Understanding the Topological Structure and Semantic Content of Darknet Communities

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Cybersecurity Group at Department of Telecommunications at Delft University of Technology

Lisa Grace Geddam
Student no. 4125754

Committee members:
Dr.ir. Christian Doerr
Dr.ir. Gerard Janssen
Dr.ir.Jan van der Lubbe

April 18, 2017

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Delft, The Netherlands

ABSTRACT

For over a decade Darknet has been gaining tremendous popularity proportional to the growing concerns fostered by lack of anonymity and privacy on the World Wide Web. In recent years, illegitimate use of the Darknet has resulted into investigation in the research community that is analogous to a domino effect further adding to popularity of this type of network. Unfortunately, higher percentages have been attributed to the illegitimate use of the Darknet rather than to the legitimate use. This is because researchers of the Darknet communities have relied on the knowledge obtained through the use of Breadth First Search crawling algorithm. Crawling makes up the main step in the exploration of these communities. Crawling is also an effective method to understand the topological and semantic structure of the Darknet communities. The algorithms chosen to crawl thus, decide the knowledge obtained from these communities.

In order to get a detailed view of the network this thesis demonstrates how these crawling algorithms spread out over the Darknet communities and how this affects what and how much we know about them. Using different strategies, the effect of using different crawling algorithms and increasing crawl size as well as the effect of crawling from different starting points is shown. This thesis proposes a method to obtain a representative view of network. Instead of crawling a bigger portion of the network using only one crawling algorithm and coming to conclusions based on it, a constructive way has been proposed and demonstrated before any such conclusions can be made. This method demonstrates that almost an entire view of the network is required before conclusive inferences can be drawn about the Darknet communities.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LDA	Latent Dirichlet Allocation
BFS	Breadth First Search
DFS	Depth First Search
RFS	Random First Search

1 Introduction

Background: What is the Darknet?

The World Wide Web (WWW) is divided into three different categories [1], namely the surface web, the deep web, and the Darknet. The surface web constitutes that part of the WWW that can be accessed by search engines. The deep web is the invisible part of the WWW, the contents of which are not indexed by any search engine. The **Darknet** is a small portion of the deep web that is hidden and not accessed by search engines. It can only be accessed by proxy servers that provide connections between clients requesting resources available on another server in an anonymous manner. In different domains, such as scientific publications and media content, the Darknet has often been known as the "Internet's dark side" or the "Dark web", and even the "Shadow Net". The Darknet is an overlay network which was originally developed as The Onion Routing or TOR by the U.S. Naval Research Laboratory to protect U.S. intelligence communications online. As of today, TOR is known as the "The Tor Project" and it is a non-profit organization [2]. It is continually gaining popularity over the years.

The reason for the increasing popularity of TOR over the years is the **lack of privacy and anonymity** available on the surface web. Over the surface web, a person can be identified from various kinds of communications conducted and online activity from any IP address. For example, work and personal email accounts from the same IP address can identify a person. This does not allow any anonymity. Any marketing website visited by users with the help of cookies on the browser can track what they are browsing and present advertisements to make them buy certain deals profitable to the marketing website. In addition to this, the ISP under which their IP address comes has details of any other website they might visit. This does not give user the privacy they need. When it comes to web users such as activists or citizens of countries that suppress free speech, this lack of anonymity does not help them to conduct their activities freely.

Anyone spying on network communication by **eavesdropping through traffic analysis** with the right tools can determine, by listening, who is talking to whom. This is possible because of the IP-based network communication whereby IP data packets contain two components, called a header and a payload. The header contains routing information and the payload contains the actual data [3]. Therefore, by observing the header information it is possible to determine the

parties involved in that communication. It is not possible to stop anyone from eavesdropping and analyzing the content but the identity of the users who are communicating with each other can be masked. This is why one cannot determine exactly who a person is, during eavesdropping. Also, the contents of the data being exchanged can be masked, thereby making traffic analysis harder. This is **where The Onion Routing, or TOR, comes in to help** in the movement of network data communication. TOR does this by mixing the data coming from various senders and forwarding them randomly to the recipients three times known as onion routing. In onion routing, the data sent by a user are encrypted and sent over any three randomly-chosen onion routers that have made themselves available to forward the data to the destination address. Each of the three onion routers peels the onion layer of encryption that contains the information of the next router in line. In this way, the first router would only know the IP address of the user and the next router to which it is supposed to forward the data. The second router would only know the information of the first and third routers. The third router would know the information of the second router and the destination's IP address. It thus becomes difficult to conduct an immediate traffic analysis and know the user and the destination as the data are mixed, and onion routers are randomly chosen and none can pinpoint both user and destination addresses. Traffic analysis may be possible when an "intruder party" controls all three routers in the chain. In practice, this is very difficult. The "intruder party" would have to control a significant portion of the onion routers making up the backbone of the TOR network. TOR conducts all the transportation of the data, like web browsing or instant messaging, in the Transmission Control Protocol layer. With TOR, several TCP streams can be shared by any given circuit established for data communication. This is how TOR helps in increasing anonymity: by handing control over to the user to decide which streams can be shared in a circuit by deciding the three onion routers before transmitting the data.

Having understood how TOR makes provision for anonymity and privacy in online communications, one can understand why it has gained so much popularity over the years. **Users of TOR** can be mainly classified in two categories: legitimate purpose users, and "illegitimate purpose" users. Businesses use TOR to view competitors' pricing without letting competitors know what they are doing. Transparency in companies is encouraged, whereby employees can report any corruption-related activities within the organization and still maintain anonymity. Bloggers who shed light on very sensitive aspects of society or government use TOR to be able

to do so freely, and still be able to influence a stratum of society with their belief systems. Law enforcement officers use it for online surveillance, to remain invisible to the illegal and suspicious websites which they are trying to observe. Activists and whistleblowers use TOR to report government corruption, protect human rights, and report abuses from danger zones and governments that monitor internal and external online activity, for example that of China. People who want the freedom to browse the Internet freely, without the risk of IP identification, which attracts a lot of online marketing and identity frauds, also use TOR.

TOR's provision of anonymity and privacy has also seen it become popular among users who use it for illegitimate purposes. Groups that cannot express themselves boldly and remain anonymous on the surface web could now be free to engage in their interests. The Tor network became a petri dish and a thriving climate for drug-related marketplaces, weapons marketplaces, credit card and important document fraud and counterfeit services, hackers, pornography, pedophilia, hiring hitmen and assassins, radical extremism, gambling sites, the occult, and cannibalism, to name a few. In addition, thousands of forums and blogs are maintained around these subjects. This is another reason that the name "Darknet" is legitimate to describe such a network.

1.1 Motivation

Several studies which have previously investigated the usage of TOR attribute high percentages to illegitimate activities. The quest to discover the truth behind the percentages allotted to the entire activity taking place within the Tor network is what initiated this research. The goal is to provide a more scientific and neutral viewpoint of the usage of the Darknet, without any biases. The question asked was if Tor was indeed used for illegitimate purposes and how exactly was such a conclusion made.

During this research, three main issues found were found in previous studies. Firstly, most of the studies used Breadth First Search (BFS) as the graph traversal algorithm while crawling for the hidden services urls. BFS is known to give complete view of some parts of the network because of its nature of choosing parent links first before exploring children links of the parent links. BFS

therefore cannot give a representative view of the network. Since most of the studies have done so the Darknet communities found were leaning more towards the dark side.

Secondly, although it is not possible to crawl the entire network of Tor Hidden Services, there still was no tangible way to obtain the representative view of the network.

Thirdly, there is no tangible research done in the context of Darknet so far that shows the influence of crawling from different starting points and uses such information to come to a representative view of the network.

This thesis proposes to expose already known limitations of BFS in the context of Darknet communities and expand the view of the network by combining knowledge obtained by Depth First Search (DFS) and Random First Search (RFS). By compiling knowledge from all three algorithms, the otherwise missed parts of the Darknet communities obtained solely through BFS were shown.

This thesis also shows the influence on the view of the network as a result of choosing different starting points. This point is significant because most of the studies done so far have chosen wiki related hidden services and used them as starting points of the crawl. But if crawl was started from an url at the periphery of the network and not so well connected the observations made about the content of these hidden services would differ.

Finally, this thesis also provides a method that can help to obtain a representative view of the network using a metric before which any inferences can be made about the network. This method gives a more detailed information about the network.

The topological structure of the network is obtained by applying crawling algorithms such as BFS, DFS and RFS. The semantic content of the Darknet communities has been studied using probabilistic algorithm called Latent Dirichlet Allocation. In this thesis both of these aspects have been combined together and applied over a fully connected network.

In order to gain understanding of the Darknet communities in the light of the above stated proposals the following research questions were considered for the investigation:

1) How does the crawling algorithm change the view of the Darknet hidden services communities?
2) How does the extent of the crawl (crawl size) affect the view of the network?
3) How does the starting point of crawl dictate the view of the network?
4) How do we get to a representative view of the network?
5) What kind of a metric, in the context of LDA, can assist in obtaining a representative view of the network?

1.3 Thesis Overview

This thesis is divided into 5 chapters. Chapter 1 provided the introduction and the motivation of this thesis. Chapter 2 gives the background and related work for the rest of the thesis. Hidden services, how they are crawled, and how semantic content can be obtained with the help of topic modeling are explained. Chapter 3 describes what LDA does, how it is used to understand the objective of this thesis, and which LDA tools have been used, and why. Chapter 4 explores the five research questions stated and introduces the methodology of the strategies used to obtain answers. Different crawling algorithms have been contrasted, the effect of increasing crawl size and choosing different starting points have been shown and finally the method proposed to obtain a representative view of the network has been explained. Finally, chapter 5 gives conclusions and presents some recommendations that can be considered to further develop this work.

2 Background and Related work

To understand the Darknet, one has to gain understanding and knowledge of the various building-blocks of the concepts involved. Chapter 1 gave a short introduction of what the Darknet is, and the history of how it came together. Chapter 2 will give core details of the working of onion routing, what the Tor network currently looks like, and how websites within the Tor network function. To understand the aim of this project, it is also important to understand how and why a certain type of analysis was chosen to examine the content. The concepts behind the constituents of the methodology chosen are also explained. This chapter aims to prepare the reader to be able to understand how and why the topological and semantic structure of Darknet communities promise to give a good view of what is taking place within the Tor network.

2.1 Hidden services overview

This section provides a background description of what hidden services are, how they operate, and what existing methods have been used so far to find them. It also provides some of the advantages and disadvantages of these existing methods.

2.1.1 Description of hidden services

Hidden services can be defined as those web services which the Tor network allows users to host in order to provide a desired anonymous service. These services are known as hidden because of the secured protocol of onion routing; in reality, however, they can be termed “onion services”. A hidden service, therefore, runs only within the Tor network and any user wishing to have access to it will require the exact url of that hidden service. It is not like the surface web, where one can look up a webpage by IP address or name. There are several ways to get around the TOR network. Often, users randomly browsing through these hidden services have to copy and paste the url of famous hidden services on the surface web, such as Hidden Wiki, which constitutes a list of those hidden services that have used Hidden Wiki’s help to advertise them and make them popular. Another way to get around the Tor network is to go to famous search engines such as DuckDuckGo or Ahmia, type in one’s search term, and maneuver through the list displayed. A third way is to continually click one link after another till it takes the user to the required page, as hidden services are often linked to one another, for example marketplaces and bitcoin web pages. The creators of the hidden services are anonymous, and so are the users of those web pages,

unless they want to give away their identities. The creators of the hidden services always have the choice of whether to advertise themselves or not. There are various types of hidden services, including Internet Relay Chat servers, Secure Shell servers, and those that provide Simple Mail Transfer Protocol for users and for communication with other hidden services.

Tor is still growing and although the extensive variety found on the surface web is not yet found in the Tor network, it still offers quite an elaborate list to choose from. Additionally, because of its anonymous nature the sites that come up need not even be found on the surface web anymore. For example, one can find blogs, information, and social networks about cannibalism on the surface web, but it would be rare that "active" participating communities feel free to express their vices unless they have a safe way to do so; this is what hidden services pertaining to that topic provide. The information presented in the abovementioned paragraph is an extreme example but it shows typical examples of what the Tor network can be used for.

2.1.2 Methods of finding hidden services

The URL of the hidden service is derived from its public key and contains sixteen characters. A client and the hidden service connect via a six-step process whereby both client and hidden service can remain anonymous to one another yet still communicate. The number of hidden services in existence is still unknown but suspected to be vast. The Tor project does not collect any statistics that would harm the privacy of either the hidden services hosting servers or the users accessing them. In its metrics portal, it only displays extrapolated figures from the statistics reported by the relays which make up the networks that have obtained the status of hidden service directories, with the assumption that at least 1% of these relays have reported the figures on the number of unique hidden services. Details of the method can be found in [4] [5]. Clearly, because of the privacy policy maintained and the extrapolation method adopted, it is not possible to know how many hidden services there really are. However, a couple of studies have been done in this line of investigation and have reported some interesting findings based on the methods and statistics collected. In addition to the scientific publications, several parties have investigated hidden services, including independent bloggers, parties converting the information they have into business solutions, and writers exploring specific types of hidden services.

When exploring the number of hidden services at any given point of time, two methods are usually used. One way is to use crawling exclusively and the other is to use crawling in combination with a method that launches relays which can receive publications from the hidden services and receive requests from the users.

The authors of Project Artemis [6] present what their crawler collected but not the crawling algorithm itself that was adopted for their study. Using an initial list of 25,000 urls generated by their crawlers, they also trained their crawlers to collect HTML content and FTP links, and to retrieve emails listed in any given URL as well as links related to i2P, which is another dark network, less popular than Tor. Over a period of time, their crawlers were repeatedly run to collect statistics on which websites are stable and which either become unreachable or inactive. The method they used to categorize the websites to allot percentages to them, and the significance they give to the keywords appearing in URLs, has not been clearly stated. This method would seem feasible only when manually labeling a smaller number of hidden services; it would be an arduous task to categorize 60,000 unique hidden services. The problem of categorization is mainly an issue of what a researcher sees as a topic or label overall. Additionally, many hidden services contain a blend of topics and it is therefore difficult to immediately label hidden services in one way only.

Another study, by Alex et al., [7], employs the method of port scanning the hidden service urls collected by using a flaw in the design of hidden services before this study was published. Tor rectified this flaw in the design as a result of the discrepancy found by this study, therefore hidden services have to publicize themselves on a given relay that acts as a hidden service directory. The flaw of the hidden services design was such that, even though only two relays can be run from one IP address, if it so happens that one of those relays is unavailable then the standby relays on the same IP address can be used as hidden service directories. By employing only fifty-eight IP addresses and the flaw described, 39,825 onion addresses were collected. Port scanning of these hidden service urls was conducted. By conducting a crawl of these addresses, the textual content was collected and categorized to present the various types of hidden services in terms of content and the languages in which they were available. The assumption of the researchers was that hidden services are not interconnected with each other, which is why this

method of port scanning was adopted. Although effective, this method depends highly on the number of ports open at any one time. Port scanning can, therefore, be seen as only one of the methods available, depending on the type of investigation at hand. The major assumption made by the current thesis however, is that hidden services are, to a large extent, interconnected, because of the onion addresses found to be appearing in the pages being crawled. A study [8] that exclusively used a Breadth First Search (BFS) crawling algorithm found 7,000 Tor hidden services; however, this figure included websites that were short-lived and therefore the number of websites considered for this study was much larger.

A recent study by Owen and Savage [9], utilized forty onion relays over a period of six months and collected publications of hidden services and requests for hidden services. This study resulted in the collection of 80,000 unique hidden services which is by far the largest set in the literature, reviewed for this current project. One must note that this number results from a fraction of relays hosted in the Tor network out of the entire pool of relays, observed over a six-month period. The only difference, methodology-wise, between [7] and [9] was the time period and the flaw in the design of relays described above.

Several other studies, [10] [11] [12], focused on sub-communities in specific domains or topics such as extremism, terrorism, and Islamic forums among the hidden services of the Darknet.

2.1.3 Advantages and disadvantages of methods adopted to find hidden services

Both the methods stated in section 2.1.2 for exploring the number of hidden services have their advantages and disadvantages. Introducing relays in the network and conducting a port scan in combination with crawling results in missing information provided by the graph structure of the Darknet. Darknet services often interconnect with each other, in most cases due to common interests, and this information gets lost when this method is used. The fraction of relays controlled out of the entire number of relays automatically defines the results in terms of the categories of the websites obtained. This leaves no scope for other categories which might be found if we use crawling exclusively, and combine different crawling techniques. Also, to get a larger number of hidden services, one would have to control a greater fraction of relays which is computationally expensive. The advantage of this method, however, is that the hidden services

that do advertise themselves to the relays but do not appear in other websites pointing to them can be found much more easily than by the crawling method. In order for the crawling method to prove more effective, a very large network topology would be required and the crawl itself would have to be run over a long period of time, keeping track of unique hidden services and the fluctuations of some during the period of crawl. Furthermore, various kinds of hidden services, such as internet chat relays, are dynamic in nature and would never present a feasible portion in such a method. Blocked sites that refuse access to crawlers by using either robot.txt or captcha also cannot be included.

The advantages of using a crawling method outweigh the disadvantages in terms of the ease with which a crawl can be conducted, as opposed to deploying a large number of relays. Because of this ease, this method is used more often, even in studies that investigate very specific domains or categories of hidden services. Given how the Tor network is developing over time, and how crawling over a long period of time helps in analyzing results, this method is more useful. This is also because as the Darknet develops it will start to show a community-based structure. Combining both these methods and combining the considerations presented in this thesis could result in a well-rounded approach to investigating the vastness of hidden services. This can be considered as a suggestion for future work. This thesis uses the crawling method exclusively. Additionally, although the exact number of unique hidden services is still unknown, this thesis aims to show some of the key elements that are often missed out but need to be considered in the equation of investigating how and what we know of the content of hidden services.

2.2 Crawling and crawling techniques

Of the different ways of obtaining hidden services within the Tor network, crawling is the most extensively used. Therefore, it is the very first step of the entire process and a crucial one upon which analysis can be conducted. This is because how we crawl, in terms of the techniques adopted, determines what we know about the Tor network.

2.2.1 What crawling is, and studies that used crawling exclusively

Crawling is an iterative process of following one hyperlink after another available on a particular webpage, and indexing them by storing them. It is an automatic method which we humans would

use to surf the web by clicking links found on one webpage to move on to another to get to the information we are looking for. With crawling, the information is then stored in a format that canlater be used for the purpose for which the links were collected in the first place.

The crawl is initiated from a starting point or a starting webpage. Every link appearing on that webpage is stored in a list. Crawling also creates an adjacency list which records very clearly which link has followed which. After this, every link is followed and the links appearing on each of those web links are stored. There are several techniques available when it comes to determining how the next page link is picked up. When the crawler is on a particular page, it will grab all the http content, excluding any images or media content, and store it under the name of the link from which it extracted the data. The url address, title, and data are extracted from the webpages and stored. The crawler also checks whether a particular link has already been visited, and stores those which have not.

The crawler can be written on any operating system; for this thesis, Linux has been used. Tor software is first installed in the system, since the crawl is being conducted from a server trying to access urls encrypted using Tor. In order to access these, a proxy server called SOCKS, Socket Secure is used. Socket Secure, is an internet protocol that allows connections between the server and the client which, in our case, is the webpage or the list of starting webpages that has been fed into the crawler's code.

For every webpage the crawler attempts to visit, there is a timer set to keep pinging that webpage. Once it times out, the crawler heads over to the next link on the list. Every webpage can decide either to respond to the crawler that is pinging it or to ignore it with a "robot.txt" on its page. Once the crawler sees this, it will drop that link and move onto another one in its list. This is the most important step in the process of the methodology used. How much time we allow the crawler to run decides how many unique hidden services can be collected. The chosen order to run the crawler is vital in deciding the dataset that can be gathered from the Tor network. As mentioned before, all these parameters also depend greatly on the websites that are available on a day-to-day basis. Of the studies which have used crawling as a method to collect hidden services, the research by Moore and Rid [13], 2016, takes the more in-depth approach of

crawling up to five hops into a particular domain and collecting up to one hundred pages from each site. However, they start with a seed list of urls provided by the popular Darknet search engines ahmia.fi and onion.city. The problem with this is that ahmia crawls with a depth of 1 and hence a Breadth First order, and there is no way of telling how deeper into the Tor network it has gone. Together, the sites gave 5,615 unique hidden services but Tor metrics for the month of November alone show an average of 60,000. The number 5,615, however, which is a mixture of various kinds of hidden services and excludes banned domains not considered in the crawl, is far less than what is shown by Tor metrics statistics. One useful aspect in the context of this thesis is the depth of the hops they chose for the crawl and consistency they maintained with number of pages per website they collected. [8] used BFS technique to crawl webpages and found 7,000 hidden services. Another study [14] used the fork-join thread method in combination with BFS, where the fork-join section aids in parallel processing, splitting the crawling tasks in such a way that part of the processing can immediately be used while the rest of the crawl carries on. This study crawled 67,602 hidden services using this method. As one can see, when it comes to crawling the entire Tor network, the crawling algorithm adopted is BFS. In order to let future researchers consider the tradeoffs involved when using only the BFS technique, this thesis demonstrates, over a given network topology, how much should actually be crawled to get a well-rounded representation of the extent of hidden services.

The crawling techniques used in this thesis are as follows:

2.2.2 Breadth First Search:

BFS has been widely used in previous studies; therefore, this thesis aims to illustrate how crawling techniques affect what we know about the network by comparing them with Depth First Search (DFS) and Random First Search (RFS). Basing the crawling on one technique introduces a huge bias. Additionally, previous studies [15] [16] have experimentally proven the occurrence of biases for each of the techniques mentioned. According to Kurant et al., [15], the BFS technique introduces a bias toward higher degree nodes in a network and the authors stated that the higher the node degree, the lower the graph diameter. Doerr and Blenn, 2013, [16] in their study in the context of social networks, present where and why BFS tends to remain in the dense part of the network. They also argue that “BFS has the tendency to move in waves through the network”. In the context of the hidden services of the Darknet, BFS leans towards that part of the

network where web pages are strongly connected, which can be seen around hidden wikis, search engines, or even marketplaces. This turns out to be the best way to know how very similar pages are generally connected in the Darknet. The argument presented in [16] has been used as a basis to argue in this thesis how important it is not to base crawling on BFS alone, due to the limited view of the network we get. It is, however, important to note that it is not possible to get a full view of the Tor network in the context of all the types of hidden services there are. This thesis only focuses on http/https pages but it is not enough to use only the BFS technique. Therefore, as far as we can crawl the Darknet, we must make sure we do not use only the knowledge of the network acquired through the use of one crawling technique alone. The pseudo code for the BFS algorithm is presented in Table 1.

```
Pseudocode for the BFS Algorithm
   1  Visited_Hidden_services ← empty
   2  Q ← Adjacency list
   3  while Q > 0 (while the queue we have is not empty yet)
   4     {      p ← removefirst from Q
   5             visited_Hidden_Services ← visited_Hidden_Services + p
   6             for random_hyperlink ∈ neighbor(p) do
   7                          { if random_hyperlink ∉ Q and random_hyperlink ∉
     visited_Hidden_services then
   8                                Q.addlast(random_hyperlink)
   9                          }
  10     }
```

Table 1 Pseudo code for the BFS algorithm

2.2.3 Depth First Search

DFS, in contrast, is rarely used – if at all – as a technique in the context of the Darknet as the goal is often to obtain a significant number of urls faster, without having to run the crawler for longer. According to [16], DFS estimates the degree of nodes wrongly and gives it a lower value as it keeps searching and will eventually move towards the outskirts of the network.

It is not possible to implement DFS on the live network and prove this hypothesis; therefore, DFS has been applied to a dataset of already crawled websites in order to demonstrate that it will tend to remain on the outskirts of the network. What this means in the context of the Darknet is, firstly, that it will pick up websites which are outside the highly-connected high degree hubs. Secondly, it will also give us a very wide variety of webpages (topics), unlike BFS, because it will keep investigating more deeply into the connected graph of unique pages. DFS gives an extremely long chain which, in this context, can be helpful if one wants to find greater variety of

web pages (topics). Although we get long chains for DFS, the clear advantage of this technique is that one can discover the webpages which do not really "directly" connect to the hubs in the network. Therefore, if such a crawl were left to run over the live network, it would yield pages which would not generally be connected to the beginning point of the crawl, which the BFS technique would certainly not find in the preliminary stages. This does not, however, apply to hidden services that do not advertise themselves by letting this url be known to the users of the Tor network. The pseudo code for the DFS algorithm is presented in Table 2.

```
Pseudocode for the DFS Algorithm
    1   Visited_Hidden_services ← empty
    2   Q ← Adjacency list
    3   while Q > 0 (while the queue we have is not empty yet)
    4      {      p ← removelast from Q
    5              visited_Hidden_Services ← visited_Hidden_Services + p
    6              for random_hyperlink ∈  neighbor(p) do
    7                        {   if   random_hyperlink      ∉   Q   and   random_hyperlink   ∉
       visited_Hidden_services then
    8                                Q.addlast(random_hyperlink)
    9                        }
   10      }
```

Table 2 Pseudo code for the DFS algorithm

2.2.4 Differences between BFS and DFS algorithms in terms of the Darknet

	Breadth First Search	Depth First Search
1)	Begins at the starting point and remains closer to it for a longer time, exploring all the parent links before the children links	Begins at the starting point and goes far from it as it explores more deeply into the children links
2)	Slower than DFS	Faster than BFS
3)	Useful in finding highly-connected communities and therefore remains in the core of the network	Useful in finding links that would be very far from network core communities
4)	Moves in waves across the network topology; therefore, types of hidden services would be fewer than in DFS	Moves as arrows across the network topology and would bring a larger variety of hidden services than BFS
5)	High branching factor	Low branching factor
6)	Overestimates degree of nodes	Underestimates degree of nodes
7)	Gives a biased representation as only highly-connected hidden services would show up for percentages attributed to various types	In combination with BFS, a wider knowledge of the network topology can be obtained

8)	More popular as more hidden services can be known in a short period of time	Less popular as it takes time to collect hidden services using DFS
9)	Misses out on peripheral network communities	Gives a better knowledge of network communities

Table 3 Differences between BFS and DFS algorithms

The RFS algorithm is mainly used to provide a middle ground, as BFS and DFS over- and underestimate the node degrees, respectively. RFS should be considered as an algorithm a human user would adopt while browsing the websites without any fixed type of structure such as those adopted by BFS and DFS. Therefore, in the pseudo code for RFS, a random link would be removed from the queue.

After analyzing the differences between BFS and DFS in Table 3, one can see that each of these algorithms has its own strengths and weaknesses. Therefore, if we base our crawl only on BFS, there will be a biased representation of the Darknet communities. However, combining and considering these differences will yield a more holistic interpretation of what we know about the Darknet.

When BFS is adopted to crawl only up to a certain extent, then the coverage yielded can be represented by Figure 1.

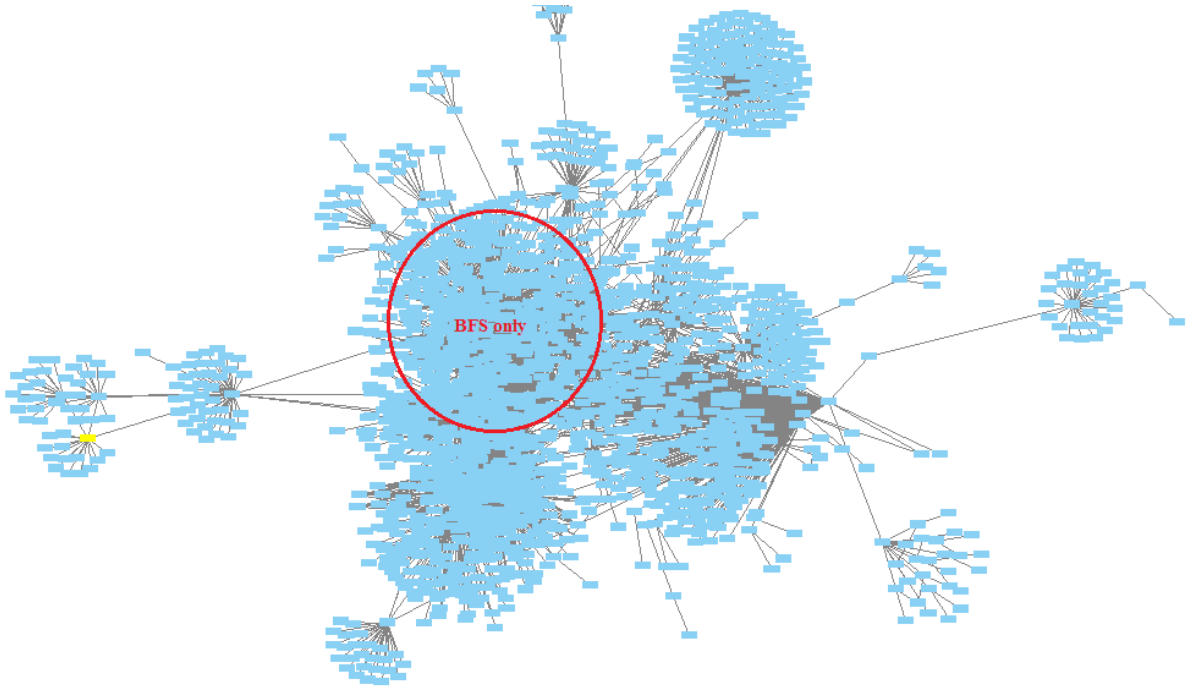

Figure 1 Representation of network topology covered when only using BFS

However, if we crawl longer and use both algorithms, the coverage yielded can be represented by Figure 2.

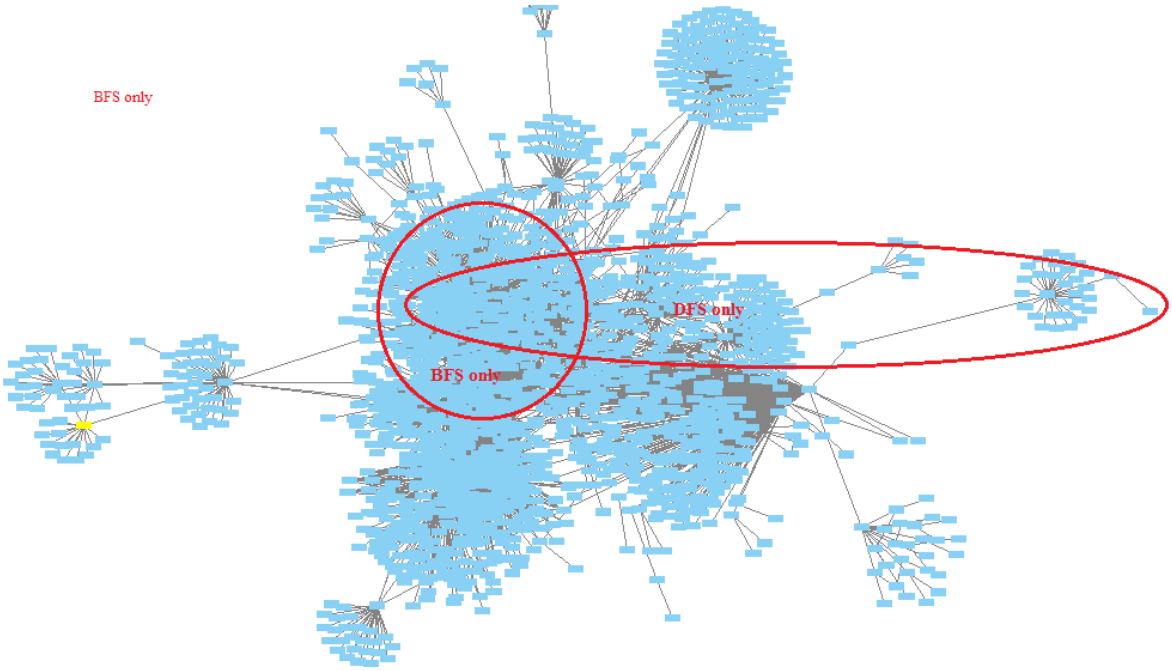

Figure 2 Representation of network topology by crawling longer using BFS and DFS

This leads to the question of whether a simple topological analysis of how the hidden services are connected is enough to understand what they are about. This is explained further in section 2.5.

2.3 Description of dataset used (overview of dataset, working model, choices made)

This section describes the dataset and the method used in this thesis. In order to demonstrate the bias introduced in the representativeness of what we know about the landscape of the Tor hidden services by using BFS algorithm alone, a specific set of fully-connected graphs of hidden service urls has been used as the "ground truth". This was also done in order to see how increasing crawl size drastically shifts the view of how much, and what, is known of the Darknet communities. This dataset has been obtained from a crawl conducted over a period of two months. The crawler found home pages of 7,153 unique hidden services. Of these, any of the hidden service urls that did not show connections with the wider set of the connected network topology were excluded from the study. This step yielded the home pages of 3,539 hidden service urls. In order to show that the representativeness of the hidden services depends on when we stop the crawl and on the techniques used, it is important to see the evolution of the crawl itself, using the techniques described. Since BFS and DFS cover the network topology differently, the study of the evolution of the crawl will show a different representation every time. Therefore, all the three crawling algorithms were applied on the 3,539 urls. Every time one thousand links were crawled, the urls collected in those thousand links were considered for representation, which was further investigated by studying the semantic structure with the use of topic-modeling techniques such as LDA. Everytime a thousand links were crawled, the results from analysis of semantic structure using LDA were compared with the results from the entire batch of 3539 urls. By choosing one starting point in this fully-connected graph, the crawl is conducted until each of the techniques covers the graph entirely. For this study, a hidden service that hosted a list of leaked urls was chosen as the starting point. The main reason for this was to be able to show the difference between BFS and DFS, as DFS would go away from the starting point more quickly than BFS. None of the hidden services was removed for post-crawling analysis as the network has to be analyzed as a whole, and removal of such pages would highlight the differences between LDA and network topology and introduce a bias.

2.4 Topic Modeling with LDA

Topological analysis using crawling has to be supplemented with a method that helps to understand what the Darknet communities are about. It is not enough to analyze the hyperlinks, as that does not give any insight into why these communities are even clustering together. To do this, text-mining analysis is required. Text-mining is a method that has been used extensively to analyze the textual data at hand and understand any connections or patterns there may be in them. Normally, when a human sees text that has been collected, he or she can already make some connections in terms of the prior knowledge he or she may carry about the context of the words in the data. Furthermore, if the words appear in a certain format, the human brain often draws inferences and makes assumptions as to what those data could be about. This is, however, humanly impossible to carry out when one has millions of words from thousands of documents. There has to be a structured and mathematical way to be able to understand the text at hand. This is what text-mining is about: being able to use certain analysis methods and algorithms to be able to organize data in such a manner that humans can understand and make sense of the connections within them.

We are dealing with Darknet communities that have sub-communities under them and, therefore, can be categorized according to the words displayed on their webpages. Since by means of crawling one can extract text, the next main goal is to make sense of all the random words clustered together. There is always an inherent structure in the way human communities are connected, and these can often be categorized by means of one main connection that makes up a community. One way to understand what connects these communities would be to collect all the words; whichever words seem to be more important for a particular webpage can be made the label under which it can be categorized. However, going through tens of thousands of webpages and labeling them manually is a tedious task. In [13], manual topic classification was employed to make an initial list of onion services by using a Support Vector Machine document classifier, which is one of the text-mining algorithms. Every other onion service found was fit into the initial classified categories set. Although the depth of crawl is maintained, the classifier chosen narrows the categories to only the ones defined and trained for every new webpage found.

This is where topic modeling comes in, this being a statistical model for extracting or exploring abstract main themes that appear in a collection of documents. The main assumption is that there

is an invisible semantic structure in the collection of documents, and that one can draw out main themes, known as “topics”, from these. This concept of topic modeling is what makes it an ideal fit to be used for the Darknet because the main communities of such a network are indeed often interconnected and interwoven. For example, drugs-related communities are often interconnected with Bitcoin- or PayPal-related webpages for transaction purposes. This is exactly how the crawler operates, by bringing back webpages that are somewhat connected for some reason or another. Having said that, this is why the algorithm used for crawling, which is dependent on the degree distribution of the network, is greatly influential in discovering such interconnected communities. A recent study [17] has used topic models to understand the content of the Darkweb by making use of structural topic models developed by [18] to analyze the political discourse within the cryptomarket forums of the Darkweb, which serve as a platform and a means to identify the general political climate.

One of the widely-used generative statistical models under the banner of topic models, called Latent Dirichlet allocation (LDA) [19], is used in this thesis. LDA makes an assumption that every document, which in this context would be a webpage, contains a mixture of several topics which is, indeed, the case because the greater the variety of words on a webpage, the greater the number of themes or topics on it. For example, a news-related webpage would contain a variety of topics as opposed to a security-related webpage, which would only contain the main topic of security. This is the reason it is safe to use LDA to understand Darknet communities. In comparison to the Support Vector Machine text-mining algorithm used in [17], LDA fares better in terms of the flexibility it offers in assigning a larger number of topics. Furthermore, if we take the case of hacking-related webpages and security-related web pages and consider that words such as hack, hire, DDoS, TCP, rent, secure, and firewall appear in our collection, the following can be understood. Secure, firewall, and TCP can be related to security-related webpages. Hire and rent can be attributed to hacking-related webpages. However, a word like DDoS can appear in both these types of webpages. This is where LDA comes in, to examine the number of times DDoS appears in both categories in which it appears; it will assign the word to either hacking or security depending on the number of webpages which relate to these. Therefore, if many documents appear to have DDoS in hacking-related webpages, this word will be placed under the

topic model generated with hacking webpages. This is another advantage of using LDA. More details of how LDA has been used in this thesis can be found in chapter 3.

The data first has to be prepared in the format which LDA accepts so that it can run the algorithm for every word and assign it statistically. The various parameters that can be varied before LDA is run have to be chosen according to the dataset. For example, one of the crucial parameters to set is the number of topics desired from the dataset. This is somewhat hard to estimate because all we have at this point is what the dataset obtained by crawling; however, we do not know what topics those webpages may contain. In general, this is a trial and error method which greatly depends on how many webpages we have, and how many different topics there are in the Tor network communities. Thus, if we say we only want to generate twenty topics out of the wide corpus of tens of thousands of webpages, we might be missing out on the other topics that come within the topic models generated. The opposite also holds true: if we choose two thousand topics to be generated from the corpus of data, then we will surely end up making too many communities that do not actually exist, and the words under each topic model will not make any sense at all for interpretation. After topic models have been generated they have to be examined and interpreted so that a human can say that a certain topic model relates, for example, to science or the military. [8] also made use of LDA to conduct their content analysis of hidden services. In the context of this thesis, it is to be noted that the global topic taxonomy presented in their study [8] is based on 1,021 unique hidden services, all of which contained English content, and on 250 topics using the LDA algorithm. Since BFS was employed in that study, it largely limits the extent of the landscape that was actually known of Tor's hidden services. Regarding the global taxonomy presented in this paper, one must keep in mind that it does not represent those percentages for the entire landscape but, rather, for only a small section of the network. The disadvantage of such a method is that one can be led to believe that the Darknet is only used for illicit and illegal purposes. This is a major bias resulting from such a method. A more refined step in this study could have been employing BFS, DFS, and RFS all together, running the crawler longer and then conducting a LDA analysis on the crawled data to come to a more conclusive global taxonomy.

When what is crawled is supplemented by the topic models to understand the Darknet overall, this thesis shows by the use of the topic models how BFS moves in waves in the network and how DFS moves as straight arrows into the network by analyzing the words that appear. In terms of the topic models, the expected result is that BFS will give few types of topics whereas DFS will give a variety of topics as it digs deeper.

In this thesis LDAvis [20] has been used to analyze the topic models, and also to demonstrate how the increasing crawl size changes what we know about the Darknet. As the crawl size increases, the words that keep adding due to newer hidden services also begin to appear.. The topic models obtained by another tool, heat map visualization, used in [21], have been used in this thesis. In this tool, topic models of two different sets have been compared against each other. In this case, the topic models of the dataset which provides the entire topology have been fixed as one batch. Every thousand links crawled in an incremental manner were fixed as the batch being compared against the main one. More details of both these tools and how they were used is given in chapters 3 and 4.

2.5 Summary

In this chapter, the background information and related work for this thesis was presented. Hidden services were described, with an overview of the various ways one can find them in the Tor network. The pros and cons of these ways have also been presented. Crawling, which is the major step that makes up the backbone of this study, has been described in detail with a focus on the different crawling techniques which can be employed. The differences between these techniques were presented. Topic modeling as a post-crawl analysis was presented in detail, and the reasons for making use of it have been explained.

3 Semantic structure using LDA

3.1 LDA and what it does

In order to understand the semantic structure, Latent Dirichlet Allocation [19] (LDA) has been chosen in this thesis. When there are only two documents and only two distinct topics can be pulled out of them, then the task is not computationally extensive. However, when the number of documents is bigger than human handling ability, then the same concept of examining each word from every document has to be automated. LDA does this by assuming that in a given dataset of documents, there is a latent structure which holds all the documents together by means of topics. Therefore, although words are being examined, the supposition is that a latent structure already exists in the collection of documents. A certain number is first given as the number of topics that are to be pulled out of the dataset. Every word from every document is first randomly assigned to any given topic. This is a temporary action in which every word has a topic assigned, and every document has topics assigned.

Next, for every word in any given document, and for every topic:

1) The probability of that topic in the given document is calculated, which translates to the words of this topic in the given document
2) The probability of the word from the topic at hand over all the given documents. At this step, the pre-assigned word to a topic is reassigned according to the product of both these probabilities.

This process is iteratively repeated for all the words in all the documents. The end result is that each document has numerous topics and each word is associated to a particular topic. For example, the word “bitcoin” appears in document X because there are other words of the same topic to which “bitcoin” belongs. Therefore, if more words on one topic are found in one document and X and bitcoin appear with them, it is strongly correlated to that topic. This is how assignments are made for every word.

In the process of preparing the given data to be fed into the LDA algorithm, key steps are involved which greatly influence a particular instance of the LDA topic models. The steps are preprocessing, choosing the value for K number of topics, and choosing hyper-parameter values. They are as follows:

3.1.1 Preprocessing

The crawled text extracted from the onion urls, (each onion url will henceforth be referred to as "document") was arranged as a "bag of words" where the words of each document were separated by a line space. This constitutes the crucial step called preprocessing. In order to get to such a bag of words, several elements had to be taken out. The goal was to have only those words which were useful in the context of the Darknet. The steps are as follows:

1. Any unnecessary content from the webpages, such as ssh banners, redirecting links, or html code, were removed.
2. A customized stop list was created as per the language constructs of the hidden services, which included all the unnecessary words to be taken out. This step speeds up the LDA runtime. This list was prepared by observing the frequency of a particular word in a corpus and deciding if it is useful in the context of the data.
3. Non-English pages were removed but the pages with occurrence of English words were kept intact.
4. Usually, as a standard practice, any webpage containing only few and insignificant words are removed to speed up the process. Another reason for this is that the words in such documents are not very valuable in the context of the entire corpus of words available. However, in this thesis work, all the webpages were kept intact to maintain the fully-connected graph structure. The details of this can be found in chapter 4.
5. Tokenization of words was also carried out. However, in order not to lose abbreviations and words that make sense as whole, smart tokenization was also used.
6. Stemming and lemmatization were also carried out. Various forms of expression of a word, such as a verb, noun, or adverb, become repetitive. Therefore, this step was performed to reduce and streamline the vocabulary. This step can be viewed as a normalization step in order to obtain only the essence of a word and avoid its tenses and other grammatical forms. With this step, however, one has to be aware of the context of the words as they can mean different things in two different contexts or forum groups. These words also constituted the customized stop list.

3.1.2 Best topics

On one hand choosing a value for the number of topics involves making an estimation and is a trial and error process. It also depends on the dataset being worked with and the requirement of the study. On the other hand, LDA can be adapted in the context of the Darknet to understand the Darknet. This is done by drawing a fixed number of topics from the corpus. Two methods are commonly used: One way is to randomly select the K value and repeat it over several runs to see if the topic instances generated are legible and provide interpretable insights. This process is then adapted according to the user's hypothesis. In this thesis, a randomly chosen K value was used. The second option is based on an automatically generated K value for a given dataset. The methodology described in [22] was adopted for the given dataset. The strategy used is fixing the dirichlet hyperparameters α and β to 0.02 and varying the number for the topics used, K. Varying the value for β yields varying results for the number of topics; therefore, a higher value of β yields fewer topics. In terms of the Darknet, this would mean that a very high value for β could broaden the topic models quite extensively, giving fewer topic models and thereby eliminating other important sub-topics under a particularly broad topic model. However, 0.02 was chosen as the fixed value. As the paper specifies, the problem of choosing a value for K is solved by taking the posterior probability over the given data (number of words, w) which, in this case, would be the log likelihood of the data integrated over all the parameters of the model, $P(w|T)$.

To demonstrate this, the above-stated method was applied to two different datasets.

The first dataset was the extracted text from a set of 1,594 urls which were crawled. This method was carried out on a total number of 42,691 words. This methodology is dependent on the dirichlet hyperparameters, whether a stop list has been considered or not, and whether all the documents have been included in the dataset - in other words, the dataset size or type. Over a range of numbers of topics K, 2-202, with a step size of forty and using the code shown in table 4 was generated.

```
llhs<-array();
counter<-1;
for(K in seq(2,202,40)){
  G <- 5000
  alpha <- 0.02
  eta <- 0.02

  library(lda)
  set.seed(357)
  t1 <- Sys.time()
  fit <- lda.collapsed.gibbs.sampler(documents = documents, K = K, vocab = vocab,
                                     num.iterations = G, alpha = alpha,
                                     eta = eta, initial = NULL, burnin = 0,
                                     compute.log.likelihood = TRUE)

  llhs[counter]<-fit$log.likelihoods[1,5000];
  counter=counter+1;

  t2 <- Sys.time()
  print(t2 - t1)
}

plot(llhs);
```

Table 4 The snippet of code used to generate figure 3

In figure 3, log P(w|T) is depicted on the y axis and the number of topics is depicted on the x axis. Therefore, the log likelihood for a 2, 42, 82, 122, 162, and 202 value for K is indicated by the line. The point at which the curve reaches its highest is considered the best estimate value for a dataset. The value 122 was found to be the best number for the value of T for this dataset. However, the topic models generated were found to be too sparse.

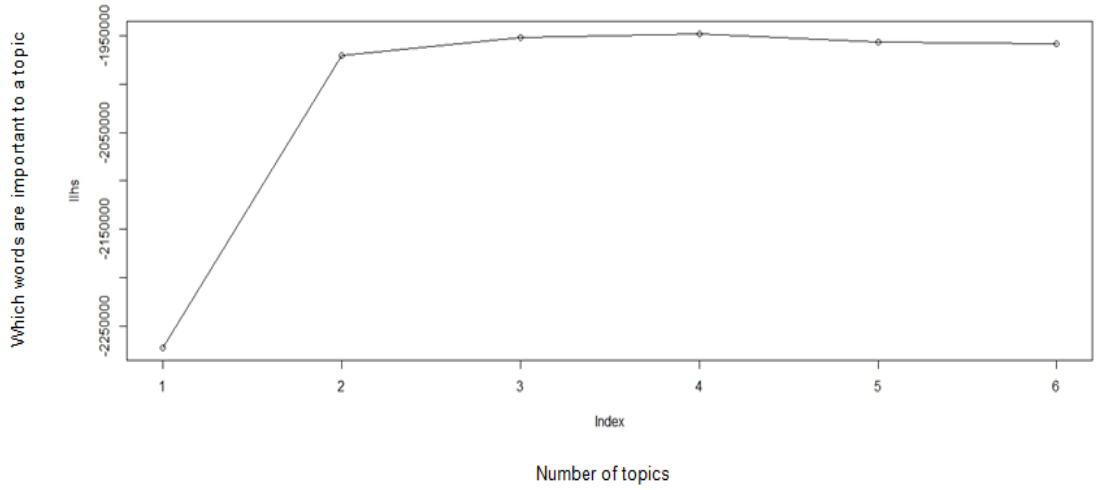

Figure 3 Log likelihood of data (words) for different values for number of topics for 1594 urls

On the second dataset, the same method yielded 162 as a closer estimate for the value of K. This can be seen from figure 4 where the curve reaches the highest point on the graph below.

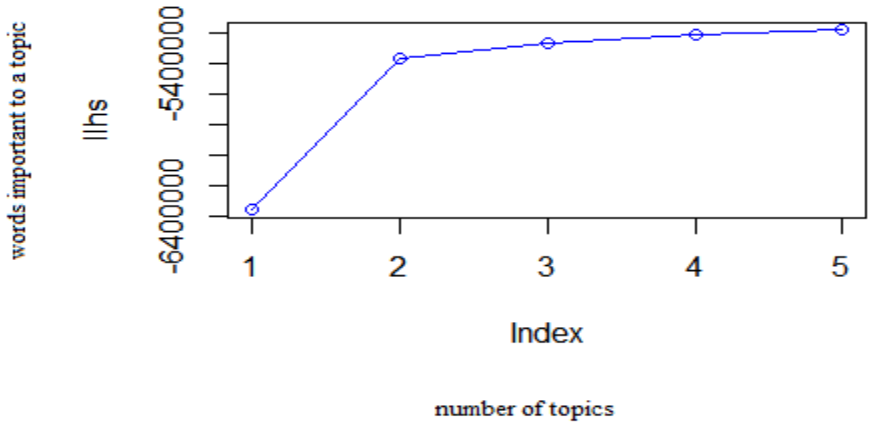

Figure 4 Log likelihood of data (words) for different values for number of topics for 3539 urls

This method of generating log likelihood of the data integrated over all the parameters of the model proves to be inconsistent with human judgments, as topics turn out to be very sparse and non-legible. Fixing 162 as the value for K turned out to be ineffective for many topics in the context of the Darknet for this particular dataset. This method also cannot be used effectively when trying to compare topic models of two different datasets because it generates non-interpretable topic models for any given dataset, therefore comparing a vast number of topic models that are not legible is not only arduous but ineffective. The goal in using LDA is highly dependent on the dataset being examined. For this reason, such a method based on log likelihood proves to be ineffective. This is described in greater detail in chapter 4.

3.1.3 Trade-offs

LDA is a tool that has to be used according to the specified domain. Results obtained over repeated runs have to be fed back to the start of the process to modify how it is used. Therefore, all the steps explained above are standard steps but they have to be used according to the Darknet-related data. When using LDA certain trade-offs are to be considered.

1. Any tradeoff involving stop list is judged in terms of what is important and specific to the communities of the Darknet. A lot of webpages found in the Darknet are hosted by independent parties which can range from hackers to forums. Some are found to contain a very specific jargon, or their English construction is not always correct. In this thesis,

these words were kept in order to preserve the insights that can be obtained from specific communities. Another reason is that the webpages would, in any case, become clustered because of these types of words occurring in most of them in a particular topic.

2. The second trade-off is whether to remove any pages with an insignificant or smaller number of words, and even the non-English pages. Given the research questions of this thesis, it is important to keep all the pages intact, regardless of these considerations. A fully-connected graph is created if hyperlinks are found in the pages. Removal of any pages for the above reasons will make the results inconsistent. This type of decision is very task-specific. Additionally, non-English pages or even English pages with other languages clustered eventually, and this allowed the analysis of the English words and understanding their context within multiple languages. This was clearly seen using the LDAvis tool.
3. The value for the number of topics is also very task-specific. In this thesis, the process is simplified and a very small number is taken, as the task is oriented towards understanding how the crawl proceeds.

3.2 Visualizing Topic models

Two tools from two different studies have been used to examine the research questions as part of visualizing the topic models. The first tool, LDAvis from [20], provides a way to understand how topics group in relation to one another and the context of the words. The second tool is the heat map visualization from [21], which has been used to understand how topics evolve over time by comparing two different datasets.

3.2.1 LDAvis

The widely-used LDAvis visualization tool helps to understand topic term connections. Visualization of the topic models helps in interpreting and providing good insights of the raw dataset. Depending on what is to be visualized over repeated attempts, more refined results can be obtained by varying K. LDAvis is used to understand the meaning of the topics, the top terms in each topic, and how the topics are related to one another based on the distance between them. Common words between two topic models reveal how similar the topic models are to each other and provide the dataset with more depth. When a certain word is hovered over, the tool

highlights the topic models in which it appears. A snapshot of how the tool works is provided below:

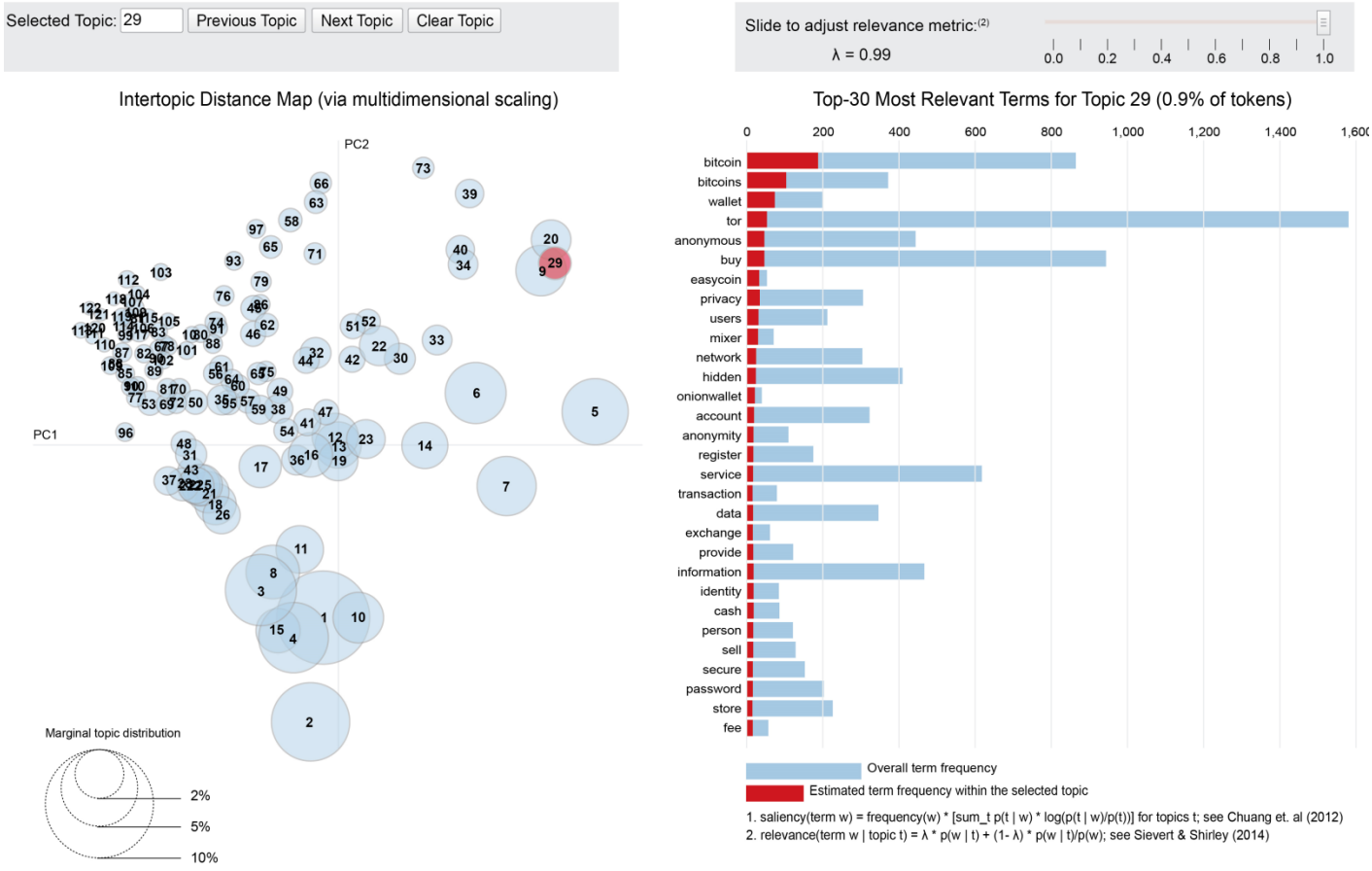

Figure 5 Snapshot of the LDAvis tool used on dataset 1

The circles on the left-hand side of the snapshot represent the topic models as per their numbers. The larger the area of the circles, the greater their prevalence in the entire corpus of data. In this snapshot topic number 1 has the highest area and, therefore, has the largest number of words from the total number of words. The degree to which two circles appear closer together is the degree to which they share the common words. For instance, if one circle is completely engulfed in another, it means all its words are found in the bigger circle. The bar chart on the right-hand side represents the words that are most relevant in a chosen circle or a topic which can be seen by hovering over a particular topic. A specific word which appears the most in a topic is represented in relation to the entire list of words, so one can also see the word's relevance in the entire vocabulary in terms of how many times it appears. The blue bar is the overall-term

frequency. The red bar is the estimated-term frequency for a selected topic. In this way, words found in each topic can be seen as the words also reveal what the topic model is about.

In addition, words common to two or more topic models give insight into the **context** of the word. This is illustrated below:

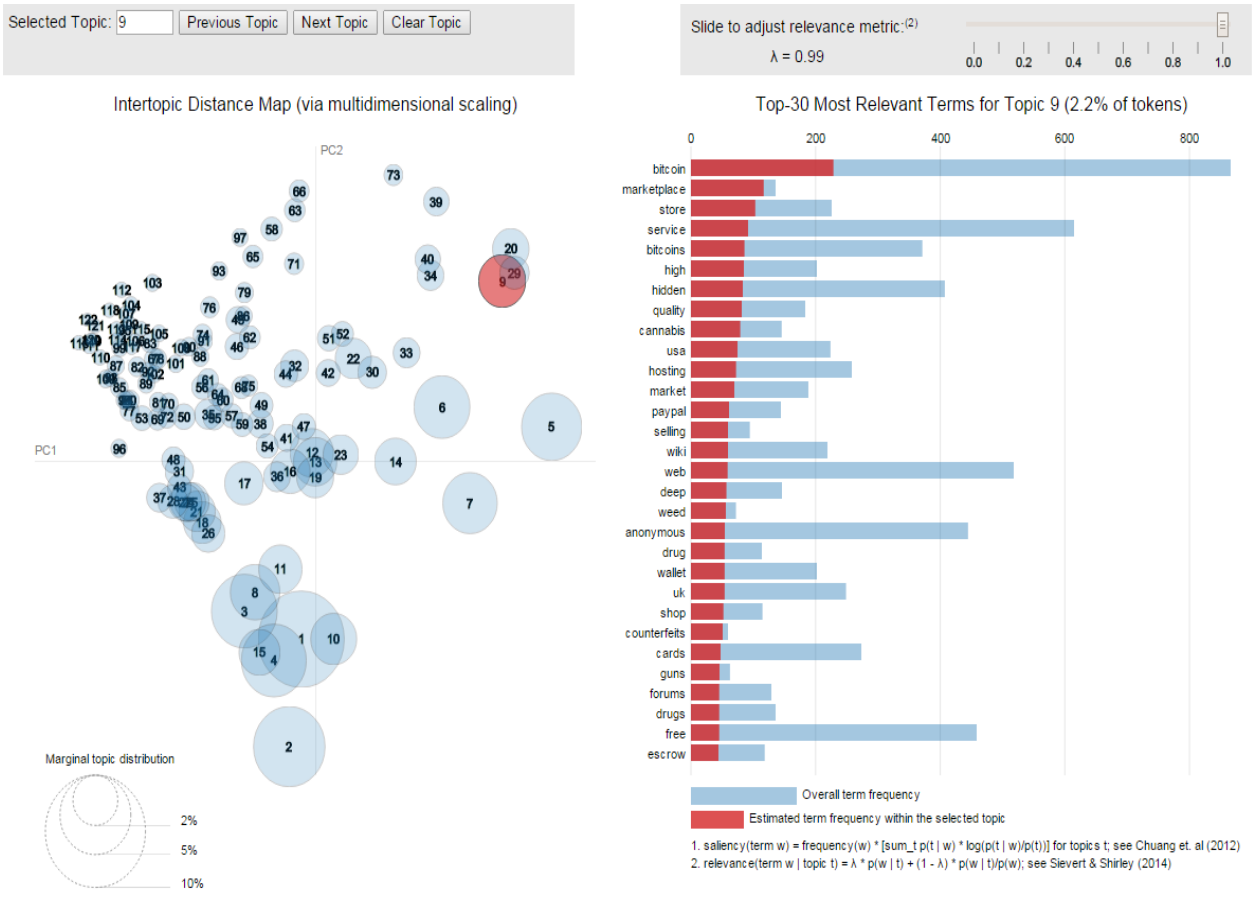

Figure 6 Snapshot of LDAvis to understand the context of the word "bitcoin"

When figures 5 and 6 are compared, "bitcoin" appears to be the most relevant term for both topic models 9 and 29. Topic model 29 is wrapped within topic model 9. Bitcoin in topic 29 relates to websites that are more security-focused, whereas topic 9 relates to websites that are drug marketplaces. Therefore, it is inferred that bitcoin's context in topic 9 is related to drug

marketplaces and the bitcoin which relates to security-related websites comes within the bigger bracket. Such inferences can be spotted with this kind of a visualization tool.

During the preprocessing step, the non-English languages that showed up in any English pages were left with the main corpus, as they gave some insights into what the websites were about. Another reason was that other languages would cluster up together anyhow. The LDAvis tool is able to capture this in figure 7, where topic 15 turned out to be Italian webpages.

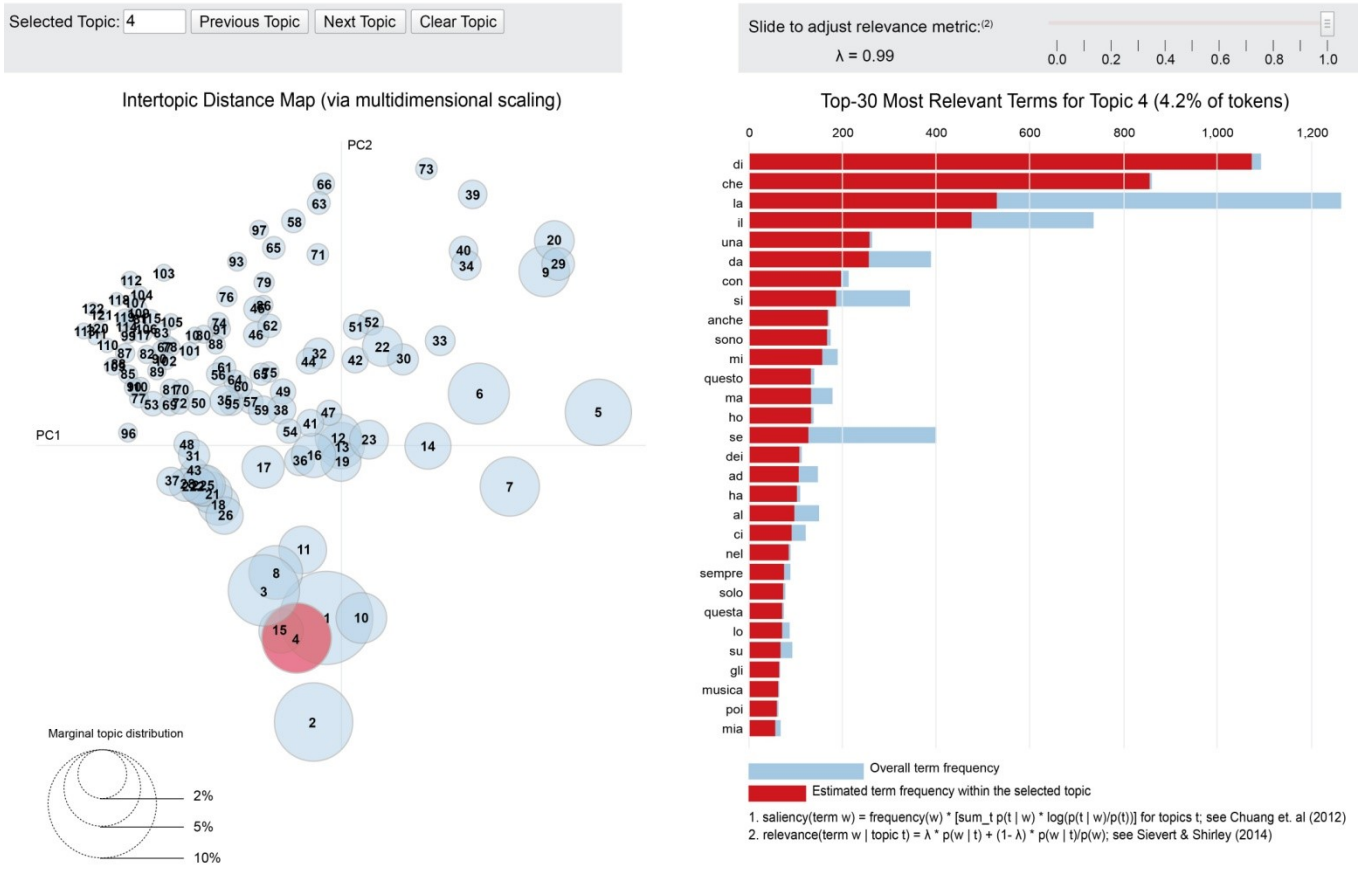

Figure 7 Snapshot of LDAvis where Italian web pages cluster together

In conclusion, several steps must be made in order to get specific Darknet topic models. The collected content of the webpages obtained from crawling was prepared to be fed into the LDA tool. Preprocessing, tokenization, lemmatization, and choosing whether to include non-English pages are some steps capable of changing our view of the Darknet communities and what can be understood of them. The number of topics required for LDA implementation also affects how

much is known of the Darknet. Using a visualization tool helps to interpret topic models so that an accurate view of them can be achieved.

3.2.2 Heat map visualization comparing topics from two different batches

This section compares how much of what is crawled with the ground truth (i.e. a fully-connected graph of hidden services). The methodology used in [21] was adapted to see what can be understood about the representativeness of topics as the crawl progresses. The main idea used in [21] in relation to being able to compare two different batches of topic models is measuring the distance between words of every topic model in one batch, and every topic model of the other batch, and representing the results in the form of a heat map. The distances are represented as a matrix. Python was used to analyze the data files, filter the stop words, and vectorize and train the data to generate a matrix file representing the distances between the topic models of both batches. The libraries used were numpy, scipy, and scikit-learn. R was used to create the visualization based on the matrix created in order to generate the heat map in this thesis. The LDA code used for this particular task was based on scikit-learn LDA implementation, available in Python. The process is as follows:

1) The **data were read** from two batches and placed into different corpuses, split by a newline.
2) The next step was to use sklearn CountVectorizer to **vectorize** the data, that is, convert the data into a matrix with tokens. This step returned the list of feature names along with the vectorized data (matrix) that was then applied to each list in each corpus. The stopwords file was also applied at this point to remove excess words. The Javascript in the files was cleaned up.
3) Two separate **LDA models** were then generated upon the data models created in step 2 using the LDA implementation with each K topic. The value for K for this thesis work was fixed at 10 for both the batches being compared. This generated trained data models and the vector of features as well as their weights for every topic was obtained. However, since the nature of the feature lists was different, they could not be compared.
4) **Normalization**: This step was taken to normalize the components because in the previous step the nature of feature lists was different. Furthermore, the assumption is that the

models have different stacks of features. Therefore, the task was to make the components of trained data models have the same list of features. This means all the weight vectors for all the topics should have the same nature. In other words, they should represent the same list of features. For example, if one topic has features [security, bitcoin] from one batch and the topic of the other batch has features [security, forum] then, in order to compare them, we should get weight vectors representing this particular list of features: [security, bitcoin, forum]. Any component missing features is assigned value 0.

5) The next step was to **calculate the distance** in the matrix. This was done using a modified version of cosine difference.
6) The library ggplot2 on R was used for the **visualization**. The matrix file was first converted into a data frame acceptable to ggplot2. This was then visualized in the form of a heat map in which the darker color indicates the highest match and the lightest color indicates a lowest match.

Logic used to compare two sets of topic models:

Step 1: Common words and unique words were picked from the topic models of two different batches. Common words from both batches are referred to as COMMON_SET, and unique words as UNIQUE_SET.

Step2: A similarity level is achieved by dividing COMMON_SET by UNIQUE_SET.
The worst-case scenario would be when COMMON_SET equals 0 (no words in common). In this case the similarity index will be 0/UNIQUE_SET = 0
The best-case scenario would be when COMMON_SET equals UNIQUE_SET (all words in common). In this case the similarity index will be COMMON_SET/UNIQUE_SET = 1 (because they are equal).
The bigger the index, the more similar the two texts.

Step 3: This comparison of step 2 is carried out over a chosen number of words. Since LDA implementation assigns a mathematical expectation of quantity to any given certain word in every topic, in both cases either

a) The expectation is 10 (which means that an article (webpage) of this topic will almost always have this word or

b) The expectation is 0.0000001 (which means that an article (webpage) of this topic will almost never have this word)

A comparison of two topic models of two different batches cannot be made by the presence or absence of a certain word but, rather, the math expectation has to be considered. If the number of topics can be simplified to two for the sake of brevity, and if we consider these to be "security" and "religion", it is highly unlikely that religion-related words would be found in the security topic. Therefore, the words of the religion topic found in it will have a higher expectation value than the words of security. Because the expectation values can vary greatly, there needs to be a cut-off value and this is the reason the comparison was made over only a few chosen words. Five hundred words were chosen for this thesis. Therefore, only five hundred words represent a particular topic, and the similarity index is calculated and represented as a heat map.

Considering the above steps, the decision had to be made regarding the similarity to be achieved using the cosine distance matrix. The trade-off involved was whether a high similarity was aimed for, or a low similarity. If a high similarity is being aimed for then a cosine comparison should be achieved not by having words in common but by having the same words in the comparison. For example, if two batches have "security" as the most frequent word but both the batches have a different expected value for that word, those models are considered different: hence the high similarity. However, low similarity was being aimed for, for which reason the comparison method was changed in such a way that when comparing two topics, the common words in two models were divided by the quantity of distinct words in both models. This was done by increasing the number of words per topic which, as stated above, was five hundred. If a very high value was chosen for K and one thousand words from both batches were chosen, the possibility

of getting hardly any matches increases greatly, as the topics would be different and have few common words. If the number of words to be extracted is a lower value and the batch has ten thousand unique words, then the similarity will be reduced. Therefore, after much experimentation with all these values, it was decided five hundred words would be extracted. Moreover, the LDA generating different topic models adds to the challenge of the matching as the words now get dispersed.

One challenging aspect observed in this process was that when using LDA, because of its probabilistic method of appointing a word to a topic, the topic models obtained are never the same every time with the same data.

Another decision to make was whether an output file can be generated which displays the topic models with their words and shows what it is that actually matches, in order to make the task of comparison easier for the human reader. This is because the problem in creating such an output is that it would be a 3D tensor and thus impossible to legibly show in a 2D matrix. For example, say we have ten topics for batch 1 and ten topics for batch 2 and that if we want to display the twenty best matches per comparison, it will result into 10*10*20 words. Therefore, fixing a low value of ten topics and increasing the number of best matches to five hundred, the heat maps were created without a matrix file.

3.3 Conclusion

In this chapter, what LDA does and the steps involved in preparing the data in the context of the Darknet has been shown. The two tools of visualization to understand LDA topic models have been explained in detail. One of the tools, heat map visualization was used to compare two different sets of topic models, one being the main batch containing all the text of the webpages used in this thesis while the other is the evolving number of webpages. This has been used to understand how topics evolve, and is presented in chapter 4. In conclusion, this chapter explains how every step of the methodology matters significantly, introduces a bias in the process, and should be accounted for.

4 Topological structure analysis using Topic models

To understand the Darknet communities from a topological point of view, there are several angles this has to be approached from. First, what kind of a crawling algorithm for graph traversal decides how we get to know the network? BFS can be used to understand highly connected nodes of the network from the network core, whereas DFS can be used to understand the communities that are found in the periphery of the network but not necessarily connected to the network core. The second angle is, how long do we leave the crawl to run as this dictates how much we know about the network? For example, obtaining one-third of the network instead of a full view will result in a bias in representativeness. But arriving at a conclusive estimate of how much of the network one should crawl before any conclusions can be made is a detailed way to approach this problem of solving the bias. The third angle concerns the differences in the view that results from starting the crawl from different starting points. One starting point can show that the network is all about one topic, but another starting point can show the network is all about another topic. This chapter explores all three angles and aims to demonstrate what considerations have to be made in order to gain a more representative view of the network.

In order to understand how the hidden services are connected to each other, LDA has been used as words found in the web pages themselves tell us the nature of these web pages. In other words the "view" of the network communities is provided by topic models generated by LDA. The comparison of these words in terms of the topic models created by LDA explain why certain words appear in the same topic model and thereby show what kind of documents cluster together in the topic models. In other words, the knowledge of the semantic content obtained from the LDA has been plugged with the topological structure of the network to expand the knowledge one can obtain about these Darknet communities.

In the related work, BFS algorithm has been used as a foundational algorithm to understand the Darknet communities. From [16], BFS tends to remain at the network core for a long period of time, and this greatly affects our view of the network. This is the reason why DFS and RFS have also been used to demonstrate how the view can vary as DFS can follow the exact opposite trend by quickly traversing to the periphery of the network. This chapter aims to demonstrate the effect of choosing only BFS as opposed to DFS and RFS.

This chapter aims to answer the following questions:

1)How does the crawling algorithm change the view of the Darknet hidden services communities?

2) How does the extent of the crawl or crawl size affect the view of the network?

3) How does the starting point dictate the view of the network?

4) What kind of a metric, in the context of LDA, can assist in obtaining a representative view of the network?

5) How do we get to a representative view of the network?

These questions are explored by comparing an already crawled dataset of urls obtained from the Darknet with the step wise crawled portion of the network. The already crawled dataset is treated as the ground truth. Two strategies are used: the first one is crawling from one starting point with step size of 1000 links and the second one is crawling from 50 random starting points with step size of 300 links.

This chapter first introduces the methodology adopted and the network topology that was used in this thesis. Treating the already crawled hidden services as a complete network was a preferred option as opposed to conducting analysis over the live network. Section 4.1 introduces the required steps in detail and with the help of one starting point, the influence of the crawling algorithms, and crawl size is then demonstrated in Section 4.2. Section 4.3 introduces the metric that helps to clearly define the representativeness of the network. Section 4.4 presents the effect of crawling from different starting points and how that determines which view of the network can be seen. Finally, the inferences and insights that can be drawn from sections 4.2 and 4.4 are presented in Section 4.5.

4.1 Introduction to methodology used and network topology

A crawler was left to crawl for a period of two months. From crawled dataset, only the html formatted unique hidden service urls were selected for the analysis. In total 7,152 were found. An adjacency list indicating all the connections was made from this dataset to obtain the network topology. Figure 8 gives a snapshot of the adjacency list3,549 links formed a well-connected graph. Ten were not useful and were filtered out as some were named google.onion or tor.onion.

Thus 3,539 unique urls made up the complete graph as can be seen in figure 10. The links that were not connected from the entire dataset of 7,152 also appear in the graph as clustered units. This connected graph of 3,539 unique urls was used for this analysis.

```
ramp2bombkadwvgz.onion →ilona4rkjw65hw6t.onion
bwin42j7wvhbeieg.onion  → help.bwin42j7wvhbeieg.onion
media7hsv7rkdwnh.onion  →help.bwin42j7wvhbeieg.onion
ramp2bombkadwvgz.onion → cheechnchong.shops3jckh3dexzy.onion
ramp2bombkadwvgz.onion →butt-head.shops3jckh3dexzy.onion
fhostingesps6bly.onion →cig7hq7ebwfvusqj.onion
ramp2uh2esdm4tac.onion → tsum.shops3jckh3dexzy.onion
ramp2bombkadwvgz.onion → jm77xadoe3yajj3c.onion
kowloon5aibdbege.onion →darkodei7qdze3pl.onion
```

Figure 8 Snapshot of adjacency list

For strategy 1, only one starting point was chosen and BFS, DFS, and RFS were run on it. For strategy 2 all 3 algorithms were run from 50 randomly generated starting points on the same graph. No matter what starting point is chosen, the crawling algorithms will travel differently in the graph before the entire graph is crawled. gotchafjkmcqdz2x.onion was chosen for strategy 1 and it had the biggest branching factor because of the number of links found on it. It is a website hosting a list of leaked onion urls sorted according to hostname and this led to 461 other unique onion urls. This starting point was chosen to show the difference between BFS and DFS. DFS tends to leave the network core faster than BFS. Also this starting point gives a full crawl of the entire network to work with, and therefore the influence of crawl size can be better demonstrated.

The 50 randomly chosen starting points were a mixture of highly connected hidden services in the network and not so well connected ones. The themes of these starting points were related to bitcoin, drugs, privacy, hacking, forums and blogs.

This thesis is based on the network topology graph represented in figure 9. Cytoscape was used for this visualization. The disconnected links were not analyzed.

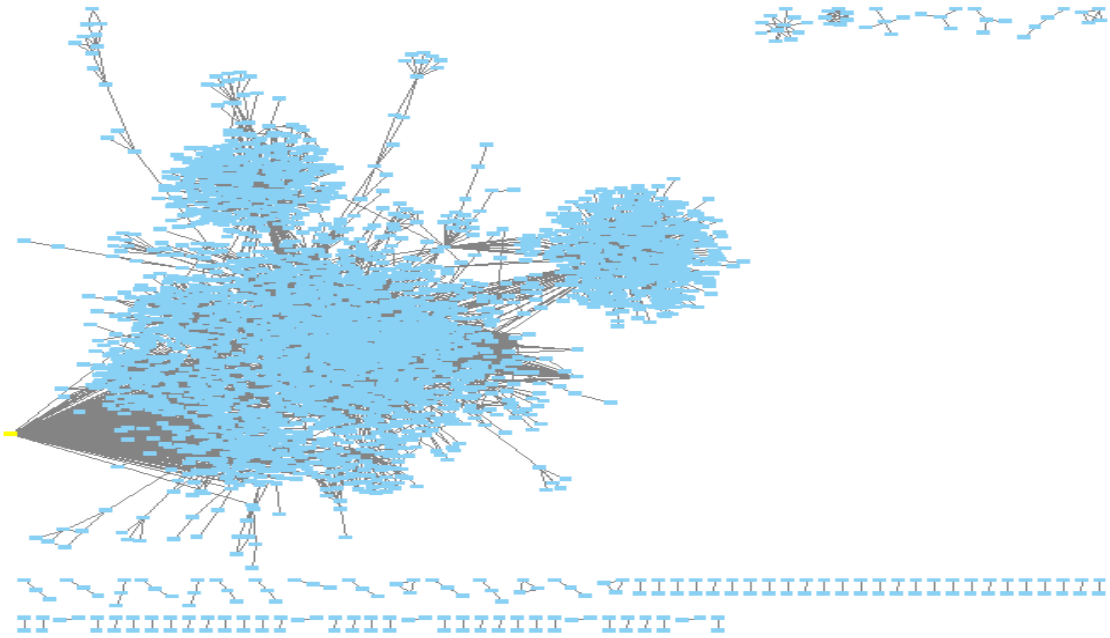

Figure 9 Entire network topology graph

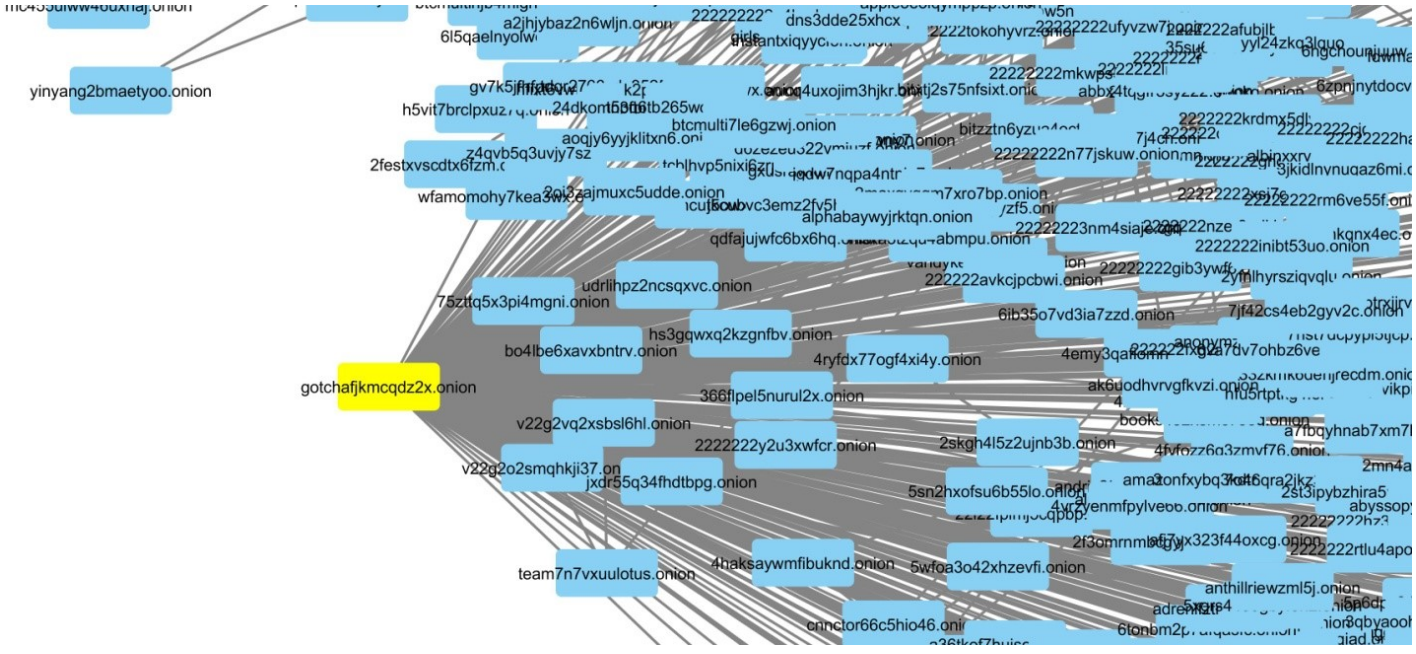

Figure 10 Network topology view showing the starting point of the crawl

4.2 Effect of crawling from one starting point

This section helps to understand how the view of the network changes with differing crawling algorithms and increasing crawl size. Starting from gotchafjkmcqdz2x.onion, three parallel

crawls were conducted. To analyze what was learnt about the network from the fixed starting point, as one progresses through all the crawling techniques, the script was run to extract the text only in batches of 100, 1000, 2000, 3000, and 3539 links and saved in their respective folders for analysis. The first hundred links were crawled in all three manners.

All the URLs for this analysis, now saved with their text alone, were fed into three different tools for analysis:

1) Document-Term similarity plots to see the difference in the behavior of the three crawling algorithms.

2) LDAvis was used to analyze the topic models obtained from three crawls.

3) HeatMapvis was used to contrast two sets of topic models to see the evolution of topics with increasing crawl size.

In figure 10, the node highlighted in yellow is the chosen starting point, and it can clearly be seen how dense its nearest neighbors are in the network. Figure 10 is a zoomed-in portion of the graph in figure 9 to show what the connections to this starting point look like.

4.2.1 Topic – Document interaction

In order to understand how techniques differ from each other over an increasing crawl size, document-similarity algorithm was used. RFS is the random selection of documents in the crawling process. For the first few links crawled, LDA was used to obtain the topic models by fixing K=10 topics in this case. The algorithm used generates three kinds of plots. The behavior of the crawling algorithms can be seen with the observations of all four kinds of plots.

To create a **non-weighted document-term similarity plot**, the algorithm ran through the first twenty words found in every topic model and compared every document with these words to give the number of documents that had at least one of the words of each topic model. If at least one of the words of topic models was found in the documents of the batch, it was considered similar or "matched". The **x axis** represents the topic models which have been sorted in such a way that they appear in descending order of size. The size of the topic model is determined by the number of words in it. The **y axis** represents the number of documents that match words with a particular topic model. One of the expected results from analysis of topic-document interaction

is that BFS would show the highest similarity or match as it remains in the network core before it starts exploring further. This can be clearly seen from the comparison of BFS curves from the plots where the first hundred links crawled are compared with the first thousand links crawled.

At this point, the topic models generated for each of the techniques are not very important. It was more important to see how many of the words found in them matched the documents because of these techniques.

Non weighted:

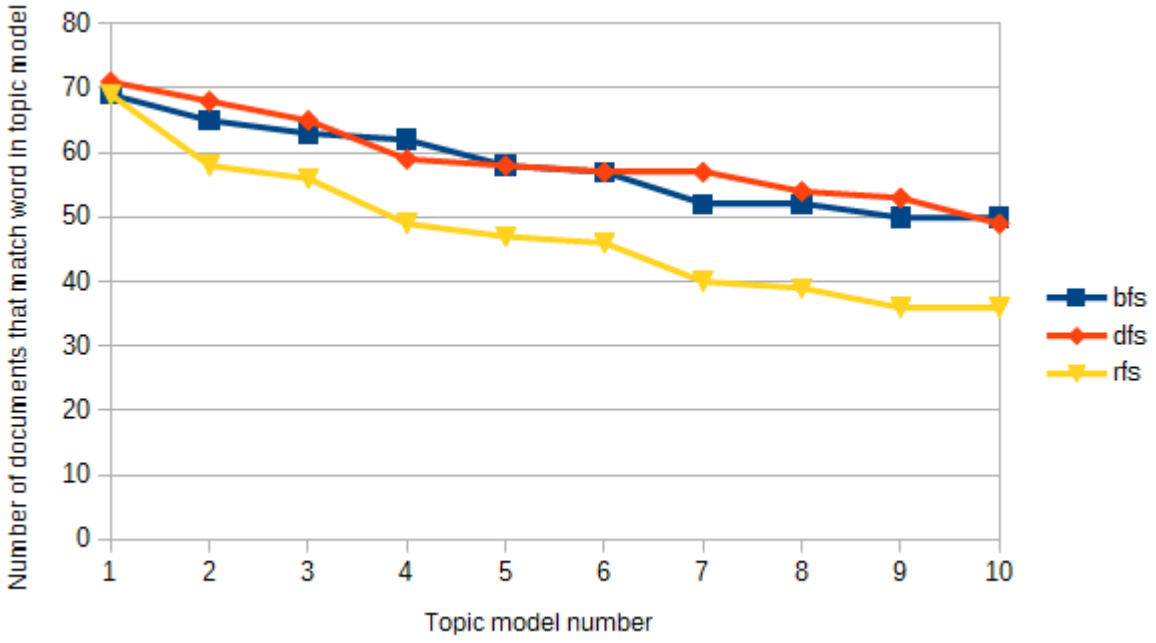

Figure 11 Non-weighted document-term similarity plot for first 100 links

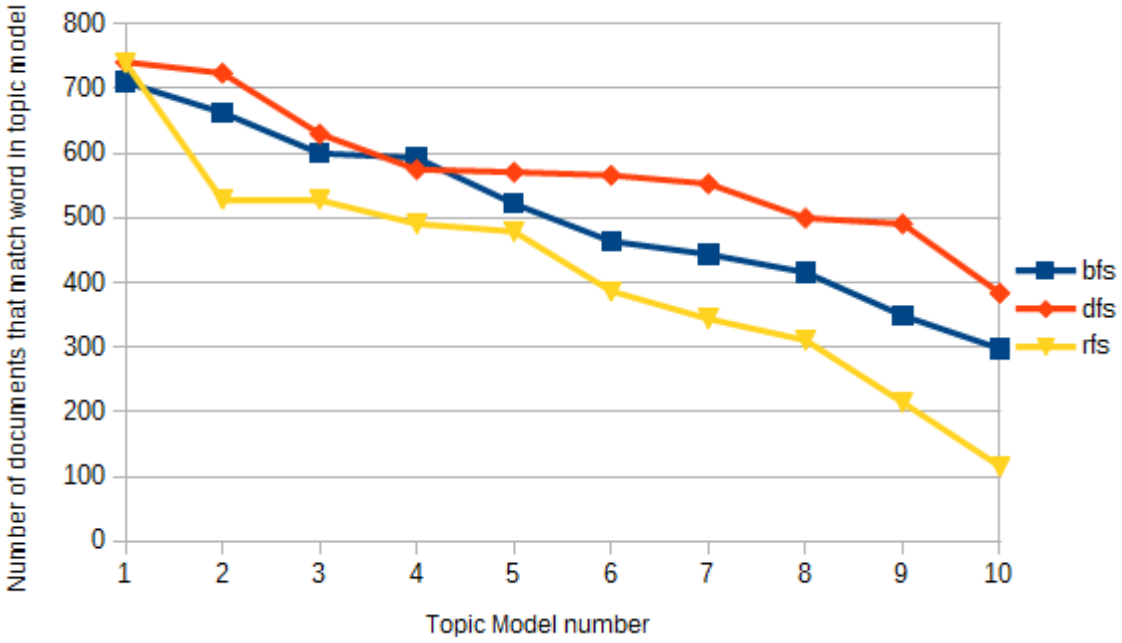

Figure 12 Non-weighted document-term similarity plot for first 1000 links

In figure 11, for the first hundred links crawled, the influence of choosing the starting point that happened to be most connected and had the highest branching factor was seen. For the same reason, little difference was seen in the biggest topic model for BFS, DFS, and RFS as there would not be much differences in the first few links irrespective of the algorithms. BFS and DFS together interchangeably showed higher matches than RFS. This shows that the similarity in topics between BFS and DFS is similar than when compared to random topics, because of RFS. This behavior also shows the importance of the network structure in terms of the similarity of webpages due to BFS and DFS.

As is evident from figure 12 there is a greater contrast when the links now number one thousand. A significant drop was seen in terms of BFS when the crawl size increased as there was a drop in the number of documents that matched the topic model words. DFS, however, seemed to fare well with increasing crawl size as there was a higher match with documents, in contrast to BFS. The fact that DFS created long chains of links, which means more variety in words, can be reflected in the way it found more matches of words than BFS when the first thousand links were considered. Additionally, BFS did not initially fare better than RFS with the thousand links

because the branching factor of the graph was not high, whereas that was not the case when only one hundred links were crawled.

One important factor to remember is that words in individual documents vary significantly as some can have fewer than five words, while others can be forums or blogs spanning across years from the date of creation of the web service.

To create a **weighted document-term similarity algorithm**, the algorithm ran in such a way that if at least one word from the topic model appeared in the document, this was counted however many times a word appeared in that document, and was considered on the **y-axis**. For example, if there were ten occurances of the word "btc" in a document it was considered as 1, but if there were "btc" and "bitcoins" and both words were in the same topic, the model occurance was considered as 2. In addition to this, the weighted aspect was because different words from a topic model that appeared in one document were given a consecutive weight; therefore, if three words of a topic model matched in one document, then its weight was 3. The number of documents that had a weight of more than 1 were counted and plotted on the y axis; therefore, the y-axis represents the sum of all the weights of the documents in terms of the words. The number on the y-axis is the number of unique words. The differences between the three techniques can be more clearly seen than with the non-weighted plot.

Weighted

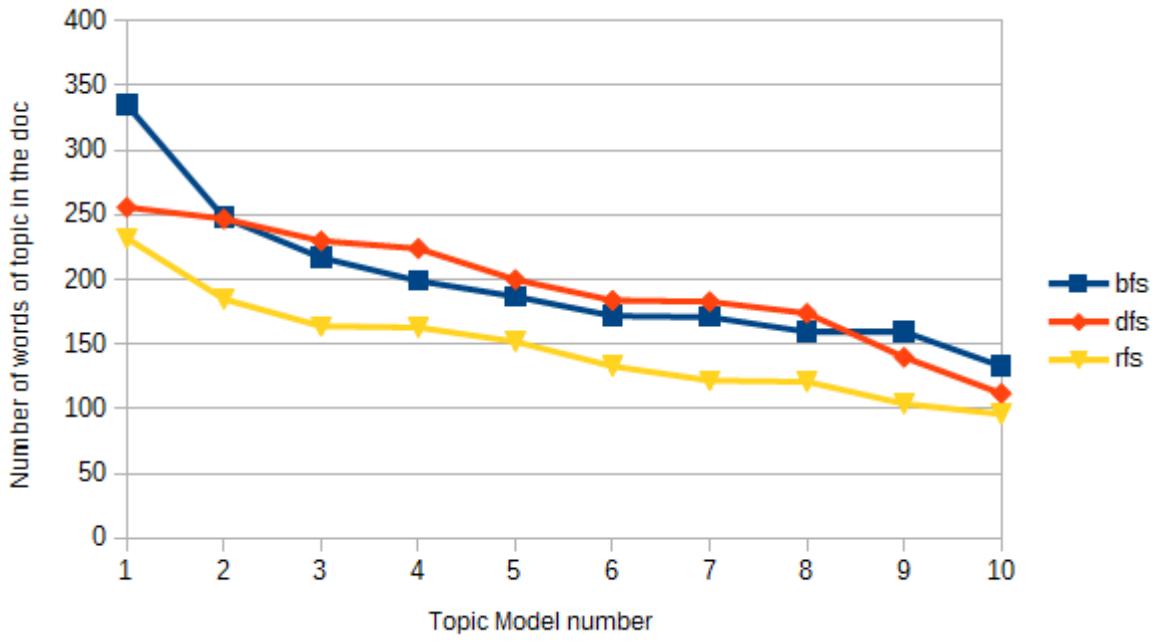

Figure 13 Weighted document-term similarity plot for first 100 links

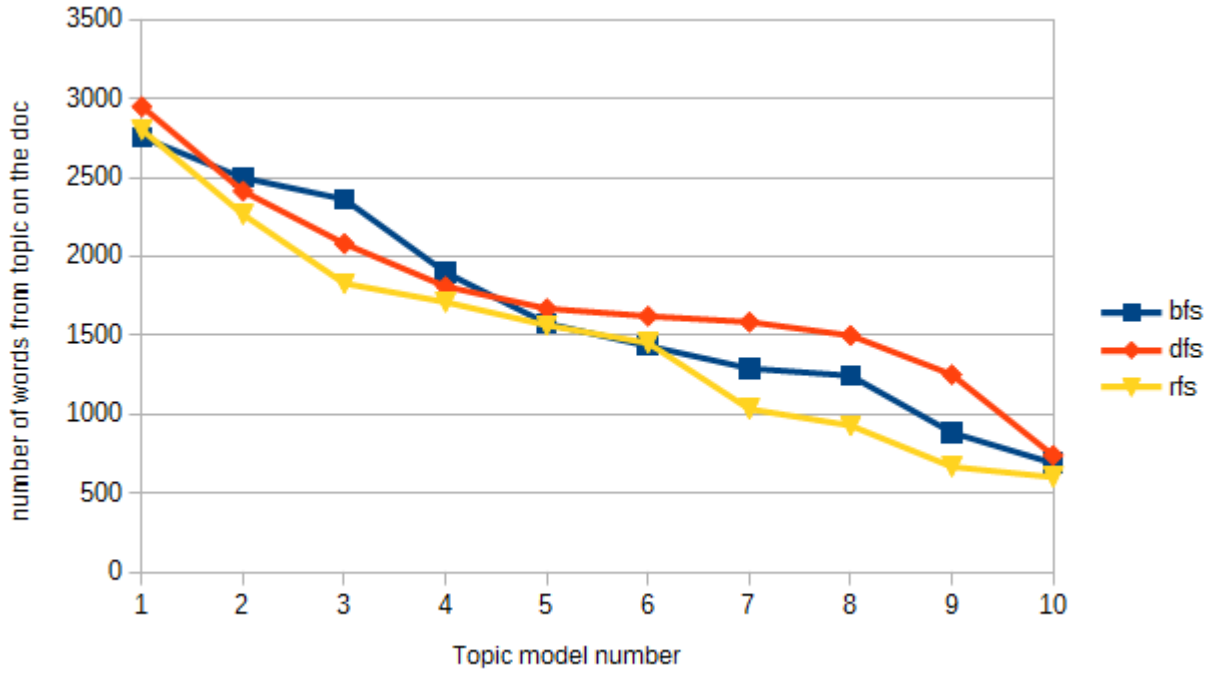

Figure 14 Weighted document-term similarity plot for first 1000 links

From figure 13 a huge difference was seen between BFS and DFS for first hundred links. The highest topic model was attributed to the BFS rather than the DFS and therefore BFS had a wider range in terms of words in the topic models because of the way it started to crawl. There were more unique words with BFS. However, as the crawl size increased, unique words started to decline as BFS started to spread.

Another point to note is the first three larger topic models had a larger value in the crawl size of one thousand links in figure 14. This indicates that, because of the BFS algorithm, more documents are talking about the same topic words, which is not the case with DFS because it runs away faster from the core points of the network due to the long links it creates. That was not the case when compared to the DFS of the first hundred links.

To understand document-term similarities for increasing crawl size, the algorithm was used to derive the document-term matches for all the batches. The batches were the first one thousand, followed by addition of the next thousand, followed by another thousand, making up three thousand links with the last batch being 3001-3539, which goes to the end of the crawl. Therefore, the **x axis** represents 1,2,3, and 4 to show they account for the abovementioned batches. On the **y-axis** in the case of BFS, the average of document-term match for batch 1 BFS_1000, batch 2 BFS_2000, batch 3 BFS_3000, and batch 4 BFS_3539 were first put in one column. The same process was carried out for DFS and RFS and put in their respective columns and all the averages were then shown on the y-axis to obtain figure 15. In this figure, the numbers 1-12 in the left most column represent the topic model numbers, and the individual columns indicate the average number of documents that have matched with words of a particular topic model.

	A	B	C	D	E	F	G	H	I	J	K	L
1	663	571	529	643	793	1083	1366	1430	1428	1046	1265	1696
2	600	491	740	1330	1290	1521	1215	1568	870	1512	2045	1016
3	522	630	116	1306	897	630	1591	2326	1643	2540	1880	1932
4	593	500	529	1124	1052	207	1348	1363	2183	1854	1384	2579
5	416	575	479	550	466	946	1362	596	1173	2101	1724	1384
6	444	384	491	881	1447	766	2356	1202	1254	1204	1098	1477
7	349	553	311	814	946	998	1424	928	2035	1384	2583	2417
8	464	724	215	1478	840	1524	1964	2156	316	2091	369	376
9	710	741	344	858	1544	1100	1123	2154	2267	1401	2405	2677
10	298	566	387	746	1233	910	1302	1359	1697	2624	2126	2003
11	**5059**	**5735**	**4141**	**9730**	**10508**	**9685**	**15051**	**15082**	**14866**	**17757**	**16879**	**17557**
12	BFS_1000	DFS_1000	RFS_1000	BFS_2000	DFS_2000	RFS_2000	BFS_3000	DFS_3000	RFS_3000	BFS_3539	DFS_3539	RFS_3539

Figure 15 Snapshot of first step to explain how document-term similarities plots for increasing crawl size were created

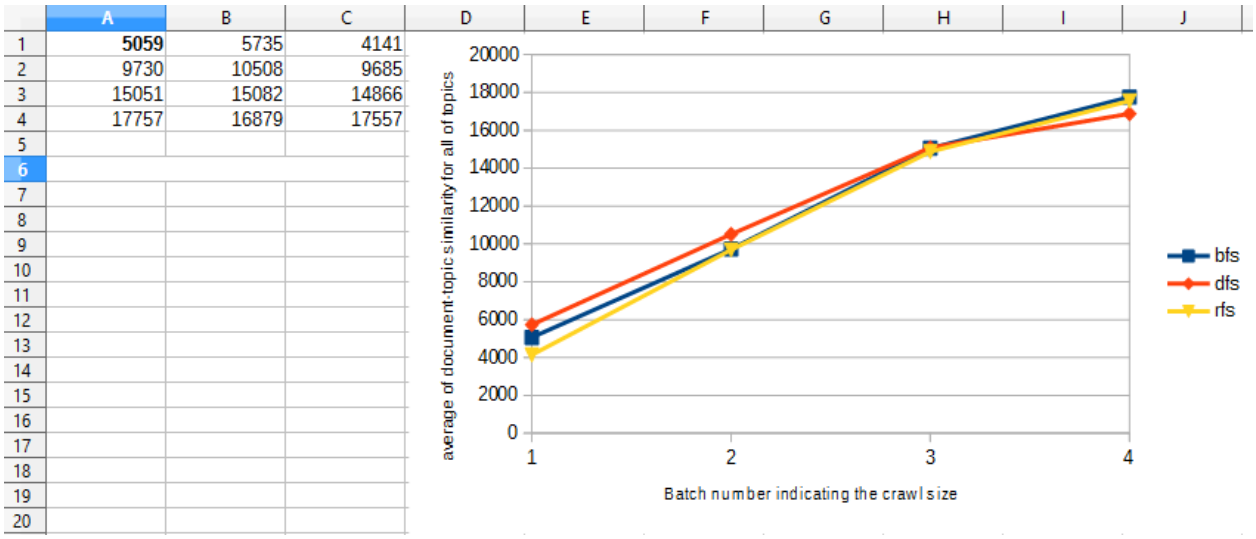

Figure 16 Snapshot of second step to explain how document-term similarities plots for increasing crawl size were created

If the document-term similarity is averaged for all topics for each of the techniques and plotted against the increasing crawl size, then how the techniques fare in contrast to each other can be seen. This is shown by figure 16. This plot should be read as BFS for crawl size of 1000 having on an average of 505 documents, DFS having 573 documents, and RFS having 414 documents that show matches with the 10 topic models.

Non-weighted document-term similarity for all batch numbers:

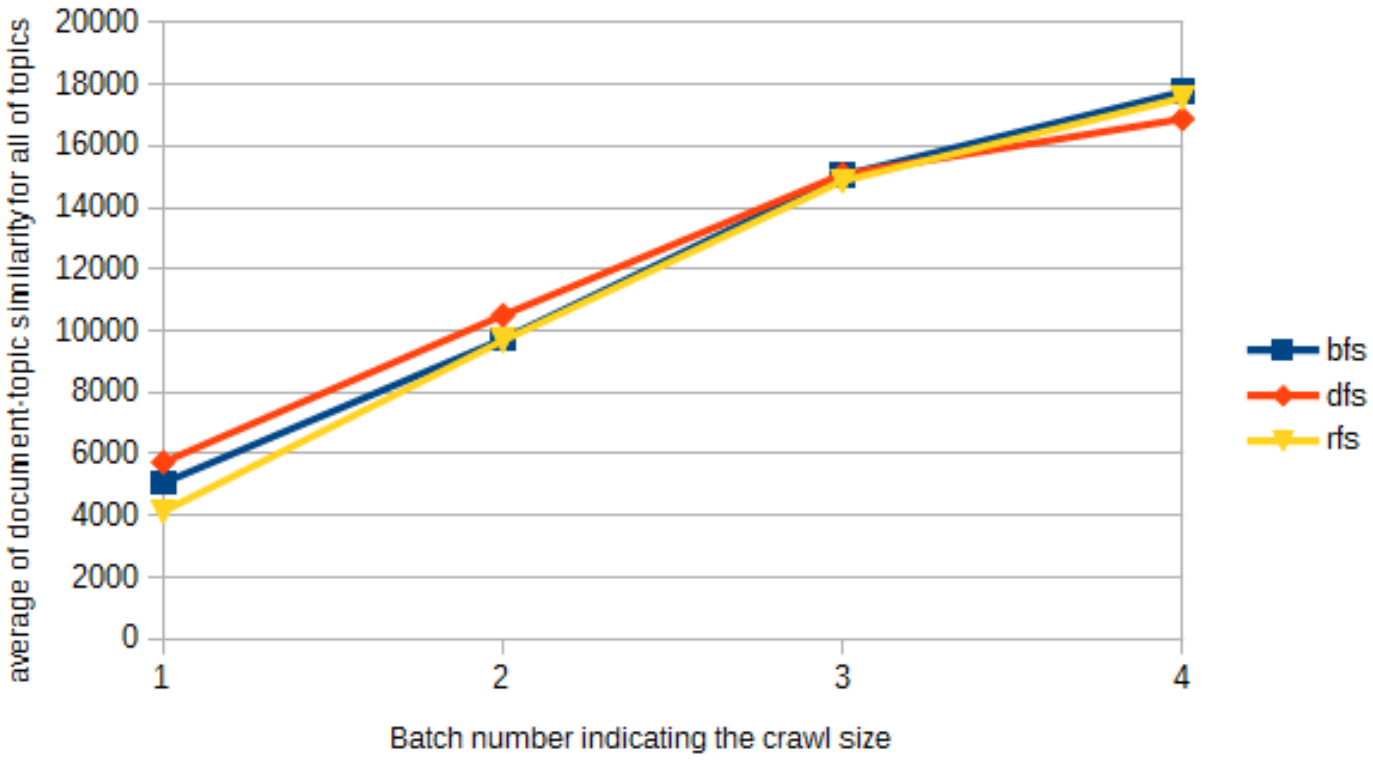

Figure 17 Non-weighted document-term similarity plot for all batch numbers

The observation made from non-weighted document-term similarity plot in figure 17 was that RFS was lower than BFS and DFS in the first three thousand links, indicating that the network structure of the graph is important. After three thousand links, RFS picked up a higher value, similar to that of BFS. Another observation is that the difference between RFS and the other two techniques was greater than when compared to an increasing number of links.

In terms of BFS and DFS, as one went far from the starting node the differences between them started to diminish whereas the differences were more clearly seen closer to the starting node. One can also argue for the phenomenon happening after three thousand links have been crawled that all three techniques show a similar behavior, but that beyond this point differences start to appear again.

Although the curves formed by the three algorithms seem to be very close to one another, they should be seen in the context of the number of documents that actually match with words in the

topic models as the crawl size increases. As the crawl size increases, documents of DFS seem to match more often with the topic models than documents of BFS do, indicating the greater variety of words matching in the case of DFS. Because of the way DFS behaves, it will be more prone to get a wider variety of words and topics faster than BFS as BFS tends to remain at the core, where documents of a similar nature or topics cluster together. At 3000 crawl size, however, all three algorithms show a minute convergence, indicating that BFS between 2000 and 3000 crawl started to get out of the core and started to have more variety in terms of words that matched the topic models; it continued to do so till the end of the crawl.

In conclusion, the following were the observations seen:

1) Closer to the highly connected starting point both BFS and DFS show similar topic models

2) BFS remains in the network core for longer period of time than DFS

3) As the crawl size increases, DFS performs better because of the variety of words found compared to BFS

4.2.2 Topic – Word relationship using LDAvis

In this section, the LDAvis-based method has been used as it gives a view of words and how large a certain topic is in the context of the entire corpus of data we have. The larger the area of a circle, the larger the prevalence of that topic in the dataset. The implementation of LDAvis have been sorted in decreasing order of prevalence. This is apparent from the area indicated by the topics. The visualizations presented in this section show topic models generated with K=10 for all batches, namely first thousand links, first two thousand links, and first three thousand links, until the end of the graph. This was done in order to maintain an achievable coherence as much as possible. Furthermore, the relevance term, λ, has been maintained at 1 for all batches. The main aim was to understand the differences resulting from different crawling techniques to contrast topics generated in relation to the words making them with this visualization.

From the batch of the first thousand links, 397 were the same in BFS and DFS, 285 links were the same in BFS and RFS, and 266 links were shared between DFS and RFS. As the starting point contained the same documents till the crawler branched out, similarities in the words were

expected to be seen in the first few topics. Words which create the "bigness" of a topic and what that looks like for the first thousand links crawled can also be seen.

Another point to take into account is that because the dataset worked upon was so varied in terms of the human topics that can be given, it was not a straightforward task to give titles to any of the topic models generated in this section. For this reason, observations have been made based on the words and the intertopical relationships between topics.

The following are snapshots of topic models generated out of the thousand links crawled in BFS, DFS, and RFS, and the observations made.

BFS_1000

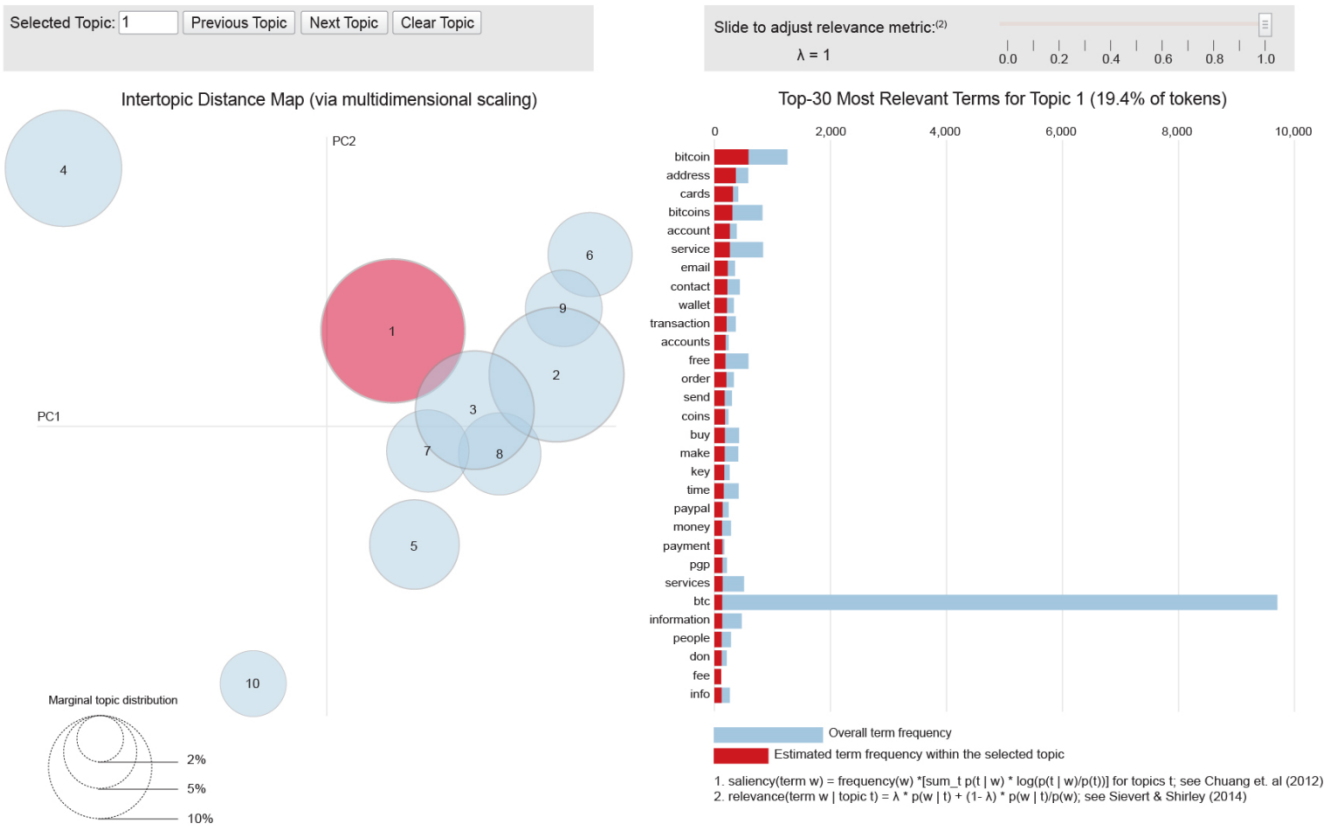

Figure 18 Snapshot of LDAvis topic model number 1 for first 1000 links crawled in BFS fashion

The top words of Topic 1, in figure 18, generated from the first thousand BFS crawled links include “bitcoin”, “address”, “cards”, “bitcoins”, “account”, “service”, “email”, “contact”, “wallet”, “transaction”, indicating that this topic relates to words that are bitcoin-payment related.

DFS_1000

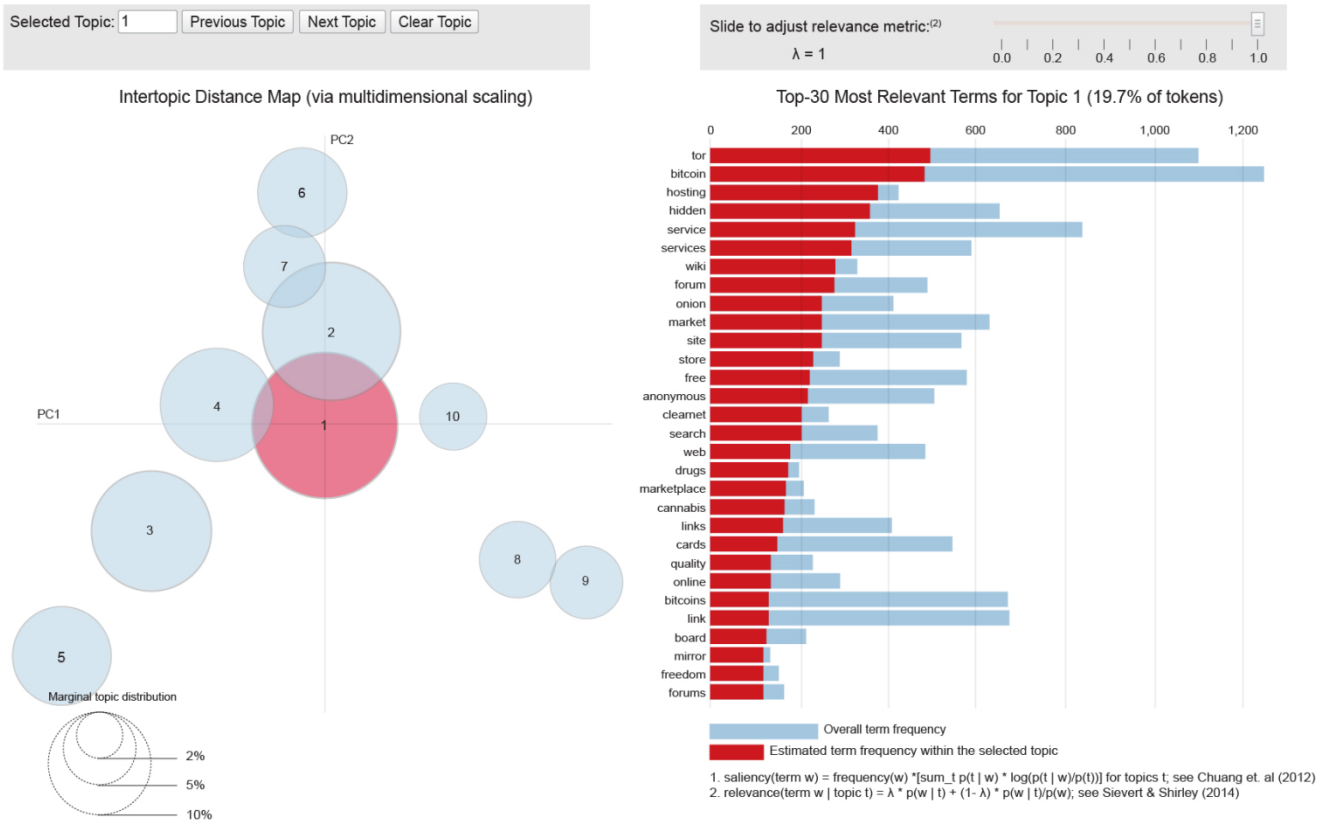

Figure 19 Snapshot of LDAvis topic model number 1 for first 1000 links crawled in DFS fashion

The top words of Topic 1, in figure 19, generated from the first thousand DFS crawled links, such as “tor”, “bitcoin”, “hosting”, “hidden”, “service”, “services”, “wiki”, “forum”, “onion”, and “market”, relevant to over 19.7% of all words, indicate this topic relates to words common to wiki.

RFS_1000

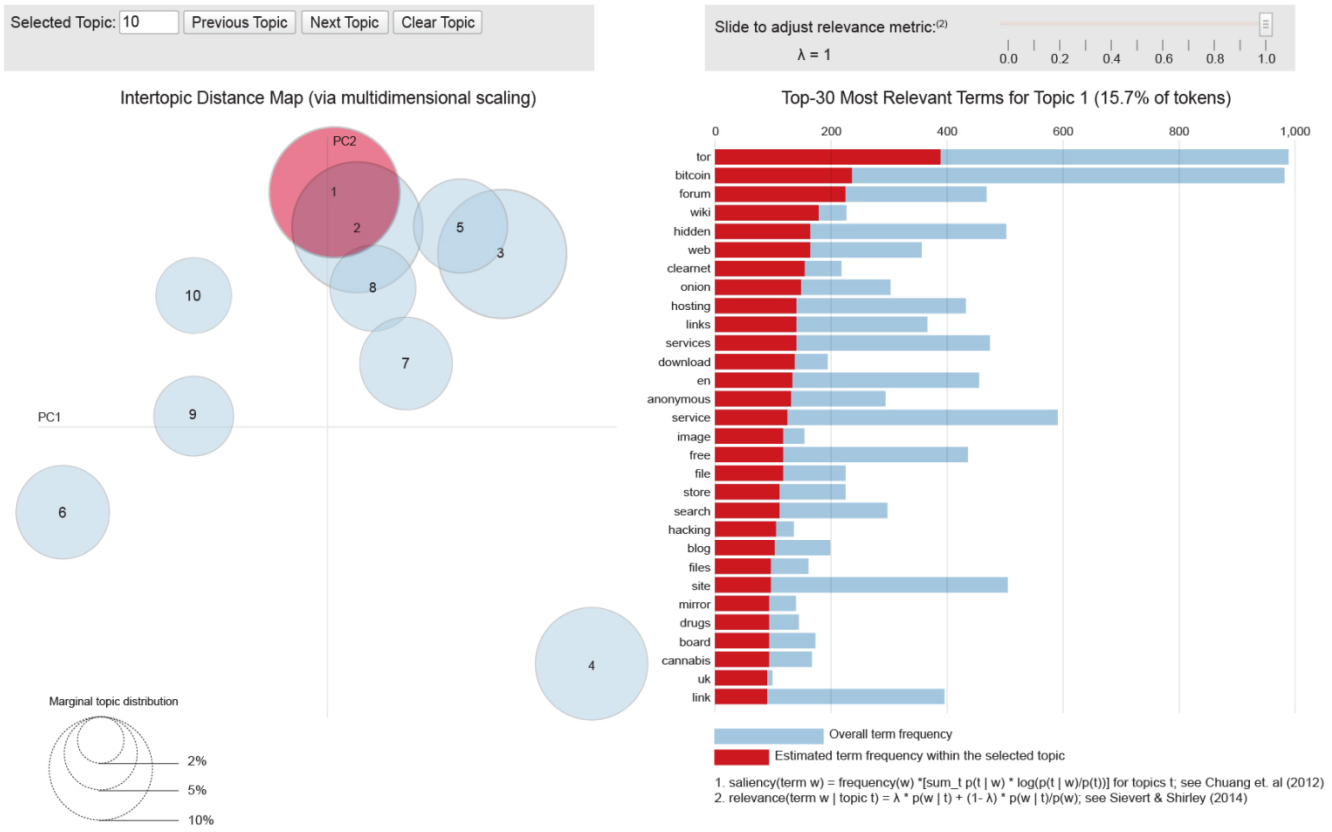

Figure 20 Snapshot of LDAvis topic model number 1 for first 1000 links crawled in RFS fashion

Another feature to note was how the topic distances differed in all these three batches. BFS and RFS showed some overlap in terms of the words they shared between topics but DFS was overall well spread. When figure 18 and 20 are compared, more words are common between topics for BFS and RFS than for DFS from figure 19, which is representative of how they differ.

One of the trade-offs is that, since the batches shared common links with one another, the occurence of the shared words found in those links, considered as documents for LDA, could affect the topic models generated. One way to analyze what the topics would be if such shared words were taken out would be pulling out such common links; however, this has not been done in this thesis as the aim is to maintain a level of coherence by keeping all the links intact, regardless of how many words overlap in the documents in the first thousand links crawled. It is also possible that the words that can be taken out are also found in the leftover documents. Another tradeoff is that since the choice of K was kept to 10, the topics could be too broad; therefore, it is possible that the top few words do not accurately represent the topic.

In table 5, a few selected topics have been shown to analyze the differences and commonalities:

Topic number	3	4	5	6	7	8	9
% of tokens/words	13.6%	13.2%	7.5%	6.6%	6.4%	6.4%	5.5%
BFS	Tor Tuesday information site security new law read posted share browser surveillance content Monday snowden	Btc bitcoins hours multiply flaw day make client transaction history hundredfold website investment amount today address found digital pending png win rich img deposit pay line notice	Usd eur buy gif card jpg products png mg amazon order images gift logo shipping cart original pcc quality bills login pure cannabis euro add blank sell shop register	Service hidden tor services server hosting und file kowloon contact links die key der support ubuntu von domain zu virtual add free keyringer ist host keybase tme rar png	Topics post posts number card view pm latest credit birth forum login information total market register account board security utc users moldova atm pin discussion day testosterone type month data	Online gb games download boy mb stries casino images iphone sex ru english boys galaxy play porn pdf videos apples free chat books image sets black cars video money motorcycles	De en la download le user forum enabled fr library na des les log ferme books linux em svar przez password version support est php login mai para
% of tokens/words	13.6%	11.8%	9.1%	7.4%	6.1%	5.5%	4.9%
DFS	Btc usd buy cards email account paypal cc card accounts order info payment euro contact shipping balance png free service jpg price guide	People time don make read money war world posted galaxy things group government issue lot years privacy porn point part comments work live real day life heroin surveillance good	Btc bitcoins bitcoin transaction address wins hours wallet make multiply send flaw amount day transaction client hundredfold website found time	Ago days apache configuration mg kamagra conf eur server default web enabled ubuntu site file debian page jpg files document hours images www pure var gram active report root located	String link iso ref actor env type server main anarplex end create diff count reference version support object error irc pony enabled post box crypto connect dir fun capability	Link phishing market de ddos alphabay protection por itima mensagem en la se cannabis es para ms lo mai una sim ver page topics bitcoin hansa acropolis	En de download directory jun le du books fr og tor til tails posted les in gg pdf mb english dir med la library die svar er linux av und cars

% of tokens/words	15.4%	11.5%	8.2%	8.2%	7.6%	7%	5.7%
RFS	Png bitcoin img voteup cards market bitcoins buy accounts card paypal credit email wallet free service contact anonymous account shipping tor search money hidden escrow	Btc bitcoins bitcoin multiply hundredfold client make transaction day website hours investment amount digital found address history win time money times hope pay returned	Post posts topics forum usd login register pm products today view board cannabis heroin latest discussion users logo online png utc cocaine shipping forums information	De itima mensagem por la en site freedom hosting mail freehosting le ver les el para di se est es des du una lo pour	Result initial ago array seed satoshi people services server hash hosting final client privacy winnings wager simplified kowoloon days days anarplex link summary link contact big tree shortlog make secret	Games play online notice line news undefined offset casino sex big live stories tags date game black rss april mr facebook top porn betting world jpg small read time gold	Gif tor gb und fr die hq img der add cart mdma lsd ist blank zv bdrip del books tab mb web das comment mit salevon de auch

Table 5 Table showing the topic models words generated by LDAvis for the first 1000 links crawled in BFS, DFS and RFS fashion

Some observations made while analysing topic-words contrast over the first thousand links are as follows:

1) While the first two topics for all three batches look similar, the differences began to show from topic 3 onwards.

2) Pages containing Italian words made a bigger appearance in the case of the RFS batch in comparison to BFS and DFS, where those words had a lower occupancy of 5.5% and 4.9%.

3) The DFS batch shows many more words related to types of drugs than that of BFS, which indicates the DFS crawler traveled further than BFS in the first thousand links.

As stated earlier, DFS in the intertopic map seems to be well-spread-out in terms of distances between the topics, except in the first two topics where it shows a slight overlap. However, in the case of RFS, topics 3 and 5 show an overlap of almost half while in the case of BFS, topic 3 shows an overlap with topics 1, 2, 7, and 8. While this is not evident from the top thirty words chosen for every topic, the overlap does show similarity in topics and therefore indicates BFS and RFS have not yet dissipated into the network like DFS has.

To understand the phenomenon as we move away from the starting point, topic 1 for all batches has been analysed from figures 21 and 22.

BFS

Figure 21 Evolution of topic model number 1 as the crawl size increases in BFS manner

DFS

Figure 22 Evolution of topic model number 1 as the crawl size increases in DFS manner

By comparing BFS and DFS batches in the context of their biggest topic model alone, the differences in the way the other topics start to arrange themselves becomes visible. Words in each of these incremental batches have increased but those words are different as BFS follows a different route to DFS. The LDA implementation used by LDAvis gives the same topic model every time it is run. How topic 1 starts to show a greater variety in words until it reaches the end of the crawl can also be seen. Table 6 shows the new words that are added to the first topic model as the crawl increases.

	1000	2000	3000	3539
% of tokens/words	19.4%	18.8%	18.1%	17.3%
BFS	Bitcoin address cards bitcoins account service email contact wallet transaction accounts free order send coins buy make key time paypal money payment pgp services btc information people don free info	Bitcoin tor address service server contact bitcoins key email password free link account public privacy send services support site page network access wallet anonymous website transaction hidden login data time	Bitcoin tor address service contact key free server site bitcoins services email account password time privacy public send page link network make pgp website log access hidden wallet don find	Tor bitcoin hidden forum service wiki onion anonymous services free kamagra clearnet market hosting web links cards store marketplace site drugs search board quality mg link escrow forums paypal high
	1000	**2000**	**3000**	**3539**
% of tokens/words	19.7%	20.9%	17.1%	17.3%
DFS	Tor bitcoin hosting hidden service services wiki forum onion market site store free anonymous clearnet search web drugs marketplace cannabis links cards quality online bitcoins link board mirror freedom	Tor contact link server service public people time services error access data site don read links internet privacy address key free make information security end string	Tor hidden service bitcoin forum onion wiki anonymous free services market links clearnet webhosting store site search marketplace board drugs cards directory link escrow quality forums paypal blog	Tor bitcoin hidden forum service wiki onion anonymous services free kamagra clearnet market hosting web links cards store marketplace site drugs search board quality mg link escrow forums paypal high

forums	files

Table 6 Evolution of words of topic model number 1 as the crawl size increases in BFS and DFS order

4.2.3 Topic-Topic interaction using heat map visualizations

Figure 21 shows the heat map visualizations obtained upon the crawled websites with BFS and DFS over every thousand links incrementally until the end of the connected graph. These visualizations **give insight into how different the view of the network can be in terms of topics** with different algorithms and different crawl sizes. Each one is generated by comparing the topic models of the main batch, which is the entire list of hidden services used in this thesis, against the batch under investigation. Model 1 in the heat maps represents the main batch. Model 2 was used for the comparison. All the words of each topic model of a particular batch have been compared against all the words of the corresponding topic model of the other batch . Since the main batch alone generated close to twenty thousand unique words, only the top few words have been chosen to get an idea of what the generated topic model in a batch is about. The colors and their corresponding values in the tables below show examples of the words due of which topic models show similarity. This section explains what happens in all the four batches crawled in BFS and DFS: one thousand links, two thousand links, three thousand links, and 3539 links.

BFS_1000

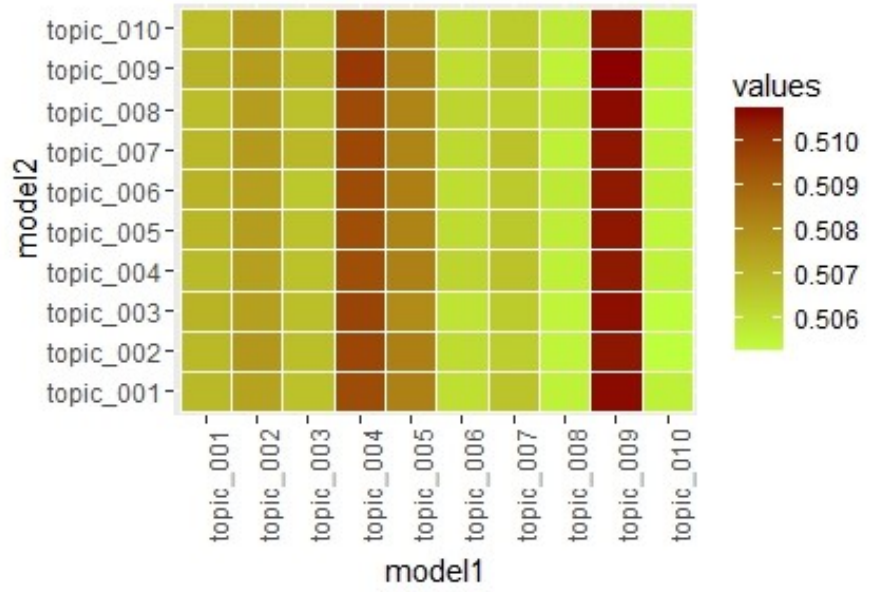

Figure 23 Heat map visualization comparing the main batch of 3539 links with the first thousand links crawled in BFS order

From BFS_1000, in figure 23, topic 9, topic 4, and topic 5 of the main batch show the highest matches with every topic model of BFS_1000 batch. This is a very interesting phenomenon which indicates that most of the topics found in the first thousand links crawled in BFS order are mostly related to **payments, hosting, wiki**, and related forums. This can be seen when the words of the tables 7 and 8 given are compared. The top three topic models of the main batch appear in all the topic models of the BFS_1000 batch. Some of the common words have been highlighted as examples. Fewer words have been chosen in this case for the tables, as the rest of the words do not show a greater variety in terms of the type of topic models.

Topic 9	png,img,bitcoin,tor,voteup,service,market,hidden,services,anonymous,free,hosting,forum,links,onion,wiki,web,store,clearnet,cards
Topic 4	tor,server,post,posts,site,link,web,contact,service,topics,page,note,login,public,key,address,hidden,support,information,search,password
Topic 5	days,ago,usd,card,buy,cards,account,credit,email,order,number,jpg,shipping,accounts,paypal,payment,contact,cc,address,price

Table 7 Topic models and their words in the main batch that show strong matches with topic models in the BFS_1000 batch

Topic 1	eur,mg,kamagra,jpg,gbp,productpricequantity,und,gb,pcs,**png**,login,add,products,von,logo,jelly,order,original,pure,cart,time,images
Topic 2	**days,ago**,**card**,cards,number,account,information,order,credit,contact,gif,security,buy,shipping,free,birth,log,email,read,site,amazon
Topic 3	**bitcoin**,bitcoins,address,wallet,transaction,service,buy,coins,send,btc,tor,email,png,hours,**account**,key,fee,time,amount,**password**
Topic 4	przez,na,postw,nie,post,dzia,ostatni,**forum**,tematw,sie,torepublic,ci,jest,fora,pgp,dla,nas,za,po,cej,torowicze,sluzby
Topic 5	tuesday,monday,larry,wednesday,porn,sex,rar,people,job,ill,hacking,big,good,counted,young,hacker,mwm,im,boy,tags,boys

Topic 6	btc,bitcoins,**bitcoin**,hours,flaw,multiply,day,client,make,transaction,history,hundredfold,pending,investment,website,found
Topic 7	usd,**buy**,browser,sell,south,javascript,worth,park,em,forbidden,act,modern,sep,jan,upload,show,enabled,de,globaleaks,productpricequantity
Topic 8	topics,post,posts,de,view,latest,la,pm,forum,en,stories,ru,**market**,board,le,total,discussion,**login**,**support**,boy,users
Topic 9	tor,hidden,service,bitcoin,links,**onion**,**free**,**services**,forum,site,download,anonymous,wiki,online,web,**hosting**,Clearnet
Topic 10	line,notice,undefined,offset,del,svar,inlgg,mnen,calibre,books,la,library,de,el,gud,inga,saizou,sedan,sista,av,mnader,**forum**

Table 8 Table showing all the topic models and their words of BFS_1000 links batch with highlighted words that show a match with the main batch topic models words

BFS_2000

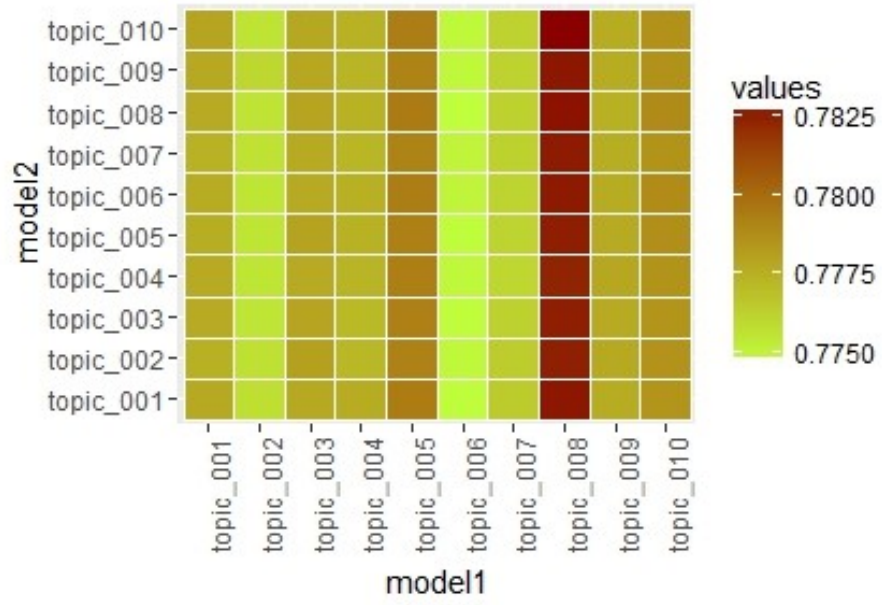

Figure 24 Heat map visualization comparing the main batch of 3539 links with the first two thousand links crawled in BFS order

Topic 8	tor,**hidden**,**service**,services,web,onion,site,hosting,free,links,forum,server,wiki,bitcoin,link,anonymous,page,clearnet,network,people,contact,files,file,search,information,internet,sites,security,blog,public,hacking,time,directory,download,board,news,email

Topic 5	ago,days,post,posts,topics,login,mdma,register,pm,forum,eur,password,view,lsd,log,hash,latest,server,result,support,array,initial,user,hq,cannabis,seed,pure,kush,version,reply,users,username,board,enabled,cocaine,online,og,total,qiwi,hours,today,posted,news,members,heroin,discussion,white,shop,faq,rules,key,times,home,products,rss,minutes

Table 9Topic models and their words in the main batch that show strong matches with topic models in BFS_2000 batch

Topic 1	phishing,service,link,hidden,ddos,protection,alphabay,market,site,freedom,tuesday,hosting,ii,hosted,freehosting,people,links,monday,tor,larry,dragon,wednesday,ubuntu,tk,hacking,big,sex,html,torture,job,ill,slavery,im,hacker,problem,apache
Topic 2	de,mg,kamagra,eur,la,le,en,gbp,pcs,les,mai,des,jelly,kush,du,apcalis,treatment,oral,productpricequantity,pure,pour,est,tablets,blood,super,erection,active,vous,lsd,pas,del,effective,il,par,sur,au,jpg,polo,citrate,nous,une,ce,ingredient,time,hours
Topic 3	jun,na,nie,ro,anonymous,przez,sie,jest,forum,postw,si,pgp,ci,google,torepublic,te,imgops,jak,kb,za,po,odpowied,dzia,exif,ale,query,aby,xd,fora,ostatni,ze,em,post,begin,tak,signature,kliknij,zobaczy,pomini,tematw,tn,czy,posty,dla,nas
Topic 4	gif,books,und,de,post,der,die,tor,topics,von,den,ist,posts,das,ltima,mensagem,por,view,mit,fr,du,latest,er,zu,auf,pm,auch,nicht,ein,en,werden,zum,wed,utc,tue,logo,dem,tpicos,da,des,hier,ber,oder,sind,eine,um,mensagens,sun,es,apr,javascript
Topic 5	btc,bitcoins,bitcoin,transaction,hours,multiply,amount,make,flaw,address,day,email,client,account,hundredfold,buy,wallet,coins,website,cc,money,paypal,pp,send,found,investment,digital,pay,deposit,time,win,acc,history,accounts,transactions,png,today,info,balance,find,rich,blockchain
Topic 6	usd,topics,posts,post,login,register,jpg,euro,pm,eur,forum,products,latest,view,cannabis,shipping,market,buy,productpricequantity,logo,bills,quality,sell,product,images,heroin,gram,news,order,discussion,info,cart,mm,users,online,today,april,total,board,shop,security
Topic 7	ago,days,online,error,og,socks,games,connect,failed,host,casino,unreachable,til,tails,inlgg,bdrip,cars,med,qiwi,rss,access,af,svar,av,det,play,kan,motorcycles,utc,localhost,occurred,authorization,challenge,generated,keybase,polipo,mnen,nbome,avc,trdar,och,som,game,dvdrip
Topic 8	tor,card,contact,address,note,free,service,cards,png,account,information,people,bitcoin,time,order,buy,email,number,don,key,services,server,site,link,public,security,credit,read,privacy,home,make,november,data,code,send,internet,network,access,password,faq,page,mail,anonymous
Topic 9	apache,configuration,enabled,server,de,default,web,file,string,page,debian,en,ubuntu,var,version,site,support,iso,conf,actor,line,el,env,php,http,centos,files,notice,main,la,user,undefined,offset,

	ref,document,error,root,es,count,log,ms,para,ver,lo,object
Topic 10	tor,bitcoin,hidden,service,forum,onion,services,hosting,anonymous,links,free,wiki,web,clearnet ,png,market,site,search,store,board,directory,img,marketplace,download,blog,files,file,cards,quali ty,page,drugs,upload,paypal,cannabis,link,archive,chat,forums,online,high,bitcoins,mirror,hacking

Table 10 Table showing all the topic models and their words in the BFS_2000 links batch with highlighting words that show matches with the main batch topic models words

When one thousand more links are crawled with BFS from the starting point, it can be noted, from figure 24 and tables 9 and 10 that some words related to **drugs**, as shown in topic 5 of the main batch words make their appearance in all topic models of the BFS_2000 batch. Topic 8 still contains words that strongly match with those in the BFS_1000 batch, as is the case with the BFS_2000 batch, indicating that the documents contained in BFS_2000 are still around the network core or, still close to the starting point. Words which could indicate greater variety are not yet seen because, even up to this point, the crawl has not reached the other types of domains that are interconnected.

BFS_3000

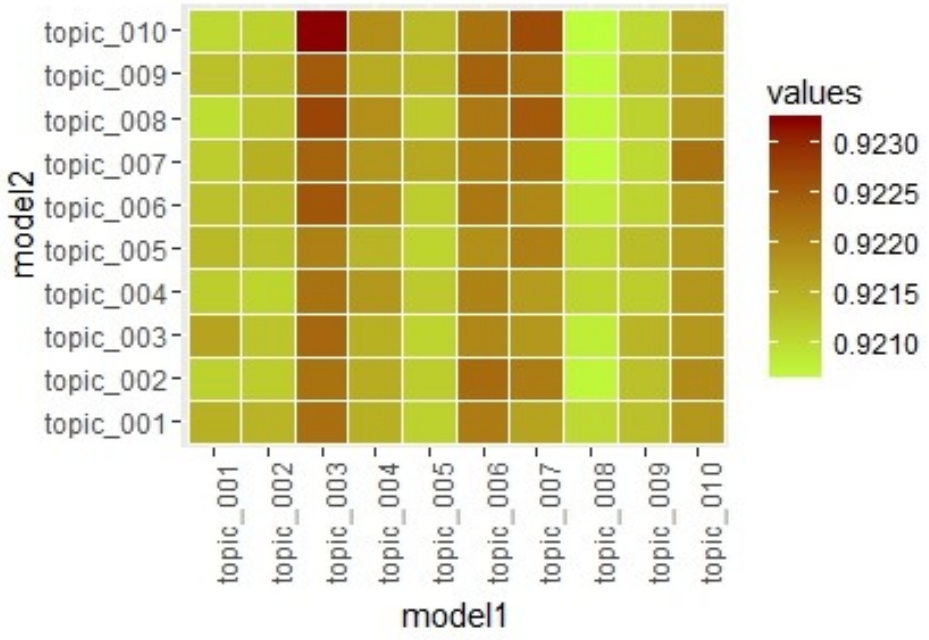

Figure 25 Heat map visualization comparing the main batch of 3,539 links with the first three thousand links crawled in BFS

Topic number	Main batch	BFS links crawled upto 3000 links
1	tor,de,hidden,onion,service,forum,wiki,books,en,links,gif,clearnet,la,bitcoin,le,web,hosting,und,board,anonymous,du,blog,les,services,des,free,chat,der,site	de,en,la,el,del,una,se,es,lo,los,para,hq,con,su,las,por,di,al,tu,si,te,pero,como,mas,il,ms,ya,cuando,solo,este,jpg,sus,le,mi,ver,imagenes,yo,ojos
2	de,la,en,el,una,es,se,lo,del,los,con,su,las,al,si,tu,di,ms,pero,le,comments,til,il,ya,jpg,webshell,leer,cuando,solo,posted,sus,mi,forbidden,imagenes,ojos	mg,kamagra,eur,people,gbp,productpricequantity,gram,pure,time,active,jelly,super,cocaine,treatment,kush,make,apcalis,oral,hours,products,work,heroin,cannabis
3	tor,post,server,posts,topics,site,web,link,public,page,contact,login,password,service,key,forum,hidden,access,security,information,address	de,hosting,site,ii,le,freedom,hosted,freehosting,la,ltima,mensagem,apache,por,mai,enabled,services,les,configuration,en,des,server,kowloon,support,du,default
4	btc,bitcoins,bitcoin,multiply,transaction,flaw,hours,hundredfold,client,make,day,website,investment,found,amount,digital,address	btc,bitcoins,bitcoin,transaction,multiply,hours,flaw,client,make,hundredfold,day,website,found,investment,digital,address,amount,win,history,today,deposit
5	mg,kamagra,eur,mdma,phishing,lsd,error,gbp,productpricequantity,gram,qiwi,hq,pure,cannabis,cocaine,pills,kush,nbome,socks,pcs,failed	pcs,mdma,lsd,na,qiwi,nbome,nie,en,og,ro,anonymous,til,forum,med,przez,er,inlgg,shop,faq,jabber,sie,de,white,det,listings,te,rss,pgp,av,svar,af,du,jest,postw
6	ago,days,hosting,site,ii,freedom,hosted,freehosting,usd,ddos,protection,alphabay,blank,gif,sell,counted,inlgg,worth,av,bdrip,displayed	hidden,service,read,comments,tuesday,posted,line,link,tor,comment,keyringer,end,html,notice,key,anarplex,post,server,links,leave,monday,public,ref,undefined,object

7	png,img,bitcoin,voteup,market,buy,cards,card,service,free,paypal,usd,credit,email,accounts,account,bitcoins,wallet,anonymous,services,order	gif,download,server,password,gb,log,tor,login,browser,array,directory,javascript,file,result,initial,access,satoshi,seed,mb,reply,jun,register,user,books,page
8	na,nie,jun,directory,sie,ro,thumbnail,anonymous,mm,forum,pgp,przez,fuer,jungs,bild,angezeigt,vollgroesse,anklicken,torepublic,jest,wtf,si,logo	usd,card,buy,account,address,email,number,cards,credit,order,contact,note,send,shipping,payment,paypal,jpg,cc,png,accounts,days,information,error,info,free,price,birth
9	de,por,ltima,mensagem,services,para,hosting,kowloon,support,mai,em,ver,mensagens,tpicos,te,enabled,se,version,virtual,contact,da,como,php,mas	ago,days,post,posts,topics,books,pm,und,gif,der,latest,view,die,forum,tor,von,ist,den,das,wed,board,zu,mit,fr,auf,today,discussion,total,de,nicht,auch,users,ein,ber,login
10	note,november,people,read,tuesday,nsa,propublica,snowden,games,play,war,foxxxydox,jpg,online,casino,money,government,gr,world,line,mr,game,june,monday	tor,png,bitcoin,img,service,voteup,market,hidden,services,free,onion,link,anonymous,forum,search,links,site,web,clearnet,wiki,hosting,online,store,information,security,network,news,wallet

Table 11 Topic models and their words in the main batch and those of the BFS_3000 batch

As shown in figure 25 and table 11, strong topic models with **Italian words** begin to appear in topics number 1 and 2 when three thousand links have been crawled. Additionally, how the words belonging to the main batch make their appearance in these three thousand links can be seen. There seems to be some dispersion in the matches between topic models where one or two strong topic models, as in the case of the first two batches, no longer show up in that manner. This is coherent with the next batch where, while we are getting close to the end of the graph, we begin to see more or less equal matches between both the topic models being compared.

BFS_3539

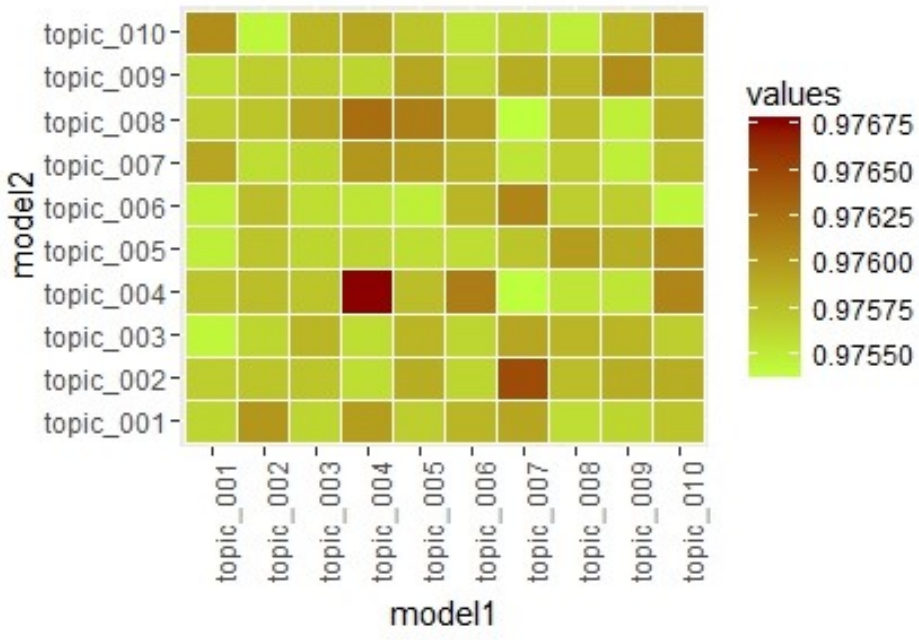

Figure 26 Heat map visualization comparing the main batch of 3,539 links with the entire network crawled in the BFS order

Topic number	Main batch	BFS crawled till the end of the graph
1	btc,bitcoins,bitcoin,multiply,flaw,transaction,client,hundredfold,hours,make,day,website,found,investment,digital	download,jun,html,nginx,access,file,torture,htm,click,rar,forbidden,nameless,stream,keybase,avc,url,title,fodor,cryptome
2	eur,mdma,error,lsd,productpricequantity,login,qiwi,hq,mg,kush,connect,pure,socks,failed,host,unreachable,access,shop	de,hosting,post,site,posts,topics,freedom,ii,hosted,freehosting,en,le,la,services,pm,les,kowloon,support,des,forum,view
3	people,read,november,time,internet,privacy,data,link,public,posted,don,government,world,law,nsa,news,security,comments	ltima,mensagem,por,mai,mensagens,tpicos,overchan,foxxxydox,ver,premier,webshell,po,przez,na,hacking,postw,abr,dog,dzia,php,fb,fora,irc,nntpchan,kidsleepy
4	tor,png,img,voteup,hidden,service,bitcoin,market,web,anonymous,links,onion,site,link,free,services,forum,wiki,server,clearnet	tor,png,bitcoin,service,img,hidden,voteup,market,free,web,site,buy,cards,anonymous,services,onion,links,email,usd,search,card,forum,link,contact
5	card,number,credit,phishing,birth,online,type,information,result,array,security,code,initial,month,seed,address,money,reply,account	nie,na,ro,anonymous,en,pgp,forum,til,counted,inlgg,av,med,jesusofrave,af,torepublic,jest,det,svar,signature,te,begin,om,jak,google,za,och
6	usd,buy,bitcoin,contact,email,account,services,cards,address,service,free,order,accounts,note,png,send,paypal,cc,shipping,payment	gif,books,und,der,die,fr,tor,seed,de,von,ist,das,den,em,zu,server,sie,mit,auf,du,eur,nicht,um,er,auch,ein,zum,ber,para,werden,onion,logo
7	de,en,le,la,books,gif,ltima,mensagem,und,por,du,les,des,mai,der,fr,tor,die,est,ist,das,von,pour,em	ago,days,mdma,lsd,gram,qiwi,hq,result,kush,nbome,initial,hash,online,cannabis,tuesday,shop,og,pills,cart,games,white,mg,casino,play,hours,winnings

	,mensagens,tpicos,para,vous,den,sur,pas,se	
8	kamagra,mg,gbp,nbome,active,ddos,jelly,treatment,oral,apcalis,ingredient,protection,anonymous,tablets,blood,ro,super,erection,alphabay,viagra,buy,effective,pill,citrate	people,mg,kamagra,time,error,read,november,connect,make,host,access,internet,don,log,comments,array,privacy,link,data,eur,active,government,jpg,gbp,law
9	de,la,el,se,una,es,lo,los,pcs,na,para,si,jun,con,las,su,por,te,jpg,al,ms,del,tu,nie,pero,como,ya,este,pgp,mas,mi,leer,cuando,comments,solo,sus,torepublic,esta,listings,rar	de,la,en,el,se,del,una,es,lo,para,jpg,los,por,si,con,las,su,al,tu,di,te,ms,mm,pero,como,este,mas,comments,ya,sim,ver,leer,mi,cuando,il,solo,le,sus,esta
10	ago,days,post,site,posts,topics,freedom,hosting,hosted,ii,freehosting,pm,forum,view,login,search,latest,register,board,users,home,vendor,news,key,today	btc,bitcoins,bitcoin,transaction,multiply,flaw,hours,client,hundredfold,make,day,website,amount,address,investment,found,digital,win,money,time,history

Table 12 Topic models and their words in the main batch and those of all links crawled until the end of the graph in BFS fashion

As mentioned in chapter 3, due to the probabilistic nature of LDA when assigning words into topics, every time LDA is run we will get a different ordering of topic models, as one can clearly see in the table 12 and figure 26. In a perfect hypothetical scenario, if topic models could be generated the same every time, this would be a way to keep the order intact and by now most of the squares in the heat map would have darker colors.

This a consistent phenomenon observed where a batch with three thousand links, as shown above, shows a midway bridge whereas when only two thousand links were crawled there was not enough representiveness. However, **the closer we get to crawling more than three thousand links, we begin to see how representativeness increases** and starts to look similar to our ground truth batch. In this case, it can be inferred that one has to crawl at least three thousand links and above to be able to get a better view of this network.

DFS_1000

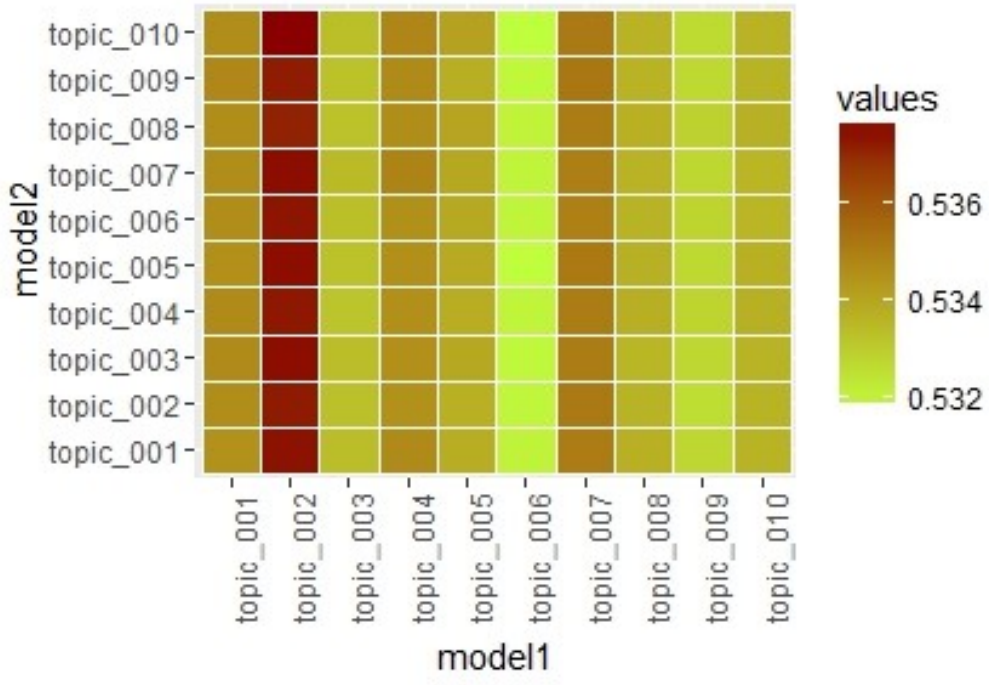

Figure 27 Heat map visualization comparing the main batch of 3,539 links with the first thousand links crawled in DFS order

Topic 2	tor,hidden,service,web,onion,forum,bitcoin,links,wiki,free,site,services,anonymous,clearnet,server,file,files,page,hosting,board,link,search,sites,store,network,support,blog,gb,apache,marketplace,forums,hacking,mirror,download,deep,market,enabled,archive,information
Topic 1	site,post,posts,freedom,topics,hosting,hosted,ii,freehosting,link,latest,pm,people,view,internet,today,discussion,users,surveillance,war,snowden,irc,html,anarplex,government,project,board,tor,secret,nsa
Topic 7	usd,propublica,foxxxydox,string,jpg,line,ro,tuesday,notice,iso,sell,ref,stories,undefined,offset,posted,series,object,worth,larry,pass,error,test,rss,type,april,cars,story,fiction,reference,motorcycles,notes,pennsylvania,capability,march,begin,february,south,bill,pdf,sets,red,message,kb,uv,january,agent,pen,cop,query,connection,entire,cgan,york,blank,september,orange
Topic 4	png,img,voteup,bitcoin,market,buy,card,cards,services,credit,service,accounts,email,paypal,free,account,contact,bitcoins,address,wallet,number,order,hosting,shipping,anonymous,link,money,price,search,tor,cc,usa,send,usd,information,payment,info,alphabay,ddos,phishing,euro,darknet,mail,full,images,balance,escrow,home,protection,good,coins,store,eur,eu,pp,kowloon,birth,hidden,links,prices,site,verified,bills,worldwide,security
Topic 10	el,una,lo,na,los,nie,con,si,su,te,la,las,del,anonymous,es,al,ms,se,pero,przez,ya,file,tu,forum,mi,webshell,leer,cuando,torepublic,solo,jest,sus,yo,rar,en,za,po,svar,jak,ci,postw,pgp,ojos,google,ella,juego,hay,donde,emacs,inlgg,mnen,imgops,imagenes,gonzo,ale,ni,puede,fora,tiene,il,eso,ma,nameless,ze,odpowied,cmo,aby,mas,sluzby,musimy,torowicze,dzia,tak,library,nas,wiadomosci,siempre,nov,mucho,dla,czy,signature,cosas,va,ostatni,autore,hacer,em,tags,italia

Table 13 Topic models and their words in the main batch that show strong matches with the topic models in the DFS_1000 batch

Topic 1	**link**,eur,**anarplex**,kush,mg,connect,passport,productpricequantity,irc,crypto,pure,**server**,realm,original,darknets,lsd,projects,tribes,agora,hand,mdma,send,connection,propublica,id,gr,marijuana,cannabis,stream,book,format,products,drivers,line,login,cocaine,logo,register,jpg,png,organic,single,related,net,haze,directly,current,nbome,order,purple,tribe
Topic 2	**anonymous**,**na**,ro,nie,books,si,library,przez,google,file,calibre,di,imgops,il,jest,kb,jak,sie,ci,odpowied,fiction,**te**,exif,sim,**autore**,ale,aby,dzia,postw,em,po,zobaczy,tak,**forum**,**torepublic**,klik

	nij,pomini,za,la,czy,posted,italia,xd,mi,posty,ze,data,bo,jpg,tylko,mnie,od,ma,dla,pgp,tego
Topic 3	ago,days,phishing,link,ddos,alphabay,protection,market,people,hours,centos,war,page,bdrip,money,website,world,anonymous,isis,make,server,porn,don,kheper,years,group,big,powered,society,friend,domain,government,fate,day,apple,read,today,power,father,forgive,anarchism,countries,dvdrip,anti,authorization,central
Topic 4	en,site,og,hosted,hosting,ii,freedom,freehosting,til,tails,inlgg,med,er,av,svar,du,af,det,posted,den,mnen,tor,trdar,och,om,som,din,kan,der,ikke,leave,onion,tekst,forum,gud,comments,reply,senaste,har,tagged,sedan,password,inga,pgp,sista,mnader,protonmail,ngle,hvis,netvrket,pc,cloud,privatliv,deres,griffin,captcha,inlggetre
Topic 5	die,und,parazite,tor,counted,bills,zu,notes,test,pass,der,das,ist,www,data,uv,ut,pen,paper,dolor,lorem,ipsum,wir,von,nic,pz,chip,fi,consectetur,amet,nicht,sit,im,fr,exit,es,cotton,mit,shim,adipisicing,tempor,sed,eiusmod,um,security,auf,ein,werden,relays,wird,reality,produced,features,aplikace,des,oder,auch,dem,sind,den,labore
Topic 6	btc,bitcoin,bitcoins,note,png,address,service,transaction,hours,coins,wallet,make,hidden,send,multiply,amount,flaw,day,website,time,buy,client,transactions,pay,hundredfold,money,found,tor,win,deposit,fee,digital,investment,rich,history,find,mix,blockchain,diff,mixer,today,ve,back,withdraw,future,times,quickly,privacy,anonymous,project,won,easycoin,accept,lot,information
Topic 7	usd,cards,buy,png,email,paypal,accounts,account,cc,bitcoin,card,free,jpg,order,market,info,pp,credit,service,euro,btc,usa,contact,days,escrow,full,shipping,payment,img,gif,acc,images,store,balance,eur,price,eu,verified,money,guide,quality,don,blank,gb,feedback,worldwide,page,prices,sell,mail,cloned,bitcoins,time,products,bills,apple,tutorial,counterfeits
Topic 8	de,en,por,le,la,post,ltima,mensagem,mai,el,se,les,para,es,du,ms,ver,tpicos,fr,lo,par,mensagens,una,des,pour,pm,vulpix,est,au,leer,con,comments,las,pas,los,da,al,mi,linux,ajout,mon,ministre,sun,sur,facture,ser,sarav,cni,vous,em,emacs,ao,si,mais,este,mega,carte,como,scurit,nous,vicious,pero,il,del,sin,sobre,ya,son,ou,passe,tu,uma,accueil,elbinario,posts,tor,forum,mot,rss,cmo,documents
Topic 9	server,apache,configuration,enabled,directory,default,web,string,jun,file,site,page,debian,iso,ubuntu,var,actor,main,conf,env,version,files,download,ref,http,support,document,create,type,root,read,count,reference,object,end,php,mb,pony,pdf,dir,user,www,motorcycles,cars,due,installed,means,respective,located,capability,error,box,test,virtual,documentation

Topic 10	tor,services,hidden,forum,service,post,search,bitcoin,hosting,links,onion,posts,topics,wiki,free,site,anonymous,web,clearnet,link,online,november,contact,network,board,security,information,blog,hacking,login,market,people,files,cannabis,kamagra,news,forums,password,email,page,list,users,home,sites,public,key,download,privacy,time,darknet,marketplace

Table 14 Table showing all the topic models and their words in the DFS_1000 links batch with highlighted words that show matches with the main batch models words

In contrast to the BFS_1000 batch, from the topic models of the first thousand links crawled in DFS as seen from figure 27 and tables 13 and 14, **drugs** related words like "cannabis", "kush", "cocaine", and "organic" start to make their appearance as the first few words, and at least four topic models, 2,4,8,5, show **other languages**. This was clearly not the case with BFS_1000. This can also be confirmed by the table of LDAvis, table 4-1. The words related to drugs were only appearing in the topic models of two thousand links of the BFS batch but in DFS_1000 they are already seen. The drug-related pages and other languages already show how many varieties of topics DFS begin to pick up, as compared to BFS, in the first thousand links alone.

DFS_2000

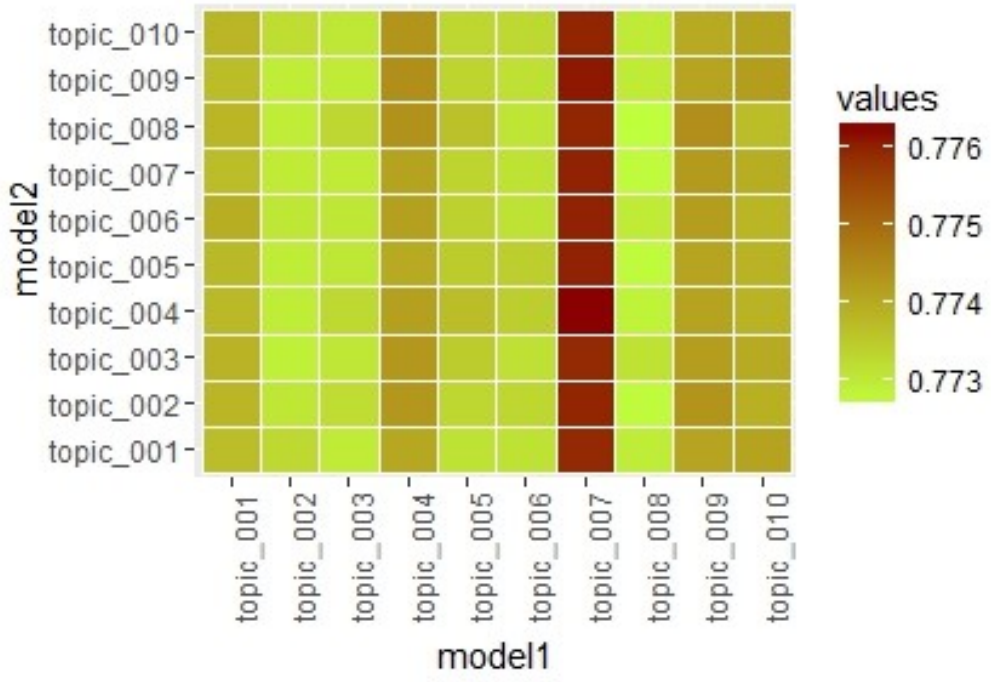

Figure 28 Heat map visualization comparing the main batch of 3,539 links with the first thousand links crawled in DFS order

Topic 7	days,ago,usd,buy,account,email,accounts,paypal,login,cc,pp,info,gif,payment,acc,full,seed,hours,satoshi,blank,balance,euro,register,usa,verified,password,guide,thumbnail,click,eu,choose,minutes,socks,cash,step,sell,wager,worth,don,png,service,username,cards,contact,free,cashout,feedback,tutorial,country,credit,page,bills,admin,odd,market,leave,money,premier,javascript,rules,tr,valid,support,send,result,bdrip,card,replace,jpg
Topic 9	card,number,credit,jpg,cards,order,shipping,november,eur,birth,product,images,price,information,products,contact,account,add,day,cart,type,address,home,gift,security,propublica,atm,tuesday,pin,logo,month,quality,bank,amazon,city,country,free,heroin,high,code,year,pgp,grams,date,cannabis,usd,worldwide,zip,state,father,money,mother,middle,visa,dragon,social,phone,cvv,send,shop,monday,vendor,exp,mm,billing,ship,png,spouse,routing,wednesday,guaranteed
Topic 10	de,la,en,el,del,se,una,es,lo,los,para,con,si,al,su,las,por,te,ms,di,comments,pero,ya,este,mi,displayed,rss,jpg,ddos,leer,solo,cuando,il,sus,cars,todo,esta,motorcycles,authorization,imagenes,ojos,esto,challenge,ella,tags,juego,sin,son,hay,posted,donde,passion,emacs,puede,sex,ni,tiene,eso,chapter,vez,va,cmo,elbinario,muy,pt,google,cock,web,view,printer,protection,era,blog,che,writes,boy,desde,siempre,porque,mucho,english,cualquier,cosas,autore,hacer,momento,irc,despues
Topic 4	png,img,bitcoin,voteup,tor,market,service,hidden,services,anonymous,hosting,free,forum,buy,links,wiki,link,onion,store,clearnet,cards,paypal,web,marketplace,search,drugs,wallet,kamagra,escrow,bitcoins,site,accounts,quality,online,high,forums,board,darknet,cannabis,email,usa,deep,selling,counterfeits,hacking,uk,alphabay,phishing,gb,ddos,mail,engine,russian,passports,fake,credit,good,apple,id,usd,mirror,kowloon,blog,sell

Table 15 Topic models and their words in the main batch that show strong matches with topic models in the DFS_2000 batch

Topic 1	de,en,la,hosting,el,le,site,ii,freedom,se,hosted,freehosting,una,les,services,del,para,lo,es,los,kowloon,des,con,si,est,su,du,las,por,tu,pour,te,al,vous,sur,pas,par,au,como,pero,il,contact,une

	,mas,jpg,virtual,cuando,qui,ya,ms,mai,ce,solo,nous,sus,ou,dans,este,di,mi,imagenes,son,ser,ojos
Topic 2	btc,bitcoins,bitcoin,png,img,voteup,transaction,multiply,hours,flaw,make,hundredfold,address,day,client,website,amount,wallet,money,found,send,investment,digital,win,pay,coins,time,history,error,rich,transactions,deposit,today,find,buy,lot,future,ve,project,won,times,returned,hope,innovative
Topic 3	link,november,mb,anarplex,nsa,snowden,html,irc,onion,darknet,pdf,june,post,mr,crypto,united,server,torture,htm,net,realm,related,cars,motorcycles,files,darknets,essay,connect,agora,americans,president,intelligence,december,unix,services,american,propublica,tribes,projects,national,hand,
Topic 4	tor,bitcoin,usd,service,forum,hidden,free,market,buy,anonymous,email,onion,services,post,links,search,topics,posts,paypal,wiki,clearnet,account,hosting,web,store,site,cards,login,eur,png,accounts,board,quality,cannabis,online,escrow,link,info,cc,usa,marketplace,page,images,drugs,register,forums,image,information
Topic 5	days,nie,na,anonymous,ro,przez,sim,sie,si,postw,jest,forum,google,jak,ci,torepublic,file,imgops,trdar,kb,te,di,library,po,odpowied,za,dzia,calibre,ore,ga,books,ale,exif,il,aby,autore,aac,pgp,ostatni,chan,fora,em,ze,data,tak,baby,kliknij,zobaczy,detective,tematw,senaste,post,pomini,
Topic 6	gif,books,kamagra,hidden,service,mg,tor,und,der,directory,jun,die,fr,ist,von,das,jelly,apcalis,treatment,oral,mit,auf,zu,den,blood,erection,nicht,tablets,ein,dragon,werden,links,blank,de,auch,logo,tab,zum,tk,super,ingredient,ubuntu,fiction,dem,polo,des,sind,ber,onion,dir,kjabber,web,citrate
Topic 7	server,apache,configuration,file,enabled,default,web,page,debian,site,files,ubuntu,var,conf,keyringer,http,key,document,user,version,root,keyring,php,rss,secret,public,installed,password,www,html,systems,respective,located,installation,support,nameless,documentation,access,tags,rar,virtual
Topic 8	ago,days,note,tor,time,people,contact,read,don,address,site,data,privacy,service,make,email,public,internet,news,free,end,website,information,security,content,access,server,world,home,network,work,services,mail,comments,back,page,find,hours,download,support,browser,love,string,create,code
Topic 9	cards,phishing,card,link,ddos,protection,alphabay,market,credit,euro,hq,shipping,price,mdma,number,qiwi,gift,lsd,counted,contact,money,im,order,visa,paypal,amazon,guaranteed,balanc

	e,atm,pin,day,bank,make,hacking,faq,mastercard,accounts,buy,regular,birth,white,people,services,bitcoin,jpg
Topic 10	mensagem,ltima,por,mai,tpicos,mensagens,ver,tuesday,en,og,til,tails,webshell,larry,med,inlgg,bdrip,monday,af,det,svar,boy,er,vulpix,du,posted,sets,boys,den,av,da,mnen,tor,php,och,em,om,som,mwm,kan,din,os,hacking,abr,dvdrip,avc,query,ikke,mr,nov,dvd,tekst,forum,ao,gud,chat,comments,programao,leave,pc,tagged,man

Table 16 Table showing all the topic models and their words in the DFS_2000 links batch to show what matches with the main batch topic models words

In comparison to the two thousand links crawled with BFS, it was observed that DFS already starts to show dispersion in terms of not giving clear distinct topic models, when tables 15 and 16 and figure 28 are analyzed. What was happening in the BFS_3000 batch, we are already beginning to see in the DFS_2000. As links increase for the DFS order this is what is expected to be seen, with strong combinations starting to decrease.

DFS_3000

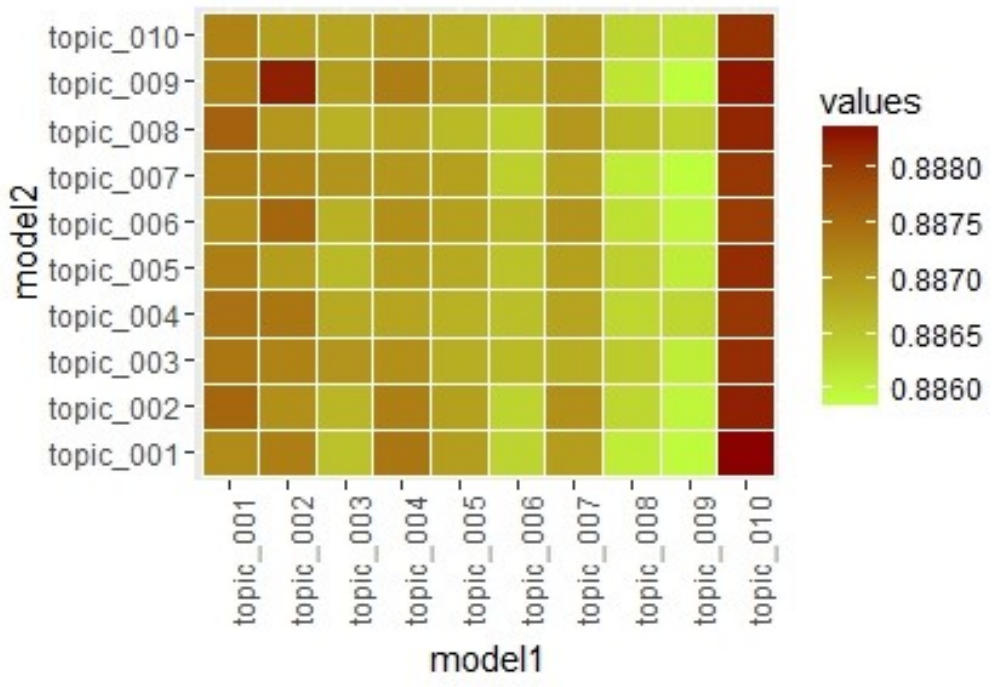

Figure 29 Heat map visualization comparing the main batch of 3,539 links with the first three thousand links crawled in DFS order

Topic model	Main batch	DFS batch crawled upto 3000 links
1	na,password,login,nie,log,anonymous,ro,register,access,file,user,med,username,google,site,rar,forum,captcha,shortlog,svar,logged,kb,onion,stream,postw,ci,replyquoted,inlgg,close,fora,click,imgops,mnen,em,te,dvd,code,po,citadel,enter,pgp,odpowied,ale,jak,aby,tak,sign,nas,month,exif,ch,jpg	ago,days,note,number,card,people,read,november,time,credit,information,key,security,address,public,don,type,world,day,month,make,birth,state,nsa,code,tuesday,year,snowden,account,order,privacy,years,end,war,date,country,government,jpg
2	tor,server,services,web,contact,support,site,public,service,page,download,hosting,key,privacy,internet,apache,read,browser,data,enabled,hidden,file,security,address,access,onion,november,network,free,version,time,configuration,information,files,kowloon,content,home,news,don,default,website,	email,account,phishing,ddos,cc,protection,pp,accounts,paypal,full,balance,result,alphabay,acc,payment,initial,array,blank,link,info,don,coins,market,service,money,guide,buy,click,address,send,verified,contact,page,amount,time
3	ago,days,foxxxydox,line,forum,undefined,notice,sie,offset,pgp,bdrip,torepublic,przez,begin,signature,av,jest,trdar,za,signed,hash,message,sluzby,torowicze,musimy,end,avc,inlgg,wiadomosci,ore,xd,dvdrip,ze,senaste,sha,query,sa,aac,rock,serwis,jezeli,tora,ich,detective,dla,dzia,jeden,mad	gif,books,und,der,tor,die,seed,von,ist,nie,den,das,na,fr,mit,server,zu,du,auf,hash,ro,sie,anonymous,er,nicht,auch,ber,ein,zum,werden,tails,forum,de,dem,es,counted,oder,onion,przez,pgp,des,jungs,logo,eine,sind,sich,wir,hier,kjabber
4	png,bitcoin,tor,img,service,voteup,market,hidden,site,usd,free,hosting,buy,cards,forum,anonymous,card,email,li	de,en,le,la,les,des,du,est,pour,vous,sur,ver,par,il,une,pas,au,ce,fr,ou,mais,mai,qui,nous,dans,je,ne,avec,di,si,ma,se,comment,site,passe,ministre,votre,avril,

	nks,search,services,paypal,onion,credit,account,wiki,freedom,clearnet,accounts,web,topics,hosted,store,wallet,bitcoins,link,board,post,ii,information,posts,usa,contact,marketplace,	mes,mot,accueil,tout,scurit,bien,aux,sont,doit,lire,nos,janvier,jpg
5	btc,bitcoins,bitcoin,transaction,multiply,flaw,hours,hundredfold,client,make,day,website,amount,address,investment,found,digital,win,history,money,time,send,today,rich,deposit,times,find,won,ve,future,project,returned,lot,pay,hope,innovative,transactions,blockchain,min,chance,error,trust,wallet	post,posts,topics,forum,ltima,mensagem,pm,login,hidden,error,service,view,latest,por,register,wed,board,connect,mai,mensagens,password,socks,tpicos,reply,utc,jun,host,failed,total,discussion,links,unreachable,overchan,users
6	gif,books,und,der,die,tor,von,ist,jun,directory,den,das,fr,zu,mit,auf,du,nicht,er,auch,ein,zum,ber,thumbnail,werden,de,dem,onion,datei,es,oder,eine,des,fuer,jungs,hier,bild,sind,vollgroesse,angezeigt,anklicken,sich,calibre,wir,kjabber,diese,wtf,im,af,logo,xmpp,wird,fiction,sie,server	png,img,voteup,bitcoin,market,usd,buy,cards,paypal,free,shipping,wallet,service,bitcoins,anonymous,store,search,quality,escrow,order,credit,accounts,price,sell,contact,cannabis,vendor,hidden,darknet,high,products,drugs,euro,marketplace
7	mg,kamagra,eur,mdma,lsd,gbp,gram,hash,qiwi,result,hq,pure,seed,kush,pills,nbome,cannabis,cocaine,shop,jpg,productpricequantity,order,online,cart,initial,products,active,og,white,super,quality,treatment,jelly,games,oral,play,heroin,tuesday,buy,apcalis,casino,high,time,ingredient,winnings,	btc,bitcoins,bitcoin,transaction,multiply,flaw,hours,client,hundredfold,make,day,website,address,investment,found,amount,digital,win,history,today,rich,time,send,money,future,find,won,ve,returned,deposit,project,hope,times,lot,transactions
8	de,la,le,en,les,des,du,pour,pcs,est,vous,sur,une,pas,par,i	de,en,la,el,se,para,una,es,lo,por,los,con,su,las,te,del,al,tu,si,ms,pero,como,po

	l,au,ce,si,fr,nous,di,mai,dans,qui,ou,avec,ne,je,site,se,passe,comment,br,votre,lire,ma,sont,mot,ferme,tout,mes,accueil,bien,aux,avril,suite,tre,val,doit,del,mais,janvier,ministre,faire,library,nos,ni,ici,mon,juillet,flag,ai,	sted,em,comments,til,mas,rss,este,ya,mi,med,ser,jpg,br,leer,cuando,inlgg,os,na,sus,solo,da,todo,uma,rar,svar,imagenes,ver,ojos,esto,ella,dos,esta,juego,yo,hay,ni,av,emacs,mnen
9	de,en,la,por,el,ltima,mensagem,para,se,es,una,lo,mai,ver,los,mensagens,post,tpicos,con,em,overchan,las,su,al,da,del,te,tu,si,como,ms,pero,os,mas,ser,este,til,comments,posts,um,uma,view,ya,mais,leer,ao,jpg,cuando,todo,sus,solo,esta,dos,sobre,mi,irc,vulpix,yo,le,imagenes	tor,site,hosting,services,service,onion,hidden,web,server,free,forum,freedom,links,link,hosted,bitcoin,page,wiki,anonymous,clearnet,file,ii,network,contact,files,support,freehosting,blog,search,sites,board,mail,list,download,directory
10	link,note,people,phishing,protection,ddos,jpg,propublica,make,war,money,post,read,life,anarplex,time,world,years,reply,big,html,don,posts,porn,real,today,things,mm,hours,series,stories,story,torture,april,power,good,day,im,chapter,pdf,problem,march,sex,posted,irc,latest,counted,american,book,americans,nginx,cars,media,girl,cryptostorm,galaxy,cops,february,trump,boy,country,slavery,group,hacking,pm,europe,made,love,january,mass,york,water,iraq,thing,year,death,politics,updated,job,long,game,published,science,society,hacker,motorcycles,government,crypto,test,lot,december,agent,america,entire,nameless,english,hand,donate,unix,pennsylvania,politician	mg,kamagra,eur,mdma,jpg,gbp,lsd,qiwi,pcs,hq,kush,productpricequantity,nbome,og,pills,pure,jelly,active,treatment,gb,super,oral,apcalis,tablets,white,blood,shop,erection,haze,ingredient,del,mm,effective,hash,jabber,polo

Table 17 Topic models and their words in the main batch and those of the DFS_3000 batch

When figure 29 and table 17 are analyzed, in the DFS_3000, a match between topic 10 and all the topic models of the DFS_3000 batch is noticed, with words making reference to **politics**, which was not even a topic found in BFS_3000. BFS showed no one strong topic model from the main batch that matched with all other topic models of the other batch. One explanation for this is that topic model number 10 generated in this LDA run happened to contain a mixture of topics, so one can clearly say it is only politics-related, and this topic model matched with all the other topic models of DFS_3000 because of how topics in general are more diverse, compared to the topic models generated by BFS_3000 links.

Other phenomenon is the number of strong matches in a random manner between both these batches; these can be clearly seen from the heat map and the reasons for it are illustrated in the table. Therefore, dispersion is seen in terms of strong matches.

DFS_3539

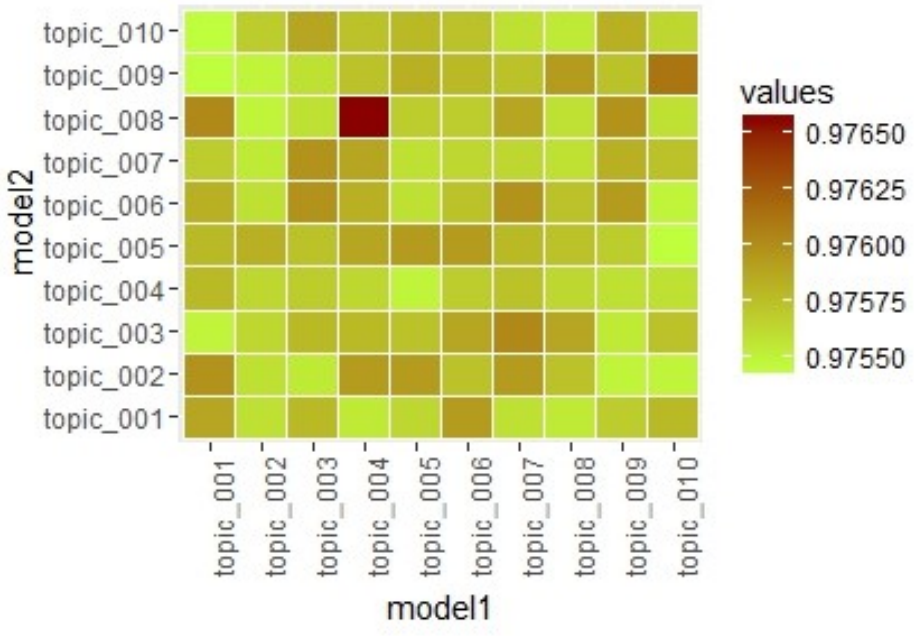

Figure 30 Heat map visualization comparing the main batch of 3,539 links with the entire network crawled in DFS order

Topic model number	Main batch	DFS crawled to the end of the graph
1	site,hosting,freedom,hosted,contact,address,ii,freehosting,key,service,public,privacy,tor,don,home,free,time,password,account,access,login,security,people,information,search,server,data,pgp,internet	days,na,nie,foxxxydox,ro,anonymous,forum,przez,htm,pgp,bdrip,torepublic,jest,za,svar,po,jak,postw,inlgg,hansa,chan,avc,mnen,imgops,ale,fora,begin,kb,ze,signature,smoketimes,odpowied,fate,aby,dvdrip
2	btc,bitcoins,bitcoin,transaction,multiply,flaw,hundredfold,hours,make,client,day,website,investment,found,amount,digital,address,win,history,money,today,rich,time,deposit,project,future,ve,find,send	btc,bitcoins,bitcoin,multiply,flaw,transaction,hundredfold,client,hours,make,day,website,investment,found,digital,amount,address,win,history,today,rich,time,money,ve,future,find,won,project,times
3	usd,eur,mg,mdma,lsd,productpricequantity,gram,logo,qiwi,cannabis,hq,pure,kush,shop,nbome,quality,pcs,products,gbp,pills,jun,og,white,jpg,login,cart,cocaine,hash,original,register,euro,faqs,haze,	del,di,til,posted,il,af,nameless,rss,library,fiction,calibre,pc,ad,rub,query,dog,che,kan,kidsleepy,ikke,autore,ut,ed,italia,ea,tekst,ma,mad,mrkim,adland,da,protonmail,mar,una,ngle,ww,detour,comments
4	tor,png,img,bitcoin,voteup,hidden,service,market,web,anonymous,onion,links,services,link,free,forum,wiki,site,clearnet,search,hosting,store,file,marketplace,paypal,cards,wallet,buy,server,files	mdma,lsd,mg,gram,qiwi,cocaine,hq,pure,kush,nbome,white,pills,productpricequantity,og,tuesday,cannabis,shop,heroin,quality,gr,jpg,original,haze,line,high,crystal,speed,drug,jabber,notice,mm,faq,monday
5	post,posts,topics,kamagra,services,forum,pm,hosting,kowloon,view,latest,mg,active,board,link,contact,total,wed,login,users,oral,overchan,jelly,discussion,treatment,register,today,home,gbp,reply,apcalis	ago,post,posts,topics,people,read,pm,time,support,latest,november,view,news,privacy,security,don,world,server,data,make,public,result,internet,users,comments,information,array,today,forum,content,initial,version

6	people,support,read,enabled,string,love,tuesday,version,propublica,comments,war,html,time,comment,mass,april,iso,line,actor,env,notice,php,stories,series,nginx,years,story,apache,rss,library,torture	png,img,voteup,bitcoin,buy,usd,cards,card,market,credit,email,account,paypal,accounts,address,service,contact,free,order,shipping,bitcoins,wallet,number,eur,search,price,pgp,cc,note,anonymous,send,usa,information
7	card,buy,cards,credit,email,number,account,order,note,jpg,cc,shipping,paypal,accounts,png,price,usa,payment,balance,pp,contact,info,birth,full,november,information,address,eu,images,acc,country	de,gif,le,la,books,und,du,des,les,en,der,eur,die,tor,von,ist,pour,pcs,est,fr,vous,den,das,sur,une,zu,sie,pas,mit,par,auf,au,ce,nicht,nous,er,auch,mai,ein,ber,dans,zum,qui,werden,logo,thumbnail,es,ou,dem,datei
8	nie,na,foxxxydox,ro,anonymous,sie,forum,przez,inlgg,av,med,torepublic,jest,nameless,za,svar,det,pgp,po,jak,postw,te,och,trdar,kb,mnen,ci,imgops,ale,som,fora,om,ze,odpowied,aby,musimy	tor,hidden,service,web,bitcoin,onion,link,forum,site,services,links,wiki,free,server,anonymous,page,market,clearnet,hosting,file,files,online,search,network,sites,board,store,blog,directory,download
9	ago,days,gif,error,connect,array,initial,server,result,seed,host,failed,socks,unreachable,utc,blank,hash,access,final,occurred,hours,client,simplified,wager,localhost,generated,polipo,admin,odd,chapter	de,en,la,por,el,ltima,mensagem,para,se,es,una,lo,mai,ver,los,mensagens,tpicos,em,si,te,con,jun,las,su,tu,al,da,como,del,ms,os,pero,fr,mas,ser,este,jpg,comments,um,uma,mais,na,mi,ya,med,leer,ao,cuando,solo,todo
10	de,en,la,le,por,books,el,se,ltima,mensagem,und,para,du,des,les,es,mai,der,fr,die,una,tor,lo,est,si,von,ist,das,pour,em,ver,mensagens,los,tpicos,vous,den,con,sur,del,pas,su,las,da,une,par,tu,zu,al	hosting,site,ii,freedom,hosted,freehosting,kamagra,mg,services,kowloon,gbp,active,jelly,oral,treatment,apcalis,contact,ingredient,virtual,blood,erection,buy,tablets,super,dragon,citrate,pill,viagra,home,effective,trial,erectile,time,sildenafil,dysfunction

Table 18 Topic models and their words in the main batch and those of all links crawled until the end of the graph in DFS order

What was noticed in the BFS_3539 batch is a similar phenomenon to what was observed in the DFS_3539 batch in figure 30. This batch couple comparison is not a good example to show the differences in terms of BFS and DFS, as similarity can be seen in the way they show matches. To understand the differences in the representativeness of topic models one must look in the range of 1,500-3000 links crawled, from which one can understand the different paths taken by BFS and DFS.

To conclude, the following observations were made:

- For BFS at 1000, topics related to bitcoin, payments, hosting, wiki were seen. These topics indicate that BFS is still close to the network core even at 1000 links away from starting point

-Drugs related pages show when 2000 links have been crawled in BFS manner whereas they show in first 1000 links for DFS already

- Italian pages start to show when 3000 links have been crawled BFS manner whereas they show already in first 1000 links crawled in DFS manner

-Other languages were detected by DFS already at 1000 links but not at all by BFS

-Politics related pages started to emerge at 3000 links crawled DFS manner but were not found by BFS

Section 4.2 covered how strategy one was adopted where only one starting point was used. This section therefore answers questions 1 and 2. The following sections explore questions 3, 4 and 5 and shows how the second strategy was used.

4.3 Effect of crawling from different starting points

This section answers the third question: How does the starting point of crawl dictate the view of the network?

It shows the difference in the view of the network over a certain extent of crawl size with a fixed algorithm. In this section, the crawl size was fixed to 100 links and BFS algorithms. The chosen crawl size was 100 links as this sufficiently presents how very easily a different starting point can give a completely different understanding of the network. Additionally, crawling over the next 100 links can make one conclude that the topic is mainly about a certain topic or another,

whereas crawling from a different starting point can give a completely contrasting view of the network. To demonstrate this, crawling with BFS algorithm over the next 100 links from four different starting points resulted in a totally different view of the network. Table 19 shows the words that could have been found in the crawl.

The plots have been scripted on R, and ggplot has been used where colors represent word distribution as per their count and have been assigned randomly based on the weight of each component. The **x axis** represents the five topics defined. The **y axis** represents the topic word count found in the 100 links that were crawled for any given starting point. The height represents the sum of all the words present in that topic. The larger the height, the greater the number of topic-words for the 100-link traversal.

Topics	Words
Topic 1	Onion, http, ago,day,upload,content,index,tor,link,home
Topic 2	Transact, card, make, multipli,day, time, hour, address, flaw, invest
Topic 3	Imag, post, data, theme, topic, onlin, book, support, onion, read
Topic 4	Service, tor, onion, site, hidden, link, forum, account, server, free
Topic 5	Thumb, default,thumbnail, host, market, buy, site, service, freedom,card

Table 19 Topics and their words for easier reference

1) **Starting point 1**: gotchafjkmcqdz2x.onion captioned "A list of LEAKED ONION Websites"

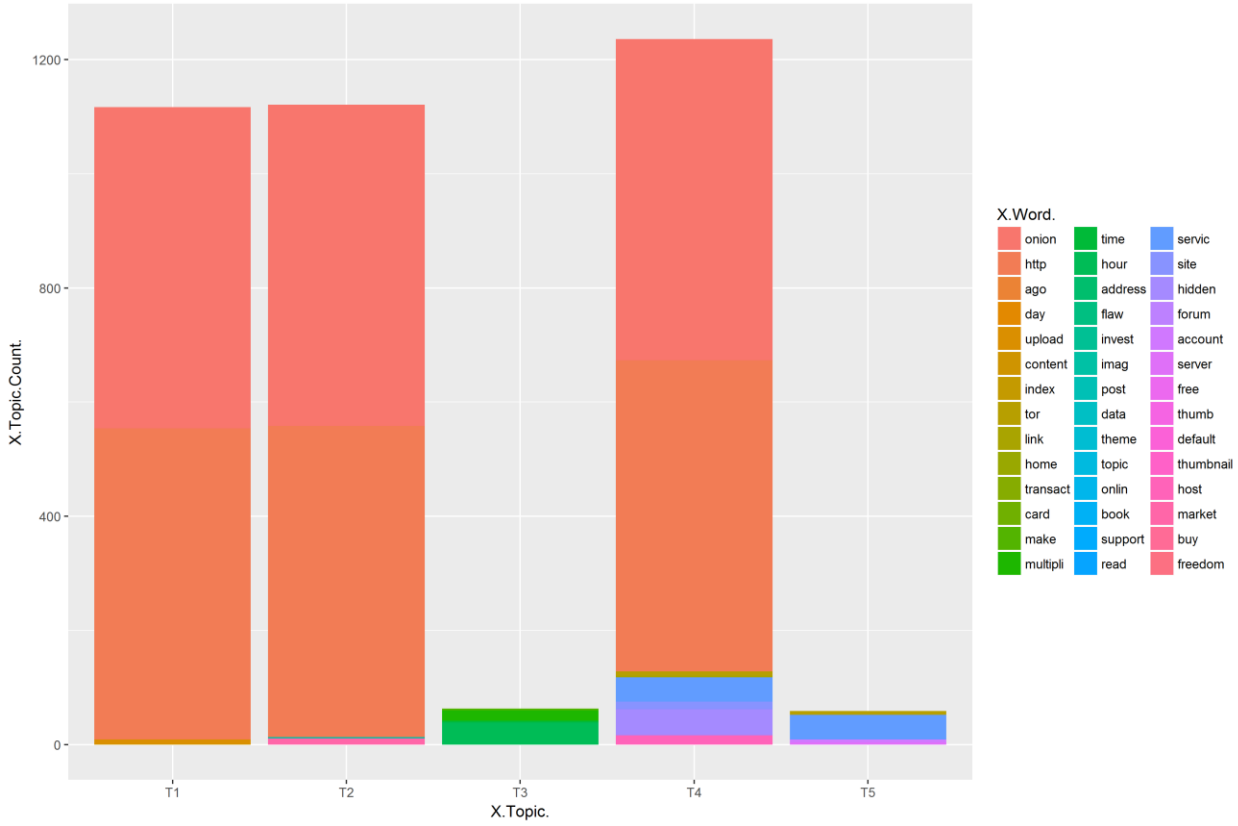

Figure 31 View of the network from gotchafjkmcqdz2x.onion

From figure 31, the heights of the topics prominent in this plot for this starting point—topic 1, topic 2, and topic 4—have the most occurring words within the 100 links crawled. Topics 3 and 5 barely have words present in these 100 links. In section 4.2, the same URL was used to understand crawling algorithms' behavior and the extent of crawl. This plot gives details of topics that can be seen over 100 links. From topics 1 and 2, words such as "http" and "onion" seem to have made their appearance among the webpages traversed from this node. Therefore, if one starts the crawl from this starting point alone, the view of the network one will have is mostly about topics 1, 2, and 4.

2) **Starting point 2**: mxzzaiahatoiyxhb.onion captioned "OnionDir - Deep Web Link Directory"

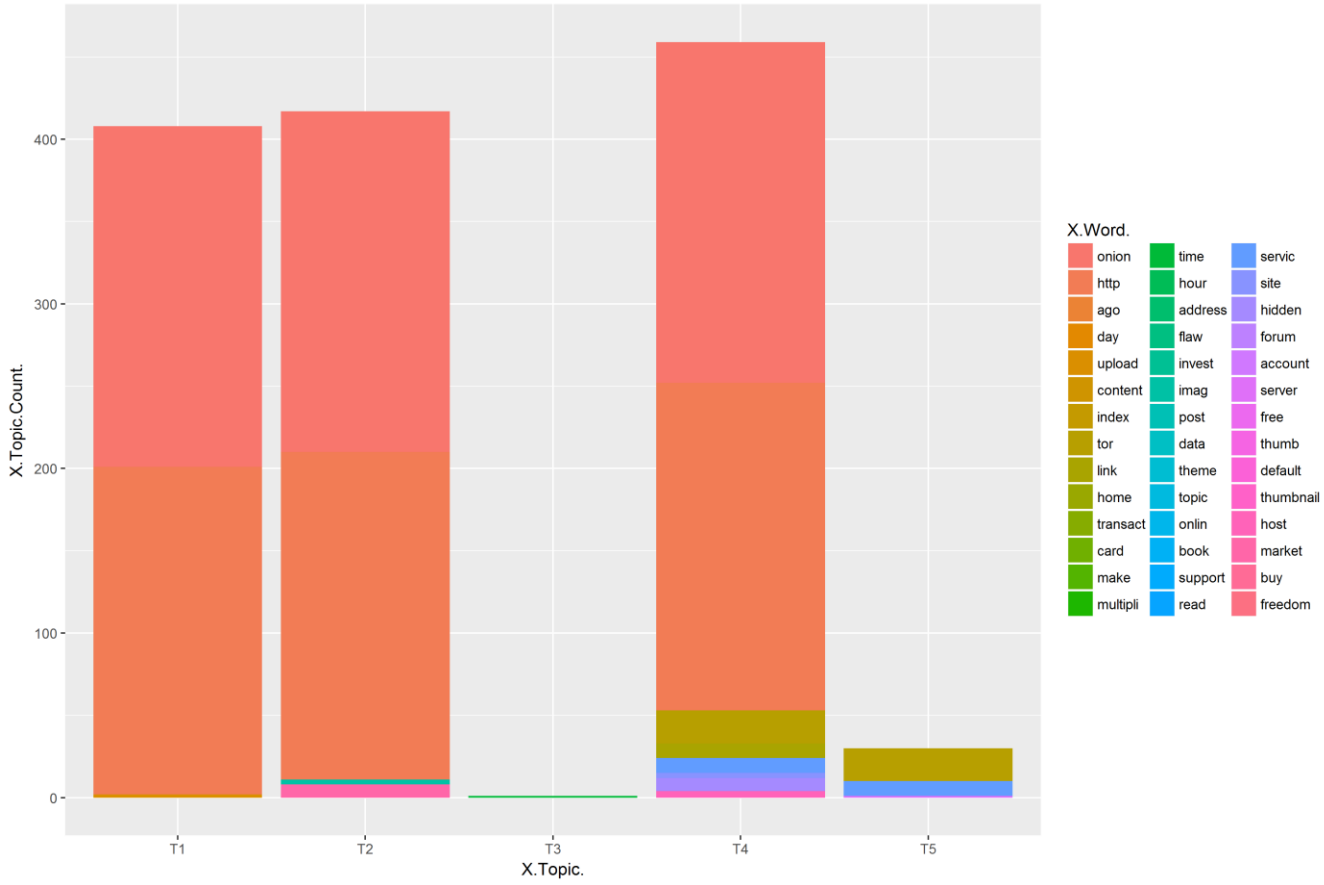

Figure 32 View of the network from mxzzaiahatoiyxhb.onion

The starting point named mxzzaiahatoiyxhb.onion in figure 32 show a similar phenomenon as gotchafjkmcqdz2x.onion did with words such as "onion" and "http". This can be explained from the similarities of the kind of web services they are, and because they crawl in BFS manner for first 100 links, they both seem to remain connected to similar kinds of webpages. The slight occurrences of other words from topic 4, such as "link" and "tor", show the different kind of webpages this starting point encountered compared to the first starting point. Another conclusion drawn from these similarities observed is that when one traverses from highly connected starting points, the view of the network will be more or less greater in terms of certain topics and therefore will give an impression that the Darknet is clearly made up of these topics alone. In the process, the view that can be observed by extending the crawl and using another algorithm is eliminated.

2) **Starting point 3**: dmzwvie2gmtwszof.onion, a Torrents related page

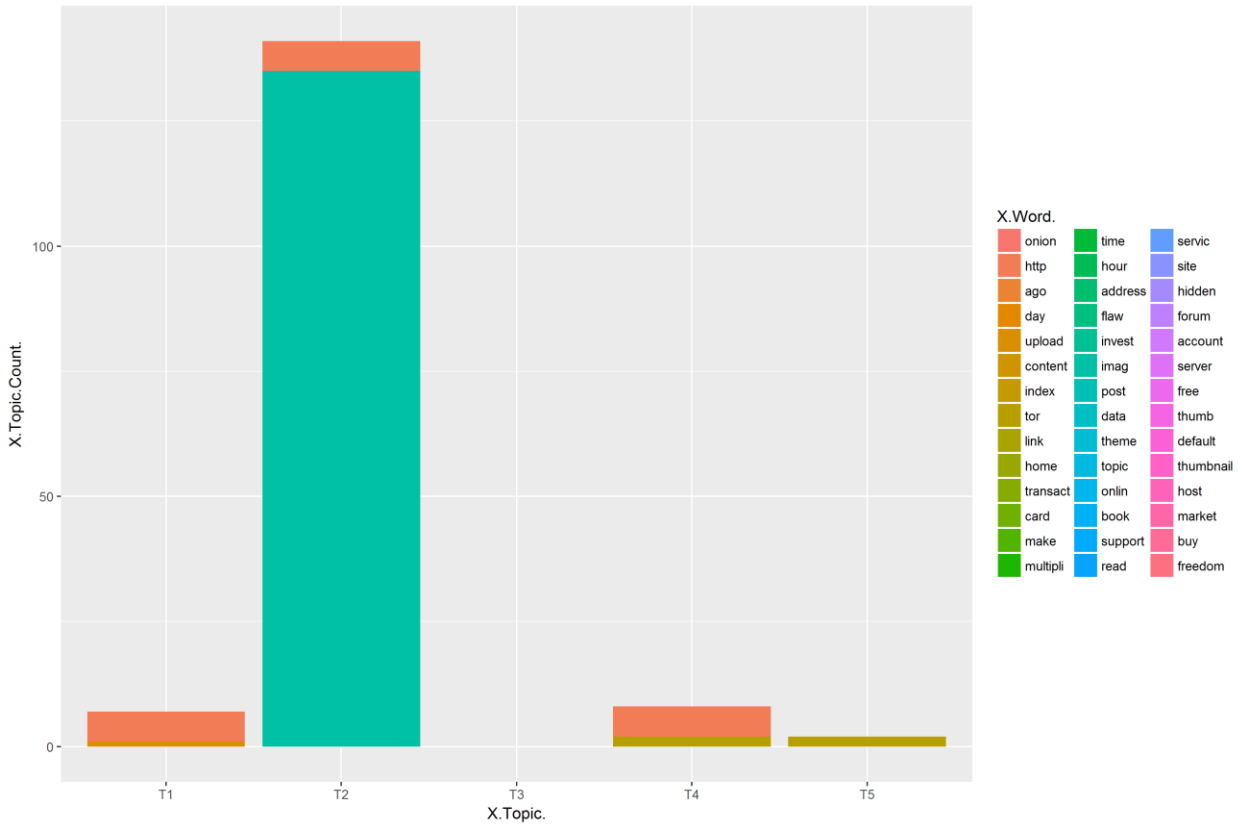

Figure 33 View of the network dmzwvie2gmtwszof.onion

This starting point in figure 33 about torrents has traversed in a completely different manner than the first two. Since the code used to create this plot did not note down words that do match and their resulting probabilities, the words appearing from topic 2 as evident from the color chart may have been either “invest” or “flaw”. The main conclusion, however, that can be drawn is that this starting point shows that for this crawl size, our view of the network is solely about topic 2.

4) **Starting point 4**: bitcoinrmnuijyli.onion captioned “100x Your Coins in 24 Hours”

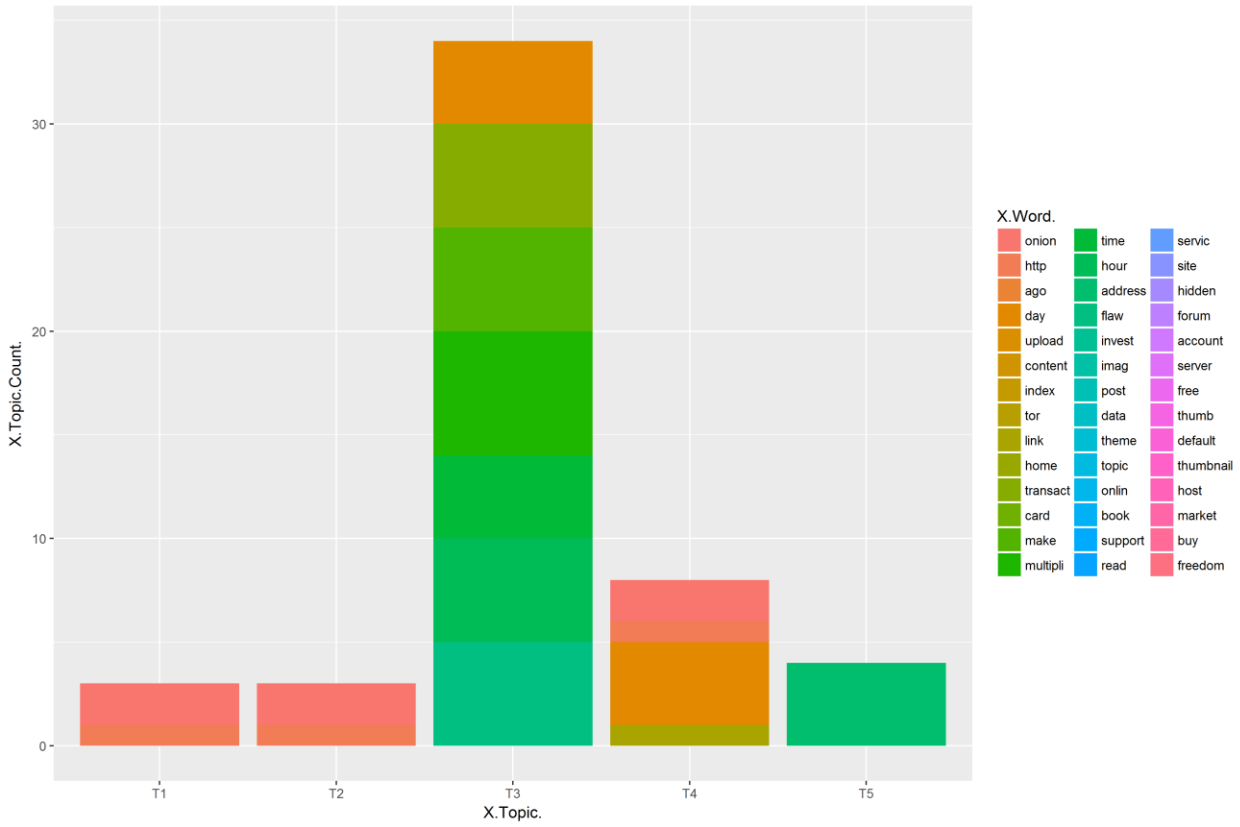

Figure 34 View of the network from bitcoinrmnuijyli.onion

From this starting point, bitcoinrmnuijyli.onion in figure 34, which mainly focuses on bitcoins, seems to have traversed webpages that show topic 3 words the most. From the words that make up topic 3, it indicates that the page traversed would also show the occurrence of this particular topic.

4.4 Metric to analyze the representativeness of the network

This section answers the fourth question. Since this thesis combines crawling and content analysis with the help of LDA, a simple metric that can be used to understand how close or far the combination of both is to representativeness is to gather the topic percentages from each of the webpages that make up the network. These topic percentages per webpage can be compared with the topic percentages of the entire network, which, in our case, is the ground truth. Every topic is made up of words with certain probabilities. Therefore, in this thesis, five topics were

chosen out of the entire network with 10 words each with their own probabilities. This is show in table 20.

Topic 1		Topic 2		Topic 3		Topic 4		Topic 5	
Onion	0.129	Transact	0.026	Imag	0.058	Service	0.017	Thumb	0.045
http	0.118	Card	0.024	Post	0.023	Tor	0.016	Default	0.043
Ago	0.022	Make	0.023	Data	0.009	Onion	0.013	Thumbnail	0.04
Day	0.02	Multipli	0.021	Theme	0.008	Site	0.01	Host	0.038
Upload	0.014	Day	0.021	Topic	0.008	Hidden	0.009	Market	0.022
Content	0.012	Time	0.018	Onlin	0.005	Link	0.008	Buy	0.021
Index	0.008	Hour	0.018	Book	0.004	Forum	0.007	Site	0.02
Tor	0.007	Address	0.017	Support	0.004	Account	0.007	Service	0.018
Link	0.007	Flaw	0.017	Onion	0.004	Server	0.007	Freedom	0.013
Home	0.006	Invest	0.016	Read	0.004	Free	0.007	Card	0.011

Table 20 Topics defined by their words and probabilities defined for metric based analysis

The reason for such a small value of K made the simulations much easier, and this will become clearer from the plots under section 4.4. This was a difficult tradeoff to reduce the computation and to obtain faster results as such a small value ended up giving very generic words under each topic. Although this was the case, the main message this thesis aims to give is still kept intact. Since the comparison is based on words having their own probabilities, there is a fixed way to measure representativeness.

The method that was utilized to make use of this metric was to crawl from a given starting point up to a defined crawl size and then note down the total presence of the word probabilities from the extent crawled while comparing this with word probabilities of the entire network. This means that some pages have the presence of some words of all topics or many words of some topics. This also means if the webpages do not have the particular topic or word, then they will not be counted, hence having the value of zero. Only the topmost important words have been chosen for each of the topics.

4.5 Getting to representative view using the metric

This section answers the fifth question: How do we get to the representative view of the network?

To obtain the representative view of the network, convergence of the metric chosen for this study was analyzed. In network graph theory, graph metrics can be used to analyze the convergence but since this study combines the topological analysis with topic models obtained, using graph metrics is not ideal, therefore topics distribution found in every web page was a simpler metric chosen which is readily available, in terms of LDA. This metric is therefore used as a function of the number of the network crawled. The point at which the convergence of the crawling algorithms with the metric as a function, will be the portion of the network that will have to be crawled before conclusive inferences about the communities can be made.

The strategy was as follows. First, over the whole network urls called nodes were treated with LDA and 5 topics with 10 words each were extracted as shown in table 20. Next, crawling was done from 50 random starting points to cover the network in all the 3 crawling algorithms over increasing crawl size. The topic-word's presence of each of the web pages was noted. Table 21 shows the pseucode of the algorithm used:

```
For step size of 300 links
{every starting point
   { for each algorithm
      { for each node traversed in the crawled network
          { presence of topic-word  over total words in that node was calculated}
      }
   }
}
```

Table 21 Algorithm used for average topic distribution while traversing

The summation of all the topic-words presence in every node traversed gave the idea of how each of the crawling algorithms actually behave. Since each of the algorithms follows its own

pattern the percentage of topic-word per node traversed is a good indicator in differentiating between them. The formula used to do this summation is as follows.

Average topic presence for different algorithms for different step size

$$\sum_{i=1}^{m}\sum_{j=1}^{x}\frac{topic\ word\ j\ count\ in\ node\ i}{total\ word\ count\ in\ node\ i}$$

Where m represents number of nodes traversed

x represents words in all topics

By using this formula the plot generated is as follows in figure 35.

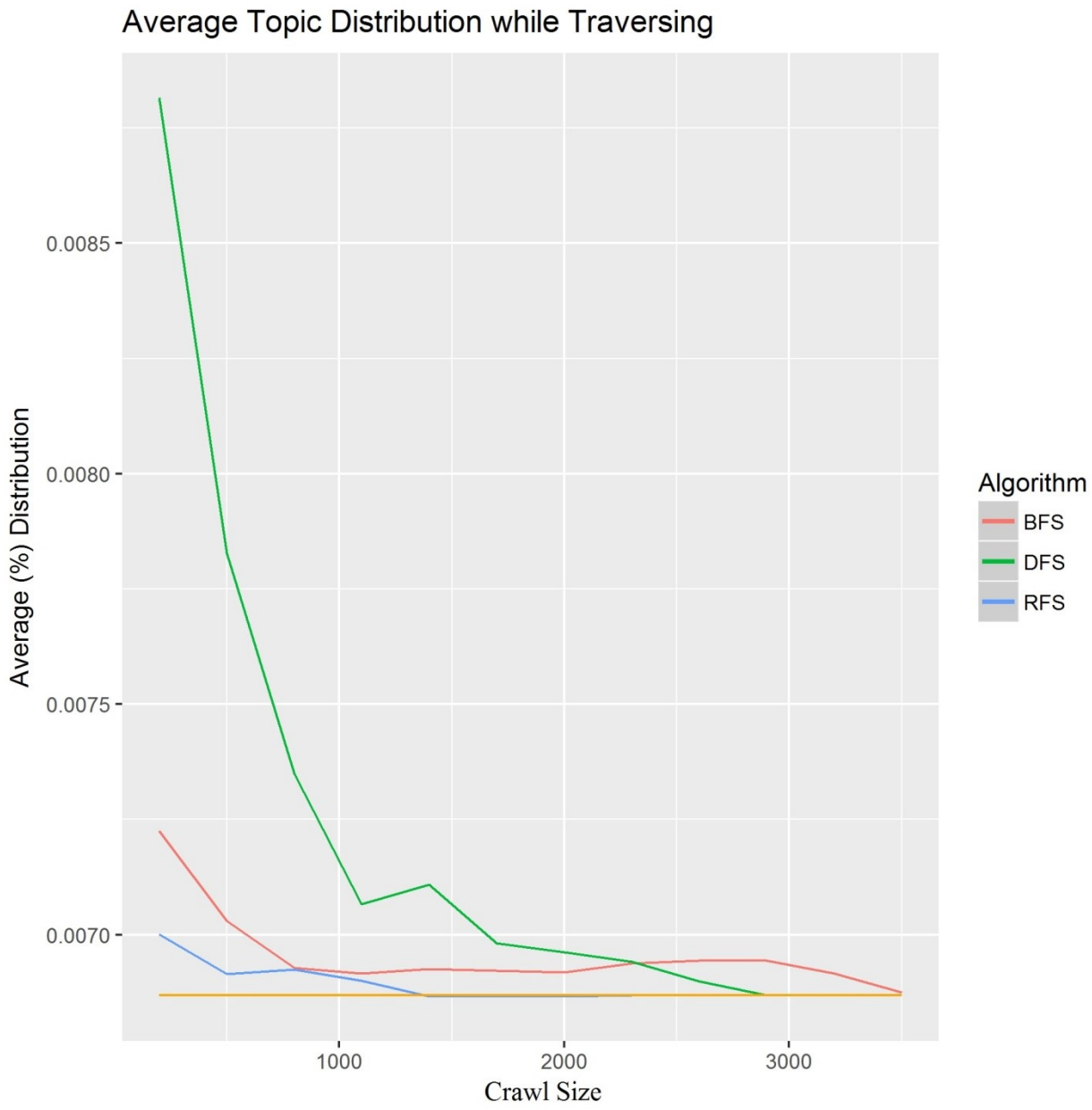

Figure 35 Average topic distribution while traversing

The orange line is the network's average topic presence. It serves as the actual metric value in the plot is the average topic presence without considering any step size or crawling algorithm where all the nodes were simply considered and the percentages of each topic-word over total words in that node were calculated and summed. The orange line is used to compare the behavior of the crawling algorithms. The value for this network turned out to be 0.00675. The formula used to calculate this is as follows:

Average topic presence (orange line)

$$\sum_{i=1}^{n}\sum_{j=1}^{x}\frac{topic\ word\ j\ count\ in\ node\ i}{total\ word\ count\ in\ node\ i}$$

Where n represents total number of nodes
x represents all words in all topics

What stands out is how low the actual metric really is and how the crawling algorithms actually show high values in comparison. Since RFS is used as a random baseline, its position compared to the actual metric is very close. However, BFS and DFS show fluctuation from this "real" value. DFS shows the greater fluctuations in the beginning but once the crawl size increases to 1300 it drops closer to the orange line. This is indicative of the behavior of DFS where it travels in straight lines across the network from all the 50 random starting points. This also shows how fast DFS actually travels away the starting points and covers a wide variety of nodes. This is the same reason BFS tends to shows closer positioning to the orange line. Because BFS stays in the vicinity of the starting points and covers nodes that have similar topics, it tends to be closer to orange line compared to DFS. But it isn't till most of the network is actually covered that BFS curve converges with the actual metric for the graph. This indicates that when BFS is chosen the network must be crawled longer. DFS converges faster than BFS but must be used in conjunction with BFS. In this case, DFS needed at least up to 2900 links that makes up for 82% of the network. In case of BFS it is almost the entire network. Although in real case scenario it may not be possible to have the entire view of the Darknet communities, these results indicate that any conclusion made before in the indicated percentage of the network is crawled will not be detailed. Therefore it cannot be concluded that Tor network is mostly used for dark purposes as it takes a longer crawl to obtain a representative view of the network. Such a method when applied on a bigger dataset would yield a better convergence.

This method was implemented to follow individual topics as well. The following plots in figure 36 are average topic distributions for all 5 topics individually.

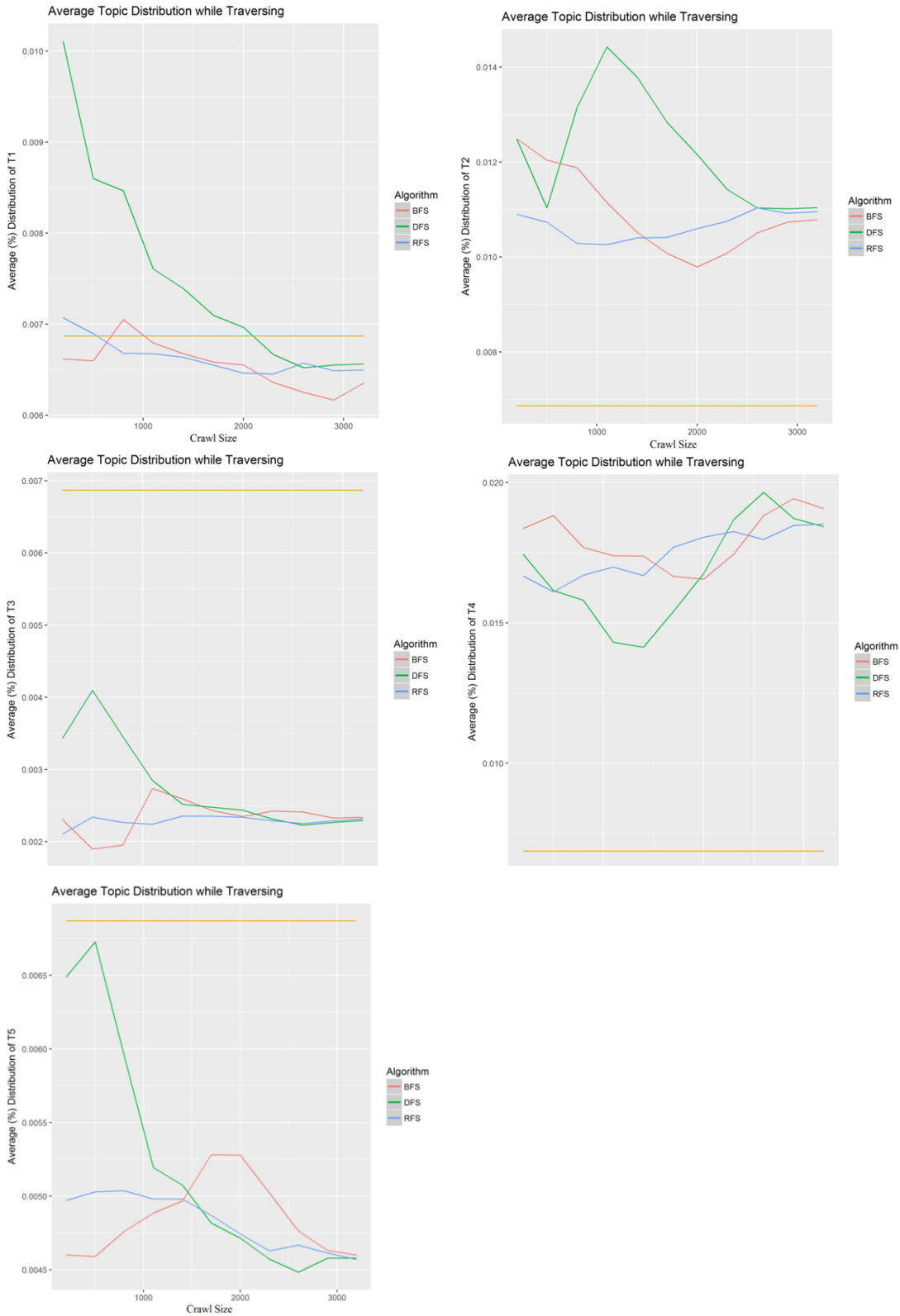

Figure 36 Average topic distribution while traversing individual topics

These plots show when the convergence takes place which indicates till what crawl size crawling needs to be carried out for each topic. When a particular algorithm converges sooner it also indicates that it is advisable to use that algorithm for a particular topic. When the curves are above the orange line, then discovery is high compared to that of the network for that particular topic (topic presence). Therefore the topics above the orange line must be targeted with the crawling algorithms that show higher topic discovery. Curves closer to the orange line, match the average topic discovery and hence called efficient algorithm.

Topics that show higher topic distribution are the popular topics compared to all topics present in the network. The higher the average topic distribution (presence) given by the crawling algorithm, the more efficient it is it. After convergence point, the crawling algorithms move in a stable manner but before that they have fluctuations.

Observations made from these plots were:

- RFS shows a stable average topic discovery than BFS and DFS. RFS shows very little fluctuations compared to BFS and DFS

-DFS gives higher average topic discovery for all topics except for T4. DFS therefore has high topic discovery compared to other algorithms

- DFS also gives a faster topic discovery and therefore recommended for discovery of topics

-T3 and T5 have lower discovery by the crawling algorithms

4.6 Conclusion

The following are some of the conclusions that can be drawn from the analysis:

1) How we crawl tells us what we know about the network:

This has been shown by understanding topic models as the crawl size increases with the use of three different traversal techniques. BFS tends to give better defined and broader topic models because it tends to remain in the network core until at least approximately two thousand links have been crawled, whereas DFS starts to give a variety of topics even after one thousand links have been crawled. Therefore, when one is analyzing only the unique domains, if one wants uniformity in terms of topics then one may start the crawl at the network core, in the manner of BFS. However, if one is aiming at a variety of topics, DFS can be selected. This shows how the hidden services of the Tor network are generally connected with one another where there are

similar domains connected in the network core; however, if only BFS is chosen and the crawl is stopped sooner than necessary, knowledge of those networked domains - or individual domains that did not come in the path of the BFS algorithm - will be lost. If DFS is added into the equation, those interesting domains which are not spotted by BFS can be found.

2) **How much we know about the Darknet is highly dependent on how much we crawl:**

If we had stopped either of the two crawls, BFS and DFS, sooner, the discrepancies introduced would be in terms of either only knowing 1/3 of the network, in the case of BFS, and conducting studies based on this 1/3 view; or finding an immense variety in terms of topics when in fact there are fewer, in the case of DFS. If BFS is stopped sooner, then what we know about the network is only as much as we have crawled. When such a concept is applied to the live network with a greater number of domains, it can be concluded that most of the studies done so far have a very limited view of the network. It is understandable that the number of hidden services in general fluctuates for many reasons; however, claims of fixed figures for what we know about the network should also consider the type and extent of the crawl. Therefore, in the case of BFS and DFS algorithms, it is necessary to crawl at least three thousand unique domains in the context of the links used as the ground truth for this thesis to obtain a representative value in terms of topics which can, in turn, help to understand what the Darknet is about. This has been illustrated using topic-topic comparison with heat maps in section 4.2.3. This figure can vary depending on the dataset worked with. When dealing with more than a million Darknet domains, this figure will surely be different. Therefore, this number is highly dependent on the dataset obtained by the crawling of the live network.

3) **Where we crawl from changes our view of the network:**

This has been demonstrated from section 4.4, where one can conclude that starting the crawl from different starting points has a huge impact on the view we get of the network. Although this was demonstrated for the first 100 links traversal, it can be inferred from the patterns observed concerning the algorithms that increasing the crawl size and using different algorithm techniques from several starting points will give a representative view of the network. This was the reason why a ground truth network was fixed—so that this comparison can be made and the point proven.

4) **Simple metric** readily available to understand if the view of network is representative or not is the topics distribution present in every node that makes up the network. How this metric was used has also been given in this chapter.

5) Using the metric defined, it was demonstrated that at least 82% of the network has to be crawled when using DFS and while using BFS almost the entire network has to be crawled in order to come to representative view of the network. Only when this is done the Darknet communities can be well represented else only a small insight of these communities will be obtained.

5 Conclusion & Future Work

5.1 Conclusion

Darknet communities need to be understood both well and thoroughly. There are many angles this has usually been approached from, such as traffic to and within the Tor network. This has often been studied through traffic analysis of various forms, either by observing the traffic to and from exit routers or by launching relays into the network by proving bandwidth. Another angle that is often used is to study how the hidden services are connected themselves as this informs us about the communities and emerging influencers within the online world over the physical world. This is the angle this thesis has explored. Although the hidden services are now rightfully called the onion services, the word "hidden" has been kept intact in an effort to demonstrate why these services should actually be called onion services and not hidden services. The term "hidden" and the research approach so far have been successful in giving the impression that the Darknet communities are mainly about illegal and illicit activities. Even the term Darknet ought to be renamed by the research community. Although it is the case that some of the hidden services are being used for "dark" purposes, this estimated conclusion streams from using BFS as the main method of graph traversal. BFS is known to introduce bias in the representation of the view of the network.

In this thesis, a comparative analysis of crawling algorithms in the context of the Darknet, studied in combination with the LDA-based topic models, has been shown. Considering that the entire network topology of what makes up the Darknet's hidden services is known, it has been demonstrated how any future studies should consider the extent they are covering the hidden services by working with the advantages and disadvantages of crawling algorithms such as BFS, DFS, and RFS.

With the use of topic models and by analyzing the words making up those topic models, it was observed that BFS gives a smaller variety of topics whereas DFS gives a broader variety because of the nature of these algorithms. Using the variety of topics as a metric, how BFS tends to remain in the core of the network before it launches out into the periphery of the network was shown whereas DFS quickly tends to go to the periphery of the network.

Additionally, how conducting crawls from different starting points greatly changes the view of the network has been demonstrated. A solution for handling this problem would be to look at the network from many different starting points and then combine the results for further analysis. This gives a closer and more refined view of the network. With the use of topics generated by LDA, the content of the hidden services has been studied, which ended up being a good tool to understand a variety of topics that can emerge from different angles. Further studies can be done that can overlap the various topics observed from every different starting point the network is being crawled from.

Another aspect defined by this thesis was a method to arrive to a representative view of the network. Topics distribution information available per webpage was used as a metric to do this. Crawling was conducted from 50 random starting points for every 300 links and the average topics distribution for every crawling algorithm was noted down. By doing so, it was found that RFS has a stable convergence to the real value of the network compared to DFS and BFS. DFS yielded higher average topics distribution whereas BFS yielded a lower value. From the behavior of the crawling algorithms it was concluded that when using DFS 82% of the network has to be crawled and when using BFS almost the entire network has to be crawled before any concrete conclusions can be made. Any inferences made before this cannot be used to conclude about the content of Darknet communities as the results would be far from the truth of these communities.

An already crawled network was used in thesis to investigate the research questions. The results and insights obtained from this study can be applied to any network that is similar to the Darknet. One possible solution to obtain a large dataset of hidden services urls is to crawl periodically from different starting points using combination of graph traversal algorithms and add to already existing list. The list indicating the number of hidden services can be cross checked with Tor metrics page which specifies reporting of Tor relays. Obtaining such an expansive list of hidden services is still a research that needs to be done.

The book of Proverbs tell us, “In all your getting, get understanding,” and this sums up the core of the message presented in this thesis. Solid conclusions cannot be made based on the view we have, but rather the ground truth should be explored as much as is possible. What we know of the

network and how, defines our understanding, which may be far from the actual truth of the Darknet communities. Clearly, this opens up various research topics, and some of the future work recommendations to gain a better understanding of the Darknet have been listed in the following section..

5.2 Future work recommendations

There are many possibilities of future work, which can be considered and presented as the following:

Investigation of better semantic structure analysis methods. In the context of the Darknet, an experimental study can be conducted to investigate which method would be better for understanding semantic structure. Different text-mining techniques could be investigated and appropriated according to the task at hand.

Automatic categorization of topics. Manual labeling with LDA is an arduous task when the number of topics is large. A useful approach can be, designing a new algorithm to train existing topic models to recognize any new crawled webpage, either to be grafted onto existing topics or to be left categorized so that the researcher can add a new category as per his or her judgment. If this can be successfully done, topic models can be used instead of search engines in the Darknet where users can go only to their topics of interest.

Clearly defining topics by investigating K value in LDA: For faster computations, the topic number was fixed in this thesis to investigate the influence on the view of the network from various starting points. However, it would be better to understand the same with clearly defined topics resulting from the Darknet so that communities can be better understood. This is important if LDA will be continued as a method to study the content of these hidden services.

Combining text mining and link analysis. Within the context of law enforcement, a combination of unsupervised text-mining tools and link analysis can be used in the context of the Darknet to target a particular type of criminals. The interconnections and communities in terms of which hidden services are connected to one another can lead to insights into the different illegal communities online. Specific focus groups being connected can also give insights into how trade operates, how gangs work, and how world events are shaped by what is taking place in these

focus groups. Adapting the work done in this thesis by further developing it and combining it with link analysis is definitely a recommended direction for future work.

Faster method based on link analysis than LDA. With the use of cluster analysis, fixing the number of clusters obtained to the K number of topics, the overlap found in the documents that match each cluster and each topic model can suggest that structured cluster analysis of hyperlinks can be a faster method than LDA. One can also study the performance of these algorithms in such a case.

References

[1] M. K. Bergman, "The Deep Web: Surfacing Hidden Value," *Taking License,* vol. 7, no. 1, 2001.

[2] Tor Project, "Tor," [Online]. Available: https://www.torproject.org/about/overview.html.en. [Accessed 2016].

[3] O. router, "Onion routing executive summary," 2005. [Online]. Available: https://www.onion-router.net/Summary.html. [Accessed November 2016].

[4] T. Project, "Tor Metrics - Unique .onion services," 2015. [Online]. Available: https://metrics.torproject.org/hidserv-dir-onions-seen.html. [Accessed October 2016].

[5] G. Kadianakis and K. Loesing, "Extrapolating network totals," Tor Tech Report 2015-01-001, January 31, 2015.

[6] P. Paganini and R. Amores, "Project Artemis – OSINT activities on Deep Web," 1 July 2013. [Online]. Available: http://resources.infosecinstitute.com/project-artemis-osint-activities-on-deep-web/. [Accessed 28 November 2016].

[7] A. Biryukov, I. Pustogarov, F. Thill and R.-P. Weinmann, "Content and popularity analysis of Tor hidden services," in *arXiv:1308.6768*, 2013.

[8] M. Spitters, S. Verbruggen and M. v. Staalduinen, "Towards a Comprehensive Insight into the Thematic," in *Intelligence and Security Informatics Conference (JISIC), 24-26 Sept. 2014 IEEE Joint*, 2014 IEEE Joint.

[9] G. Owen and N. Savage, "Empirical analysis of Tor Hidden Services," *IET Information Security ,* vol. 10, no. 3, pp. 113 - 118, 2016.

[10] L. Yang, L. Feiqiong, J. M. Kizza and R. K. Ege, "Discovering topics from dark websites," in *Computational Intelligence in Cyber Security, 2009. CICS '09. IEEE Symposium*, 2009.

[11] S. Ríos and R. Muñoz, "Dark Web portal overlapping community detection based on topic models," in *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, Beijing, China, 2012.

[12] G. L'Huillier, H. Alvarez, S. A. Ríos and F. Aguilera, "Topic-Based Social Network Analysis for Virtual," in *ISI-KDD '10 ACM SIGKDD Workshop on Intelligence and Security Informatics*, Washington, D.C., 2010.

[13] D. Moore and T. Rid, "Cryptopolitik and the Darknet," *Survival : Global Politics and Strategy,* vol. 58,

no. 1, pp. 7-38, 2016.

[14] D. A. Wondyifraw, "Design of a Dark Web Crawler and Offline Language Identifier for Amharic Documents," Addis Ababa University, Feb 2016.

[15] M. Kurant, A. Markopoulou and P. Thiran, "On the bias of BFS," *International Teletraffic Congress, arXiv:1004.1729,* 2010.

[16] C. Doerr and N. Blenn, "Metric convergence in social network sampling," in *Proceedings of the 5th ACM workshop on HotPlanet*, New York, NY, 2013.

[17] R. Munksgaard and J. J. Demant, "Mixing politics and crime - the prevalence and decline of political discourse on the cryptomarket," *International Journal of Drug Policy,* vol. 35, no. 0955-3959, pp. 77-83, 2016.

[18] M. Roberts, B. Stewart and D. Tingley, "STM: An R Package for the Structural Topic Model," *Journal of Statistical Software,* vol. 2, no. 55.

[19] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[20] C. Sievert and K. E. Shirley, "LDAvis: A method for visualizing and interpreting topics," in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, Baltimore, Maryland, USA, 2014.

[21] E. Alexander and M. Gleicher, "Task-Driven Comparison of Topic Models," *IEEE Transactions on Visualization & Computer Graphics,* vol. 22, no. 1, 2016.

[22] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences,* vol. 101, no. suppl 1, pp. 5228-5235, 2004.