19], M.E.J Newman defines modularity as

Definition 2.7.1 (Modularity)**.** The modularity is, up to a multiplicative constant, the number of edges falling within groups minus the expected number in an equivalent network with edges placed randomly.

There are multiple ways of calculating the modularity of a network. However, in this thesis, we present the method introduced in the book *Networks: An Introduction* by M.E.J Newman [18] as this is the method we implement. We remember that the number of edges connecting to a vertex i, or the degree centrality of the vertex $\mathcal{C}_{\text{deg}}(v_i) := d_i$, is calculated as in Eq. 2.3. Assume that we are working with an undirected, complex network G with N vertices, M edges and an adjacency matrix A, for which we would like to calculate the modularity. We introduce some grouping on the vertices of the network, either that all vertices belong to their own unique group or that they are grouped in any other way. We mark the vertices with a label c_k, determining which group they belong to, where $k \in \mathbb{N}^+$ with $k \leq N$ is the group index. Imagine an equivalent random network G_R, we look at vertices i and j which have corresponding degrees d_i and d_j. There are two ends to an edge, which means that for a network with M edges, there are $2M$ edge ends. Now, if we pick a random edge end incident to vertex i in G_R, the probability that the other end of the edge is connected to vertex j is the degree of vertex j divided by the number of total edge ends in the network, i.e.,

$$p = \frac{d_j}{2M}, \tag{2.10}$$

which applies for all $j \in [1, N]$. Since there are d_i possible edge ends connected to vertex i there are d_i ways of an edge connecting edge ends from i to j. This translates to the following expected value

$$\mathbb{E}[\text{Edges connecting } i, j] = \frac{d_i d_j}{2M}. \tag{2.11}$$

We expand on that by introducing $\delta(c_i, c_j)$, the Kronecker delta, which is 1 if vertices i and j belong to the same group and 0 if not, so that we can write the expected value of edges connecting i to j in case they belong to the same group as

$$\frac{d_i d_j}{2M}\delta(c_i, c_j). \tag{2.12}$$

Iterating over all the vertices, without counting edges twice, we arrive at the total number of expected edges between all pairs of vertices belonging to the same group in the network

$$\frac{1}{2}\sum_{i,j}\frac{d_i d_j}{2M}\delta(c_i, c_j). \tag{2.13}$$

Now, as we know the expected number of edges connecting pairs of vertices in the same group in G_R, we see from Definition 2.7.1 that we need to find the number of actual edges connecting vertices in the same group in G. Since we have the adjacency matrix A, which contains all edges in G, we can sum over all its elements, making sure to divide by 2 to avoid counting edges twice, while again imposing $\delta(c_i, c_j)$ to arrive at our expression

$$\frac{1}{2}\sum_{i,j}A_{i,j}\delta(c_i, c_j). \tag{2.14}$$

To find the difference between expected and actual edges we subtract eq.2.13 from 2.14 which results in

$$\frac{1}{2}\sum_{i,j}\left[A_{i,j} - \frac{d_i d_j}{2M}\right]\delta(c_i, c_j). \tag{2.15}$$

We scale the sum by the number of total edges in the network M to get the fraction of such internally connecting edges in groups

$$Q = \frac{1}{2M}\sum_{i,j}\left[A_{i,j} - \frac{d_i d_j}{2M}\right]\delta(c_i, c_j), \tag{2.16}$$

which is the final equation for the modularity Q of the network G. Suppose there are more edges between vertices of the same groups of G than expected from G_R. In that case, Q will be positive with a maximum value of 1 (all the edges of the network are between vertices belonging to the same group, so c_i is equal to c_j for all i, j when $A_{i,j} = 1$). If there are fewer edges between vertices of the same groups of G than expected from G_R, Q will be negative.

For example, one could imagine the network built up of all the students attending the University of Oslo. The students of physics, social sciences, theology, etc., at UiO have dense connections to other students in their own fields but sparse connections to students in other fields. If we rephrase the groups of students belonging to a particular field as a student cluster, this would likely indicate that the entire student network of UiO, built up of all its attending students, has a positive modularity score.

The Louvain Method

The Louvain method [2] is an unsupervised method for modularity optimization. It is unsupervised as one does not need to know the number of clusters nor the cluster sizes beforehand. Assume we have a network of N vertices. The first step of the Louvain method is to assign a cluster to each vertex so that we start off with N clusters in our network. We begin by removing a vertex i from one cluster and placing it in another cluster belonging to a neighbouring vertex j. Then, we calculate the modularity to see if our move has increased it. After looking at all possible neighbouring clusters, we move vertex i to the cluster that maximized the modularity. The cluster maximizing the modularity could be the cluster the vertex originated from; thus, staying put is an allowed move. We then repeat the process for all the vertices in the network until reaching a local maximum where moving a vertex no longer increases the modularity score. Phase one of the method is now concluded. In phase two, we construct a new network where the vertices represent the clusters found in phase one. In the new network, the edges between the vertices are weighted as the sum of the weight of all edges connecting the clusters from phase one. Edges linking vertices in the clusters are represented by self-loops in the new network. They are weighed as the sum of all internal edges in the clusters from phase one.

We now have a new network where we can reapply phase one. These two phases can be repeated until we find the best possible grouping of clusters to maximize the modularity of the network. In the experiments conducted in this thesis, we will only use phase one of the Louvain algorithm, not constructing new networks from the partitions we find. This is because we are interested in tracking the evolution of the communities across time through re-identifying their members, thus making the memory of the original vertices important.

The Louvain method is an example of *greedy optimization* of the modularity, which, compared to methods like simulated annealing or spectral optimization, has shown to provide less accuracy [20].

The Leiden Method

The Leiden method [1] is an improvement of the Louvain method. It was developed after experiments showed that the Louvain method could, at times, yield clusters that were poorly connected or, in other cases, was shown to disconnect clusters altogether. An example is that the Louvain algorithm allows for "bridging"-vertices, i.e., vertices that are central to the connection of vertices in its old cluster, to be moved into a new cluster and disconnect the old cluster. This unfortunate move can lead to other vertices from the old cluster being moved into new clusters and the original cluster being lost. As the goal is to maximize the modularity, when taking no extra precautions, the Louvain method tends to lose smaller clusters to larger ones, yielding clusters that have "eaten" smaller clusters that might contain useful information about the network.

The idea behind the Leiden algorithm is to ensure convergence to a

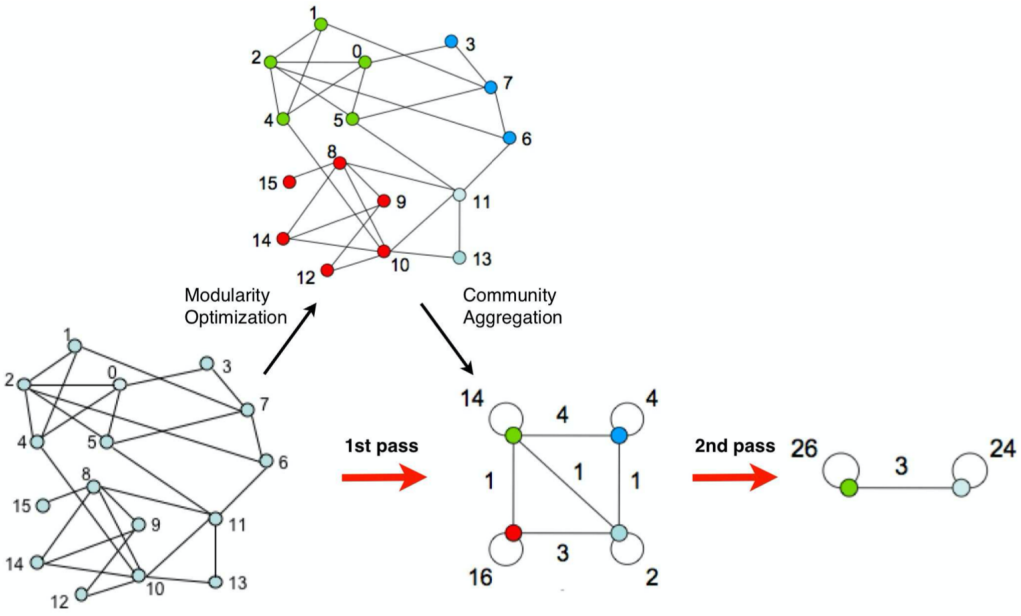

Figure 2.5: This figure illustrates the different phases of the Louvain method. During phase one, vertices are moved to new clusters to maximize modularity. Phase two constructs a new network from the clusters of phase one. Figure borrowed from *Fast unfolding of communities in large networks*, Vincent D Blondel et al. [2].

network where all clusters are locally optimized with regard to modularity. This is achieved by introducing an additional step to the Louvain method. We start similarly to the Louvain method, moving vertices to neighbouring clusters while calculating modularity. However, the difference is that the Leiden method uses a fast local move procedure.

In random order, all the vertices in the network are put in a queue. The vertex at the start of the queue is then removed, and the modularity score is evaluated for moving the vertex to any of its neighbouring clusters. After moving a vertex to the cluster with the greatest positive change in modularity, we put all the vertices whose neighbourhood changed to the end of the queue. After all vertices have been visited and removed from the queue once, only affected vertices remain in the queue. This streamlines phase one by avoiding unnecessary testing and moving vertices, making phase one faster than in the Louvain method.

Before initiating phase two, an additional phase is introduced to refine the partition P found during phase one. In this phase the goal is to identify a partition P_{refined} which is a refinement of P. Here, P_{refined} is initially set to be a network where all the vertices from the underlying network are clusters on their own. From here, the algorithm performs a version of phase on again. However, instead of assigning a vertex to the neighbouring cluster with the largest increase in modularity, it can be randomly assigned to any neighbouring cluster with an increase in modularity. The larger the increase,

the larger the probability, so it is still likely to be moved to the cluster with the highest increase in modularity, but not definitely. This often leads to clusters in P splitting into more clusters in P_{refined}. In other words, the refinement phase allows for a broader exploration of the partition space. As with the Louvain algorithm, we will not be initiating phase two of the Leiden algorithm. We will only make use of phase one and the refinement phase.

2.7.2 Label Propagation Algorithm

In this thesis, we also use a label propagation algorithm for community detection proposed in *Near linear time algorithm to detect community structures in large-scale networks* [3]. We again imagine a network consisting of N vertices and M edges. The first step is to initialize all vertices with a unique label signifying which community they belong to. At the beginning of iteration 1, there are N unique labels. Then, we start propagating through the network's vertices, letting each vertex take on the label most of its neighbours have. This updating can happen one of two ways

- *Synchronous updating:* A vertex at iteration t takes on the label most common among its neighbours at iteration $t-1$.
- *Asynchronous updating:* A vertex at iteration t takes on the label most common among its neighbours at iteration t, whereof some have already been updated in iteration t and some have not.

The challenge with synchronous updating is that if there exists bipartite[3] subgraphs in the network, the label updating can begin to oscillate. In other words, the vertices jump back and forth between two labels in subsequent iterations resulting in a loop. If the graph contains vertices of degree 1, i.e., with only one neighbour, which by definition is a bipartite subgraph, this immediately becomes a problem. As a consequence of this, asynchronous updating is used.

An important aspect of the algorithm is that the vertices are queued randomly for updating at each iteration. As the propagation goes on, there will form more and more groups of vertices that agree upon a label, expanding in members up until some point. The number of unique labels declines over the iterations until reaching the number of final communities. There is a possibility that vertices whose neighbours have two equally popular labels exist, resulting in the vertices bouncing back and forth between two communities forever. Thus, the breaking point of the algorithm cannot be when all vertices stop changing labels. Instead, the algorithm stops when all vertices have one of the labels belonging to the maximum of its neighbors. When the process is finished, the vertices are put into communities with the other vertices that share their label. All vertices sharing a unique label make up one community, and the final partitioning is then guaranteed to fulfill the following criteria

[3]Bipartite means that all the vertices can be divided into two independent and disjoint sets where all edges connect a vertex from set 1 to a vertex in set 2 [58].

- No single vertex can belong to more than one community (disjoint communities).
- A vertex with label L_i has more (or equally many) neighbours with label L_i than neighbours with label L_j for all $j \neq i$.

These communities are very similar to the definition of *strong communities* proposed in the paper *Defining and identifying communities in networks* [59]. The main difference is that this algorithm results in communities with the possibility of a vertex with label L_i having more or equally many neighbours with label L_i as neighbours with label L_j whereas for strong communities, a vertex with label L_i has strictly more neighbours with label L_i.

There are multiple solutions as nothing is being maximized or minimized to determine the algorithm's stopping point. An example is if one has two different solutions and relabels the vertices so that all vertices that had a label L_a in solution 1 and label L_b in solution 2 are given the same label L_c. Then one can repeat the iterative process over the labels, receiving yet another solution which can again be aggravated with the formerly aggravated solution so that the process can be repeated.

Note that as long as all vertices are initialized with unique labels at the beginning of the algorithm, the algorithm is unsupervised. Another noteworthy aspect of the algorithm is that the average time for each iteration goes as $\mathcal{O}(M)$, and for homogeneous networks (no community structures), the algorithm can result in one single community.

2.8 Phase Transitions

To put it simply, a phase transition is a change of state of some variable of a system. The most known phase transition happens in almost every household daily, the change in state of water from liquid to gas through boiling. When a sufficient amount of energy is added to the system, and the liquid water reaches its boiling point of 100 degrees Celsius, the water undergoes a non-continuous phase transition, or what is known as a first-order phase transition [60]. One might assume that it is continuous, as most have experienced that the entire pot of liquid water does not spontaneously turn to vapor but rather, over time, evaporates. The quality that determines if the transition is of a first-order or high-order (continuous) is the change of state. When a water molecule reaches a temperature of 100C, it immediately goes from liquid to gas, so the transition is not continuous. The same goes for liquid water freezing to solid water, more commonly known as ice, at 0 degrees Celsius. The boiling- or freezing point of water is more known as the critical temperature of the system in relation to the phase transition. Such critical quantities are the points that, when reached by the driver of the phase transition (temperature for water), the change of state occurs, either continuously or abruptly, depending on the system. Another phase transition, famous to at least physicists, is the magnetization of the Ising model. This phase transition is also driven by temperature, where

we observe that the spontaneous magnetization per spin is not equal to 0 (can be either positive or negative depending on the direction of the spin) for temperatures below the critical temperature, $T < T_c$, and equal to 0 for temperatures greater than T_c. This transition is of second-order, as the magnetization decreases while the material is heated before reaching zero at T_c, thus not happening abruptly. One of the more interesting qualities of phase transitions are *critical phenomena*, which are phenomena occurring close to high-order phase transitions [61]. This often entails systems exhibiting particular behaviours around critical points, sometimes enabling us to identify a phase transition in a system before it happens.

Percolation Transitions

Let us introduce a 2-dimensional lattice where all states are occupied. The edges are present with probability p or absent with probability $1-p$. Here, a *cluster* is defined as all nearest-neighbour states, that is to say, all states connected through an edge. A *percolating* cluster is defined, for the 2-dimensional case, as a cluster spanning either top to bottom or left to right in the lattice. The critical probability for such a cluster existing in a lattice is called the percolation threshold, p_c. This threshold is the value of p where an infinite cluster first appears in an infinite lattice [61], and it is the point at which a percolation transition occurs. This percolation transition is commonly known as "bond percolation", but there exists other percolation transitions such as "site percolation" [15]. From this concept, parallels can be drawn to network theory. Imagine a critical probability p_c where for $p < p_c$ a network is made up of many smaller and isolated clusters, and for $p > p_c$, a gigantic cluster spanning the entire network emerges. This idea is very similar to a bond percolation transition in a lattice, but instead of only demanding a cluster connecting two opposite sides of a lattice, we demand a cluster where starting at any vertex, all other vertices can be reached. This is the same as achieving a probability of a vertex belonging to the infinite cluster equal to 1 [15].

Chapter 3

Approach

The main focus of our approach chapter is to thoroughly introduce the reader to the methods developed for this thesis. This includes clarification of the process of data acquisition and how we filter and enrich the dataset. Moreover, we explain the graph-building process, how we section our underlying graph into different sets of sub-graphs (slices), and the purpose for why we partition the graph in the ways we do. Lastly, we provide an overview of the concept of vertex activity. Our goal with everything we introduce in approach is to extract as much information from the dataset we have containing the DW as possible, and the natural starting point of this process is to build the underlying network based on the interaction of Twitter users (see Section 2.4). From there we define the types of slices we will be producing as sets, while emphasising on the reasons behind us looking into them and the differences between them. Finally, we define vertex activity as the degree centrality, counting both in-bound and out-bound edges, of a vertex and explain why we believe this measure can provide insight into the dynamics of the DW through central or in other ways important Twitter users.

Table 3.1: Timeline of the data collection.

earlier	All tweets ever tweeted.
2000	Our data starts. Thread completion leaded to finding very old users.
Dec 2019	Corona virus identified. Started collecting data/ looking for historic data..
Dec 2019	Started collecting data containing Covid related key.
21.01.20	First tweet linking 5G to Covid (mentioning of both in the same tweet, not necessarily the first).
Feb 2020	We find out about the conspiracy.
03.04.20	First towers burn.
15.05.20	Collection ended, investigation of data stopped.
later	Collection of more data.

3.1 The Data Base & Extracting Temporal Data

Data Acquisition

Since Twitter's Terms of Service prohibit storing large datasets, we choose a streaming-based approach, first introduced in [62], storing metadata only. We keep only in-stream-anonymized user Ids, and the corresponding timestamps to create the tweet-retweet-user mapping. Moreover, we do neither store or process any other information. The data collection took place using a custom build framework for Twitter graph analysis [43], and a custom scraping strategy.

Between December 2019 and May 2020, about 1 billion Twitter statuses related to the COVID-19 pandemic were collected by querying statuses containing the keywords presented in table 3.2. The data collection timeline is visualized in Table 3.1.

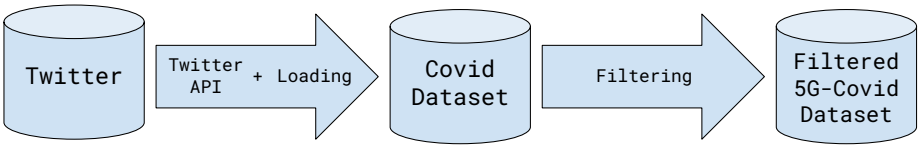

Figure 3.1: **Building the dataset:** The first steps of our pipeline. Further description can be found in the text.

Loading and Filtering

For loading a sufficient amount of Twitter data related to the COVID-19 pandemic, a framework for handling the massive amounts of data, the "Fact-Framework" [43], was created by *Daniel T. Schroeder*. The entire COVID-19 related dataset, produced by querying the API for the keywords presented in Table 3.2, was loaded into the framework that was later finished through collaborative efforts. The data was then filtered once more, searching for keywords related to the 5G-network, e.g., *"5G, 5g, 60Hz, #5G, ..."* to obtain a set of tweets likely to be part of the 5G-Corona conspiracy. This step reduced the number of statuses of interest to $364,325$.

Enrichment

As described in Section 2.4, a status can be replied to or retweeted, which leads to threads starting from one status and propagating down a "sharing-ladder". As mentioned earlier, searching for tweets containing keywords is limited by a 2 week window, but if one possesses the **id**-attribute of a status, it is possible to find it no matter how old it is. This allows restoring "parent"-statuses that did not contain the keywords we previously searched for on Twitter. This enrichment process restored more of the conversations connected to the DW. Since the Twitter API does not enable finding "child"-statuses, the thread completion process is limited to searching for child-tweets only in the data we already possess. As a result, we deal with a multitude of incomplete threads.

The restoring of statuses connected to the ones from the filter resulted in the base dataset containing all the potentially interesting tweets found during the entirety of the collection process. We will not be using the entirety of the base dataset in our experiments. We consider anything outside of the period $01.02.2020 - 11.05.2020$ as negligible for a quantitative analysis due to the small number of relevant tweets posted before and after this window.

Contact Extraction

Given the now filtered and enriched dataset, we start extracting information about interactions between users. Therefore, we go through all the statuses in our dataset and count the contacts between two users. We define

$$Z_u = \text{set of all users}$$

and

$$Z_s = T \cup R \cup B \cup Q$$

Table 3.2: This table shows an overview of the key words we looked for in the tweets we acquired using the Twitter API. DR stands for *Directly Related*, G stands for *General*.

Neutral DR	**German**	**Neutral G**	**Negative**
coronavirus10	coronar_allesoeffnen	vaccination	coronapanik
corinavirus6	allesoeffnen	vaccine	covidiot
corona	allesöffnen	epidemic	
coronaoutbreak	coronadeutchland	pandemic	
coronavirus	xj621	quarantine	
coronavirusde	machtbueroszu	quarantined	
coronavirusoutbreak	machtdiebueroszu	mutation	
covid	bueroszu	wuhan	
covid19	büroszu		
covid2019	diebüroszu		
covid_19			
covid-19			
wuhancoronavirus			
wuhancoronovirus			
wuhanvirus9			
coronavírus			
coronavirus7			
coronavirus8			
coronavirus9			
zerocovid			

with Z_s being the entire set of statuses, T being the set of tweets, R being the set of retweets, B being the set of replies and Q being the set of quotes (see Section 2.4). A contact between two users is defined as any user j interacting with any user i through either

1. User j retweeting a status by user i,
2. User j replying to a status by user i,
3. User j quoting a status by user i.

We can represent the set of contacts as a symmetric adjacency matrix A where $A_{ij} = 1$ represents a contact while $A_{ij} = 0$ represents no contact. Contacts are directed and weighted after the type of interaction. However, throughout this thesis, we treat contacts as unweighted and undirected edges. In other words, these contacts are the edges of our networks, which represent the connection between users. Each of these contacts is accompanied by a time-stamp telling us when the contact was made, which alongside the adjacency matrix, is the basis of the graphs we build.

3.2 Graph Building

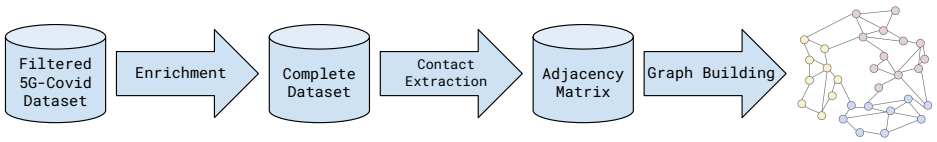

Figure 3.2: **Building the dataset:** The second step from our pipeline which can be subdivided into three/four smaller steps.

We build interaction networks (see definition in Section 2.4) based on the adjacency matrix induced by the contacts. Thus, the vertices represent statuses, and the edges represent a *contact* between them. Note that all statuses produced by an individual user have the same ID in our dataset. Thus, they are collapsed into one vertex, i.e., all the contacts associated with the statuses of one user become edges connected at one end to only one vertex or, in other words, one user. As previously stated, we define a contact as any two users interacting through the act of retweeting, quoting or replying (see Section 2.4). At the same time, we store attributes to the statuses to use them as vertex attributes.

We define the *underlying graph* $G_\downarrow$ as the temporal graph containing the entirety of vertices (Twitter statuses) and edges (contacts) in our dataset

$$G_\downarrow = (V_\downarrow, E_\downarrow), \tag{3.1}$$

where $V_\downarrow = \{v_i\}_{i=0}^{N}$ and $E_\downarrow = \{(u, v, t), u, v \in V_\downarrow, t \leq T\}$. Here T is the time window of data collection, so that $T = t_\mathrm{f} - t_0$. We want to emphasise that each edge has a unique timestamp t which is the exact datetime of the contact taking place.

3.3 Extracting Slices

A collection of interactions over a large window of time cannot by itself provide us with much information about the dynamics of said interactions over time. However, partitioning the information into slices enables us to study how the network changes across them. To examine the evolution of the network, either temporal or as a result of contacts, we divide $G_{\downarrow}$ into sets of "static" slices.

3.3.1 Defining a Slice

We define a slice as a sub-graph of $G_{\downarrow}$ so the set of these slices is $S = \{G(V_{\downarrow}, E_i), i \in L, E_i \in E_{\downarrow}\}$ where L is the number of slices. We wish to produce multiple versions of sets of such slices. Some sets are purely temporal, and others use a "sliding window" approach where we determine slices after the evolution of the neighborhoods of vertices. The sets of slices may be temporal, but each slice in a temporal set is a static "snap-shot" of some time period in the interaction network. The following subsections introduce three different types of slices; temporal slices, accumulative slices, and contact slices. Our motivation for producing multiple types of slices is the possibility of extracting different types of information from them.

Pure time-slices provide information about the temporal evolution of the network as a whole. However, they make it hard to track the intersect of clusters across time as vertices that are not active in a time period will be removed. Accumulative slices make it possible to track clusters but become large quickly, making them difficult to work with, especially when looking into the temporal changes. By saying temporal changes, we refer to how the conversation's discourse changes over time, the addition of or loss of users active in the conversation, and how the dynamics of user interactions evolve. Contact-based slices are a new way of producing sub-graphs. Their purpose is to provide a better picture of how conversations or communities within the graph evolve and how that affects the dynamics of the network as a whole. In a Twitter interaction network, edges represent contacts between statuses. Connected statuses are, in reality, conversations that are observed as communities in the network. Statuses with outbound contacts to other statuses are responses, so a vertex's neighborhood is by definition a conversation.

3.3.2 Accumulative Slices

The accumulative slices are defined as the set of all contacts made in the time interval $[0, t_i], \quad 0 < t_i < t_{i+1} \leq T$, where T is the time window of data acquisition. This means that slice s_{i+1} contains all contacts in s_i plus all other contacts made on the time interval $[t_i, t_{i+1}]$

$$\begin{aligned} s_{i+1} &= \{u, v \subset V_{\downarrow} : \exists (u, v, t) \in E_{\downarrow} \wedge t \in [0, t_{i+1}]\} \\ &= s_i \cup \{u, v \subset V_{\downarrow} : \exists (u, v, t) \in E_{\downarrow} \wedge t \in [t_i, t_{i+1}]\}. \end{aligned} \tag{3.2}$$

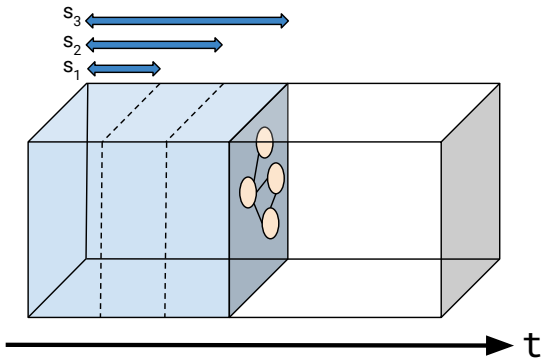

Figure 3.3: Illustration of accumulative slices. Slices contain all vertices and edges from the previous slices.

We define the distance between two subsequent times, $t_{i+1} - t_i$, as Δt which is equal for all i. The last slice of the experiment is, by definition, the entire underlying graph, $G_{\downarrow}$.

3.3.3 Temporal Slices

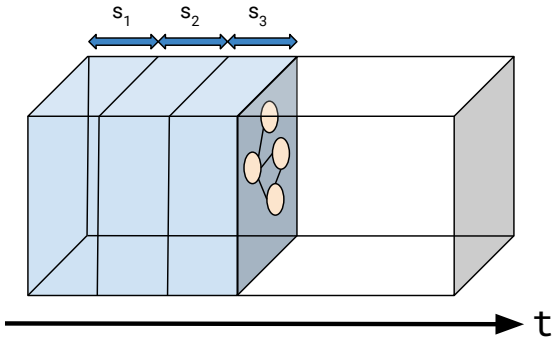

Figure 3.4: Illustration of temporal slices. Slices do not contain the vertices and edges of previous slices.

We divide our graph into slices of sub-graphs defined by the timestamp of each edge. We remind the reader that edges are contacts between users (retweets, comments etc., see Sections 2.4 and 3.1) and thus associated with a timestamp. We divide our set of edges into intervals of time, e.g., one day, a week, which we call $\Delta t = (t^s, t^e)$ so that we get L slices in total and the entire time interval where we collected our dataset $T = t^e_L - t^s_1$. This results in a temporal graph $\mathcal{G} = (V_{\downarrow}, E_1, ..., E_L)$. Now each slice $s_i \in S$ contains all edges added to the network in the time period Δt_i with

$$S = \{G(V_{\downarrow}, E_i), i \in L\}, \tag{3.3}$$

and $\bigcup_{i=1}^{L} s_i = G_{\downarrow}$.

3.3.4 Contact Slices

Another way of defining a slice is to look at the number of new contacts a vertex makes. We define k as the maximum number of neighbours a vertex is allowed to have at any given time. This scalar parameter can be a constant value for all vertices in G or a vertex-varying value defined by the level of "activity" displayed by each vertex individually. A general definition of the neighbourhood of a vertex is

$$N(v) = \{u \subset V_{\downarrow}, u \neq v : \exists(u, v, t) \in E_{\downarrow}\}, \tag{3.4}$$

which is all vertices directly connected to vertex v through an edge.

It becomes evident that the formalization of these slices must be done inductively as there is no way of knowing beforehand when a neighbourhood grows larger than k, i.e., it is not possible to start at the end and slice as we move backward.

For all slices we start by transforming our underlying network $G_{\downarrow}(V_{\downarrow}, E_{\downarrow})$ into a temporal graph $\mathcal{G} = (V_{\downarrow}, E_1, ..., E_M)$ where M is the total number of edges in $E_{\downarrow}$ and $E_i = (u_i, v_i, t_i) \in E_{\downarrow}$ contains exactly one edge with $t_{i-1} < t_i < t_{i+1}$. Furthermore, $\bigcup_{i=1}^{M} E_i = E_{\downarrow}$ applies. This means that we have ordered the edges linearly by time. Subsequently, we start to merge the edges into a unified graph, one by one, until some criteria function, for which we will present examples of in the following, of the neighbourhoods at time t_i, $N_\tau(v)$, and k is is activated. We introduce a new time dependent definition of the neighbourhoods as

$$N_\tau(v) = \{u \subset V_{\downarrow}, u \neq v : \exists(u, v, t) \in E_{\downarrow} \wedge t \leq \tau\}, \tag{3.5}$$

so that after merging edges with timestamps up to t, $N_\tau(v)$ describes the neighbourhoods in the graph $\mathcal{G} = (V_{\downarrow}, E_1, ..., E_\tau)$ where $E_\tau = (u, v, \tau)$.

The criteria function, $C(N_\tau(v), N_\tau(u), k)$, determines when a new slice should begin. How the new slice is initialized and what the criteria function is can be varied, so let us begin by looking at the case of a constant k.

Criteria Function: Constant k

A new slice is initialized as the old slice minus the oldest edge in the neighbourhood, which has grown larger than k. We denote the removed edges as *forgotten edges*, e^o. We also require agreement between the vertices connected to an edge proposed to be forgotten, that is to say, that the neighbourhoods of both vertices have to be larger than k. We define the criteria function in the case of a constant k as

$$C(N_\tau(v), N_\tau(u), k) = (|N_\tau(v)| > k) \wedge (|N_\tau(u)| > k), \tag{3.6}$$

a function that returns true if both neighbourhoods connected through an edge are larger than k and false if not. In short, edges are merged until the criteria function returns true, and a new slice is initialized. After running through this process we end up with a set of slices $S = \{s_j\}_{j=1}^{L}$ where each slice s_j, for $j = 2$, is a subgraph of $G_{\downarrow}$ containing the entirety of slice s_{j-1}

except from the forgotten edge, e^o_{j-1}. This means that at the end of the process remains a set of forgotten edges

$$F = \{e^o_1, ..., e^o_L\} \tag{3.7}$$

The reason for slicing the network in the proposed manner is to analyze the evolution of the network by keeping track of the movement of vertexes into new clusters. We want to see if removing the oldest contact after a user has made k new contacts is a possible way of doing this. More explicitly, we assume that **by slicing according to contacts, we can gain information about the users joining and leaving sub-conversations in the DW**.

Criteria Function: Vertex-Varying k

Moving on to the vertex-varying k, we imagine defining a non-constant k uniquely associated with each vertex. We wish to determine a set of scalars $K = \{k_1, ..., k_N\}$ using various methods, the first of which is the mean sizes of the communities of the vertices in the previous slices. We begin by initializing all k's as the same value, so $k_v = k$ for $v = 1, ..., N$. As with the constant k we start merging edges into one graph until $C(N_\tau(v), N_\tau(u), k_v)$ returns true. Before initializing the next slice, we calculate the size of all neighbourhoods $N_\tau(v)$ and find the mean of the initial k-values and the new neighbourhood sizes, so

$$K = \left\{ \frac{k_v^{t=0} + k_v^{t=\tau}}{2} \right\}_{v=1}^{N} = \{k_1^\tau, ..., k_N^\tau\}. \tag{3.8}$$

Now we initialize the next slice in nearly the same way as we did when using the constant k. The only difference is that the criteria function is now vertex dependent as the value of k uniquely depends on the individual neighbourhoods. We repeat this process, calculating the new mean each time the criteria function returns true, initializing a new slice. At the end of the process, the set of k values will be

$$K = \left\{ \frac{k_v^{t=0} + k_v^{t=\tau} + ... + k_v^{t=T}}{L} \right\}_{v=1}^{N} = \left\{k_1^T, ..., k_N^T\right\}. \tag{3.9}$$

The reason for looking at another approach than the constant k is that some users communicate a lot while others communicate very little. Thus, imposing the same size condition on all the neighbourhoods would be naive, especially concerning the agreement requirement. If the neighbourhood of v, a very active user, grows larger than k, but the oldest edge in the neighbourhood is connected on the other end to a user who has only once been a part of the conversation, the criteria function would return false as that neighbourhood would be smaller than k. In reality, we would want to remove that edge as it is very plausible that that single contact is no longer a critical part of the conversation within the neighbourhood of the active user v. The goal is that a vertex varying k will pick up on this and lead to us forgetting these non-essential edges.

3.4 On Vertex Activity

As described in Section 2.5, centrality measures give us insight into which users or vertices contribute most to the flow of information in a network. As we are dealing with an undirected network, we can not distinguish between if a vertex is highly active (e.g., comments on other statuses with a high frequency) or is made highly active by others (many other statuses are responses to this status). We therefore define vertex activity as

Definition 3.4.1 (Vertex Activity)**.** The number of occurrences where a vertex is a part of a contact.

In other words this is the number of edges connected to a given vertex and so equal to the degree centrality (see Section 2.5) of the vertex. This can be calculated for each slice s_i or for the entire underlying network $G_{\downarrow}$. This measure is of interest as we wish to explore the correlation between vertex activity of a group of vertices and other properties of the system such as the size of the largest clusters and number of overall contacts. The goal of which is to gain insight into how the vertices affect the evolution of the network, are all vertices important or will we see that a smaller group of vertices are driving the conversation surrounding the 5G-corona misinformation event (see Section 2.2).

Before looking into the vertex activity we formed some hypotheses going off previous experience about how Twitter works and how people use it to communicate. We know that twitter allows for conversations between users though the comment section attached to each status, so it is fully possible to have conversation back and fourth between two or more users. There is also a possibility, especially regarding influential people with large followings, that a user will post a status that gain a lot of traction through other users retweeting/commenting on it where the user who originally created the status does not respond further to those interacting with it. At least there is a large probability that the amount of others interacting with the status far outnumbers the number of responses back from the user who created it. Take the twitter account **@BarackObama** (former President of the United States) as an example. As of 21.03.22 the account has a following of 131.3 million users and looking at a tweet posted by the account on 13.03.22 we count $29,8$k retweets, $6,672$ quotes, 31k comments and $354,7$k likes. It is safe to assume that with an amount of responses of this size, Barack Obama does not have time to respond to every single one each time he tweets something, and so in a network context the amount of inbound edges far outnumber the outbound.

Chapter 4

Experimentation, Results & Analysis

The following chapter presents the results of the experiments. We begin by recapitulating the different types of sets of slices we defined in Chapter 3 before we move on to parameter configuration for the sets we produce. At this point, everything we need to set up our experiments is given. We start our experiments with a preliminary analysis of the data, in particular, a look at the vertex distribution across the sets of slices. This seems to be a natural starting point as the evolution of the density of vertices and edges in the network can indicate where the activity in the DW is significant.

The next step in the process is to look into the communities within the DW through cluster detection. We investigate the distribution of cluster sizes and turn our focus to the magnitude of the largest cluster across slices. The reason for this approach is that we assume that large clusters are more significant to the overall discourse of the conversations contained in the DW and thus more important than the smaller ones. Furthermore, we track the largest cluster through re-identification of vertex members to explore the origins of the DW. If the largest cluster stays the largest from beginning to the end, it could indicate a single "important" origin of the DW.

Finally, we take a closer look at the temporal slices, producing more sets with finer and courser time-grids. The former is in an attempt to identify more local behaviours of the DW and the latter to see if we can observe more general patterns in the DW. We compare the activity of the vertices with the highest degree centrality to the total number of contacts in the graphs. This approach allows us to determine the significance of these vertices in terms of the overall discourse. Our very last experiment is extracting the average nearest neighbour degree (ANND) function of slices to investigate its assortativity.

In the following, we propose four main hypotheses regarding what we expect to find from our network in particular

Hypothesis 1: Centralized Core

A minor fraction of users drive the DW. (H1)

The reasoning for proposing hypothesis H1 is driven by our intuition about and experience with how Twitter works, as elaborated on in Section 3.4. In short, the conversation threads of Twitter are not always conversations back and forth but rather a set of responses to one engaging status. Hypothesis H1 can be reformulated as a proposal for a driver of the DW, drawing parallels between the dynamics of the spread of the DW during its peak on Twitter and a phase transition (see Section 2.8).

Hypothesis 1a : The Influencer Transition

When an influential user tweets, the number of users active in the digital wildfire increases largely. (H1a)

We cannot with a 100% certainty differentiate between the importance of the act of tweeting, when the tweet is posted or what is the content of the post. However, the effect is that many new people participate in the DW by responding to the influencer.

Hypothesis 2: Dispersed Core

A large fraction of the users contribute to driving the DW. (H2)

Hypothesis H2 was developed from the idea that the DW possibly has multiple origins, e.g., that more than one initial conversation about the connection between 5G and COVID-19 could be significant to the evolution of the DW.

Taking more inspiration from phase transitions, more specifically percolation transitions (see Section 2.8), we propose another hypothesis for a driver of a transition in the network.

Hypothesis 3 : The "Black Hole" Transition

When the size of the largest cluster S reaches a critical size S_c, the cluster begins acting as a "black hole". (H3)

By black hole behaviour, we refer to the parallel of an immense gravitational pull. For an interaction network, this essentially means that a cluster is gaining new vertices at a substantially higher rate than when $S < S_c$. This hypothesis connects size to change in size, i.e., comparing a function to its first derivative.

Hypothesis H3 differs from the gigantic cluster parallel we drew under percolation transition (see Section 2.8). Here, we are not assuming that the entire network becomes fully connected when reaching the percolation threshold but rather that a large part of the network at some critical largest cluster size, S_c, becomes more connected at a substantially higher rate than for $S < S_c$. This idea of a transition is more similar to a percolating cluster in a lattice, without imposing a definite border or similar criteria at which the transition occurs. To be clear, this is not an actual phase transition but comparable to the concept.

Before presenting the results, to help the reader remember the terminology, we recapitulate the different types of slices produced in the thesis. We produced multiple sets of slices based on the different methods of slicing our underlying graph $G_{\downarrow}$. The following are definitions of these slices

- **Accumulative Slices**
  A slice i contains contacts that occurred in the time interval $[0, t_i], t_i < t_{i+1} <= T$, where T is the entire time frame in which the DW occurred. All slices $i + 1$ contains all the contacts from slice i plus contacts made in the time interval $[t_i, t_{i+1}]$. Furthermore, we define the distance between two subsequent times, $t_{i+1} - t_i$, as Δt which is equal for all time intervals in a set. The last slice in the slice-set will by definition be the entire underlying graph.

- **Temporal Slices**
  A slice i contains contacts that occurred in the time interval $[t_{i-1}, t_i], t_{i-1} < t_i <= T$. Subtracting the accumulative slice i from the accumulative slice $i + 1$, results in the temporal slice $i + 1$. The first slice in the temporal slice-set is equal to the first slice in the accumulative slice-set as long as Δt is the same in both experiments.

- **Contact-Slices**
  Here, slices are determined by a criteria function considering the number of allowed neighbours for each vertex so that $C = f(k)$. Starting with an empty network, we add contacts after their time-stamp until the criteria function determines a neighbourhood is too large. At this point, the oldest edge in the neighbourhood is removed, and a new slice is initiated. We continue this process until all the edges of the underlying graph are added, resulting in a set of slices. This is an inductive process, which means that we do not know the cardinality of the set beforehand.

4.1 Parameter Configuration for Slices

The very first step in our experiments is to produce the sets of slices. To accomplish this, we must decide on the parameters governing the slicing processes.

Temporal- & Accumulative-Slices: Time-Intervals

We decide a reasonable time-interval to be $\Delta t = 1$ day because we assume that conversations stay more or less stable within a day with few significant changes to the central message. Moreover, we believe a conversation continues across days, but pauses during the night when Twitter users are sleeping. Thus, we use this Δt for both the temporal and the accumulative slices. For the temporal set of slices, this ensures that each slice represents a "snapshot" of all contacts that occurred during one day. For the accumulative set, every slice will contain all contacts made up until one day in the Twitter interaction network related to the DW.

Contact Slices: k-values

To identify candidates for the maximum neighbourhood size in our criteria function, Equation 3.6, we looked into the distribution of neighbourhood sizes in the entire network. We present the distribution of degree centrality in the underlying graph $G_{\downarrow}$ in Figure 4.1. Here, we see that most of the vertices have neighbourhoods of size $k \lessapprox 1000$. These results motivate us to create contact-based slice-sets using criteria functions with discreet values of $k \in [10, 800]$. We initialize the first set using $k = 200$ and quickly observe signs of problems like a very large amount of slices and the production being very time-consuming. A great number of slices leads to a much larger set of data, and so they would require substantially more time spent on calculations. Considering the scope of this thesis we decided for using larger values of k. After testing values $200 < k \leq 800$ we observe that sets produced using $k = 600$ and $k = 800$ give us a reasonable amount of slices within a sensible amount of time. For contact-based slicing, we found that the major issue of using a constant k-size for all vertices, while not imposing other criteria, is that small values of k produce a massive amount of slices, while the higher values converge towards the accumulative slices.

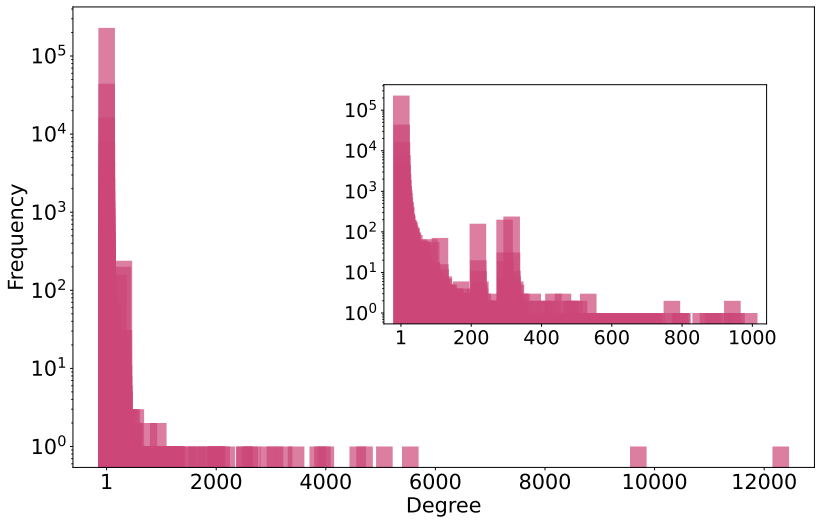

Figure 4.1: The distribution of node activity in the complete underlying network, $G_{\downarrow}$. We observe the distribution to be much denser around small neighbourhood sizes and quickly grows sparser towards higher ones.

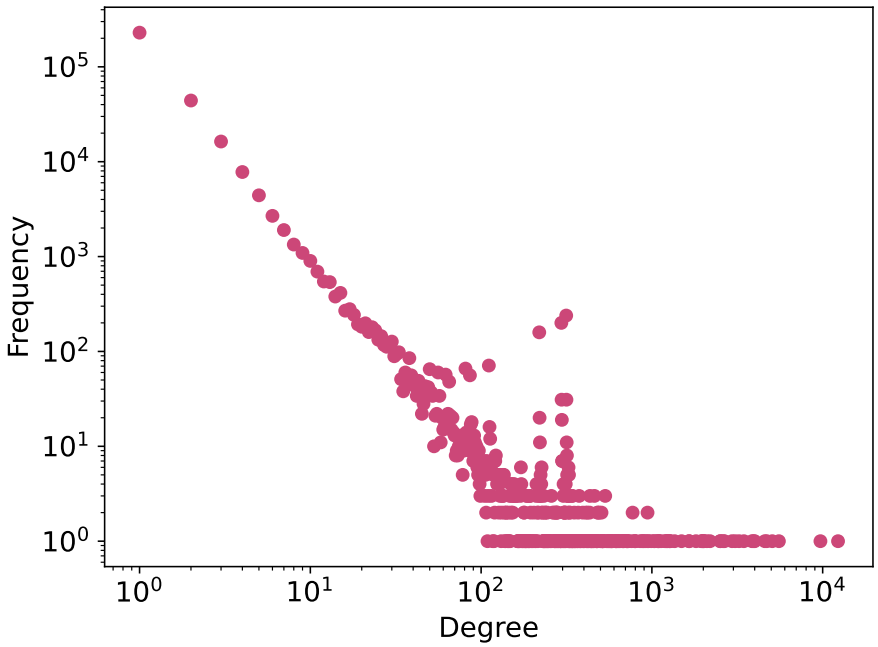

Figure 4.2: The distribution of node activity in the complete underlying network, $G_{\downarrow}$. This is another way of visualizing the distribution from Figure 4.1, here with both axes logarithmically scaled to reveal a weak scale-free nature of $G_{\downarrow}$.

4.2 Preliminary Analysis

To familiarize ourselves with the dataset and slice-sets, we begin by looking into the more general attributes of the network. It is important to note that statuses produced by one Twitter user are collapsed into one vertex, while edges belonging to those statuses are kept. We wish to utilize the information obtainable from this as a guide for later experimentation, to identify possible phases of the evolution of the DW and to formulate some initial expectations.

4.2.1 Data Exploration

A natural starting point is to examine how the network's number of vertices and edges, respectively representing Twitter users related to the COVID-5G misinformation event and the contacts between them, vary throughout the sets of slices. Figures 4.1 and 4.2 both show the degree distribution in the underlying graph $G_{\downarrow}$. Figure 4.1 clearly shows the general qualities of the distribution without requiring much interpretation. The majority of degrees are of degrees $< 10^2$ an the distribution seemingly decreases exponentially except for some outliers between degrees of $200 - 400$. This exponential decline is clearly shown through the linear pattern seen in the log-log plot in Figure 4.2, which apart from the outliers between $200 - 400$ very closely mimics a power law of the degree with negative exponent. This indicates that our network is at worst semi-scale-free.

The Vertex and Edge Distribution Across Slices

We looked into how each slice's number of vertices and edges varied across the experiments through visualization of the distribution of users and contacts in Figures 4.3 and 4.4. By looking at the temporal slices in Figure 4.3 we identify the peak of the DW as April 2020, which correlates with the expected timeline (see Section 2.2). Since each slice represents one day in the period $[01.02.2020 - 11.05.2020)$, we exchange the labels on the x-axis to represent a timeline, which we show in Figure 4.5. Furthermore, we deduce that slices $[60, 80]$ in the bottom of Figure 4.3 translates to the period $01.04.2020 - 21.04.202$ shown in Figure 4.5, which is right around the peak of the DW. The accumulative slices indicate the same peak by the slope being steepest around the same indices as the peak in the temporal slices, which is expected from us using the same Δt we produced the sets. Comparing Figure 4.4 to the accumulative slices in Figure 4.3, we find that the distributions are very similar, and any discrepancies are likely to come from the varying number of slices. Taking a closer look at the temporal slices in Figure 4.5, we can identify three distinct areas by analyzing the vertex distribution over time, which we categorize as follows

- Phase 1: **Before** the peak of the DW on Twitter,
- Phase 2: **During** the peak of the DW on Twitter,
- Phase 3: **After** the peak of the DW on Twitter.

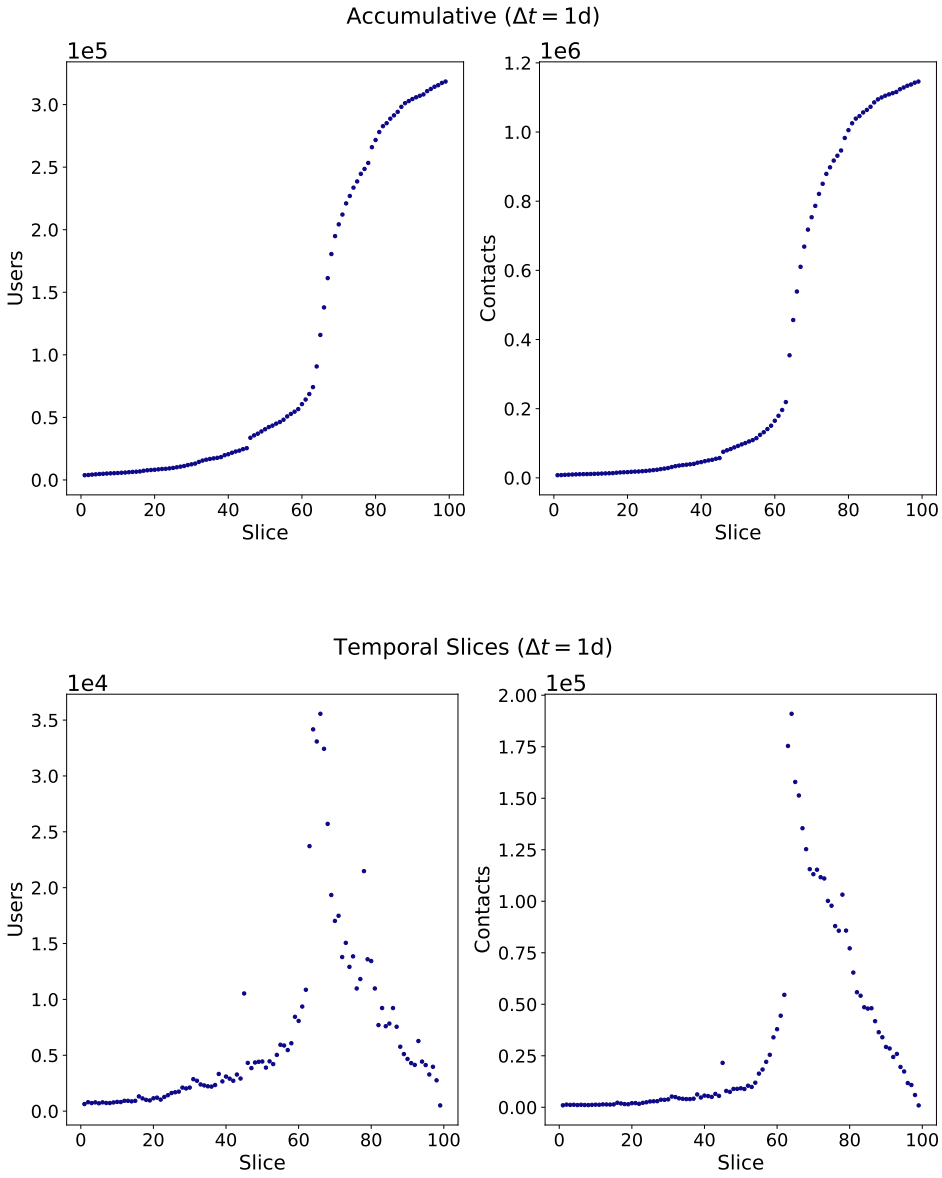

Figure 4.3: **Accumulative (top) & Temporal (bottom) Slices:** Distribution of users and contacts related to the DW across slices. We identify the peak of the DW on Twitter between approximately slices $[60, 80]$, in the accumulative as a steep slope and in the temporal as clear peak in the area in question.

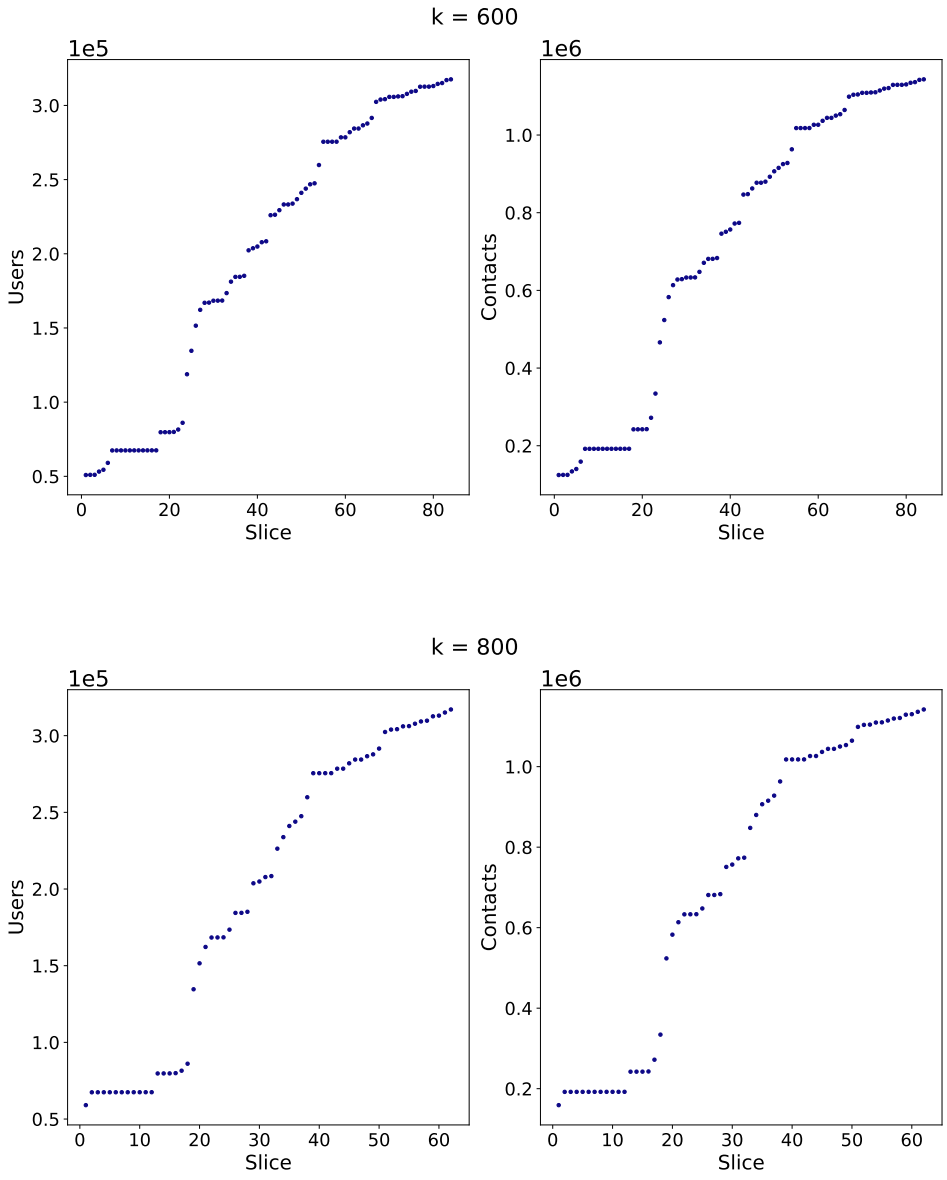

Figure 4.4: **Contact slices $k = 600$ (top) & $k = 800$ (bottom):** Distribution of users related to the DW and contacts across slices. As with the accumulative slices in fig. 4.3 we observe a strictly increasing distribution, only here less smooth. This comes from the way the criteria function slices the underlying graph, which differentiates the contact slices from the accumulative ones.

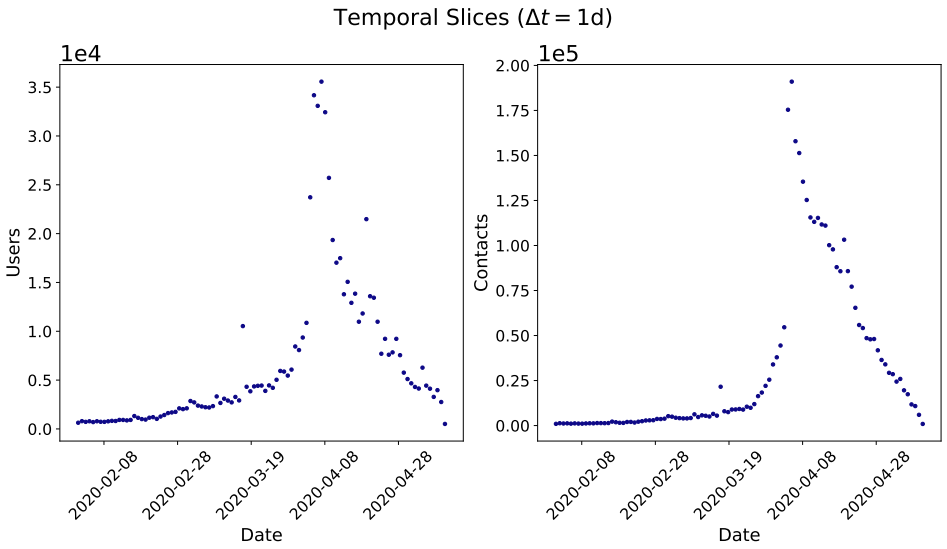

Figure 4.5: Distribution of users related to the DW and contacts over time in temporal slices of $\Delta t = 1$ day. This figure shows the same result as the bottom plots in fig. 4.3, only now plotted against the corresponding timeline the slices represent.

For the reader familiar with some statistical mechanics, the distributions of vertices vs. time for the temporal slices shown in Figure 4.5 closely resembles what we expect from phase transitions in critical phenomena physics, with an example being the specific heat per spin vs. temperature for the 2-dimensional Ising-model. More generally, the system is undergoing a transition. To observe a behaviour of this nature provides us with confidence in moving forward with our quest of trying to define and quantize this behavior. One way of doing so is to look at how groups within the network behave and evolve. In the following, we look at cluster detection and tracking expecting that the clustering of the network can help us understand the underlying reasons for this transition and provide suitable suggestions for drivers.

4.3 Cluster Detection

We explored the extraction of communities in our experiments through three algorithms; Leiden [1] and Louvain [2], whom are modularity based, and Label Propagation [3]. The resulting distribution of cluster sizes from each method is presented in Figures 4.6 to 4.8. Right away we see that the distributions for accumulative and contact slices are very similar, which could be expected from the similarities in the distribution of nodes and contacts shown in Figures 4.3 and 4.4. The same similarity can be seen from the fraction of clusters belonging to a certain size-group, shown in Figures 4.9 and 4.11.

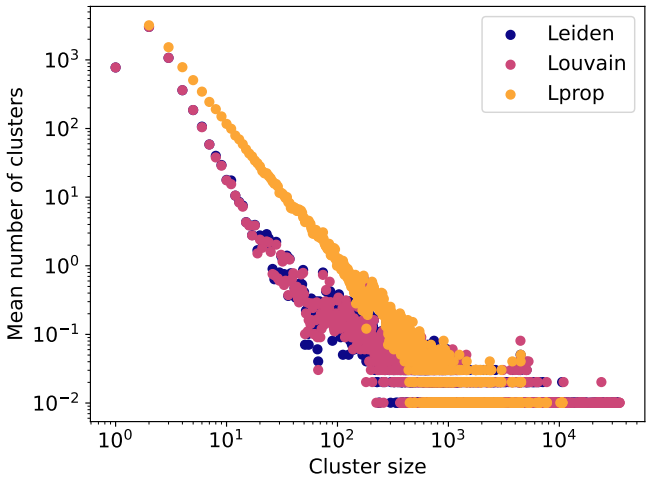

Figure 4.6: **Accumulative Slices:** Distribution of cluster sizes over all slices. For sizes $< 10^2$, the distribution is densest and seemingly follows a power law. As there are very few clusters of size $> 10^2$, the distribution grows more erratic, and there are no longer any apparent fitting approximations. This behaviour applies to all three algorithms.

The distribution for the temporal slices, shown in Figures 4.7 and 4.10, show, in general, the same trend, but with less difference between the clustering methods as well as having a smaller maximum size of the clusters. This reduction in maximum size can be explained by the temporal slices not being accumulative, and therefore contain fewer vertices per slice. However, slice 1 is an exception, as it is equal to the accumulative slice 1 (see Section 3.3). For all sets of slices, we observe a distribution that favors smaller cluster sizes, $< 10^2$, which is in line with what we expect from the degree distribution of the underlying graph, shown in Figure 4.2, being close to scale-free.

We observe that all three methods produce distributions that, for the most part, follow a power law with a negative exponent. Both Leiden and Louvain produce many clusters of size 1, which is not a good representation of a network with no vertices of degree 0 such as ours. This multitude of clusters of size 1 is part of why the distributions from Leiden and Louvain do not follow a true power law. On the other hand, Label Propagation produces

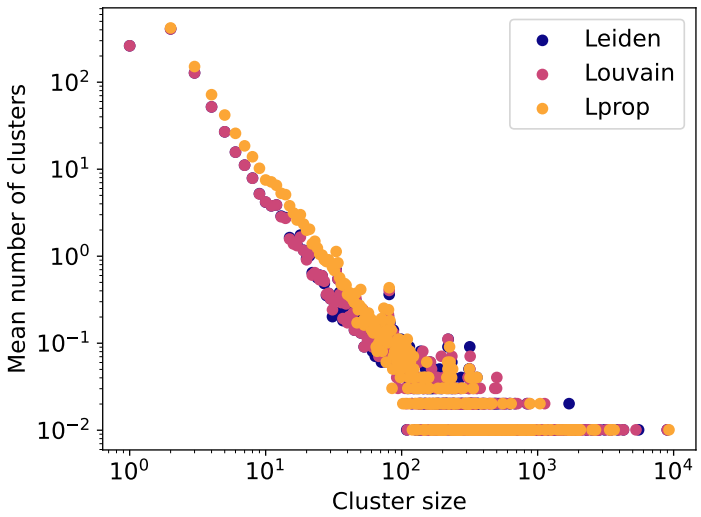

Figure 4.7: **Temporal Slices:** Distribution of cluster sizes over all slices. For sizes $< 10^2$, the distribution is densest and seemingly follows a power law. As there are very few clusters of size $> 10^2$, the distribution grows more erratic, and there are no longer any apparent fitting approximations. This behaviour applies to all three algorithms.

no clusters of size 1, which better represents our network and makes for a distribution that nearly perfectly resembles a power law with a negative exponent.

4.3.1 Vertices in the Largest Clusters

To study the temporal evolution of the overall discourse, we seek to gain more insight into the largest cluster. In a Twitter interaction network, the largest clusters represent the conversations or narratives that have the largest number of contributing users. We do this as we assume that the larger clusters are more dominant than smaller clusters in affecting the overall dynamics of the system. We presume, as presented in hypothesis H2, that a driver of DWs is that a significant fraction of the users converse back and forth, over time compelling more people to join the conversation, thus resulting in growing clusters. As stated in H3, we suppose that there could exist a critical size at which a cluster would start to gain new vertices at a substantially higher rate, i.e., displaying a nature similar to a black hole. Thus, to test these hypotheses, we need to examine the behaviour of the largest clusters in our network.

The fraction of vertices in the largest cluster in each slice, relative to the total number of vertices, is visualised in the left parts of Figures 4.12 to 4.15, while the right side displays the fraction of vertices belonging to the top 10% of the largest cluster of each slice. The figures show that Louvain and Leiden assign more vertices to the largest clusters than Label Propagation. The difference between the algorithms appears much more significant for the

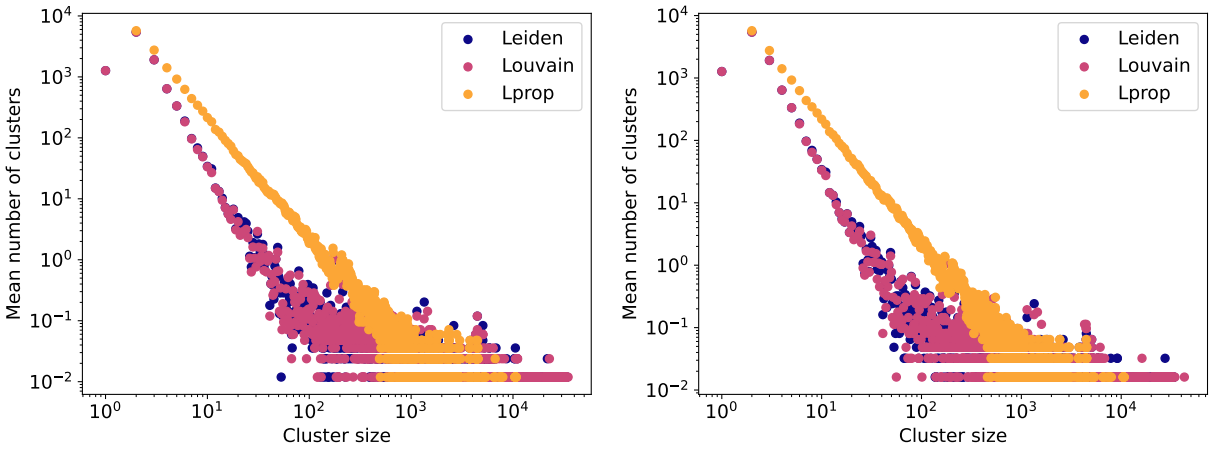

Figure 4.8: **Contact slices $k = 600$ (left) & $k = 800$ (right):** Distribution of cluster sizes over all slices. For sizes $< 10^2$, the distribution is densest and seemingly follows a power law. As there are very few clusters of size $> 10^2$, the distribution grows more erratic, and there are no longer any apparent fitting approximations. This behaviour applies to all three algorithms.

accumulative and contact-based slices than for the temporal slices. However, if we pay attention to the difference in the scaling of the y-axes, we see that this only really applies to the right-handed plots.

The difference only appears great for the left-handed plots because of the difference in magnitude on the y-axis, but there is a large difference for the right-handed plots. We presume this is due to the fact that the Label Propagation algorithm favours smaller clusters and produces less large clusters (clusters of size $>$ 1000 vertices) than Leiden and Louvain (see Figures 4.9 to 4.11).

When using the Leiden and Louvain algorithms, the number of vertices in the largest cluster stays below 15% of the total for the accumulative and contact-based slices. In contrast, the number of vertices in the top 10% largest clusters increases to around 50% towards the final clusters. These results indicate that during the period close to and during the peak, there existed more than one significant conversation dominating the DW. Moreover, the results indicate that the top 10% largest clusters represent what the algorithms believe to be all statuses belonging to multiple densely connected conversations. Provided this, there is reason to look into more than the single largest cluster in future work on the subject.

For the time-based slices, shown in Figure 4.13, both the number of vertices in the single largest cluster and the number of vertices in the top 10% of largest clusters lie between $0 - 70\%$. Especially in the left figure, displaying the fraction of vertices belonging to the single largest cluster, when comparing the scale of the y-axis, we see that the increase in the relative number of vertices across slices is much more significant than in the accumulative and contact-based slices. As expected, the results for the temporal slices are different from the accumulative and contact-based ones,

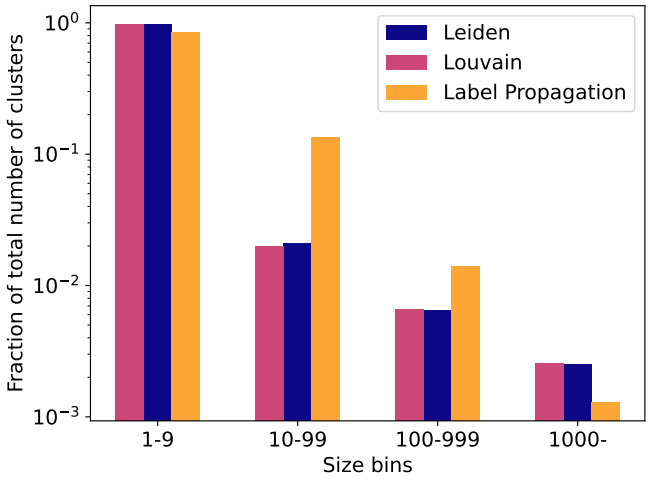

Figure 4.9: **Accumulative Slices:** Fraction of clusters whom belong to a size group. An alternative visualization to Figure 4.6 which more clearly shows the over-all distribution of the size-magnitudes of clusters produced by the three clustering algorithms. It is clear from the figure that the great majority clusters produced by all algorithms have sizes between $1-9$ vertices.

as they include statuses from earlier days of the DW. Most often, the number of vertices in the largest cluster is relatively small $< 20\%$, but in a seemingly non-ordered fashion, when comparing it to the distribution of vertices over slices, Figure 4.3, it fluctuates towards higher values in specific slices. To support our results we extend to not only the largest cluster, but the 10% largest clusters. From the right-handed figure, displaying the number of vertices in the top 10% largest cluster, we observe that it acts much more in aligning with what we expect. There we observe a tendency of a peak that more or less aligns with the position of the peak of the DW on Twitter. **This result shows a centralization of the conversations revolving around the DW.** This phenomenon has to the best of our knowledge never been observed before and seems to be an ideal candidate for the prediction of DWs.

We observe a very high increase in the relative number of vertices in the largest cluster for the last slice in the temporal slice set, Figure 4.13. If we compare this result to Figure 4.5, we see that the last slice contains very few vertices in total. Moreover, we know that Twitter banned statuses and users promoting attacks on 5G infrastructure after the peak of the DW in April 2020 (see Section 2.2), which, amongst other efforts, led to the decline in statuses relating to the DW after April 2020. In light of this, we argue that the most likely cause for the spike in the last slice in Figure 4.13 is that there were not many other active conversations relating to the DW at this point.

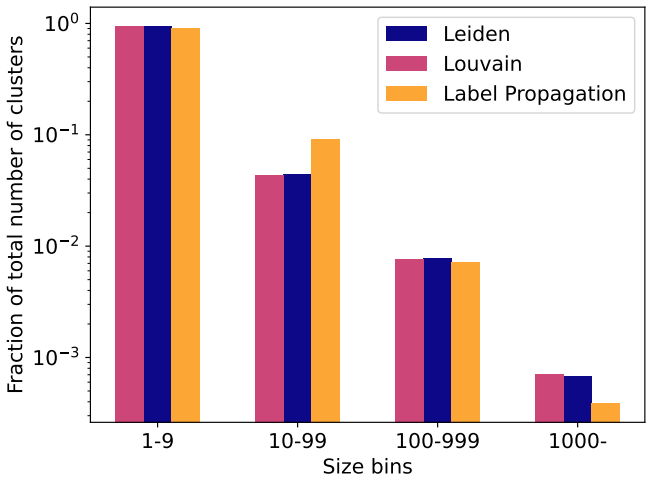

Figure 4.10: **Temporal Slices:** Fraction of clusters whom belong to a size group. An alternative visualization to Figure 4.7 which more clearly shows the over-all distribution of the size-magnitudes of clusters produced by the three clustering algorithms. It is clear from the figure that the great majority clusters produced by all algorithms have sizes between $1 - 9$ vertices.

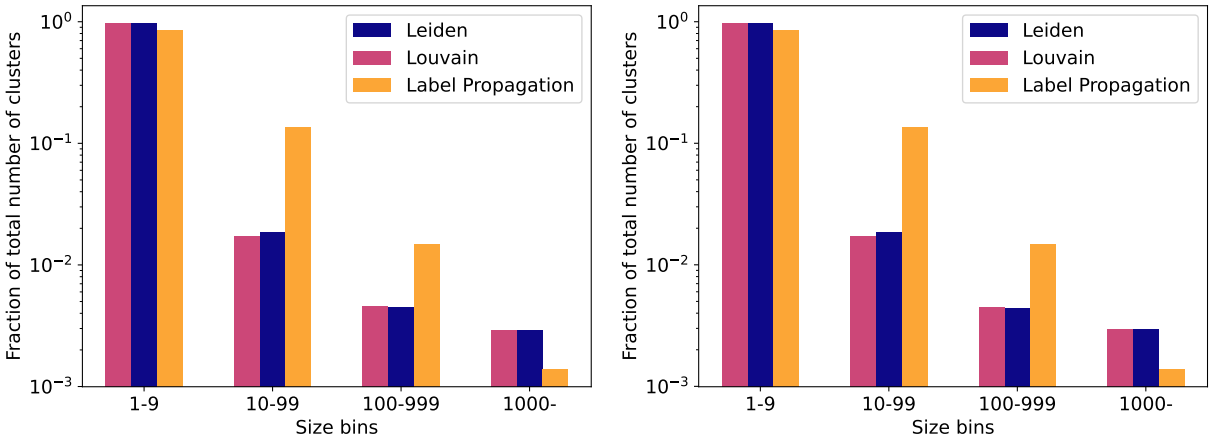

Figure 4.11: **Contact slices** $k = 600$ **(left) &** $k = 800$ **(right):** Fraction of clusters whom belong to a size group. An alternative visualization to Figure 4.8 which more clearly shows the over-all distribution of the size-magnitudes of clusters produced by the three clustering algorithms. It is clear from the figure that the great majority clusters produced by all algorithms have sizes between $1 - 9$ vertices.

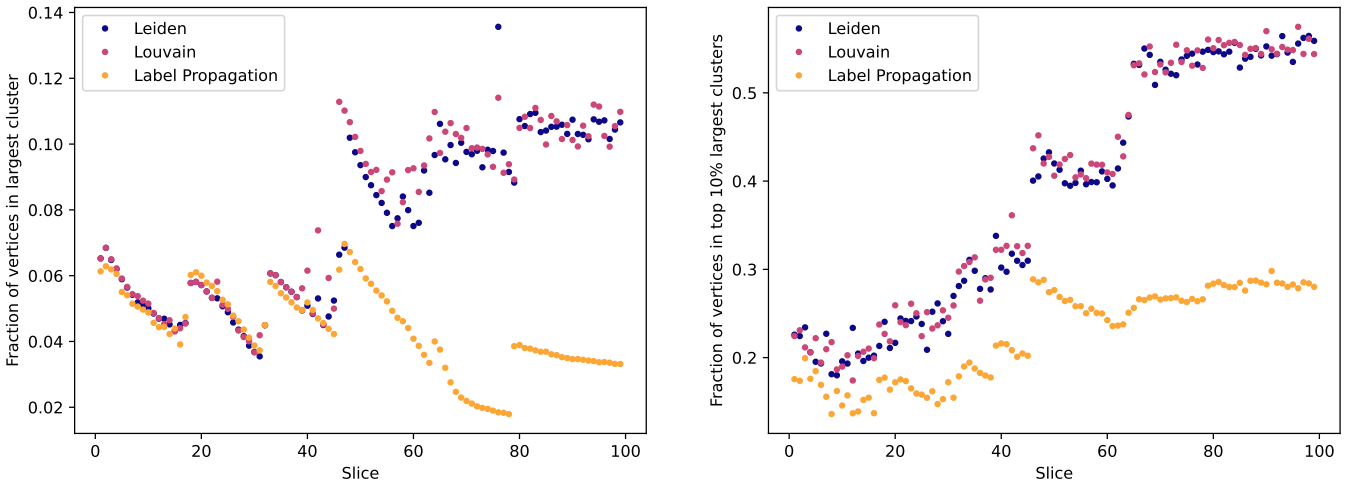

Figure 4.12: **Accumulative Slices:** Number of vertices relative to the total number of vertices in slice. Left: Fraction of vertices in the largest cluster of each slice. Right: Fraction of vertices in the top 10% largest clusters of each slice.

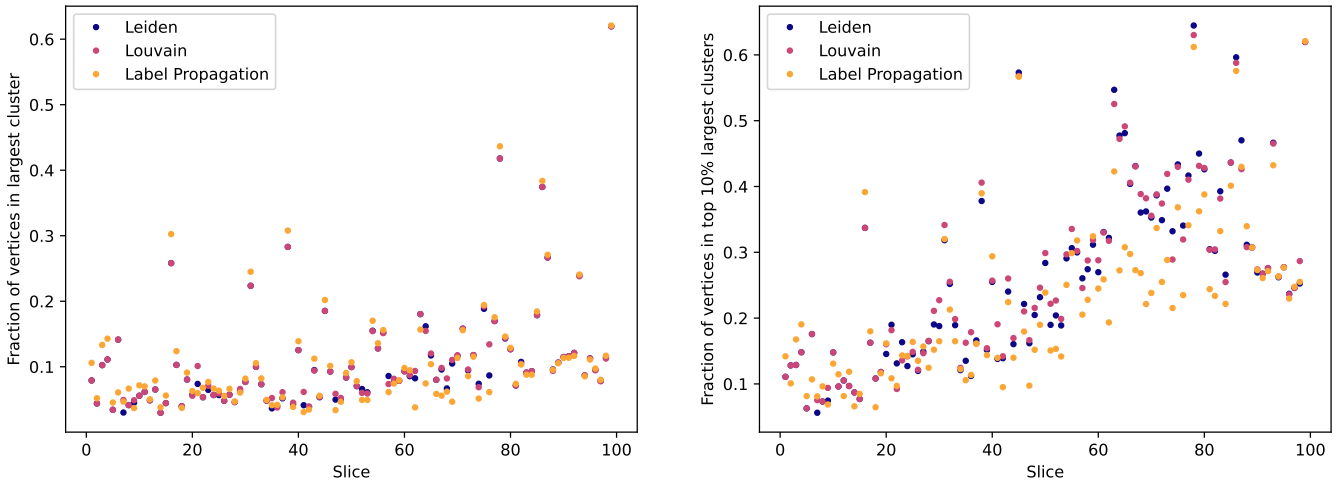

Figure 4.13: **Temporal Slices:** Number of vertices relative to the total number of vertices in slice. Left: Fraction of vertices in the largest cluster of each slice. Right: Fraction of vertices in the top 10% largest clusters of each slice.

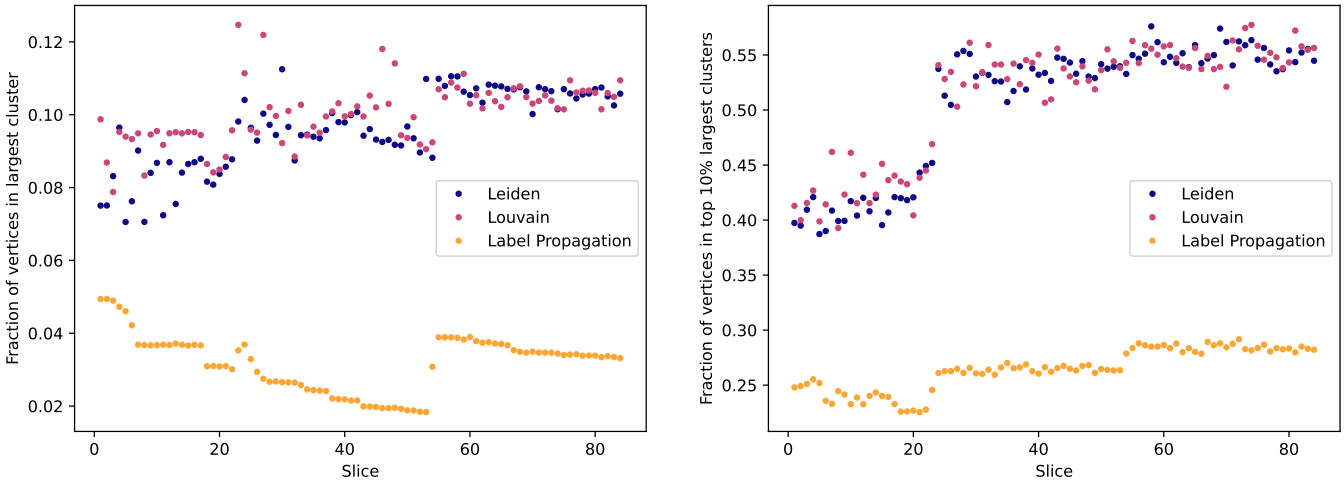

Figure 4.14: **Contact Slices** $k = 600$**:** Number of vertices relative to the total number of vertices in slice. Left: Fraction of vertices in the largest cluster of each slice. Right: Fraction of vertices in the top 10% largest clusters of each slice.

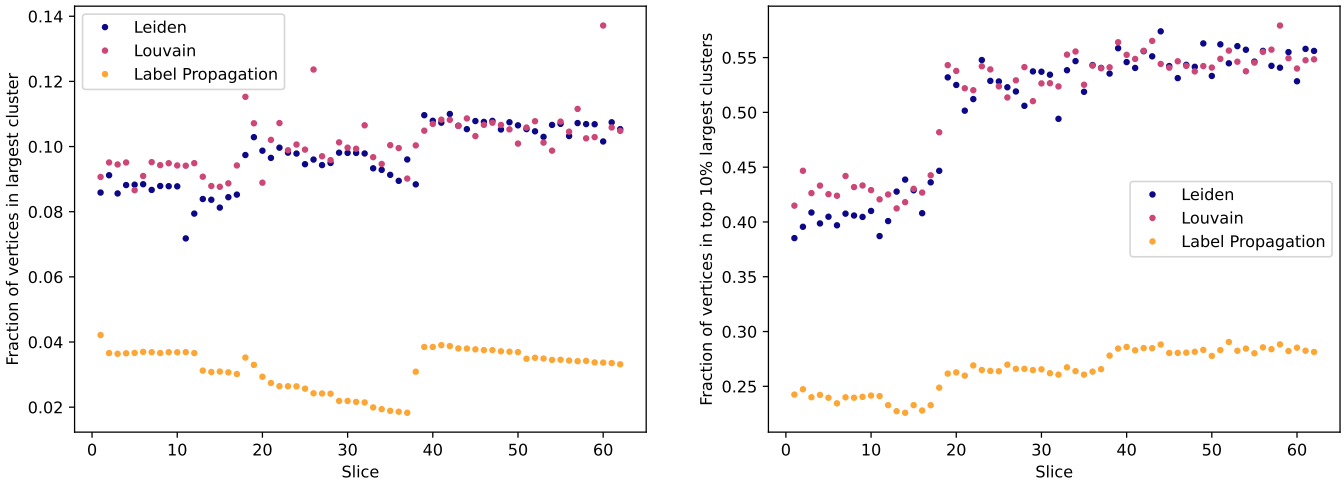

Figure 4.15: **Contact Slices** $k = 800$**:** Number of vertices relative to the total number of vertices in slice. Left: Fraction of vertices in the largest cluster of each slice. Right: Fraction of vertices in the top 10% largest clusters of each slice.

4.4 Cluster Tracking

As emphasised in Section 4.3.1, we assume that the behaviour of the largest clusters provides information describing the evolution of a DW. This assumption prompts us to continue exploring by seeing if the largest cluster from the beginning of the DW stays significant through time or if other previously small or non-existing clusters, e.g., the emergence of new narratives or growing popularity of previously non-engaging narratives, start dominating the conversation. Suppose the largest cluster from the very beginning of the DW stays the largest across all slices. In this case, finding the largest cluster is a key quality for early identification of DWs as it presents features unique to the spreading of a DW before it leads to harmful consequences. Another fruitful outcome would be the results enabling us to identify another point during the evolution of the DW from where the largest cluster stays the largest. This result would indicate a critical point in the DW and provides information about the structure of the DW.

We track the absolute largest cluster in each slice across slices through re-identification. We re-identify clusters across slices by looking at the intersect calculated by the fraction of vertices with the same user ID, where we deem the cluster in slice $i+1$ which has most intersect with a cluster in i as well as containing at least 50% of the vertices of in cluster i as the re-identified cluster. As we base intersect on the ratio of similar vertices, this method only works for the accumulative and contact-based slices. Moreover, the method does not work for temporal slices, as they do not remember the vertices in previous slices. Simplified, this means that for the temporal slices, there is no way of knowing if a cluster in slice $i+1$ is the same cluster as any of the clusters in slice i.

4.4.1 Tracking the Path of the Largest Cluster Across Slices

We begin by identifying the largest cluster in each slice and re-identifying it through every following slice. This process creates a path through each slice, except the last one. If the largest cluster in any slice $i+1$ equals the re-identified cluster from slice i, the paths from the two slices are the same, and we only draw it once. We present these paths in Figures 4.16 and 4.17. Here, we mark the start of new paths with ($\times$), as well as the instances where a cluster is no longer re-identified with ($\bullet$) and when the largest cluster of a slice is a part of a path other than the path of the largest cluster in the previous slice with ($\blacktriangledown$). To clarify, when the largest cluster in slice $i+1$ is re-identified as the largest cluster in slice i, this is not a new path, and we do not mark it; we just continue on the path from the previous slice. Thus, $\blacktriangledown$ does not represent a new path, it indicates that this largest cluster is not the same as the largest cluster in the preceding slice, but it is the re-identified cluster from some previous slice before that.

All the largest clusters are re-identified throughout the experiments, except for the accumulative slices using the Leiden and Louvain algorithms, where there are some instances where a cluster is not re-identified. In general, the label propagation algorithm returns fewer paths. In other words, the

largest cluster stays the largest through slices for a longer time and has much fewer paths starting on older paths than the one from the closest slice on average.

Except for the paths produced by the Leiden algorithm for contact slices using $k = 800$ (top right in Figure 4.17), we observe gatherings of ▼ in the figures showing the results of the Leiden and Louvain algorithms. These indicate an oscillation of the largest cluster, i.e., two clusters competing for the title of largest. These oscillations represent two or more clusters close in size that, how we interpret it, can represent two things. One explanation is that the clusters alternate in gaining members over time. The real-life interpretation of this is that there are two popular narratives at the same time, both drawing new users to the DW. The other is that new vertices have entered between slices that disrupt or merge clusters per how the modularity-based algorithms produce clusters. These vertices can be users who participate in the conversations around multiple narratives, creating a link between the narratives. This corresponds with the behaviour we observed in Figures 4.12, 4.14 and 4.15, that indicate that there is reason to look into more of the large clusters than the very largest.

There is no evidence in the results that support the idea that the largest cluster in the early stages of the DW stays the largest. Usually, for the modularity-based algorithms, the first largest clusters stay large and grow as the number of vertices in the network increases (see Figures 4.3 and 4.4). We interpret this as a thread or conversation with much traction at the beginning, which either lost traction or that there were other threads with more pull for a while. However, after a period, the thread gained the attention of more people and became important for the DW once more. Only for the results produced when using the Louvain algorithm on the contact-based slices with $k = 800$ (middle plot to the left in Figure 4.17) does the largest cluster from the first slice end up as the largest cluster in the last slice. Thus, it is clear that this is not a constant attribute across slice-sets and algorithms and, therefore, not an idea worth entertaining further.

In some instances, e.g., when using the Louvain method on the accumulative slices depicted in Figure 4.16, as time traverses, the largest cluster in a slice can no longer be recognized. This effect probably comes from the fact that when adding vertices and edges to the network, the network's modularity, at some point, is maximized by dispersing the vertices to other clusters, not by keeping the cluster in question. The real-world cause of this is apparently that a conversation on Twitter over time can split into several sub-conversations that are more densely connected than the conversation as a whole.

The largest clusters resulting from the label propagation algorithm behave differently than those resulting from the modularity-based algorithms. One possible explanation lies in the way the algorithm works. According to the theory presented in Section 2.7.2, the label a vertex gets is determined through agreement with its neighbours. One of our hypotheses, presented in H1, is that a few influential people drive the DW. Combining this with the distribution of vertex activity (see Figure 4.1) showing that many vertices have a degree centrality of 1, i.e., many vertices only have one nearest-

neighbour, and that we, based on experience with using Twitter, believe that the majority of the contacts of influential people are inbound, we can assume that a substantial amount of all new vertices added to the network over time connects to only one other vertex. Adding 1-degree vertices does not break up any clusters produced by the label propagation algorithm. The new vertex would automatically agree with its nearest-neighbour about its label. There are simply no other contending labels.

In some of the plots displayed in Figures 4.16 and 4.17, we observe abrupt jumps in the size magnitude of the largest cluster between two consecutive slices. This behaviour happens sporadically for all algorithms, both for the accumulative and the contact-based slices. We hypothesize that, in rare cases, one or more vertices that share strong bonds to more than one cluster are added to the network. These can create "bridges" between previously separate clusters and combine them into one cluster. For Leiden and Louvain, this means that the modularity of the graph is no longer maximized by keeping the clusters separate but rather by combining them, and for the label propagation algorithm that the "bridge" is substantial enough for the clusters on each side to agree on one shared label.

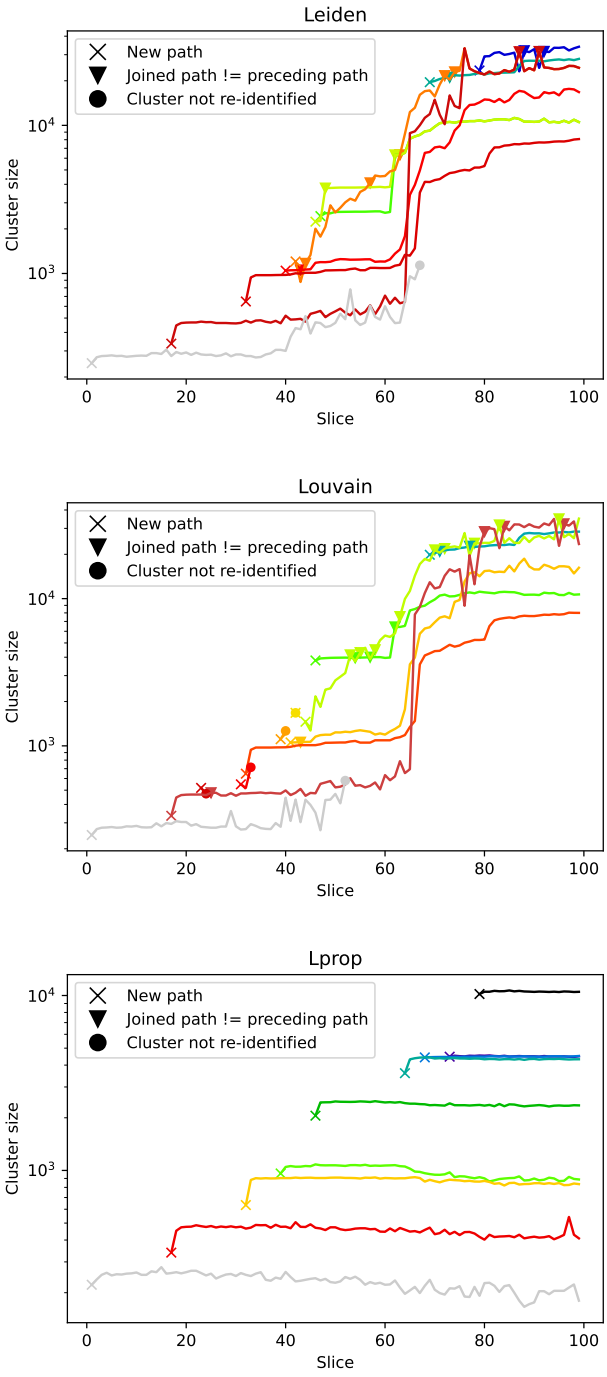

Figure 4.16: **Accumulative Slices:** Path of largest cluster in each slice through all slices. "×" marks the start of a new path, "▼" marks where the largest cluster of a slice lies on an already existing path (but not the same path as the largest cluster from the preceding slice), and • marks where a cluster was no longer re-identified in succeeding slices.

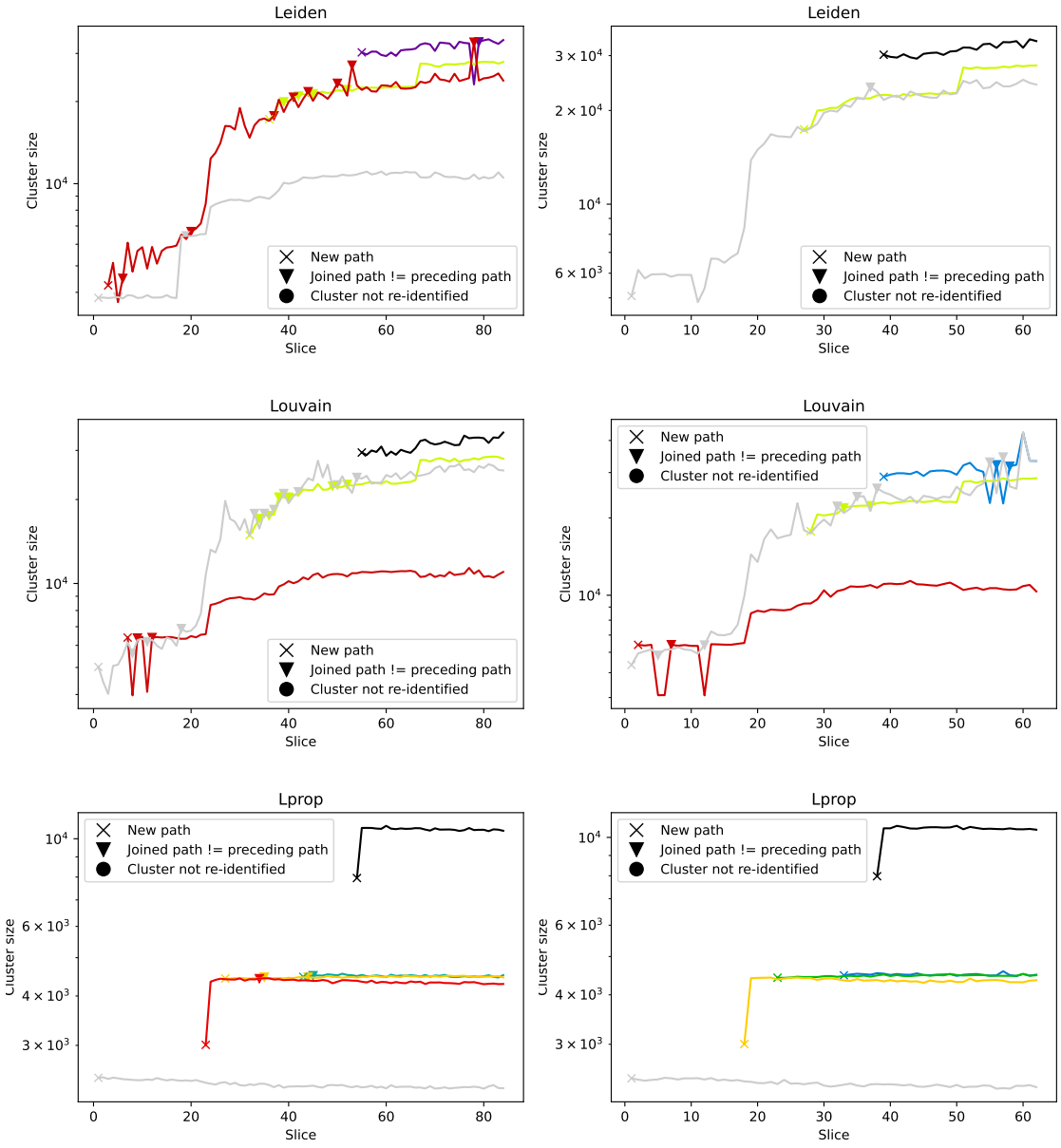

Figure 4.17: **Contact Slices $k = 600$ (left) and $k = 800$ (right):** Path of largest cluster in each slice through all slices. "×" marks the start of a new path, "▼" marks where the largest cluster of a slice lies on an already existing path (but not the same path as the largest cluster from the preceding slice), and • marks where a cluster was no longer re-identified in succeeding slices.

4.5 A Closer Look at Temporal Slices

As seen from the results in displayed in Sections 4.2 to 4.4, the contact-based slices made with a constant and high k-value gave results very close to the accumulative slices, and did not provide us with sufficient amounts of new information. For this reason, we will not pursue it further in this thesis. We emphasise that we did not have the opportunity to implement the contact-based concept entirely because of time limitations. However, we firmly believe that this way of slicing an interaction network could provide helpful insight into the interaction dynamics and the temporal evolution of the network. Except for tracking clusters through re-identifying vertices, we can extract the same information from the temporal slices as from the accumulative ones. In this section, we will be focusing on vertex activity, a feature of the network which, for our purposes, can be thoroughly examined using the temporal slices. The reasons stated above prompt us to discard the accumulative and contact-based sets moving forward, as there is no more information to be gained from them. Most of the results in this section depend only on the sub-graphs in the slices and not the clusters from community detection. However, at one point, we will be comparing vertex activity to the largest cluster produced by the Label Propagation algorithm. When making the comparison for the largest cluster produced by Leiden and Louvain, the results were very similar, so we will only be showing the results from Label Propagation as an example. Moreover, our underlying graph is constructed of pairs of vertices, i.e., there are no entirely disconnected vertices in our network, so the minimum size of a partition produced by a community detection algorithm should be 2. Leiden and Louvain tend to produce single vertex partitions whenever the modularity score does not sufficiently increase by moving the vertices into larger communities, resulting in multiple single vertex "islands". An artifact of the label propagation algorithm is that all vertices connected to at least one other vertex will agree on a shared label. Thus, the minimum size of a partition in our graph is always 2. We now take a closer look at temporal slices. We produced two new sets of slices; one using a finer time-step, $\Delta t = 4$ hours, to see if we could identify more local variations in the data, and one using a coarser one, $\Delta t = 3$ days, to see if there is additional information to be found from a potentially smoother distribution.

As re-identifying vertices across slices does not work for the temporal set, we look into centrality measures to get an overview of vertex activity.

The Vertex and Edge Distribution Across Slices

As we move forward with only the time-based slices, in figures where we have previously plotted the slice number, we now switch it out with the corresponding timeline as previously displayed in Figure 4.5. As one slice represents a given time period of activity concerning the DW on Twitter, it directly translates to a date. We see that the distributions presented in Figures 4.18 and 4.19 closely resemble the shape of the distribution in Figure 4.5, which we expected as all are purely time-based slices with

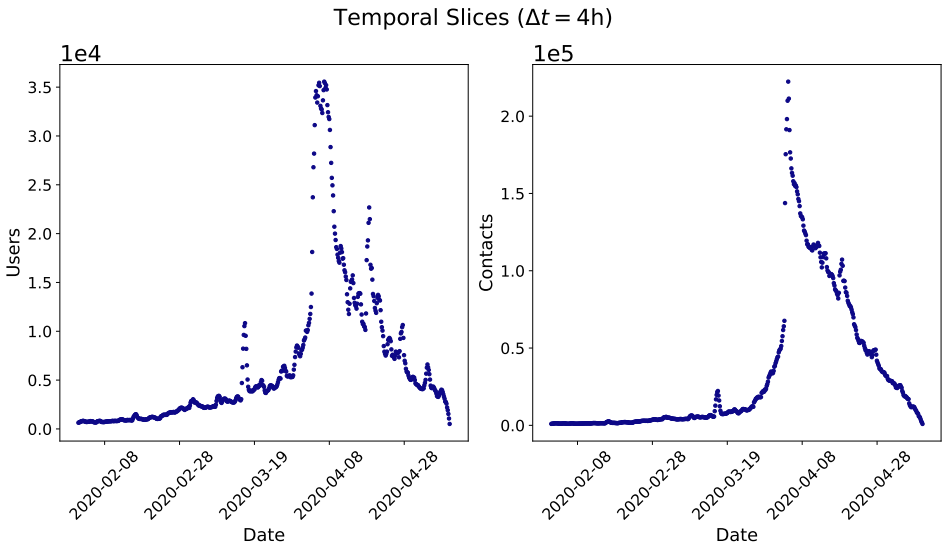

Figure 4.18: Distribution of users and contacts related to the DW over time in temporal slices of $\Delta t = 4$ hours.

different values for Δt. The main difference is the resolution which increases with a decreasing Δt. The finer resolution in Figure 4.18 lets us identify more local maxima in the number of users in the period leading up to the peak of the DW on Twitter as well as some after the global peak. If this is a general pattern for DWs spreading in OSNs or a behaviour unique to this DW in particular can not be confirmed as we do not have other datasets of DWs to compare our results to. If it is a general behaviour, such local maxima can be a property to look for when trying to identify DWs before they go viral in the future.

The general shape of the distributions of both the number of contacts and users display a transition happening around the peak of the DW on Twitter in early April 2020.

Vertex Activity

To investigate our hypotheses on whether it is the sum of many conversations between a significant fraction of all users (H2) or only a few influential users (H1) who contribute most to the evolution of the DW, we look into the degree centrality of the vertices. One way to explore the relationship between active users and the fraction of those who contribute the most is to compare the total number of contacts to the fraction of contacts made by the most active users. Therefore, we compared the sum of the contacts of vertices with a higher degree centrality than 3-std over the mean degree of each slice to the total number of contacts in our network. The cut-off for what we categorize as "the most active users" (above 3-std from the mean) was chosen through trial and error, where we tested multiple intervals to see which value would produce a sufficiently small set of vertices relative to the total.

The results are presented in Figures 4.20 to 4.22. The figures show that,

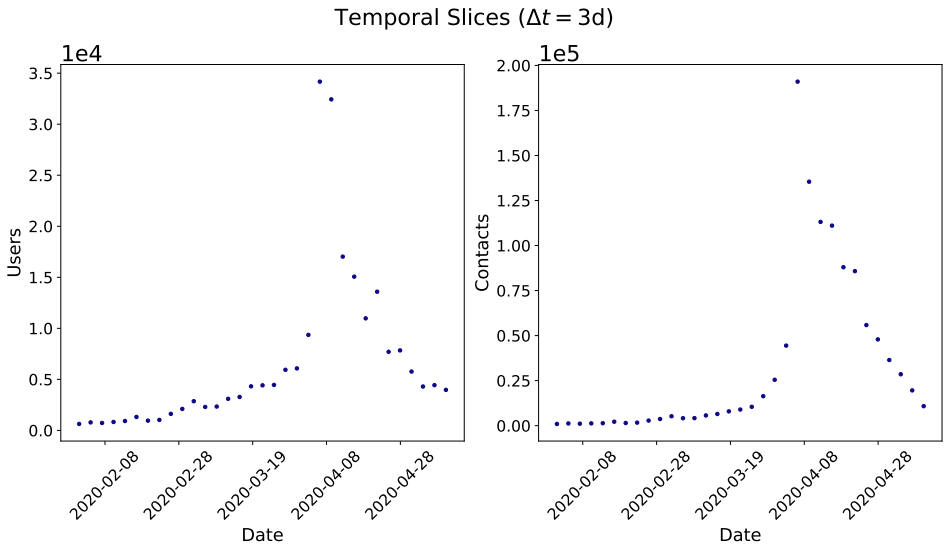

Figure 4.19: Distribution of users and contacts related to the DW over time in temporal slices of $\Delta t = 3$ days.

at times, less than 4% of the total users active in the conversation on Twitter account for more than half of the contacts made. As our graph is undirected, this measure must not be confused as 50% of the edges originating from only 4% of the users; it simply means that over 50% of the edges are either outbound from or inbound to 4% of the vertices. As pointed out in section 3.4, even though the network is undirected, it is safe to assume that the most active vertices have largely more inbound edges than outbound from the nature of Twitter as an OSN. Thus, we assume that the majority of this activity is users responding to the statuses of influential users.

These results incline us to further investigate the correlation between the most active users and the evolution of the DW by comparing the degree centrality of the single most active node in a slice and the size of the largest cluster produced using the label propagation algorithm for every slice. The figures displaying this comparison can be found in Figures 4.23 to 4.25. Straight away, we see that the two measures nearly perfectly overlap for all slices, indicating that one central user fully drives the largest cluster we observe in a given slice. Within the limits of this thesis, this result gives us strong support of hypothesis H1 as well as indicating that the driver proposed in hypothesis H1a are worth investigating more in future work on the subject of phase transitions in DW.

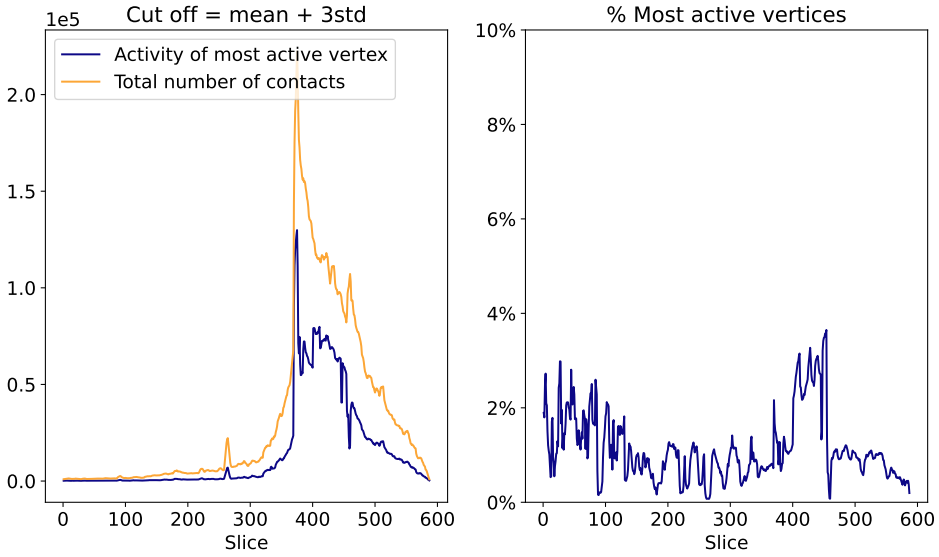

Figure 4.20: **Temporal Slices (Δt = 4h):** Comparison between the total number of contacts and the fraction of contacts attributed to the most active users of the network. The left figure displays the percent of vertices relative to the total number of vertices in a slice responsible for the contacts plotted in the right-handed figure.

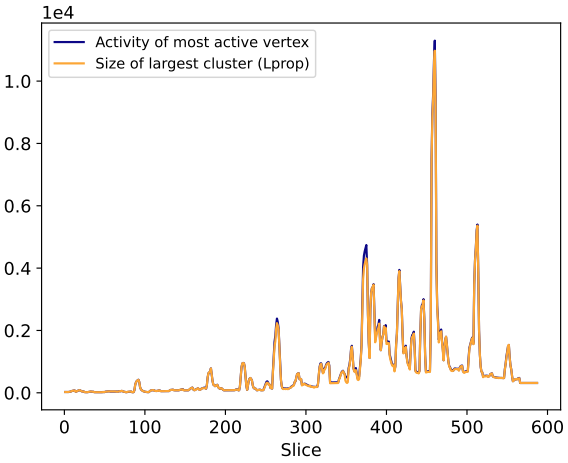

Figure 4.23: **Temporal Slices (Δt = 4h):** Comparison between the centrality degree of the single most active user in a slice and the size of the largest cluster in a slice.

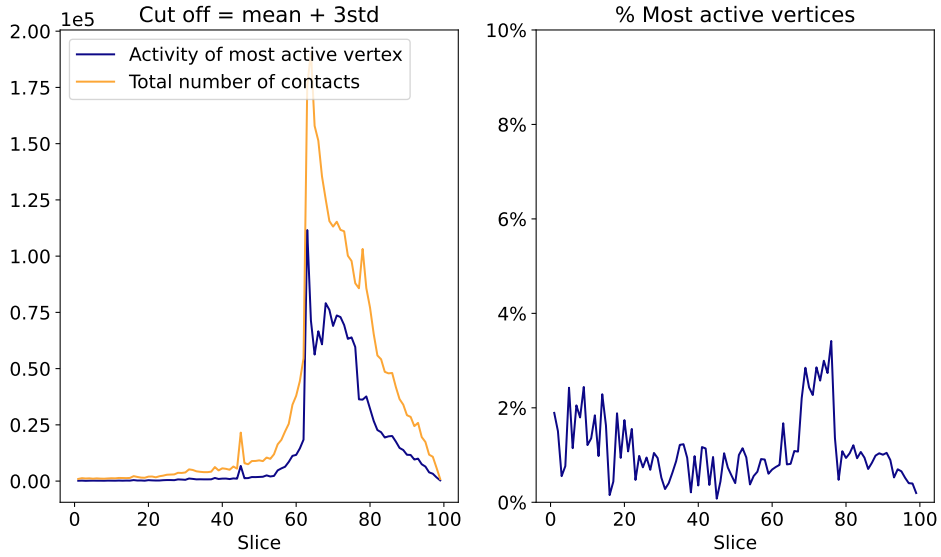

Figure 4.21: **Temporal Slices (Δt = 1d):** Comparison between the total number of contacts and the fraction of contacts attributed to the most active users of the network. The left figure displays the percent of vertices relative to the total number of vertices in a slice responsible for the contacts plotted in the right-handed figure.

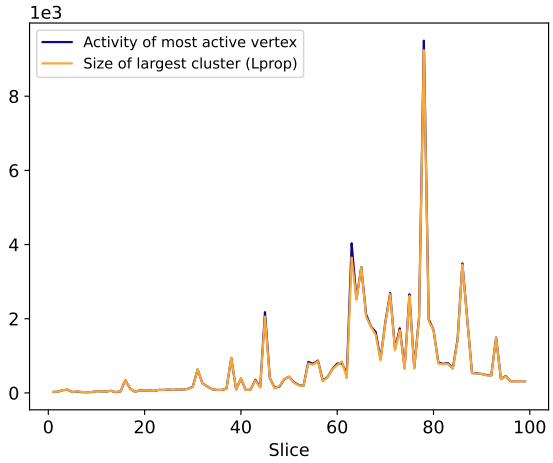

Figure 4.24: **Temporal Slices (Δt = 1d):** Comparison between the centrality degree of the single most active user in a slice and the size of the largest cluster in a slice.

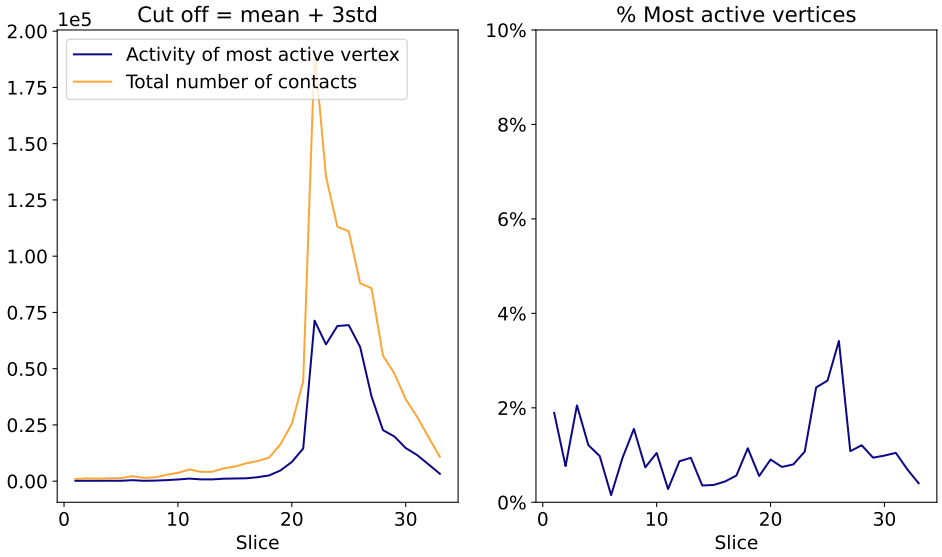

Figure 4.22: **Temporal Slices (Δt = 3d):** Comparison between the total number of contacts and the fraction of contacts attributed to the most active users of the network. The left figure displays the percent of vertices relative to the total number of vertices in a slice responsible for the contacts plotted in the right-handed figure.

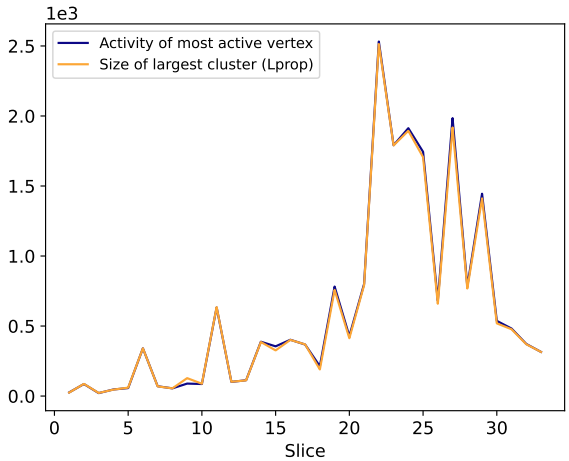

Figure 4.25: **Temporal Slices (Δt = 3d):** Comparison between the centrality degree of the single most active user in a slice and the size of the largest cluster in a slice.

4.5.1 Average Nearest Neighbour Degree

As described in Section 2.6, average nearest neighbour degree (ANND) can provide insight into the nature of how vertices connect in a network. We calculated the "average" ANND-function for the sets of slices, $K(k)$, by averaging over the individual ANND-function for each slice, i.e., the $K(k)$

is both an average over the neighbourhoods but also an average over time. We divided our timeline into the three phases we found in Section 4.2.1 and performed the same calculation on each of the phases to see if the ANND-function changed significantly during the progression of the DW. The figures showing the average ANND-function over the entire sets of slices are shown in Figures 4.26 to 4.28. Note that both axes are logarithmically scaled. Comparing the results for the different Δts, we see that they all display the same overall relation but with varying levels of resolution, as expected. The ANND-function indicates that the network leans towards a disassortative nature but is not clear enough for us to identify a suitable approximation. To be able to define the networks degree assortativity strictly, we would want to be able to approximate the ANND-function as some relation $K(k) \approx ak^{\mu}$, which would enable us to interpret the sign of μ to reveal the nature of the degree correlation [50]. It is clear from the figures that the data does not seem to follow such a trend or any power law at all, and so attempting to fit it would, at best, provide a very uncertain fit.

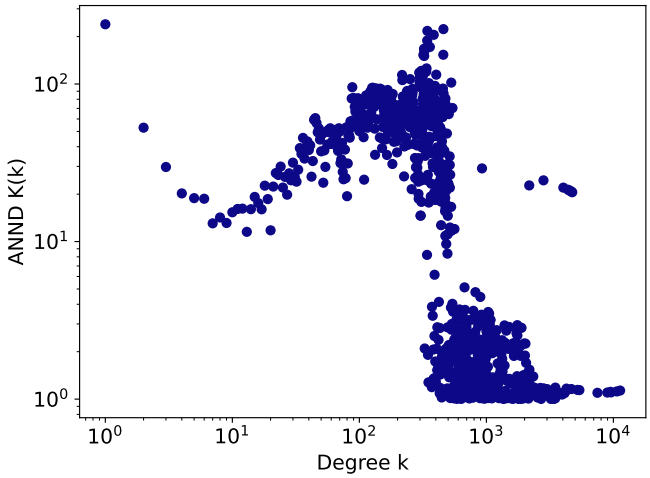

Figure 4.26: **Temporal Slices (Δt = 4h):** Average nearest neighbour degree (ANND) function calculated as average of all slices in set. Fit using ordinary least squares.

What we will do is analyze the trends shown in the figure, for which we turn to the results from the ANND-function for the different phases shown in Figures 4.29 to 4.31. From all figures displaying phase 0, before the peak on Twitter, the data follows a much straighter line than when looking at the entire time interval. This pattern indicates that in the early phase of the DW, the degree connectivity of its underlying network follows a power law with a negative exponent, i.e., it has a disassortative nature. As previously mentioned in Section 2.6, the article *Why social networks are different from other types of networks* by M. E. J. Newman and Juyong Park reports social networks as typically assortative in terms of degree correlation [53]. That result is interesting in light of our results, in general, showing the network being disassortative, and as by intuition, one might assume that

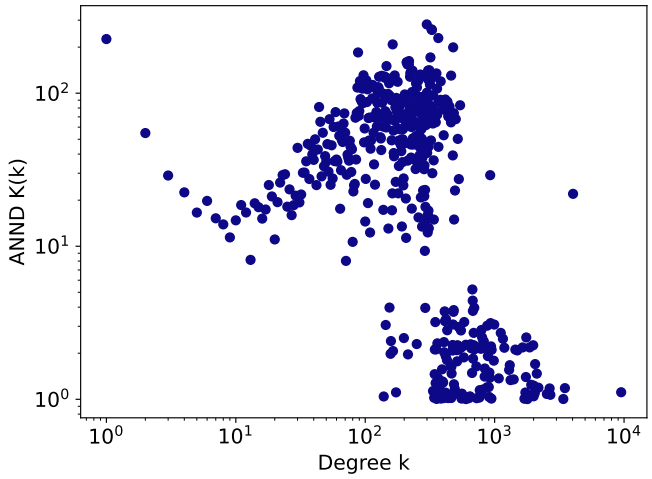

Figure 4.27: **Temporal Slices (Δt = 1d):** Average nearest neighbour degree (ANND) function calculated as average of all slices in set. Fit using ordinary least squares.

social networks should display a disassortative nature just from there being less very well connected people than sparsely connected people in the world. Imagine the following pool of a well-connected individual, e.g., a politician or a celebrity on an OSN. There is a reasonable probability of this individual being connected to other well-connected people. Still, this number relative to the number of individuals from the general population following them is likely very small. Individuals of the general population are likely more sparsely connected, so the average connectivity of the people connected to densely connected people should be relatively small. Suppose we now imagine the contact pool of a sparsely connected individual, where the total number of contacts is small. In that case, it will only take one or a few densely connected contacts to skew the average degree centrality of its neighbours. Combining the expectation for densely and sparsely connected individuals argues for a disassortative nature of the network's degree correlation. We base this hypothesis on intuitions made from our own experience of the nature of humans interacting on OSNs, the distribution of "sheep vs. shepherds" in the general population, and the assumption that this nature of the following network translates to interaction networks. The only results we can use to test the hypothesis are those obtained from our unique dataset. Therefore, we suggest this as a possible area of interest for future work on the subject. In our dataset, the trend is overall disassortative, with the phase before the peak on Twitter seemingly following a power law. If what we expect from the interaction network of Twitter, in general, is an assortative degree correlation, such disassortative tendencies could be something worth looking into as an early indication of a DW, but this would require more research and more data to be confirmed as a distinct trait of a DW evolving on an OSN, as it could be a mere coincident or not correlated to the evolution of the DW at all.

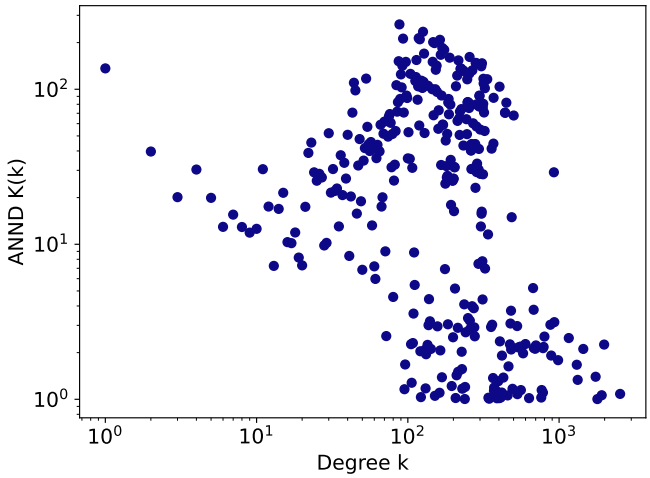

Figure 4.28: **Temporal Slices (Δt = 3d):** Average nearest neighbour degree (ANND) function calculated as average of all slices in set. Fit using ordinary least squares.

Looking at phase 1 (during the peak on Twitter), the ANND-function does not seem to follow a power law. The relationship is seemingly disassortative for vertices with $k < 10$, but this trend does not continue for higher degrees. Instead, the correlation turns assortative before we observe a rather abrupt change in ANND for vertices with degrees close to $10^{2.5}$ or approximately $k \approx 300 - 400$. Here, the ANND-function drops before flattening out for higher degrees. This result is not straightforward to interpret, as to the best of our knowledge, there is no clear explanation as to why the ANND-function should drop by nearly a magnitude of 10^2 between vertices of degree ~ 200 and vertices of degree ~ 400. This behaviour can be an indication of some "critical" area of degree values for the underlying network that make up a DW during its peak, where the nature of the network's degree correlation changes and so be of interest for future research.

The period after the peak, phase 3, shows for all Δt the most erratic behavior of all the phases. This thesis aims to provide an initial foundation for predicting dangerous conspiracies in complex networks. As a result, our focus has been chiefly on the period leading up to a DW going viral on an OSN and the period during its peak, but not the period after. Since the information obtainable from phase 3 can not help us towards this goal, we choose not to analyze further or interpret the result.

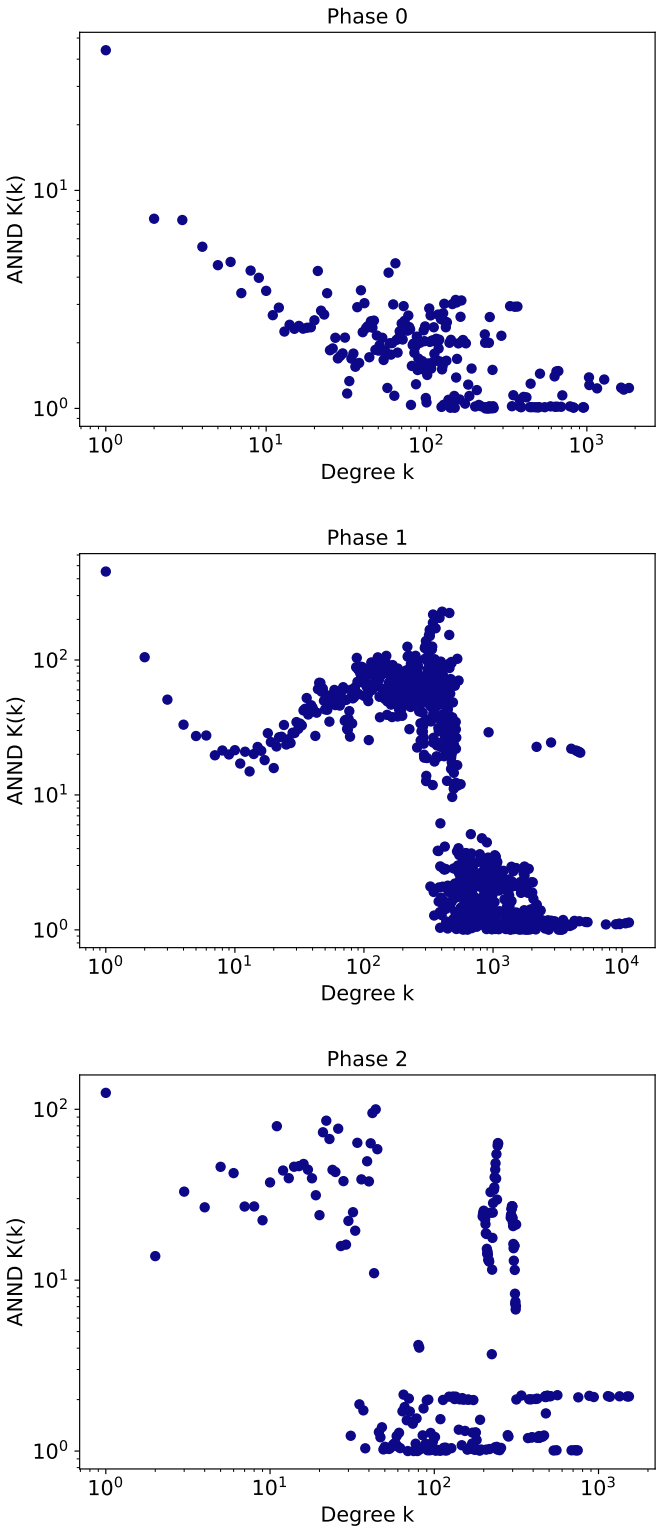

Figure 4.29: **Temporal Slices (Δt = 4h):** Average nearest neighbour degree (ANND) function calculated from three separate phases as described in the text. Fit using ordinary least squares.

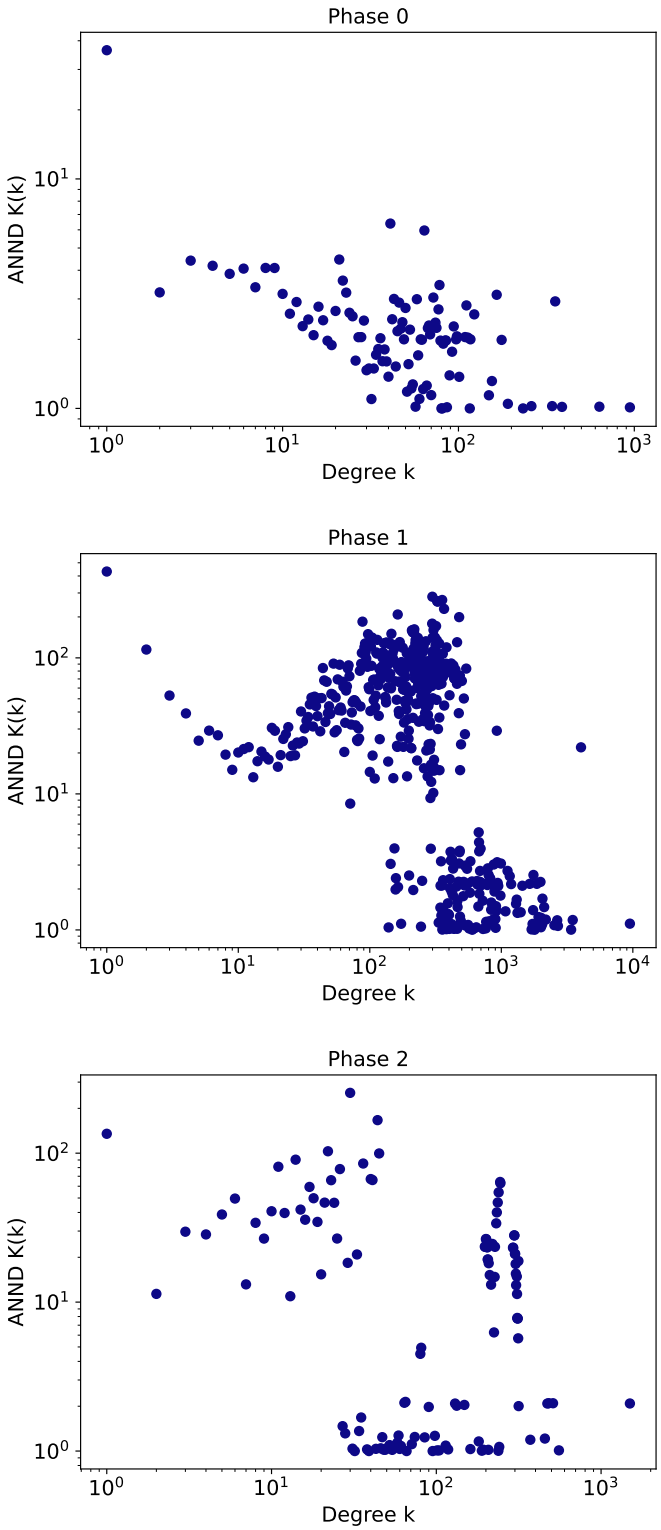

Figure 4.30: **Temporal Slices (Δt = 1d):** Average nearest neighbour degree (ANND) function calculated from three separate phases as described in the text. Fit using ordinary least squares.

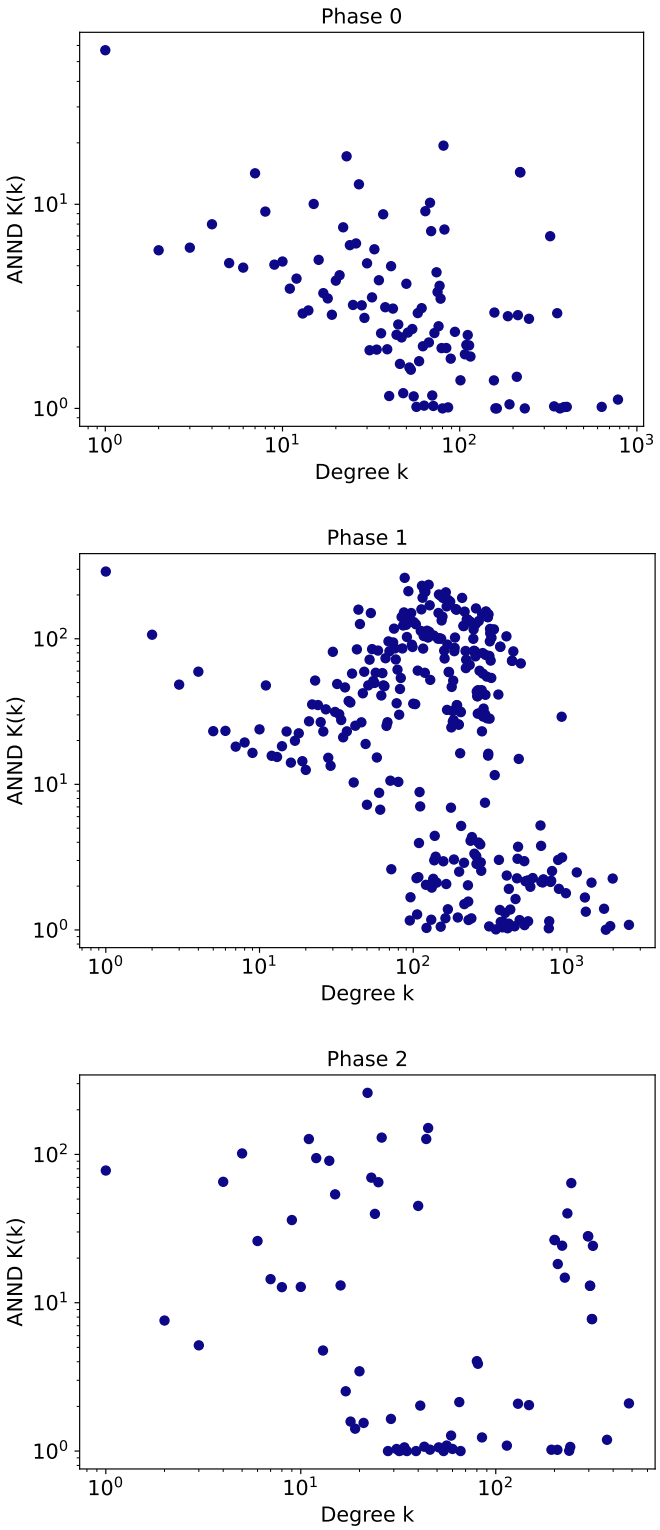

Figure 4.31: **Temporal Slices (Δt = 3d):** Average nearest neighbour degree (ANND) function calculated from three separate phases as described in the text. Fit using ordinary least squares.

4.6 Summarizing Discussion

Having introduced all the findings of this thesis, we present in the following a summary aiming to gather the overall analysis of the results.

The main finding of this thesis lies in the exciting behaviour we observed in the distribution of statuses and contacts over time, Figure 4.5. The shape of the distribution displays a transition during the period where the DW peaked on Twitter. This feature indicates a transitional nature of the DW which supports the idea of modeling such spreading of misinformation with harmful real-world consequences as a phase transition in the underlying interaction network.

The distribution of vertex centrality degrees in Figure 4.1 approximately follows a power law with a negative exponent, which argues for a close to scale-free network. This indicates that the overall distribution of degree centrality in a DW is not a unique quality as social networks often are weakly scale-free [63, 64].

The distributions of cluster sizes in the different sets of slices, Figures 4.6 to 4.8, all follow a power law but wee see it most clearly from the results produced with the Label Propagation algorithm. This result correlates well with the degree distribution following an apparent power law. In Figures 4.9 to 4.11 we observe that all three of the cluster detection algorithms we implemented favour small clusters of size < 10. This is likely an artifact of there being many more vertices with a low degree centrality, and so the algorithms perceive these as many small clusters. On the other hand, as the results for vertex activity show that the largest cluster consists of one vertex with high degree centrality connected to many vertices of low degree centrality, we cannot confidently draw any conclusions. As all three algorithms result in a magnitude of small clusters, it would be interesting to investigate further their impact on the dynamics of the overall discourse.

Both the results depicting the number of vertices in the largest clusters, Figures 4.12 to 4.15, and the paths of the largest cluster, Figures fig. 4.16 and 4.17, show that there is not one single cluster with a size of magnitude much greater than all other. The peak in Figure 4.13 corresponds with the peak of the conversation on Twitter and thus implies a possible centralization of the conversation. This centralization can be an interesting feature in modeling and predicting these phenomena, as it could provide an early indication of smaller misinformation events turning into a DW. The results in Figures 4.16 and 4.17 show that the largest cluster generally does not stay the largest, nor are we able to pinpoint a specific area where a new largest cluster that stays the largest throughout appears. This points to a DW not emerging from a single origin but rather a multiple of origins. We interpret multiple origins in terms of a social network as multiple conversations with similarities in the main message but with different overall narratives. These figures also display oscillations, which we interpret as two or more large clusters going back and forth in being the largest. This strongly indicates the existence of multiple important narratives in the DW, which argues for a focus on more than only the single largest cluster in future investigations.

The community detection algorithms and centrality measures imple-

mented are well-tested methods, and the comparable results achieved for the clusters produced by Leiden, Louvain, and Label Propagation, confirm successful implementation. The Label Propagation algorithm ensures no single-vertex clusters for a network containing only vertices of degrees > 1, while Leiden and Louvain do not. This lets us determine Label propagation as more suited for the task based on attaining a more realistic distribution of clusters. The underlying cause of this attribute is that the Label Propagation algorithm only stops iterating when there are no more vertices that change labels. Leiden and Louvain stop after the system's modularity does not increase more than some tolerance when moving vertices. Even when this tolerance is set to a very small number, the algorithms can, like in the case of our network, prematurely stop, resulting in many 1-vertex clusters. Using Label Propagation is thus a method of optimizing the cluster distribution without adding external supervision.

We do not find evidence to support H3 in the results. Observing the number of vertices in the largest cluster in the temporal slices, Figure 4.13, at no point, do we see a significant growth in cluster size across slices to indicate a critical size. In certain slices, the size of the largest cluster increases greatly from the former slice, but it does neither stay significantly large nor strictly increase in size in the next slices. However, even though it is not evident that the biggest cluster, i.e., main discussion or narrative, "eats up" smaller discourses when reaching a critical size, we can show a centralization for the biggest 10% of clusters (see Figure 4.13). This could be another quality worth investigating as a potential driver of the observed transition.

The activity of the most active vertices vs. the total number of contacts in each slice, Figures 4.20 to 4.22, show that, at times, under 4% of the users account for over half of the total contacts. This comparison must not be interpreted as one person tweeting a lot, but rather that only 4% of vertices are, most likely, inbound to over 50% of the edges. This result underlines the importance of the contributions made by influential users in the DW, and along with the activity of the most active vertex nearly perfectly overlapping the size of the largest cluster in Figures 4.23 to 4.25. Furthermore, it provides support for hypotheses H1 and H1a, while undermining hypothesis H2. In other words, influential users are central in drawing other users to the conversation. Moreover, it is not necessarily conversations back and forth between users that drive the conversation but rather the activity of a small set of influential users that provokes responses from big groups of others. Based on these results, we suggest that future work should be conducted focusing on the activity of central vertices.

The disassortative nature of the ANND-function, shown in Figures 4.26 to 4.31 contradicts expectations from literature [53]. Disassortative nature means a strong tendency for well-connected users to connect to sparsely connected users and visa versa, which we, in the context of Twitter statuses, see as an artifact of many users responding to the post of an influential user. We do not believe it to be influential users having back and forth conversations with many other users. Suppose this behaviour of the ANND-function is unique and characteristic to a social network during a DW when compared to the ANND-function of the network on average. In that case, it

can be an interesting feature for modeling and predicting DWs.

We found that the naive way of slicing using a constant maximum neighbourhood size did not provide more information about the network dynamics than can be gained from accumulative slices. Furthermore, we assume that given more time to develop the contact-based slicing method, it could provide new information about the structure and evolution of the network. These developments include implementing a more complex criteria function based on individual vertices (see Section 3.3) and adding an aspect of time.

We intended to look at more centrality measures for vertices, more specifically betweenness and closeness centrality, that we presented in Section 2.5. Calculating these measures for our network proved too time-consuming, and we did not have the opportunity to obtain results for them during this project.

Chapter 5

Conclusion

In this thesis, we first introduced the reader to the concepts and methods needed to follow the experiments and understand the analysis of the results. This included defining terms related to Digital Wildfire (DW) and online social network (OSN), defining networks and methods related to complex network theory, and a walk-through of the applied community detection algorithms.

After covering the necessary background, we presented our contributions in terms of methods and the overall experimental setup. We specifically focused on the graph representation of Twitter's underlying interaction network. Furthermore, we went into detail concerning the methods we developed for graph slicing. In our experiments, we analyzed the derived network slices using Leiden, Louvain, and Label Propagation algorithms. This approach allows for identifying possible key qualities in the structure of the conversations. In addition, we looked at measures such as degree centrality and assortativity to describe vertex activity, which allowed for uncovering more possibly characteristic patterns and behaviours of a DW.

In the following, we present our main findings and proposals for future work, as well as our final thoughts and comments regarding the ongoing threat posed by DWs and the importance of developing methods that enable us to predict and stop them.

Conclusion

This thesis is the first aiming to model the dynamics of the spread of misinformation with harmful real-world consequences, known as Digital Wildfires, in complex temporal interaction networks. Hence, this work is also the first attempt to understand and extract more universal patterns from the communication underlying a DW, beyond analyzing individual information cascades. The ultimate goal is to develop methods for predicting the spread of misinformation with harmful consequences, something we are convinced is only possible through understanding these phenomena on a societal scale.

Because of the exploratory character of the thesis, the results uncovered during the course of this work are diverse and of a general nature. However, the main the main finding of the thesis is:

> *The dynamics of the communication underlying Digital Wildfires show similarities to phase transitions.*

We suggest that modeling DWs as phase transitions in their underlying interaction networks is a promising strategy. However, we first aim to understand the characteristics of DWs and the transitions that transpire in them. Thus, in Chapter 3 we introduce four hypotheses based on our understanding of the nature of DWs. These hypotheses are carefully chosen to identify patterns in the underlying communications.

We found evidence supporting the **Centralized Core** hypothesis (H1) and **The Influencer Transition** (H1a), which we argue is sufficient to propose a possible driver, namely, influential users. The degree of the vertex with the largest degree centrality nearly perfectly overlaps with the size of the largest cluster.

Along with results showing that a minor group of vertices, at times, are inbound to over half of the edges, the overlap of activity and cluster size indicates that central vertices are major in drawing users to and driving the conversation on a large scale. Moreover, these results undermine the **Dispersed Core** hypothesis (H2). One can argue that an influencer is not influential without a large number of individuals to influence. However, the influencer is the catalyst in this equation, and the increasing number of active users is only the reaction.

Our findings undermine **The "Black Hole" Transition** hypothesis (H3), i.e., we do not observe a critical size at which a cluster starts "consuming" vertices at a substantially higher rate. However, there are oscillations between the largest communities, i.e., the main narratives, and an apparent centralization of the discourse into the largest clusters. Moreover, we observed that the largest cluster at the beginning of the DW does not tend to stay the largest as the DW transpires, indicating that a DW has its origins from multiple sources. Thus, contrary to expectations, the rapid growth in the number of tweets with misinformation content cannot be explained by a single rapidly growing narrative that individuals are constantly joining. Instead, the rapid growth of the overall phenomena is likely caused by different narratives connecting.

The main limitation of the thesis is the uniqueness of the dataset. It requires a substantial amount of data, either from other DW in online social networks or from sources who can capture the essence of a DW and synthesize realistic data, to enable us to identify unique features that are general only to the spread of misinformation on OSN. These features are essential to produce a comprehensive model; thus, it would be impossible at this point to directly model a DW as a phase transition.

To conclude if we answered the main research question, we once more break it down into smaller questions.

Focusing on the *modeling of social networks and temporal data*, we produced an underlying graph consisting of all the contacts and statuses in the data, from which we produced three types of sets of slices; accumulative, contact-based, and temporal. The accumulative and contact-based sets provided insight into the movement and evolution of the cluster in the network. Given the time window of this thesis, we could not finalize the concept and implementation of the contact-based slices, and using a constant and high k-value in the criteria function resulted in sets of slices too close to the accumulative one to provide any new insight. The temporal slices allowed us to investigate the overall temporal evolution of the network's vertices and their activity, providing insight into the different stages of a DW spreading in an OSN.

In terms of *analysis*, we again highlight the main finding and discussion around the hypotheses in the preceding paragraphs. In addition, we divide the DW into three distinct phases; one before the peak on Twitter, one during the peak on Twitter, and one after the peak on Twitter. Although all three displayed an overall disassortative nature, the average nearest neighbour degree of these phases differed significantly, something we believe could be an exciting area for further investigation and possibly a unique trait of a DW.

A secondary finding is identifying Label Propagation, in our case, as the superior algorithm for cluster detection. Label Propagation guarantees a minimum cluster size of 2 in a network with no isolated vertices, resulting in a more realistic distribution of clusters. Thus, allowing for optimization without adding supervision.

To summarize, discovering transitions in the underlying follower network of a Digital Wildfire is easy, but identifying the qualities and the critical phenomena necessary to model it as a phase transition is difficult. However, we argue that modeling and predicting misinformation spreading is a problem worth investing.

Sadly, today, the reality of misinformation on the internet is grim. A very current example is the ongoing war in Ukraine, showing that modern warfare is fought on all fronts, including online. The Russian government is preventing the truth about war crimes committed in Ukraine from spreading to the east and to the Russian population while spewing out false claims of righteousness. This is only one of the scary examples of how the censorship of factual information and the spreading of misinformation can be used to such a degree that nearly an entire population is kept in the dark.

Another example is how digital communication impacts the development

of conflicts and the spread of hate speech in Ethiopia and Mali, countries where people are killed daily over conflicts linked to inequalities and ethical- and cultural backgrounds. The dark realities referenced above underline the importance of developing methods helping us to prevent misinformation spreading online from causing harmful consequences. However, it is important to note that understanding the dynamics of information spread can be a powerful tool, both for those wanting to use it for good and for those with malicious intents. In the best case, it is used by journalists and content moderators in democratic societies to spot the emergence of a new extremist group or a novel disinformation campaign. However, it could also be used by a totalitarian regime to identify and control democratic uproar in social networks early on. Technology alone cannot solve social problems. Solving the problem of DW and the resulting harmful consequences will necessarily involve technical tools. Still, education, media competence and an informed, critical public are just as essential. Most importantly, we need to maintain a democratic society that keeps those in power of these tools accountable so that they are used for the common good.

5.1 Future Work

As conveyed by the title of this thesis, the main objective of our work is to take the first steps toward predicting the spread of misinformation with harmful consequences. Our goal is to lay the foundation for future work on the subject and provide suggestions for directions to take. Mainly, we take the first steps towards modeling a Digital Wildfire (DW) as a phase transition by looking for evidence of transitions and potential drivers.

The addition of more data containing real DWs or realistic, synthesized data could, in the future, enable us to generalize our findings and discover more traits characteristic to a DW. The final goal is a mathematical model that can capture the general nature of a DW, which can be used to identify discourse on online social network (OSN) with the potential for real-world harm.

A model can be expanded by combining interaction-based networks with natural language processing. Analyzing the content of conversations along with the information about the dynamics of the interaction network promises a more detailed picture including the kind of misinformation spreading. Are there topics that are more prone to attract the attention of the masses than others? Do "conspiracy theorists" write in a different style than the average OSN user, and can we quantify these differences?

We propose the idea of a more complex function for slicing, which considers the running average of the neighbourhood size of each vertex. Since the scope of this thesis allows only the development of a primitive prototype, we suggest refining. However, we assume that when properly executed, the contact-based slicing of interaction networks provide useful information about how individuals join, leave, or switch between sub-conversations during a DW.

In addition, we propose applying a learning algorithm to optimize k-

values and enable us to find a "k-function" that yields the optimal values for creating slices that best describe the evolution of communities. At least, we imagine it could help us decide if we should focus more or less on the effect small neighbourhoods have on the large communities and the overall discourse.

Lastly, concerning contact-based slices, we wish to introduce the idea of a function that adds yet another level of complexity, namely time. The goal is to define a function dependent on both time and number of neighbours, $\lambda(t, k)$, which decides whether or not to forget an edge based on both the number of neighbours a vertex connected to that edge has and the time. This means that we combine the vertex-varying k with a time t_{old} so that even if the k-criteria is not fulfilled, an edge deemed "too old" will still be removed. This approach assumes that conversations that happened, for example, a month ago, will not still be influential to the current dynamics of the spread of the network.

Taking a more Bayesian approach to modeling the spread of misinformation in OSN, we believe it could be interesting to combine follower networks and interaction networks. Using the degree centrality of users in the follower network as a prior, we can continuously update the prior using the degree centrality of users in an interaction network as a likelihood to form a posterior, hopefully letting us predict the importance/centrality of users. This could even be extended to a special case of a compartmental model, similar to those used to model disease epidemics. Imagine the users of an OSN are divided into categories in relation to a conspiracy spreading on the platform. For example, users who have not yet heard of the conspiracy are placed in one category, users exposed to the conspiracy are placed in another category, users actively participating in spreading the conspiracy are placed in yet another, etc. Assigning individual probabilities based on vertex- and edge attributes and location in the network, we could try to identify fitting differential equations to describe how users move in and out of the different categories and the spread of the conspiracy in the network.

A quality of the DW we deem of interest for future research is the tendency of centralization we observed when looking at the size of the largest network relative to the total number of vertices. This apparent centralization or synchronization of the communities could be an essential quality in identifying a DW before it turns significant. We would like to see answered in the future whether singular individuals or larger groups drive this tendency.

Bibliography

[1] V A Traag, L Waltman, and N J van Eck. "From Louvain to Leiden: guaranteeing well-connected communities." In: *Scientific Reports* 9.1 (2019), p. 5233. ISSN: 2045-2322. DOI: 10.1038/s41598-019-41695-z. URL: https://doi.org/10.1038/s41598-019-41695-z.

[2] Vincent D Blondel et al. "Fast unfolding of communities in large networks." In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2008/10/p10008. URL: http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[3] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." In: *Physical Review E* 76 (3 Sept. 2007), p. 036106. DOI: 10.1103/PhysRevE.76.036106. URL: https://link.aps.org/doi/10.1103/PhysRevE.76.036106.

[4] Board of Regents of the University System of Georgia. *A Brief History of the Internet*. [Online; accessed 09-Jun-2022]. URL: https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml.

[5] Bernard Marr. *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read*. [Online; accessed 16-Dec-2021]. May 2018. URL: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/.

[6] Shakuntala Banaji et al. "WhatsApp vigilantes: An exploration of citizen reception and circulation of WhatsApp misinformation linked to mob violence in India." In: *London School of Economics and Political Science* (2019). [Online; accessed 03-Apr-2022].

[7] Nikolaus von Twickel. *Fake twitter account pumps up oil prices*. [Online; accessed 03-Apr-2022]. Aug. 2012. URL: https://www.themoscowtimes.com/2012/08/07/fake-twitter-account-pumps-up-oil-prices-a16852.

[8] Soutik Biswas. *Social Media and the India Exodus*. Aug. 2012. URL: https://www.bbc.com/news/world-asia-india-19292572.

[9] Marc Fisher, John Woodrow Cox, and Peter Hermann. "Pizzagate: From rumor, to hashtag, to gunfire in DC." In: *Washington Post* 6 (2016), pp. 8410–8415.

[10] *The Digital Services Act: ensuring a safe and accountable online environment*. [Online; accessed 16-Dec-2021]. URL: https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/digital-services-act-ensuring-safe-and-accountable-online-environment_en.

[11] Lee Howell. *Digital wildfires in a hyperconnected world*. [Online; accessed 22-Mar-2022]. 2013. URL: http://reports.weforum.org/global-risks-2013/risk-case-1/digital-wildfires-in-a-hyperconnected-world/.

[12] Johannes Langguth et al. "COVID-19 and 5G conspiracy theories: long term observation of a digital wildfire." In: *International Journal of Data Science and Analytics* (May 2022). ISSN: 2364-4168. DOI: 10.1007/s41060-022-00322-3. URL: https://doi.org/10.1007/s41060-022-00322-3.

[13] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks." In: *Physical Review E* 69 (2 Feb. 2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[14] A.-L Barabási. *Network science*. Cambridge University Press, 2016. ISBN: 9781107076266.

[15] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks." In: *Reviews of Modern Physics* 74.1 (Jan. 2002), pp. 47–97. DOI: 10.1103/revmodphys.74.47. URL: https://doi.org/10.1103%5C%2Frevmodphys.74.47.

[16] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks." In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509. eprint: https://www.science.org/doi/pdf/10.1126/science.286.5439.509. URL: https://www.science.org/doi/abs/10.1126/science.286.5439.509.

[17] M. E. J. Newman. "Assortative Mixing in Networks." In: *Physical Review Letters* 89 (20 Oct. 2002), p. 208701. DOI: 10.1103/PhysRevLett.89.208701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.89.208701.

[18] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010, p. 224. DOI: 10.1093/acprof:oso/9780199206650.001.0001. URL: https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780199206650.001.0001/acprof-9780199206650.

[19] M. E. J. Newman. "Modularity and community structure in networks." In: *Proceedings of the National Academy of Sciences* 103.23 (May 2006), pp. 8577–8582. ISSN: 1091-6490. DOI: 10.1073/pnas.0601602103. URL: http://dx.doi.org/10.1073/pnas.0601602103.

[20] Santo Fortunato. "Community detection in graphs." In: *Physics Reports* 486.3-5 (Feb. 2010), pp. 75–174. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2009.11.002. URL: http://dx.doi.org/10.1016/j.physrep.2009.11.002.

[21] Wikipedia contributors. *Social physics — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Social_physics&oldid=1059955197. [Online; accessed 19-Jan-2022]. 2021.

[22] Daniel Thilo Schroeder et al. "WICO Graph: a Labeled Dataset of Twitter Subgraphs based on Conspiracy Theory and 5G-Corona Misinformation Tweets." In: 2021.

[23] Konstantin Pogorelov et al. "Fakenews: Corona virus and 5g conspiracy task at mediaeval 2020." In: *MediaEval 2020 Workshop*. 2020.

[24] Soroush Vosoughi, Deb Roy, and Sinan Aral. "The spread of true and false news online." In: *Science* 359.6380 (2018), pp. 1146–1151. DOI: 10.1126/science.aap9559. eprint: https://www.science.org/doi/pdf/10.1126/science.aap9559. URL: https://www.science.org/doi/abs/10.1126/science.aap9559.

[25] David Sayce. *The Number of tweets per day in 2020*. [Online; accessed 20-Sep-2021]. Dec. 2020. URL: https://www.dsayce.com/social-media/tweets-day/.

[26] Wikipedia contributors. *Misinformation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Misinformation&oldid=1044421500. [Online; accessed 20-Sep-2021]. 2021.

[27] Wikipedia contributors. *Disinformation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Disinformation&oldid=1044688161. [Online; accessed 20-Sep-2021]. 2021.

[28] David M. J. Lazer et al. "The science of fake news." In: *Science* 359.6380 (2018), pp. 1094–1096. DOI: 10.1126/science.aao2998. eprint: https://www.science.org/doi/pdf/10.1126/science.aao2998. URL: https://www.science.org/doi/abs/10.1126/science.aao2998.

[29] Merriam-Webster.com. *Conspiracy Theory*. https://www.merriam-webster.com/dictionary/conspiracy%20theory. [Online; accessed 13-Oct-2021]. 2021.

[30] Johannes Langguth et al. "COVID-19 and 5G conspiracy theories: long term observation of a digital wildfire." In: *International Journal of Data Science and Analytics* (2022), pp. 1–18.

[31] Leo Kelion. *Coronavirus: 20 suspected phone mast attacks over Easter*. [Online; accessed 22-Mar-2022]. Apr. 2020. URL: https://www.bbc.com/news/technology-52281315.

[32] Katie Collins. *Violence, arson, abuse: The real-world consequences of those false 5G conspiracies*. [Online; accessed 11-Oct-2021]. July 2020. URL: https://www.cnet.com/tech/services-and-software/fake-5g-coronavirus-theories-have-real-world-consequences/.

[33] Paula Gilbert. *Vodacom, MTN towers burnt in SA by alleged 5G conspiracy theorists*. [Online; accessed 11-Oct-2021]. Jan. 2021. URL: http://www.connectingafrica.com/author.asp?section_id=761&doc_id=766499&.

[34] News Staff. *Police say cell phone tower fire in Scarborough considered arson*. [Online; accessed 11-Oct-2021]. Mar. 2021. URL: https://toronto.citynews.ca/2021/03/31/fire-crews-battle-burning-cell-phone-tower-in-scarborough/.

[35] Yan Holtz. *Network Diagram*. [Online; accessed 11-Oct-2021]. URL: https://www.data-to-viz.com/graph/network.html.

[36] Wikipedia contributors. *Social network — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Social_network&oldid=1045629582. [Online; accessed 11-Oct-2021]. 2021.

[37] Jongkwang Kim and Thomas Wilhelm. "What is a complex graph?" In: *Physica A: Statistical Mechanics and its Applications* 387.11 (2008), pp. 2637–2652. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2008.01.015. URL: https://www.sciencedirect.com/science/article/pii/S0378437108000319.

[38] S. Boccaletti et al. "Complex networks: Structure and dynamics." In: *Physics Reports* 424.4 (2006), pp. 175–308. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2005.10.009. URL: https://www.sciencedirect.com/science/article/pii/S037015730500462X.

[39] A.-L Barabási. "Network science." In: Cambridge University Press, 2016. Chap. The scale-free property. ISBN: 9781107076266.

[40] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks." In: *Nature* 393.6684 (June 1998), pp. 440–442. ISSN: 1476-4687. DOI: 10.1038/30918. URL: https://doi.org/10.1038/30918.

[41] Petter Holme and Jari Saramäki. "Temporal networks." In: *Physics Reports* 519.3 (Oct. 2012), pp. 97–125. DOI: 10.1016/j.physrep.2012.03.001. URL: https://doi.org/10.1016%5C%2Fj.physrep.2012.03.001.

[42] *Twitter Developer Platform: Tweet Object*. https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet. [Online; accessed 03-Sep-2021].

[43] Daniel Thilo Schroeder, Konstantin Pogorelov, and Johannes Langguth. "Fact: a framework for analysis and capture of twitter graphs." In: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2019, pp. 134–141.

[44] Luk Burchard et al. "Resource efficient algorithms for message sampling in online social networks." In: *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2020, pp. 1–8.

[45] Luk Burchard et al. "A Scalable System for Bundling Online Social Network Mining Research." In: *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2020, pp. 1–6.

[46] Konstantin Pogorelov et al. "WICO Text: A Labeled Dataset of Conspiracy Theory and 5G-Corona Misinformation Tweets." In: Oct. 2021, pp. 21–25. DOI: 10.1145/3472720.3483617.

[47] Andrea Landherr, Bettina Friedl, and Julia Heidemann. "A critical review of centrality measures in social networks." In: *Business & Information Systems Engineering* 2.6 (Oct. 2010), pp. 371–385. DOI: 10.1007/s12599-010-0127-3.

[48] Ulrik Brandes. "On variants of shortest-path betweenness centrality and their generic computation." In: *Social Networks* 30.2 (2008), pp. 136–145. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2007.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0378873307000731.

[49] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a Feather: Homophily in Social Networks." In: *Annual Review of Sociology* 27.1 (2001), pp. 415–444. DOI: 10.1146/annurev.soc.27.1.415. URL: https://doi.org/10.1146/annurev.soc.27.1.415.

[50] A.-L Barabási. "Network science." In: Cambridge University Press, 2016. Chap. Degree Correlation. ISBN: 9781107076266.

[51] Dong Yao, Pim van der Hoorn, and Nelly Litvak. *Average nearest neighbor degrees in scale-free networks*. 2017. DOI: 10.48550/ARXIV.1704.05707. URL: https://arxiv.org/abs/1704.05707.

[52] Gnana Thedchanamoorthy et al. "Node Assortativity in Complex Networks: An Alternative Approach." In: *Procedia Computer Science* 29 (2014). 2014 International Conference on Computational Science, pp. 2449–2461. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.05.229. URL: https://www.sciencedirect.com/science/article/pii/S1877050914004062.

[53] M. E. J. Newman and Juyong Park. "Why social networks are different from other types of networks." In: *Physical Review E* 68.3 (Sept. 2003). DOI: 10.1103/physreve.68.036122. URL: https://doi.org/10.1103%5C%2Fphysreve.68.036122.

[54] S Kirkpatrick, C.D Gelatt Jr, and M.P Vecchi. "Optimization by Simulated Annealing." eng. In: *Science (American Association for the Advancement of Science)* 220.4598 (1983), pp. 671–680. ISSN: 0036-8075.

[55] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications." In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97. URL: https://doi.org/10.1093/biomet/57.1.97.

[56] Liang Yang et al. "Modularity based community detection with deep learning." In: Jan. 2016.

[57] Martin Niss. "History of the Lenz-Ising Model 1920–1950: From Ferromagnetic to Cooperative Phenomena." In: *Archive for History of Exact Sciences* 59.3 (Mar. 2005), pp. 267–318. ISSN: 1432-0657. DOI: 10.1007/s00407-004-0088-3. URL: https://doi.org/10.1007/s00407-004-0088-3.

[58] Reinhard Diestel. "Graph theory." In: 5th ed. Graduate Texts in Mathematics. Springer, 2017. Chap. Bipartite graphs. URL: https://link.springer.com/content/pdf/10.1007%5C%2F978-3-662-53622-3.pdf.

[59] Filippo Radicchi et al. "Defining and identifying communities in networks." In: *Proceedings of the National Academy of Sciences* 101.9 (2004), pp. 2658–2663. DOI: 10.1073/pnas.0400054101. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0400054101.

[60] Omri Sarig. "Continuous Phase Transitions for Dynamical Systems." In: *Communications in Mathematical Physics* 267.3 (Nov. 2006), pp. 631–667. ISSN: 1432-0916. DOI: 10.1007/s00220-006-0072-7. URL: https://doi.org/10.1007/s00220-006-0072-7.

[61] Kim Dr. Christensen. *Percolation Theory*. MIT. [Online; accessed 16-May-2022]. Oct. 2002. URL: https://web.mit.edu/ceder/publications/Percolation.pdf.

[62] Daniel Thilo Schroeder et al. "The connectivity network underlying the German's Twittersphere: a testbed for investigating information spreading phenomena." In: *Scientific Reports* 12.1 (2022), pp. 1–13.

[63] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. "Scale-free topology of e-mail networks." In: *Physical Review E* 66 (3 Sept. 2002), p. 035103. DOI: 10.1103/PhysRevE.66.035103. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.035103.

[64] A.L Barabási et al. "Evolution of the social network of scientific collaborations." In: *Physica A: Statistical Mechanics and its Applications* 311.3 (2002), pp. 590–614. ISSN: 0378-4371. DOI: https://doi.org/10.1016/S0378-4371(02)00736-7. URL: https://www.sciencedirect.com/science/article/pii/S0378437102007367.