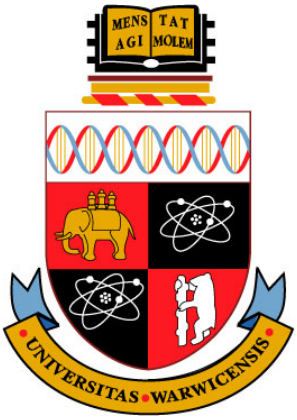


Online User Privacy Investigation Using Social Profile Seeding

by

Adam Coles

Department of Computer Science

University of Warwick

2016–17

ABSTRACT

Social media is the most popular activity to engage in whilst online. As the number and purposes of social media expands the view on an individual's life becomes more detailed, allowing users to connect with new people who share similar interests or friendship circles. However, people are often unaware of just how much data they make available for the world to see. This project is an investigation into the type of data social platform users provide on their public profiles, and how this data can be used to find meaningful connections between people as well as link the same user's profiles on separate social medias. Through a combination of preliminary research, manual exploration and developed tools, interesting results were collected, focusing on the Facebook, Twitter, Instagram and LinkedIn platforms. The final results come in the form of recommendations provided to both users, to reduce the risk of being attacked online, and platforms, to protect their users from being targeted.

ACKNOWLEDGEMENTS

I would like to acknowledge a number of individuals for their help and support during this project. Firstly, I have to extend my gratitude to my project supervisor Dr. Matthew Leeke, for his tremendous contribution to the project. Without his guidance and support the project would not have even come into fruition, let alone be completed to, what I believe, the high standard it is today. I also must give my thanks to Dr. Alexander Tiskin for providing his time out of a busy schedule to both mark my work and appraise my presentation.

The project would not have been possible without the assistance of all 50 members of the test set, and therefore I am grateful to my friends and family who took part in the investigation. Of particular note is Alex MacPherson and Inez Gill, who were willing to have their social profiles openly examined during my presentation. I'd also like to thank Inez a second time for her support during the early stages of my project, always listening to my ideas and providing open access to her social profiles for experimentation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER
ONE

INTRODUCTION

With access to the internet becoming increasingly vital in the modern world it is of no surprise that globally, as of June 2016, 49.2% of the population are considered to be active internet users [25]. Across all users, social media is the most popular activity to engage in whilst online, and a staggering 31% of the global population own a profile on at least one social platform [66]. All this activity produces a vast amount of personal data, with Facebook alone currently holding over 300 petabytes of information across their warehouses [79]. Despite the risks of an open profile being broadcasted to the public constantly, there are no specific guidelines for users to follow on how to keep themselves and their identities safe whilst using social media.

1.1 Motivation

The term 'active digital footprint' refers to the personal data an internet user gives permission to be accessible online, and all users of social media will have one [49]. As the number and purposes of social media expands this digital footprint becomes more detailed, allowing users to connect with new people who share similar interests or friendship circles. However, people are often unaware of just how much data they make available for the world to see, a fact utilised by the police to track down criminals and known associates [70]. A larger social media presence gives more chance to find additional information about a person that they have not explicitly shared. Whilst the police use open data to benefit society, if they have that data available to them so do potential criminals or stalkers, and social media consumers must be aware just how much can be found out about them online [51]. Despite attempts to make people aware of the dangers of open online profiles, in general there is a blase attitude towards online privacy. In 2012, a survey showed that 26% of American Facebook users shared their entire profile publicly, including all wall posts [50]. Without locating key attributes of social media that can be exploited there will not be a change in attitude towards this issue.

1.2 Project Aims

The main goal of the project is to locate potential exploits within social media that put user's privacy and identity at risk. Through a variety programmed components, the methods attackers could use to impersonate or abuse people through social networks should be identified, however not necessarily be pursued beyond proof of concepts. Results from the project may assist in a variety of fields, from public safety to use in law enforcement.

1.2.1 Connect Platforms

With the number of social medias ever expanding, the purpose of social media is shifting from simply connecting people. Platforms such as LinkedIn that provide a professional network or Instagram which focus on photos alone allow uses to share more detailed information about a certain aspect of their lives. Therefore, a key part of the project will be to examine links between platforms, as every new profile provides additional insight into an individual. Even though this stage of analysis is crucial it will not be fully automated, as some manual work will give higher accuracy results as well as present a closer to real life situation.

1.2.2 Connect People

Many platforms have obvious links between users, such as Facebook friends or Twitter followers. However, in general these links are kept hidden or the number of links is so large that the majority are insignificant. The project will aim to find the key connections for a target user on certain social medias, and the implications of learning these connections. Along with this there will be an attempt to examine the persistence of these connections over time, and whether these links are shared across platforms.

1.2.3 Use Simple Techniques

Exploring privacy exploits is not a new field of research and many methods for pulling information out of the various sources have already been applied and tested. In an attempt to differentiate from previous work as well as approach the problem for a realistic scenario the techniques employed by the project should be relatively simple, allowing for not only small automated tools but also outputting data in a format that a human can draw inferences from. Not only does this give greater leniency to the developer but this methodology should produce more accurate results.

1.2.4 Provide Recommendations

Statistical data works well for analysis, yet for actual users is not applicable enough to follow. To match both the motivation and overarching goal of the project it has been decided that the final result of the project will be recommendations. For users, the main security flaws for multiple platforms will be shown with evidence, and how to avoid these pitfalls will be suggested. For platforms,

examples of how other competing social medias keep users safe regarding specific information will be provided.

1.3 Change in Focus

The aims of the project listed above have changed drastically from the original conception of the project despite the theme remaining intact. This will be explored in more detail later in the report. In summary, it was felt that creating an application was too ambitious of a goal, and created time-consuming, uninteresting tasks such as user interface development. Modifying the goals of the project to be more research and results based allowed for more interesting discoveries to be made yet has not damaged the motivations behind the project.

1.4 Stakeholders

Throughout the course of the project a few people have invested both their time and their knowledge into the project and can therefore be seen as stakeholders. Dr. Matthew Leeke had a major part in creating the idea behind the project as well provide insight throughout in how the project should progress. Adam Coles, the single developer and researcher, has invested both time and effort into ensuring the project's completion. Finally, it could be argued every person who provided their profiles for testing have some interest in the outcome of the project, since the recommendations provided will be tailored towards them.

1.5 Report Structure

This report is an attempt to collect all the information about the project and is split into three major areas: exploration, investigation and reflection.

1.5.1 Exploration

Sections 2-4 of the report explore the main concept of the project. They begin by researching existing academic work regarding social media and potentially useful tools for development of investigative components. In section 3 the legal, social, ethical and professional issues which may hinder the project, and the measures prepared to combat these issues. To complete this area of the project, the key objectives of the project are defined, by looking at both the motivation of the project and the current consensus of academics researched in section 2. This is the only area of the report spoken in future tense instead of past tense, as when the foundations of the project were set out it was prior to the main investigation.

1.5.2 Investigation

Sections 5-7 explain how the main discoveries of the project were made. In 5, the approach to the investigation is explained, looking at how a test set was made, the key attributes of this set, an outline and justifications of the methods of examining the data provided and concluding with the tools that

must be developed to perform these methods. Following this in section 6 how each component was programmed and their purpose is explained in far greater detail. Finally, the results of the investigation are shown, as well as other interesting findings, ending with the recommendations.

1.5.3 Reflection

Sections 8-10 look back upon what was achieved in the investigation and whether the project could be called successful. In 8, the project management is discussed, explaining the methodology during the investigation and development, displaying the project timeline, and describing the tools used that were critical to the project's success. The aims and objectives of the project are revisited in 9, where each one is evaluated in detail to determine if it has been met or not. Ultimately this is the main criteria for the project's success. The main conclusions of the investigation are given in 10, along with possible future work and further extensions.

CHAPTER

TWO

RESEARCH

Examining potential privacy exploits in social media is already a hot topic in academia. Most of these existing results look at specific cases on certain platforms; few attempt to generalise the issue. However, combining the methods from these papers could be a way of effectively attacking individuals. Research for this project was split into three main sections: data targeting, looking at existing techniques to produce a list of key attributes required from a user's profiles; data harvesting, the methods and tools that collect this data from various platforms; data enhancement, how can the collected data be augmented to generate more inferences about a person.

2.1 Data Targeting

This area of research was an attempt to identify what information should be targeted from a user's profile. By no means is this a new filed of interest; De et. al created a process of detecting impersonation on a social network (figure 2.1) which looked at matching fields between victims and potential attackers [8]. They used four main attributes in the following order: basic profile data, photos, friendship networks and social activity. Although this method exists, Facebook do not currently employ an automated fake profile detection. Whilst not for exactly the same purpose as the project, this system provided a strong base for further research.

Images are one obviously crucial requirement in both impersonation and learning about individuals. The former is already rife within Facebook, with Fong, Zuang and He finding 8.7% of all Facebook users are using other people's images [17]. When it comes to learning, it is obvious that even with just a handful photos a human should be able to learn an individuals face, and there are plenty of face recognition algorithms to allow computers to do the same [72] [80]. However, pictures shared on social profiles can convey far more. Celli, Bruni and Lepri found that using the bag of visual words feature extraction technique on a large dataset of Facebook profile pictures allowed for traits, such as extraversion and dominance, to be predicted with reasonable accuracy, above 65% [5].

Using social media images for purely identification is not an untouched issue.

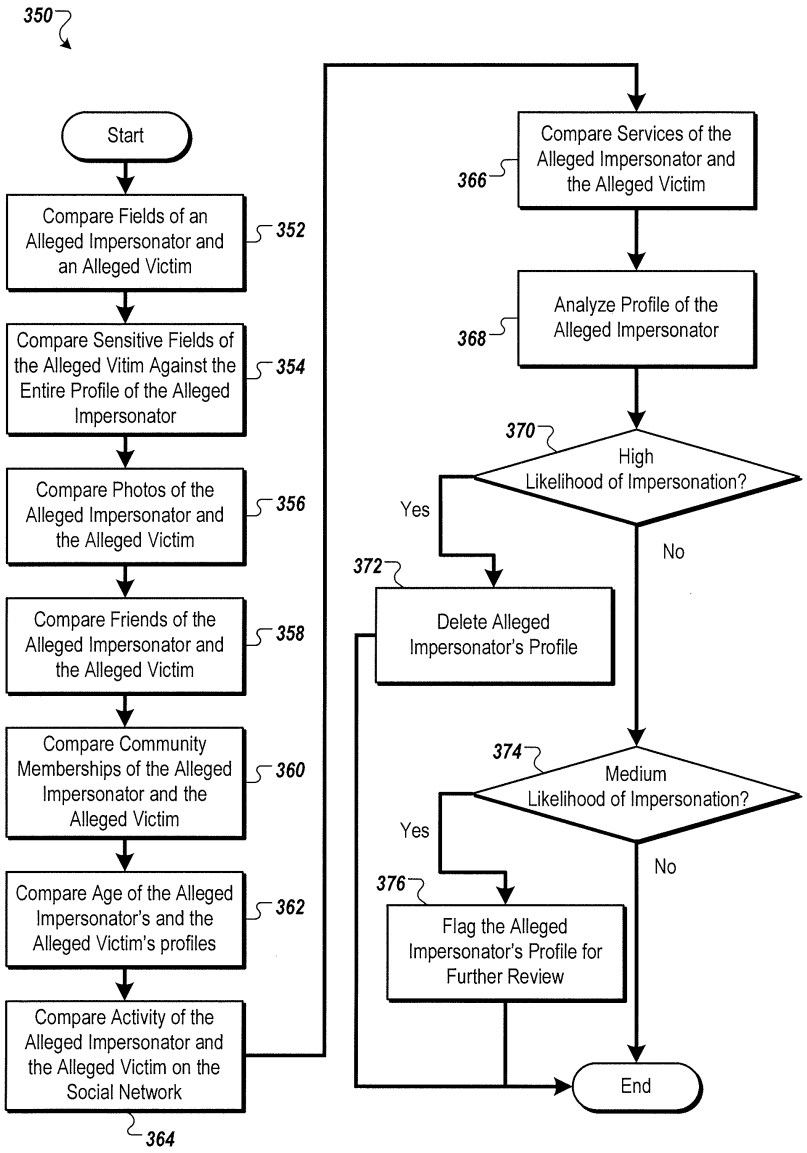


FIG. 3B

Figure 2.1: Method employed to detect impersonation on social media [8]

Launched in 2016, FindFace is a website that allows people to locate profiles of people on the popular Russian social platform VK through any photograph [16] [19]. By using a well-trained complex neural network, FindFace compares the face in the image uploaded to every face on the VK platform. Since every profile picture on VK is public it is impossible to hide from this attack. Whilst, due to to nature of neural networks, there is a chance the profile will not be located, FindFace have cited 70% accuracy with their algorithm, with other

sources citing even higher [32]. These types of applications show the dangers of having an open social platform with data exposure that can be exploited easily.

Nearly all social platforms have some way of explicitly connecting users, for example 'friends' in Facebook and 'followers' in Twitter. This can be seen as step one of understanding friendships on social media. However, finding meaningful links between people can be far more difficult. Roth et. al found a way of grouping friends in a network based on interactions between users [63]. They used a clustering algorithm on a weighted friendship graph based on some interaction, in their case emails, where the weight considered frequency, direction and precedency. Although they used emails, Roth noted that this method could be expanded to any interaction in a social network.

Arguably the hardest trait of a user to accurately obtain and replicate is their personality; how they represent themselves online. Identifying people in a conversation or debate context can be relatively straightforward, Zheng et. al found that with just 30 messages a specific user from 20 could be predicted with 83% accuracy [81]. Some of the features used in the algorithm for this could be directly used for impersonation, such as sentence structure or complexity of language. Sentiment analysis, particularly with Twitter due to the vast amount of text data produced, is also a common area of interest with many different approaches developed [57]. Even simple techniques can prove to be effective, with Fiaidhi et. al finding a basic K* classifier generates a sentiment rating with 91% accuracy [15]. Beyond sentiment analysis, any text content produced by users online can be mined for keywords or phrases, which Kapanipathi et. al used to create hierarchical interest graphs [39].

Within the scope of the project the techniques described in this section do not need to be implemented; they provide an understanding of what data the investigation should be looking to gather. As there is no final system, simply the knowledge of what can be achieved with what information is enough to rate the severity of data exposure and therefore provide reasonable recommendations to prevent this exposure.

2.2 Data Harvesting

Once there is an understanding of what data to target numerous different tools are available to harvest that information from online sources. To start, the majority of social platforms provide REST (representational state transfer) APIs (application programming interface) to developers that allow them to access user data [14] [35] [75] . Google also provides an API for people to use their search features, although this is heavily rate limited [23]. Nearly all social networks require user agreement for most of their information to be disclosed, this is not feasible for this project. Of all major platforms, Twitter is the only one that readily allows access to public user profiles, and even though this is rate limited, it should be pivotal for the project later.

Beyond social media there is a high possibility that more user related data appears elsewhere on the web. Whilst not the focus of the project it is important to know more robust means of attacking individuals. It is likely that anybody who is interested in finding out anything about a person using the internet will turn to a search engine. In 2013 Google had approximately 30 trillion websites indexed, so even searching using detailed parameters produces thousands of

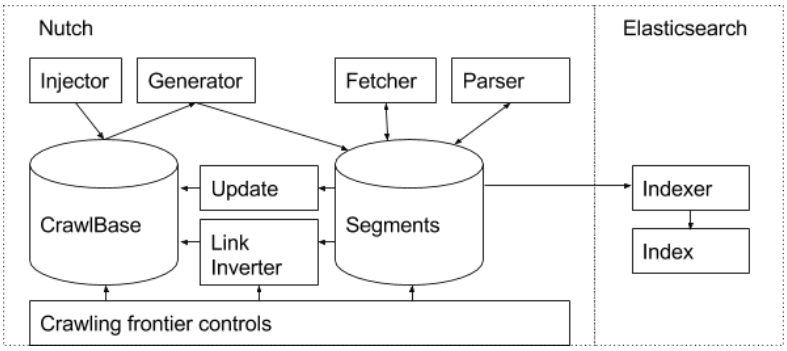


Figure 2.2: Main components of Nutch and its relation to Elasticsearch [42]

results [41]. However, search engines are not the only way to scour the web. Customisable or specialised web crawlers such as Apache Nutch, a Java based modular crawler, can output more desirable results than Google [18].

For specific open source web crawlers, Apache Nutch is highly scalable, robust Java web crawler. As it is built upon the Apache Lucene framework, Nutch is modular and therefore can be combined with other packages to expand upon the features available, such as search or indexing with ElasticSearch (figure 2.2) [11]. Norconex is another Java based crawler with similar functionality to Nutch. It comes with a plethora of documentation and a handful of tutorials, including how to crawl with the assistance of Facebook API [12]. However, due to being less established than Nutch, Norconex has less support from other tools and is not maintained as well.

In the context of this project it is unlikely large scale crawling and scraping will be required, since at any one time only one individual is being searched for. For this case, simple HTML (hypertext markup language) parsers will perhaps be more useful, since a handful of destination URLs (uniform resource locator) will be known, and these parsers can be customised to pick out specific information. One such parser, JSoup, is an open source Java library for manipulating HTML elements [38]. This library is well known for being simple and well documented. A more robust Java parser, HtmlUnit, provides additional functionality such as a browser emulator, allowing for cookies to be stored [29]. These cookies provide the ability to login to various websites, which could come to be particularly useful with social platforms.

As previously mentioned many social networks provide means for developers to access user data, and from this, many libraries have spawned to assist with using REST. In the case of Twitter, due to the slackness on privacy, libraries such as twitter4j can be used in the project to retrieve all public data from user profiles [76]. With Facebook, the user need to give a code to the application before most of their data be viewed. Despite the ability to scrape directly from Facebook this goes against their terms of service (TOS), although due to the educational nature of the project this should not cause problems. Should this TOS violation become a major issue, or scraping is time consuming and infeasible, this user code will be needed instead, and then the Java library restFB can be integrated into the system [61].

Ultimately the focus of this project will be on data retrieved from social

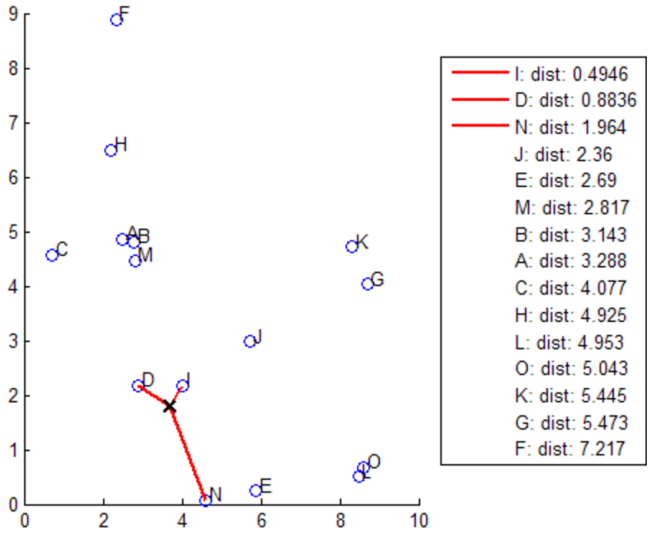


Figure 2.3: Selecting neighbours in the 3-nearest neighbours algorithm [52]

media, and therefore use of the Twitter API and the simple web parsers seem most likely. Although Nutch and the Apache Lucene toolkit provide a vast amount of scalability, the amount of time required to learn and initialise them seems too large a trade-off.

2.3 Data Enhancement

Various techniques can be employed after data has been collected to infer new results. These methods are used extensively in the work referenced in section 2.1, and this section will go over some of those in more detail. Whilst previous work up to this point has mainly been for reference of what is possible, the techniques discussed here could plausibly be deployed during the investigation.

"Text mining, also known as text data mining or knowledge discovery from textual databases, refers to the process of extracting interesting and non-trivial patterns or knowledge from text documents", Tan et. al [68]. Whilst text mining has many uses, for instance cataloguing documents in the publishing and media sector, the sheer amount of data produced in social media rapidly accelerated the use of text mining for brand sentiment and marketing purposes [1] [26] [54]. There can be many steps to text mining beyond the initial data collection, depending on the purpose. It is possible apply extensive natural language processing to identify speech or to syntactically decompose a corpus of text, which in recent times have been highly successful using deep learning techniques [30] [22]. On a simpler note, pattern recognition using regular expression template matching can be deployed to extract data such as email addresses or website links [36].

One of the more classical uses of text mining is for sentiment analysis, taking a section of text and analysing its subjectivity and positivity [48] [58]. This task comes with a variety of pitfalls, such as the definition of subjectivity not being set in stone [67]. When accomplished accurately though this technique

is a powerful tool in analysing individuals agendas and has been used in such fields as marketing and political campaign analysis [71]. Whilst not a perfected technique, despite only needing to achieve roughly 79% accuracy to be on par with human analysis, this method is too complex and powerful to be used in this investigation, although it may be worth keeping in mind for extension work [55].

When handling statistical over textual data there are more refined and accurate techniques for extrapolating patterns. For the purposes of this investigation, supervised learning, the method of using pre-existing data to predict values or classify a set of target data, seems the most useful machine learning approach, since there will be plenty of historical data available to create a training set [53]. Listing all possible procedures in this field would be infeasible, although a few can be highlighted. A decision tree is a graph of likelihoods given the values of certain parameters, used in both classification and regression tasks [4]. They use a metric, generally Gini impurity or information gain, to determine the split of data around a certain attribute and select the best possible split at every level [62]. Even though they can be powerful alone, decision trees are often combined, known as ensemble, to create more robust and complex learning tools. One such method, random decision forests, produce multiple trees in an attempt to fix the problem that decision trees often over fit to the training data [20] [28].

Another well-known supervised learning methodology is the k-nearest neighbours algorithm, which can be seen in figure 2.3 [44] [52]. This works by arranging the training set of data on a set of parameters axis; when new data arrives, it is placed on the axis and is classified based on the closest neighbours by some weighting metric, generally Euclidean (straight-line) or Manhattan (total difference on each axis) distance, although different metrics can be used dependant on purpose [40] [78]. The algorithm is popular due to it being extremely simple and flexible, using different values for k, and therefore approximating with a smaller or larger subset of the training data, can cause drastically different results. Despite it's simplicity, there have been many successful cases of using k-nn effectively, such as sentiment analysis performed by Fiaidhi et. al discussed in section 2.1 [15] [46].

This chapter covered the preliminary research in the general field of the project. In the chapters to follow the focus shifts towards the project specifically, examining the issues that may occur during it's course and what the project is trying to achieve.

CHAPTER

THREE

ETHICAL, SOCIAL, LEGAL, AND PROFESSIONAL ISSUES

When handling personal data, even for research purposes, certain precautions must be taken to protect the people involved. The aim of this section is to discuss problems that may arise in the ethical, social, legal and professional fields, and how preparations have been made to resolve them. For a base during development, the British Computing Society Code of Conduct will be maintained [65]. This is to make the code reusable in the future should a full system be produced.

3.1 Ethical Issues

Due to the highly personal nature of the intended investigation, a plethora of ethical issues emerge, the key issue being the privacy of the individuals used for testing. Any test users will have to give their express permission for their profiles to be examined, and if they are selected to be an example for displaying the potential results of using the methods discovered, they must give further permission for their details to be shown to stakeholders or other third-parties. At any point the user has the right to remove themselves as a test case if they feel uncomfortable with the information found.

Despite the many potential ethical issues with this project, by following the simple rules outlined above and without attempting to access non-open data all these issues can be avoided.

3.2 Social Issues

Continuing from the protection of privacy of individuals involved, there is the chance that some evidence found may lead to somebody feeling victimised. To prevent this as much as possible, the investigation will avoid examining a persons sexual, religious or political preferences. Even though some of this data may not be private for all users, its best to abstain across the entire test set to avoid accidental discovery. On top of this, there will be no conclusions drawn using race or disability, and the set of users will try to be as diverse as possible.

Should methods be found that successfully extract important data from profiles, the view on social media may change for some users, hopefully leading them to be more aware of profile security.

Social issues are harder to analyse than the other as they are often subjective to individuals. Looking at where these issues may arise and refraining across all users, in combination with the solutions to the ethical issues, should be enough to prevent problems.

3.3 Legal Issues

Since the tools developed will inevitably use some third-party software or libraries the developer must ensure that they have the correct licensing. A major aim of the project it to only use open data and open software so all third-party sources should be covered by some open source license [31]. As personal data will be mined and stored at some time during the program operation, the system must comply to the terms of the Data Protection Act [24]. Following from this, the data collected must come from legitimate sources that have the right to own the data to begin with, although this may be hard to verify.

The majority of legal issues faced will be tackled during any development that takes place. Listed above are the simple rules to adhere to that many programmers are trained to follow instinctively.

3.4 Professional Issues

It is critical for the investigations success that the results presented are deemed to be trustworthy. All recommendations will be justified by a combination of statistical results as well as patterns observed during the exploration of various platforms. Also, developers will follow the BCS Code of Conduct which ensures they are working for the public's interest and for the profession [65]. Since it is likely that other academic's ideas will be used throughout the project, they will be referenced accordingly.

Similarly to legal issues, industry standards are taught to professionals during their learning experience, and therefore should be straightforward to follow.

Within the first three chapters the project foundations have been laid, giving some insight into the general aims, the current related work and available tools, and finally with this chapter the issues that may arise from the project. With these covered, the next chapters look at how the project began to form, beginning with the objectives and moving onto how the investigation was approached.

CHAPTER
FOUR

PROJECT OBJECTIVES

Breaking down the main aims of the project into detailed objectives gives direction to the investigation and ensures time is being used effectively. However, these targets are flexible and still give some room through generalisation for the investigation to change its focus. The objectives can be split into two areas, investigative and technical.

4.1 Investigative Objectives

The following objectives cover what is trying to be achieved from the focus of the project. They guarantee that conclusions drawn from the results can be trusted, that the results and conclusions relate to the motivation and focus of the project, and the conclusions made are somewhat new in their respective field.

I1: *Compile an unbiased test set of people of at least 30 who own profiles on a variety of social media accounts. Every member of it must have given their consent.*

I2: *Examine user privacy on the following social platforms: Facebook, Twitter, LinkedIn and Instagram.*

I3: *Identify information which is made publicly available on each of the chosen platforms.*

I4: *Find at least two methods of linking the same person's profiles across different social medias, and assess which platform is hardest to reach.*

I5: *Determine at least two methods of finding the connections between different people.*

I6: *Make assumptions on the strength of connections between different people, and be able to group these connections into 'friendship' groups, using at least one customised technique.*

I7: *Estimate the chance of identifying or replicating a social platform's user's speech patterns based on the average amount of text mined from that platforms profiles.*

I8: *Estimate the chance of imitating a social platform's user based on the average number of images scraped from that platforms profiles.*

I9: *Provide recommendations to users of social medias to keep their personal information secure.*

I10: *Provide recommendations to social platforms to keep their users secure.*

4.2 Technical Objectives

The following objectives explain what will be required from the developer when producing the tools used to assist the investigation. They envelope both the functional and non-functional areas of development.

4.2.1 Functional

Functional requirements are what the tools should be able to do, and the outputs expected from them.

T1: *Create an automated batch script that will apply all tools written to multiple users over time.*

T2: *Scrape all possible information from as many platforms as possible and store this information locally in a database for further examination.*

T3: *Reduce the effect of any rate limited APIs by locally caching data, allowing for rapid development iterations.*

T4: *Create some way of using Google's search tool to perform wide search on a user, and pull these search results into a text file to be examined.*

T5: *Automatically detect when a search result has found a user's social media profile.*

T6: *Use twitter4j or equivalent library to access the open API provided by Twitter and accumulate useful data from this.*

T7: *Implement at least one supervised machine learning technique, and analyse the consistency of this method.*

T8: *Produce social graphs for a given user, and either create software to display these or format the data to be used by third-party software.*

4.2.2 Non-Functional

Non-functional requirements look at the attributes of the produced software, and how they have been developed to ensure they work in a real-world scenario.

T9: *Only harvest open data from social platforms that does not compromise user privacy.*

T10: *Ensure all third-party libraries used are open source.*

T11: *Tools written should be extensible. They should be modular, and therefore expanding their purpose or repurposing them will be easier.*

T12: *Tools written should be efficient. Whilst it is unfeasible to put a limit on their runtime, the longest tool should still be able to run overnight.*

T13: *Tools written should be well maintained. They should be well documented as to make the reusable for extension to the project in the future.*

Since an agile approach was chosen for the project these objectives are subject to change, however the modifications made should not drastically alter the core concepts.

4.3 Transitioning From Specification

The aims of the project have changed drastically from what was proposed in the specification, and therefore most requirements set out there have now become defunct. Despite this, the general ideas originally presented remain in the project. There are some clear links between the two sets of the objectives and the initial requirements. For instance, functional requirement 2 in the specification discusses finding links to other social medias once Facebook has been completely scraped, a similar concept to investigative objective 4. Also, since non-functional requirements are generally comparable across development, these change little between the two documents.

4.4 Hardware, Software and Research Constraints

To keep the developed tools maintainable and usable some constraints must be made on the resources used. The final tools should be able to run on home computers as this is what is available to the investigator, as well as giving room to extend the project into a complete system at a later date. Since collecting results will be done iteratively, many components in the tools will have to be fine-tuned. No tool can take longer than 12 hours to run; this would prevent them being run overnight. As mentioned previously, the software should use only open license sources. This is to add no extra costs to development and make the tools accessible to any interested third-parties.

When it comes to the results of the investigation, all conclusions drawn should be fully justified. No specific single person should be mentioned in the report to maintain their privacy, therefore case studies mentioned will be anonymised, unlike the project presentation where the people used gave consent. Authors of algorithms and third-party libraries used during development should be fully acknowledged and referenced to abide by the solutions to the professional issues of the project.

This chapter marks the end of the exploration stage of the report, with the end result being clear goals that the project must reach. In the following chapters how the investigator approached the objectives discussed in this chapter and the tools created to accomplish them are discussed.

CHAPTER

FIVE

INVESTIGATIVE APPROACH

With any investigation, a lot of time is devoted not to the development of programmed tools but to the exploration of the idea, to understand in greater depth the specific areas of interest. This section of the report will cover: the creation of a test set, which became the foundation of the project; patterns initially recognised in this test set; identification of what knowledge can be gained from each platform; and the reasons why Twitter became a focus of the investigation.

5.1 Creating a Test Set

To progress further in the project at this stage, sample social profiles were required. People were reached out to over Facebook, where they could 'like' a status to give their consent to be part of the investigation. This meant every example user in the project had, at the very least, a profile on Facebook, which is sensible as Facebook is the most popular of platforms, with 1.23 billion daily users as of December 2016 [10] [13]. The status was made available for 3 days during which 85 people gave their consent. Fifty of these were selected pseudo-randomly to be part of the investigation, with favour towards people better known by the investigator, as therefore facts could be checked with greater accuracy.

Outside of Facebook, a large proportion of the test set had additional profiles on the chosen platforms, the exact breakdown displayed in figure 5.1. As shown, the number of people who share certain social medias is slightly higher than average, represented in table 5.1. There are numerous reasons for this: people with greater activity on social media were more likely to see the advert, as this was the medium it was posted on; selection was not entirely random and individuals with more profiles were preferred; and the advert was shown to connections of the investigator who's age group (18-25) is one of the larger users of social media, shown in figure 5.2. This bias does not alter the validity of the investigation, using people owning multiple social profiles expands understanding and the results are targeted at this type of person. Additionally, to keep statistics produced from the project realistic, seven users in the investigation only own Facebook profiles, greatly limiting the amount of data that can be

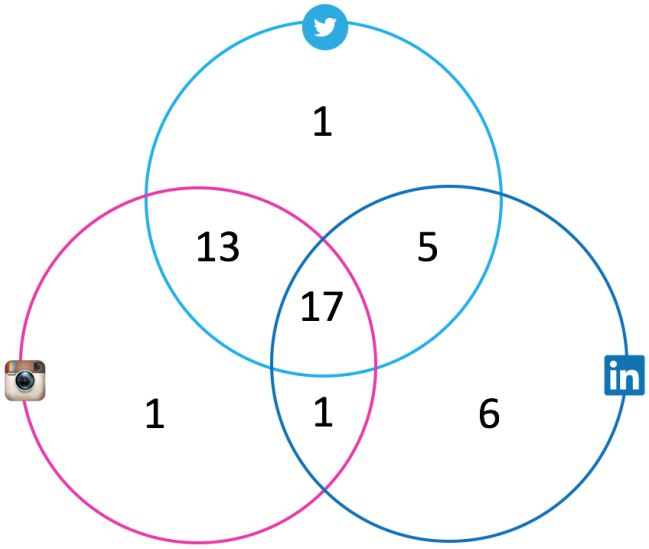


Figure 5.1: Breakdown of profiles in test set

	Twitter	Instagram	LinkedIn	Facebook
% Twitter Users Using	-	58	47	91
% Instagram Users Using	52	-	38	94
% LinkedIn Users Using	39	35	-	86
% Facebook Users Using	29	34	33	-

Table 5.1: Social media users who use a different platform (2014) [9]

harvested about them.

Other attributes of the test set can be seen in figure 5.3. As expected the gender ratio is close to even, however the age ratio is skewed heavily towards 18-25 year olds. This is due to again the advert being promoted on the investigator's social media, who has connections a similar age to themselves. Similarly, the amount students in the set is high, which is related to the age and occupation of the investigator. Despite the clear biases in the demographics of the set this should not hinder the validity of the investigation, as this age range and occupation typically use social media regularly, and therefore the results of this investigation are targets to them. There are a few outliers in the set which hope to bring insight into the habits of working professionals.

5.2 Initial Analysis

After the database of users profiles was curated some trends were visible straight away. It was these patterns that steered the course of the project further, as they piqued the interest of the investigator, which led to the project moving away from heavy development. Whilst ultimately the test set is quite small, it focusing on one demographic improves the chance that the patterns are not random

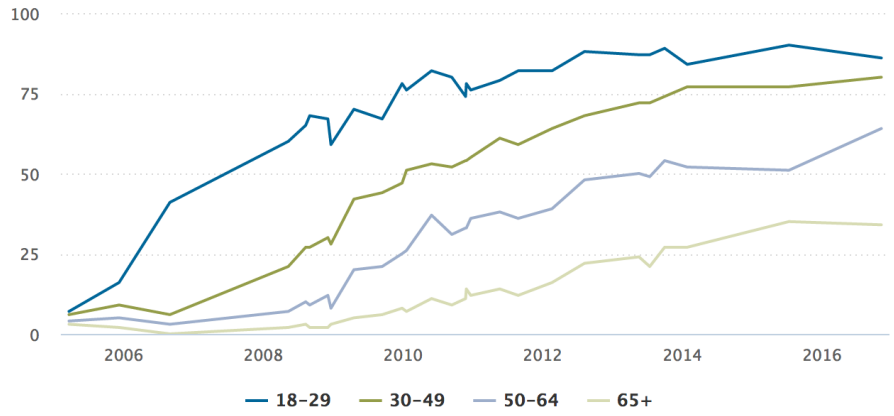


Figure 5.2: Percentage of U.S adults who use at most one platform by age [6]

chance, and the foundations of development were made from some preliminary exploration into them.

As mentioned previously each social media serves a different purpose. Facebook is a general platform, serving as a way of connecting friends and family to share what is important in their lives [13]. Twitter serves to share snippets of ideas and information quickly to a network of friends or anyone else who is interested [73]. Instagram is a way of sharing information through a series of photos and images [34]. Finally, LinkedIn is a professional network for finding new work opportunities and searching for people to fill available positions [47]. The more specialised platforms can be targeted specifically for the data they provide. If a reliable link between two platforms can be formed, the amount and variety of information greatly increases.

One immediately noticeable property in the test set is the relation between Twitter and Instagram; only 0.06% people who owned an Instagram profile did not own a Twitter profile, and 17% vice versa. With Instagram being a photo sharing platform there is a high chance that users would share the same photo on Twitter, potentially linking the Instagram account directly when doing this. This was used as a means of connecting the two platforms, therefore finding more profiles of a given individual. On a similar note, LinkedIn had the most unique users at 6 (ignoring Facebook), with half of these being over 30 years old. Since LinkedIn is a platform targeting working professionals this was as expected, although does imply the older generation prioritise owning a LinkedIn profile over other social medias. In fact, of the eight people aged over 30, five had a LinkedIn profile and the remaining three only owned a Facebook profile. Although there is a potential assumption to make here, due to the lack of numbers it is impossible to provide concrete evidence.

Another interesting point was the lack of protection on the majority of social profiles. Across all 36 Twitter profiles, only 4 had private mode active, with a slightly higher 13 in 32 when it came to Instagram [7] [33]. When these profiles are in private mode the only data typically available to extract from them are a single image, a name and some alias. The numbers also provide information about what the public believe is more important to hide; clearly photos, like

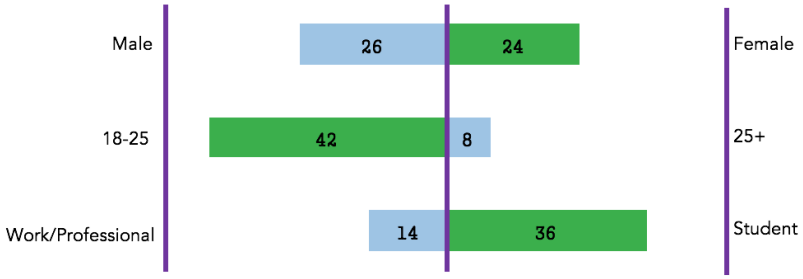


Figure 5.3: Broken down demographics of the test set

that displayed on Instagram profiles, are more important to hide than habits and opinions, like that provided on Twitter profiles. Furthering from this not a single LinkedIn profile had maximum privacy constraints enforced, probably due to its use to promote oneself in a professional environment.

Beyond the obvious data available from each platform's purpose, there is additional data hidden in the intricacies of some social medias. For instance, each platform has a way of connecting users, and in the case of Twitter their follower system is public; a key part of the investigation. Facebook profiles display connections depending on a privacy setting, however where available this did not prove particularly useful. Many connections are Facebook are not meaningful to the users, and whilst there were signs that the first handful of displayed friends were more likely to be close to the user, this was challenging to prove. Of six users surveyed, displayed Facebook friends scored above average in terms of 'closeness' in five cases, although for each user at least one displayed friend was deemed as insignificant to the user.

From collecting the links to profiles, known on most platforms as screen names, it became clear that many users reuse their online alias across accounts. By being able to predict likely alternate names for users once one has been obtained would provide an opportunity to link to a new platform. This idea has been used before to some success, Pennise et. al used the similarity between screen names on Adam4Adam, a dating social network, and Facebook to locate an individual in a STD investigation [59]. Whilst in this particular case the ability to find the person in question was a positive, in most cases when using dating applications people do not want to be found without their consent. Common changes to screen names include the addition of numbers to the end of the name and adding full stops or underscores to separate words; this was exploited later in the investigation.

5.3 Twitter Analysis

In section 2.2 the various APIs of platforms were discussed, with Twitter having the only open access to user data without explicit user consent. This led Twitter to be a focal point of the investigation, especially considering the lack of privacy settings enforced by nearly the entire test set, as mentioned in section 5.2. Whilst the implementation and results came about later in the investigation,

firstly it was important to identify what data is available and how that may be used. By combining the data accessible on Twitter with the methods discussed in section 2.3, the amount of private information that can be discovered grows vastly.

A *'tweet'* on the Twitter platform is a message of no more than 140 characters in length that a user sends onto the network. They can contain links and images, with any shared link only counting as 23 characters in length. A word in a tweet beginning with the hash symbol is known as a *'hashtag'*; when multiple users start to tweet the same hashtag, it becomes *'trending'* and displayed to all users in that region. Users can *'retweet'* tweets they particularly like, sending the same tweet out onto their network, or alternatively *'like'* tweets they are fond of. To connect to others and expand their network, a user can *'follow'* people to see their tweets on a feed, or be followed by people interested in their content [74].

The first entry point to understanding a user on Twitter is their tweets. Whilst this could be text mining for interests or sentiment analysis on the contents, techniques examined in section 2.1, looking for links could be equally as useful. With Twitter users having a high chance to own a Instagram account, discussed in section 5.2, there is a reasonable chance they would link to the other platform in a tweet to inform their followers of their new profile. Beyond social media, people will often link to meaningful websites in their lives, which could be personal projects or projects they are heavily invested in. Expanding the follower graph is another way of finding information on a user, their key friendships and groups. Since any user in the graph could be private, there are flaws to doing this, however in the majority of cases a near complete graph can be constructed. Most Twitter following relations are one-way, with only 22.1% of connected pairs having a reciprocal relationship [43]. Focusing on these two-way pairs will not only shrink the follower graph to a manageable size but also remove the connections less likely to be part of friendships.

The main two experiments performed on Twitter during the investigation were attempting to segment the follower graph into key friendship groups and observing if it was possible to extrapolate friends from a user's likes. The latter began with two concepts; a user is more likely to like a friend's tweet over a different tweet and the people who like tweets with few likes are probably friends of the tweeter. To extend this further, when a user likes many tweets by a particularly popular user or celebrity some of the user's interests can be assumed. During the project these theories were attempted to be proved through supervised learning.

With a direction established for the investigation and the objectives associated to various social platforms, how tools that can extract and manipulate data from these platforms can now be discussed.

CHAPTER

SIX

TOOL IMPLEMENTATION

Even though the project is more focused on research and results instead of developing a system, there was still significant time invested in the development of programmed tools which assisted the investigation. This chapter of the report covers all the decisions made during this time, the libraries used, how each tool was programmed and the reasoning behind that tool being required. Also covered are various gaps in the implementation and how those gaps would have been filled given more time.

6.1 Framework and Management

The first decision to any programming task is selecting the right language. This was relatively trivial for this project; the developer has the most experience using Java and there are many libraries available for Java to achieve just about any task. One of the main flaws of Java is it can be quite time consuming to create clean, elegant interfaces, and when the project was initially system based this was a major concern. However, now that no polished system is required, using Java was a far more sensible choice. Another bonus Java brings is it's cross-platform compatibility. Even though at no point was a second device required, having the ability to recover from disaster was comforting.

For managing the codebase, Apache Maven was used throughout development. Maven is a Java software management tool that allows for exterior dependencies to be imported and handled with ease [2]. There is a huge database of available open-source libraries online, with many developers providing a Maven link on their project websites. These links are added to an XML file which is the backbone behind the inner workings of maven. Despite a steep learning curve, by following a handful of online references importing libraries for use became trivial. Should testing have been required, Maven also has useful tools to perform repeatable unit tests with libraries such as JUnit, although these remained unused.

Despite not having software iterations, version control software was still used. In part, this was to back up work and data on the GitHub cloud, to prevent major setbacks should disaster occur [21]. Also, it is standard programming practice to use version control when programming and the developer feels more

```
#!/bin/bash

rm -rf results
mkdir results

input="./input.txt"
while IFS=:"," read -r f1 f2 f3 f4 f5
do
    mkdir "results/$f1"
    java -jar ./target/online-privacy-1.0-SNAPSHOT-jar-with-dependencies.
        jar $f1 $f2 $f3 $f4 $f5
done < "$input"
```

Figure 6.1: Bash Script to Execute the Toolkit (run.sh)

comfortable when using git. This became especially useful when performing minor tweaks to the machine learning tools or attempting to overhaul some of the scraping tools, providing the developer a safe return point when manually reverting code became impossible.

All code was written in the Atom text editor [3]. In previous documents, it was said that Sublime Text was going to be used, but Atom provided far more possibilities in the form of add-ons, such as an in-built terminal which was used regularly during bug fixing [69]. Other tools in Atom used were vim-mode, switching editing to the style of the Vim editor; java-linter, a way of checking syntax errors in real-time for Java code greatly speeding up development through less wasted compilations; and beautify, a code indent and brace fixer particularly useful when adding dependencies to the Maven XML file.

To execute the code a simple bash script was created. This looped through a CSV (comma-separated values) file of test user information, generated folders to store any data outputted from the system and executed the toolkit with the correct parameters. This script is shown in figure 6.1. Only a single jar was used for the entire tool suite; before compilation the tools not in use would be commented out to prevent waiting extra time for needless results.

6.2 Alias Analysis

Comparing aliases required no data from the internet, as all profile names had been collected beforehand, so this seemed like an easy starting point for development. The comparison approach was to use Levenshtein distance, a well-known technique dating back as far as 1996 [45]. This method looks at the number of single character edits between two strings, which include insertions, deletions and modifications. For example, the strings 'frog' and 'dog' have a Levenshtein distance of two, remove the *f* and replace the *r* with a *d*. Whilst not the most robust string comparison algorithm, in this context it can be powerful, as many common aliases have a distance of only one or two. It has been refined over the years to a point where it is both efficient in space and time, the pseudo code can be found easily online [27].

Defining exactly when two aliases matched was the most difficult part of designing this tool. After much trial and error, it was decided that a distance

of 1 represented a very similar alias, a distance of 4 or less represent similar aliases, and a distance of more than a third of the combined length of both aliases showed a clear difference between them. The first two distances are rather self-explanatory; the last was achieved by altering the fraction until only all the human-checked different aliases remained. Originally development began in a bash script, yet due to the syntax of if statements being slightly larger and considering the amount of comparisons required simply for equal checking aliases a small Java program was created instead.

6.3 Scraping Facebook

Since Facebook was the social media that tied in every user of the test set, it was also the first to be examined. As mentioned in section 2.2, Facebook, unlike Twitter, does not provide completely open access to user data through the use of REST APIs without a token private to each user. Therefore, a custom scraper had to be made to target the data on these profiles. Whilst this does go against the TOS - automated tools are banned on Facebook - it was felt that this was directed more towards crawlers than scrapers. Given that the project was for educational purposes, and the user's being targeted had given their express permission, it was deemed reasonable to create this tool and only cease if demanded to by Facebook themselves.

To begin with public, Facebook pages that can reached through any search engine without logging into the platform were looked at, however it was immediately apparent that these were inconsistent. The term 'internal profile' will now be used as the phrase for the profile viewed when logged in to a platform. For some users, all information provided on the internal profile was the same as on the public profile, yet on others the public profile had little to no data, whilst the internal profile had a reasonable amount. Also in many cases, it was difficult to locate the public profile, as some are hidden from search engines. This meant that the tool had to log in to Facebook before scraping. A fake profile was created for the scraper, since the investigator was connected to all members of the test set and therefore would have greater access to their data than a standard user.

Some potential libraries to assist in data harvesting were mentioned in section 2.2. Of these HtmlUnit, a Java based browser emulator, was chosen to be used as it provides a means of logging in to platforms prior to accessing the profiles. HtmlUnit is extensively documented, yet parsing pages using it seemed to the developer as a little overly complex, so Jsoup was used to parse the pages fetched by HtmlUnit [29] [38]. A code extract of how logging into Facebook is done shown in the first half of figure 6.2. Some initial settings should be made to the web client, namely enabling cookies to store the authentication token and disabling javascript, as HtmlUnit struggles to deal with the number of AJAX (asynchronous javascript and XML) calls made during login. The authentication token is the cookie which keeps a user logged into a platform when changing web pages.

The second half of figure 6.2 shows which pages of a users profile are targeted for scraping. There are three main areas of interest: occupation, living and relationship. The first covers both work and education, and proved vital in linking to other social medias as well as simply understanding more about a per-

```
public static void loginFacebook() throws IOException
{
    webClient.setJavaScriptEngine(new JavaScriptEngine(webClient));
    webClient.getCookieManager().setCookiesEnabled(true);
    webClient.getOptions().setJavaScriptEnabled(false);

    final HtmlPage facebook = webClient.getPage("http://www.facebook.com"
        );
    final HtmlForm form = (HtmlForm) facebook.getElementById("login_form"
        );
    final HtmlSubmitInput button = (HtmlSubmitInput) form.
        getInputsByValue("Log␣In").get(0);
    final HtmlTextInput textField = form.getInputByName("email");
    final HtmlPasswordInput textField2 = form.getInputByName("pass");
    textField.setValueAttribute("c**********s@gmail.com");
    textField2.setValueAttribute("*******");
    final HtmlPage loginPage = button.click();

    webClient.getOptions().setJavaScriptEnabled(true);
}

private void scrapeProfile(String id) throws IOException
{
    String baseUrl = "https://www.facebook.com/" + id;

    String education = getHtml(baseUrl + "/about?section=education");
    String living = getHtml(baseUrl + "/about?section=living");
    String relationship = getHtml(baseUrl + "/about?section=relationship"
        );
    String all = education.concat(living.concat(relationship));

    PageScraper mainPage = new PageScraper(getHtml(baseUrl), PageType.
        FB_PROFILE, targetProfile);
    PageScraper infoPage = new PageScraper(all, PageType.FB_EDUCATION,
        targetProfile);

    String photos_albums = getHtml(baseUrl + "/photos_albums");
    Document doc = Jsoup.parse(photos_albums);
    Elements profiePicLink = doc.select("a:contains(Profile␣pictures)");

    if (!profiePicLink.isEmpty())
    {
        String link = profiePicLink.first().attr("href");
        PageScraper photosPage = new PageScraper(getHtml(link), PageType.
            FB_PHOTOS, targetProfile);
    }
}
```

Figure 6.2: Login and Targeting with HtmlUnit (FacebookHarvest.java)

son. An individual's current or previous home location is interesting in its own right, however when shared on one profile is likely to be shared on other's and again, this information can be used to connect to new platforms. Relationship and family information does not directly provide that much data, bar the one or two confirmed strong connections to other users, however having these guaranteed contacts provided means of reaching new profiles through them. Finally, the profile pictures of a user were scraped. To begin the database of photos for identification.

Not displayed in figure 6.2 is the 'PageScraper' class. This class used JSoup to parse the HTML provided in the form of a string generated by HtmlUnit. The HTML is split into a tree structure by JSoup where elements can be accessed using their id or class property. In Facebook container (div) ids were seemingly random strings and it was difficult to keep track when coding which id had the desired data. Thus, alongside JSoup a custom HTML tree class was created, which allowed divs to be renamed into a human friendly notations. Once the data had been identified and labelled it was stored in a profile object for later use. Without listing extensive edge cases it was impossible to label data provided entirely accurately, for instance an educations finish date could have many different notation including 'September 2015' and 'Class of 2015'. Regular expressions were used to categorise data as well as possible.

Following the scraping, all the data was outputted into a text file in a unique folder for each user. This allowed the use of bash scripts to extract key information from these files, for example counting the number of lines in the photo file, 'facebookPhotos.txt', gave the number of profile pictures available for that user, with each line consisting of a URL to that photo. A similar concept was implemented for counting the educational or occupational institutes per person. In chapter 7 the results from this will be discussed further.

6.3.1 Locating LinkedIn Using Google

The first platforms to attempt to link together were Facebook and LinkedIn. This is due to them sharing many properties: both use real full names; both have a single main profile picture; there will be at least some shared friendships; and education/occupation history is present on both. Whilst the shared data can be harvested from Facebook and stored offline, there still needs to be some means of searching for a LinkedIn profile using this data. The developer turned to the search engine Google. Similarly to Facebook, Google's provided developer API is constrained, both being rate limited and requiring an explicit purpose to use. Once again HtmlUnit was used to work around this.

A standard user of Google goes to the main website and types a query into the search box. This generates a URL which the server processes to return results. There have been attempts, with much success, of deconstructing the generated URL into components, so advanced users do not have to use the search bar at all [77]. For this project though, it was decided to emulate the typical user, in an attempt not to break Google's TOS. The browser provided by HtmlUnit allows for form submission, so by navigating to Google, entering the desired search query and submitting the form results could be collected. There were some performance costs by doing this, yet these seemed negligible even when processing multiple users.

The code for Google querying can be seen in figure 6.3. Initially this code

```
private GoogleResult performSearch(int remove) throws IOException
{
    HtmlPage google = webClient.getPage("http://www.google.co.uk");
    HtmlForm searchForm = (HtmlForm) google.getElementByName("f");
    HtmlTextInput searchBar = searchForm.getInputByName("q");
    HtmlElement button = (HtmlElement) google.createElement("button");
    button.setAttribute("type", "submit");
    searchForm.appendChild(button);
    searchBar.setValueAttribute(generateQuery(remove));
    HtmlPage searchPage = button.click();

    GoogleResult result = new GoogleResult();
    String html = searchPage.getWebResponse().getContentAsString();
    Document doc = Jsoup.parse(html);

    Elements check = doc.select("div.med");
    if (check.size() != 0)
        return result;

    Elements elements = doc.select("div.kv");

    for (Element ele : elements)
    {
        String removeExtra = ele.text().replace("␣-␣Similar","");
        removeExtra = removeExtra.replace("␣-␣Cached","");
        result.addResult(removeExtra);
    }

    return result;
}
```

Figure 6.3: Searching Using Google (GoogleQuery.java)

was much shorter, however when using the tool there were some major issues. For instance, occasionally a slightly modified version of Google would appear where the submit button had a different id; to combat this a new button is now created and appended to the form, to guarantee submission. Occasionally, the query would be overly complex and return zero results, so a 'remove' parameter had to be added, which determines the number of attributes of the query to be used in search. An extension to this would have been the ability to judge the correctness of a search based on the number of parameters used, yet this was never implemented. Whilst not robust and with room for refinement, the search tool was capable enough for the requirements of this project.

To locate LinkedIn the following, where available, were added to the Google query in order of importance:

site:linkedin.com/in/ - *Ensured only LinkedIn profiles would be returned*

Full Name - *Results must contain the full name of the target user*

Most Recent Education - *Typically university, always displayed on a LinkedIn profile*

Most Recent Occupation - *Likely displayed on a LinkedIn profile*

Living Location - *Sometimes displayed on a LinkedIn profile*

Using only these four data points to locate profiles seemed sufficient, as it was highly unlikely two profiles shared the same for all of them. Only the top three results would be used in the investigation for any search query, as any more were likely not relevant. In many cases, it was found that the results from this query would often contain both the target profile and friends of the target profile; this is discussed in greater detail in chapter 7.

6.3.2 Extending Scraping to LinkedIn and Instagram

Both LinkedIn and Instagram share similar policies to Facebook when it comes to access to user datao on their API; they both require tokens private to the user. With this is mind, it makes sense to extend what has been built so far to collect the data from these additional social platforms. Yet this seemed unecessary, both platforms have only specific data available (in occupation history and photos respectively). It can be assumed that, once links to these platforms have been obtained, this data can be acquired trivially, and would not add much in to the searching power to what has already been used to locate them in the first place.

Despite this, some minor attempts were made to collect basic information from these sites, however problems occurred early. LinkedIn regularly demand that users on their site login to access the data, and their fake profile detection is more advanced than Facebook's. The investigator did not want to use their own personal profile, and although an alternate profile was made it was never used. With Instagram, almost all the content generated on the page is via AJAX calls. There were endeavours to tune the HtmlUnit parameters to handle this amount of AJAX, nonetheless this was abandoned before a solution was made.

6.4 Accessing Twitter

Mentioned numerously throughout this report, Twitter provides open API for developers to extract user data. This comes in the form of REST, where a developer connects to a specific REST endpoint to receive data in the form of JSON. In order to authenticate the developer they are provided with an OAuth token and both a private key as well as a key related to their personal Twitter account. To hide the complexities of using these REST services, the Java library twitter4j was used, which provides access to the data in a developer friendly format. For instance, users are contained in a 'User' class, with all their information accessible via getters.

To start with, the developer had to learn how to use the tools available in twitter4j. With clear documentation and the use of self-explanatory naming conventions this was achieved quickly. There are also extensive resources on help forums, for example Stack Overflow, that could be used as reference when trying to accomplish specific tasks [56]. One limitation of the Twitter REST APIs is that they are limited to 15 requests per 15 minutes for each separate request type. There are some ways to bypass this - it is possible to have multiple OAuth tokens - however for this project simply waiting until the requests

```
for (Status status : statuses)
{
    if (!CodeUtils.checkDate(today, lastYear, status.getCreatedAt()))
        break;

    if (!status.isRetweet() && status.getText().length() > 80)
        largeTweet.add(status.getText());

    int hour = status.getCreatedAt().getHours();
    times[hour]++;

    Place loc = status.getPlace();
    GeoLocation geo = status.getGeoLocation();
    String source = status.getSource() == null ? null : Jsoup.parse(
         status.getSource()).text();

    if (source.equals("Instagram"))
        System.out.println(status.getSource());

    if (loc != null)
        if (hour <= 8 || hour >= 19)
            if (home.containsKey(loc))
                home.put(loc, home.get(loc) + 1);
            else
                home.put(loc, 1);
        else
            if (work.containsKey(loc))
                work.put(loc, work.get(loc) + 1);
            else
                work.put(loc, 1);

    if (geo != null)
        geoTagged.add(status);
}
```

Figure 6.4: Extracting Data From Tweets (TwitterHarvest.java)

refreshed sufficed. Generating multiple OAuth tokens through having multiple applications associated with Twitter is risky, since it break Twitter's TOS and therefore could lead to termination of all the accounts.

Initially only the contents of tweets were analysed for basic information extraction. The last 200 tweets - up to a year ago - were requested and iterated through. Some key information was then taken: all locations of tweets and what time tweets with locations were posted; any geo-tagged coordinates; and any link to Instagram. Tweets sent between 8am and 7pm were deemed to be 'at work', therefore a location of work could be inferred, and vice versa for home location. The code for this can be seen in figure 6.4. It was found that most people do not location tag tweets and consequently this tool was relatively useless, yet when it did work provided strong results.

A somewhat similar approach was taken for like analysis, where the last 200 likes - up to a year ago - were taken and iterated over. However, for likes, each

```
private RandomDecisionForest createTree() throws IOException
{
    final BufferedReader br = new BufferedReader(new FileReader("testData
        .csv"));
    final List<ClassifierInstance> instances = new LinkedList<
        ClassifierInstance>();

    String line = br.readLine();
    while (line != null) {
        String[] splitLine = line.split(",");
        Double[] values = new Double[splitLine.length - 1];

        AttributesMap attributes = AttributesMap.newHashMap();
        for (int x = 0; x < splitLine.length - 1; x++)
            values[x] = Double.valueOf((String)splitLine[x]);

        values[2] = Math.log(values[2]);
        values[3] = Math.log(values[3]);

        for (int x = 0; x < values.length; x++)
            attributes.put(headings[x], values[x]);

        instances.add(new ClassifierInstance(attributes, splitLine[
            splitLine.length - 1]));
        line = br.readLine();
    }

    return new RandomDecisionForestBuilder<>(new DecisionTreeBuilder<>())
            .buildPredictiveModel(instances);
}
```

Figure 6.5: Creation of Random Decision Forest (FriendPredictor.java)

individual 'tweeter' was aggregated and stored in a single object. From there a machine learning technique could be applied to the aggregated data, for a prediction on how strongly affiliated that user was to the target user. To begin a set training data was required. Five users from the test set were asked to rank their liked users on a scale of 1 to 5. The classes were as followed:

1 - *The user is unrelated*

2 - *The user is a favoured celebrity or online personality*

3 - *The user is a 'real life' acquaintance*

4 - *The user is a 'real life' friend, coworker, peer*

5 - *The user is a very close connection*

The theory was that the liked users properties would cluster in such a way that supervised learning could predict their class. The parameters that were decided on to predict users are: is the target user following the liked user; vice versa; the number of people following the liked user; the number of people

the liked user follows; the number of likes from the target user to the liked user; and the average number of likes of these tweets. A machine learning library was required as implementing any technique from scratch would have been needlessly time consuming. QuickML was the library of choice, due to it's large suite of tools, even if the documentation was somewhat lacking [60]. Available already on QuickML was an example of using a random decision forest to predict attribute data, which had to be slightly modified. Shown in figure 6.5 is the creation of the forest. A QuickML default random decision forest uses 8 separate decision trees of max depth 5. Whilst alternatives, for example just using a standard decision tree, were tested the initial random decision forest achieved the only promising results. Predictions made were outputted into a text file for later analysis.

Some time was dedicated to the tinkering of the parameters of the random decision forest tree. Originally the complete number of followers and followings of users were inputted, however this was changed to be a natural log of the value for each, in order to reduce the spread of data, which originally could be anywhere between 1 and a million. By adding the following or follower boolean parameter, being able to differentiate between classes 1 and 2, as well as 1 and 4, became far more accurate. It seemed that the following boolean being false was the primary indicator of users of class 1. Since every new run of the Twitter analysis took a day of time, development was particularly slow.

The final step to Twitter analysis was examining follower relations. As a follower network is a large graph a Java graph library was required, and for this JGraphT was used [37]. This library provides not only the data structure to create and store graphs but also many algorithms that can applied onto a graph; of interest being the clique algorithms. To begin with, the followers and followings of the target users are compared such that only users that exist in both remain. Discussed in section 5.3, this should have reduced the number of nodes in the graph down to 22.1% of the original. These links are then added to an undirected graph, such that every vertex is now only connected to the target's vertex. This is then repeated for every linked user, however no new vertices can be created, so only connections to users that link with the target can be made. What is outputted from this is typically a well-connected graph with the target user as the 'centre' that all of nodes connect to.

From this graph two things can now be achieved. The graph can be displayed so that any human can make their own inferences. To do this third-party graph displaying software was used, Gephi, that can be seen in figure 6.6. Gravity was applied on the graph so that connected users attract and unconnected users repel. Of note are not the well-connected users, but in fact the users who are not well connected. While there will be well connected users that are important to the target user, those who are part of smaller clusters are unique to the graph, so may have a special connection to the target - this is discussed more in chapter 7. Another technique which was examined, but not fully explored, was clique analysis. Using the Bron-Kerbosch clique detection algorithm to identify the set of all maximal cliques - subgraphs were all nodes are connected and no new node can be added such that the first property holds - it may be possible to identify the core clique of friends who are part of the most maximal cliques [64]. The Bron-Kerbosch algorithm is a recursive backtracking algorithm, which treats the graph as 3 sets, R, P and X, where P and X are disjoint such that P union X contain all the vertices connected to every vertex in R. After many recursive

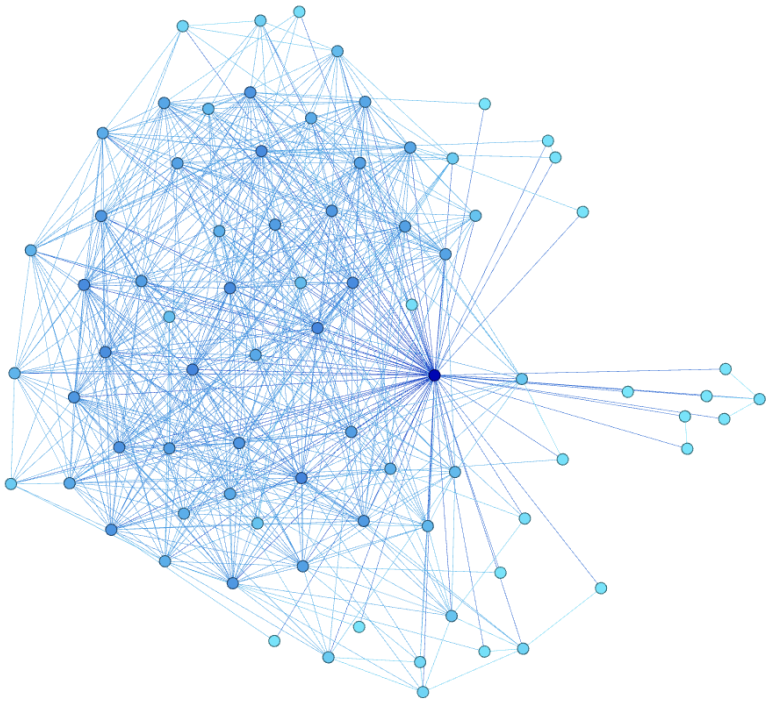


Figure 6.6: Twitter Graph for a User Displayed Using Gephi

calls, P and X become empty and therefore R is a maximal clique. Even though there was some progress made in using maximal cliques, ultimately all tested users' networks contained too many cliques to tweak the algorithm effectively.

6.4.1 Extending Like Analysis to Instagram

Instagram - much like Twitter - has a similar like system, yet when the number of likes exceeds 10, a regular occurrence, this is no longer retrievable. What can be accessed from Instagram are comments, which also occur less frequently than likes. Even though Instagram was not tackled during this project for the reasons mentioned in section 6.3.2, the investigator sees no reason why the supervised learning technique used for Twitter likes could not be extend to Instagram comments. To achieve this the photos of the target user would be examined and the people who commented on it would be compared. The following parameters could be used: average number of likes on photos, average number of comments on photos, number of photos commented on and number of hashtags used in photo description. The final parameter is because hashtags on Instagram greatly increase exposure, and therefore the more hashtags used the higher chance somebody unrelated to the target would comment.

To summarise, this chapter discussed how the implementation of the tools used throughout the investigation occurred. Covered were the libraries used, a hand-

ful of examples from the codebase and comments on the decisions made regarding specific algorithms. In the following chapter the results from the investigation using these tools will be discussed, and some inferences that have been made.

CHAPTER

SEVEN

RESULTS

During the investigation many interesting discoveries were made, and certain inferences can be deduced from them. This chapter covers what was discovered, how these affected the focus of the project, the implications of said findings on user privacy and the recommendations proposed to users and platforms to protect user privacy. It will also look at how the developed tools were used as well as how the tools could be expanded to further solidify claims made and uncover more data.

7.1 Alias Analysis

The first tool created was the alias comparison tool using Levenshtein distance. It was expected that there would be some similarity between different alias names across platforms, due to people wanting to be recognisable when communicating to the same friends on various social medias. Although possible to change both the Facebook and LinkedIn aliases, Twitter and Instagram use screen names over the real name of the user, so therefore were a larger focus for this section of the investigation.

Displayed in figure 7.1 are the shared aliases between platforms. As expected, the platforms which share the same alias are nearly only Twitter and Instagram, where 11 out of 30 users use the same screen name on both profiles. This trend continues when examining the Levenshtein distance. Of the remaining 19 profiles on Twitter and Instagram, 3 have 'very similar' aliases, 8 have 'similar' aliases and only 4 have definitively unique aliases on both. From this, it can be assumed that once a Twitter or Instagram profile has been found the other can be reached with relative ease using search engines. The other results from using the string comparison tool were: Facebook and Twitter were at least similar on 8 accounts; Facebook and Instagram were very similar in five instance and similar in five; LinkedIn shared no similarities beyond exact matching. Even though the Facebook alias matched or was close to the other platforms, since the final percentage is relatively low, the link between profiles cannot be established reliably.

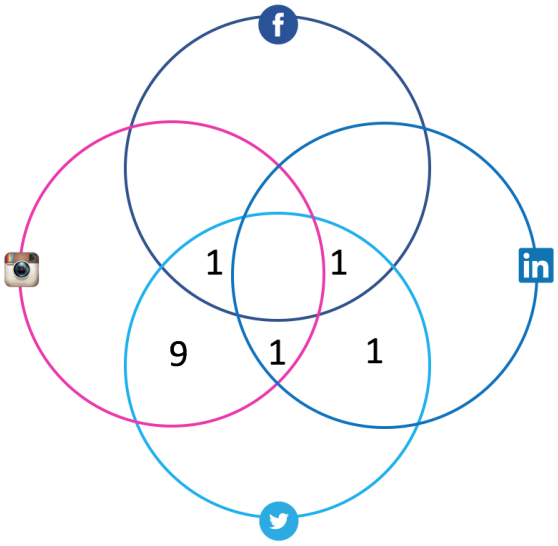


Figure 7.1: Social Platforms That Share the Same Alias

7.2 Facebook Data

Facebook is the world's most used social media and was the starting point of major development. All users in the test set had Faceboook and as mentioned in section 6.3, the first online tool created scraped information off the profiles on the platform. These results were stored in text files which were analysed by bash scripts, counting and comparing the various data pieces collected. In this section the following will be discussed: which were the most common details shared; the possible exploitation of users from what was found; and general statistics about the findings.

7.2.1 Image Availability

To begin, the number of photos of the target user available was analysed. The scraper worked by looking for the commonly used 'Profile Pictures' album, which stores all the public profile photos of a Facebook user. As these images are used to identify the person to their connections, generally they contain the user's face at a time relatively close to the upload date of the photo. This not only allows attackers to impersonate the target user by harvesting recent images, but also gives a history of the user, and photos from further back in time would be harder track whilst also providing a potentially greater threat. Although not required, the tool returned links to the photos which could have been used for image analysis.

Across the fifty users tested the average number of photos returned was approximately 10 - surprisingly high. Despite this, two users did not have the easily accessible 'Profile Pictures' album, and many had only a small handful of images. A chart of the number of photos against frequency can be seen in figure 7.2. As shown, the majority of users in fact provided 10 or less photos, and a handful of users brought the average higher. The maximum number of photos

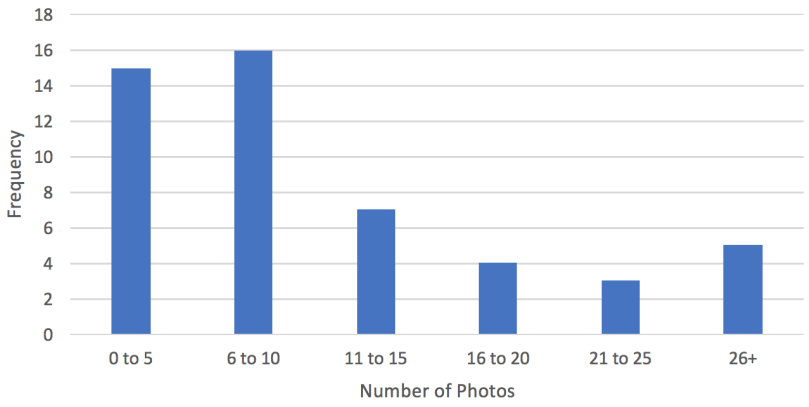


Figure 7.2: Chart of Number of Photos Scraped Against Frequency

scraped from one user was 28. Obviously not all of these were recent, with the eldest photo going back 8 years, however as previously mentioned even old photos add value to learning about an individual. It's worth noting that current age seemed to have no effect on the average number of photos of users, although male users had less photos than female users, with 7.8 to 13 respectively. There were many outliers in both cases though so it is difficult to be comfortable in this assumption.

7.2.2 Education and Occupation

As they were used later when attempting to connect to LinkedIn, collecting the educational and occupational history of a Facebook profile was critical to the investigation. Similarly to the photo scraper, this information was taken from the target profile and stored in a text file to be examined via bash scripts. The output was human readable, yet could be extracted via regular expressions to count the data. This data can be found either on the main profile page of user or in the about section under the heading 'Work and education'.

	Zero	**Only 1**	**Only 2**	**3 or More**
Number of Educations on Profile	11	16	10	13
Number of Occupations on Profile	28	11	8	3

Table 7.1: Educations and Occupations on Facebook Profiles

The number of each attribute was split into four buckets, 0, 1, 2 and more than 2. The results can be seen in table 7.1. As shown, there is roughly a 50% chance of finding some occupation history, and a higher chance of finding education history, for this test set. However, since the test set was heavily weighted with students, it is likely that using users with professional jobs would see an increase in occupation and a decrease in education. Only 6 users were found that did not have any work or education history on their internal Facebook profile. Of the 39 users who had an educational history present, 35 of them shared their university as one of the institutions. In fact, only 2 current students out of 36 did not share their university information. Breaking this down further,

as a reason for capturing education was to connect to new platforms, out of the 35 people who shared their university, 23 had a LinkedIn profile.

7.2.3 Searching for LinkedIn

The primary attributes required when searching for a LinkedIn profile were previous educations and occupation roles. Google was used to locate the profiles since it is the most commonly used search engine in the world by some margin. As discussed in the previous section, 23 of 29 people in the test who had a LinkedIn account shared their university on their internal Facebook profile. Of the remaining 6 users, only 2 provided any educational or occupational history at all. However, those 4 who provided limited data had distinct enough names that, on the smaller platform in comparison to Facebook, these people could be located via name alone and identified by images.

For every member of the test set who did provide either a university, high school or job placement, their LinkedIn account using the parameters described in section 6.3.1 was at worst third placed in the Google search results, and was first returned in over 60% of cases. This means it can be safe to assume that given a Facebook profile locating the corresponding LinkedIn profile is almost guaranteed. The only case where potentially this will not be accomplished is where the Facebook profile provides at most only a name; with no image or additional details to verify the user on the alternate platform the link could not be established.

Tested with a handful of users was the reverse connection; transitioning from LinkedIn to Facebook using only the details given on the former. Due to Facebook rarely returning a direct link to profiles, this was accomplished far less often, with only 2 in 10 users tested being found. Arguably users would be more concerned with protecting this connection over the Facebook to LinkedIn link, since they would not want potential employers trawling through a personal social media profile, where as they are unlikely to be troubled if friends find their professional profile. On the other hand, LinkedIn does contain a detailed job history of a user, which could be taken by potential attackers for a variety of purposes.

7.2.4 Additional Data Collected

A handful more attributes were scraped from Facebook profiles and were used by the investigator for checking details or linking to new platforms. While nothing stood out as a definite connection between social medias, often the occasional detail would be shared across profiles. One to note was every Facebook was accessible by the toolkit, so therefore no user in the test set had maximum privacy settings. This setting prevents anybody from viewing a user's internal profile without the user initially sending a friend request. Due to the purpose of social platforms being to connect people it seems counter-intuitive to enable this feature; however, it was surprising that not a single user had that level of awareness.

Hometown and current location were both relatively commonly displayed on profiles. In total, 68% of users shared either one of these attributes, and 45% of users showed both attributes. This shows people value them the same in terms of privacy, although arguably knowing the current residence of the user

is far more damaging. Public records available in libraries such as the electoral roll could reveal the exact address of users once the general area is known. For linking to other platforms, the current location was used to some success when finding Twitter accounts, however with only a handful users tested and fewer successes this link is tentative at best. What can be said is if trying to separate profiles users should simply reduce the amount of information made available.

Relationships were the final characteristic collected from Facebook profiles. For family connections, 32% of profiles gave at least one, and for a romantic relation, 25% had one listed. To note, it is difficult to affirm the correctness of these associations as it is common on Facebook to list friends as both relatives or partners. These were used in the same way though, since a strong bond can be confirmed either way. In the cases where it was difficult to find a profile on a new platform, the relation links were used. These friends are maintained regardless of medium, so by discovering a friend's additional profiles the target user can be found through them. This was covered in the project presentation, but to maintain test user privacy in this report will be divulged further.

7.3 Twitter Data

The investigation into Twitter was arguably the more interesting and technical half of the project. A major constraint though when collecting and analysing data was the rate limits on the open API. Minor tweaks to tools led the code being run overnight, and if the code failed or no improvement was shown additional tweaks then had to wait for the next night to run. Additional, for the graphing tool, only a handful of users could be analysed due to each user taking a whole night rather than the entire test set taking a whole night, since some users had many follower connections. This caused a large gap in knowledge to remain when it comes to follower review.

7.3.1 Tweet Content

The first tool created for Twitter examined the contents of a user's tweets. Large tweets - those that exceed 80 characters - were taken and stored in a text file. This could have led to both sentiment and interest analysis, however these were not implemented due to them already existing, as mentioned in chapter 2, and so not producing new results. An average Twitter user in the test set had 25 large tweets, although 4 were protected. Removing these users, so only analysing the active Twitter accounts in the set, increased the average to 28, slightly under the amount Zheng et. al used for their identity detection, discussed in section 2.1 [81].

When trying to link to Instagram, 4 out of the 32 non-private users had a direct link to the platform in one of their tweets. This means at best 1 in 8 cases, additional to alias analysis, Twitter and Instagram can be freely moved between. One of these 4 users had an excessive number of Instagram tweets at 10, leading them to be particularly open, despite them having only moderately similar aliases on both platforms. The discovery continues to emphasise that Twitter and Instagram are two heavily related social medias, even though they are owned by two separate corporations.

Locations were rarer to find on Twitter than on Facebook. Only 3 out of the 32 users provided any location on tweets, and only 2 of those gave a geo-tagged location at the time. This shows it is difficult to determine user's location using tweets, which was a promising result. In one user's case their city of work and home could be deduced with accuracy, however this is a rare occurrence for a general user.

7.3.2 Friendships Through Likes

For the investigator, the most hopeful part of the project to locate new and compelling results was through the analysis of peoples' Twitter 'liking' habits to determine their friendships. As discussed in section 6.4 the supervised learning technique known as a random decision forest was used to separate relations between users into five buckets. The first stage was assembling a training set, a collection of parameters and matching classifications that have been provided by humans, so can be seen as correct. By passing this training set into the random decision forest the model would try and split the data into their classes based on the patterns in the corresponding parameters. To recap the classes were decided to be the following:

1 - *The user is unrelated*

2 - *The user is a favoured celebrity or online personality*

3 - *The user is a 'real life' acquaintance*

4 - *The user is a 'real life' friend, coworker, peer*

5 - *The user is a very close connection*

Five users agreed to be part of the training set, generating a set of 233 different classifications. Unfortunately, this was about a fifth of what was wanted, however with few compliances and both the time of the investigator and the time of the users being limited this was all that could be accomplished. What this led to when classifying relations was a large amount of ambiguity. For example, the prediction map for a typical connection had this appearance: {1=0.089, 2=0.132, 3=0.151, 4=0.331, 5=0.297}. This caused the average for a prediction to tend towards 3, away from the extremes of 1 and 5. To combat this, a majority rather than an average was used, so to analyse the effectiveness a confusion matrix will be used over a more traditional loss function. However, this was also not without its downsides, as using a majority restricted the classifications to their five discrete buckets without any concept of certainty.

One pattern in the training data can be seen in figure 7.3. This chart plotted the number of people a liked user followed, known as followings, against their number of followers. The actual numbers in both this chart and in the random decision tree were natural logged to drastically reduce the scope of the data. As shown by the chart, classes 3, 4 and 5 tend to cluster to the lower centre of chart due to them all representing general users of social media. Classes 1 and 2 spread much wider, with class 2 trending to the left, but with their average followers being much higher. With these classes representing celebrities this result is expected, as celebrities follow few people and yet have many thousands of followers. People who are not celebrities, however are still within class 1, are

seen in the upper right of the chart, due to them following lots of users in an attempt to generate more followers.

A second pattern is portrayed in the figure 7.4. This chart plotted the number of likes by the target to tweets of the liked user, against the average likes of said tweets. Classes 1 and 2 were removed from this chart, as well as any outliers, entries where either parameter was greater than 50, to isolate the difference between acquaintance, friend and close connection. As shown, class 4 and 5 tend to be towards the bottom of graph with fewer average likes, however can have any number of likes. Class 3 on the other hand typically has a higher number of average likes, and remains low when it comes to number of likes. There are more classification for 5 then any other class in this chart and in the training set in general, which is to be expected as users have a higher chance to 'like' their close connections.

	Predicted				
	Class 1	**Class 2**	**Class 3**	**Class 4**	**Class 5**
Class 1	12	4			
Class 2	4	26		2	
Class 3		3	4	6	2
Class 4		3	1	20	17
Class 5	1	1		3	19

Table 7.2: Confusion Matrix for Actual Classification Against Friend Prediction

Five users were randomly selected, who were not included in the training set, to be the test set for the system. In table 7.2 the confusion matrix for the results is shown. Two users had a much lowers ratio of true positives than the others, which is explained by them as being cause by their infrequent use of Twitter. Classes 2 and 5, which are the centre point for investigation as they uncover user interests and strong connections respectively, both have good true positive ratios, indicating the majority will be classified correctly. Whilst class 4 has just under a 50% true positive ratio, most the misses are for class 5. Despite this weakening the performance of class 5, the difference between the classes is a nuance so the error is understandable. Easily the worst performing class is class 3, however there few example of people liking class 3 users, therefore this does not affect the results as much. Class 1 has a reasonably good true positive score, and all misses are classified as the related class 2.

A major flaw of using this classification method is that the classes are subjective. This means between any two users their slight interpretation of what is a friend over an acquaintance or good friend changes. For instance, one user had a large number of misclassifications of class 4 as class 5, due to their profession establishing many friendships so it being harder to assign good connections. The subjective nature of the classes did not only affect testing but also creating the training set. Beyond simply broadening the definition of each class or discussing with each different user how to classify certain relations, which in some instances was done, it is difficult to fix this flaw, and may just be a downside of using supervised learning.

If you were to merge the classes 4 and 5, the number of true positives greatly increases. At this point the friendship analysis can be used to connect users on Twitter, even by using information on different platforms. One use of the like

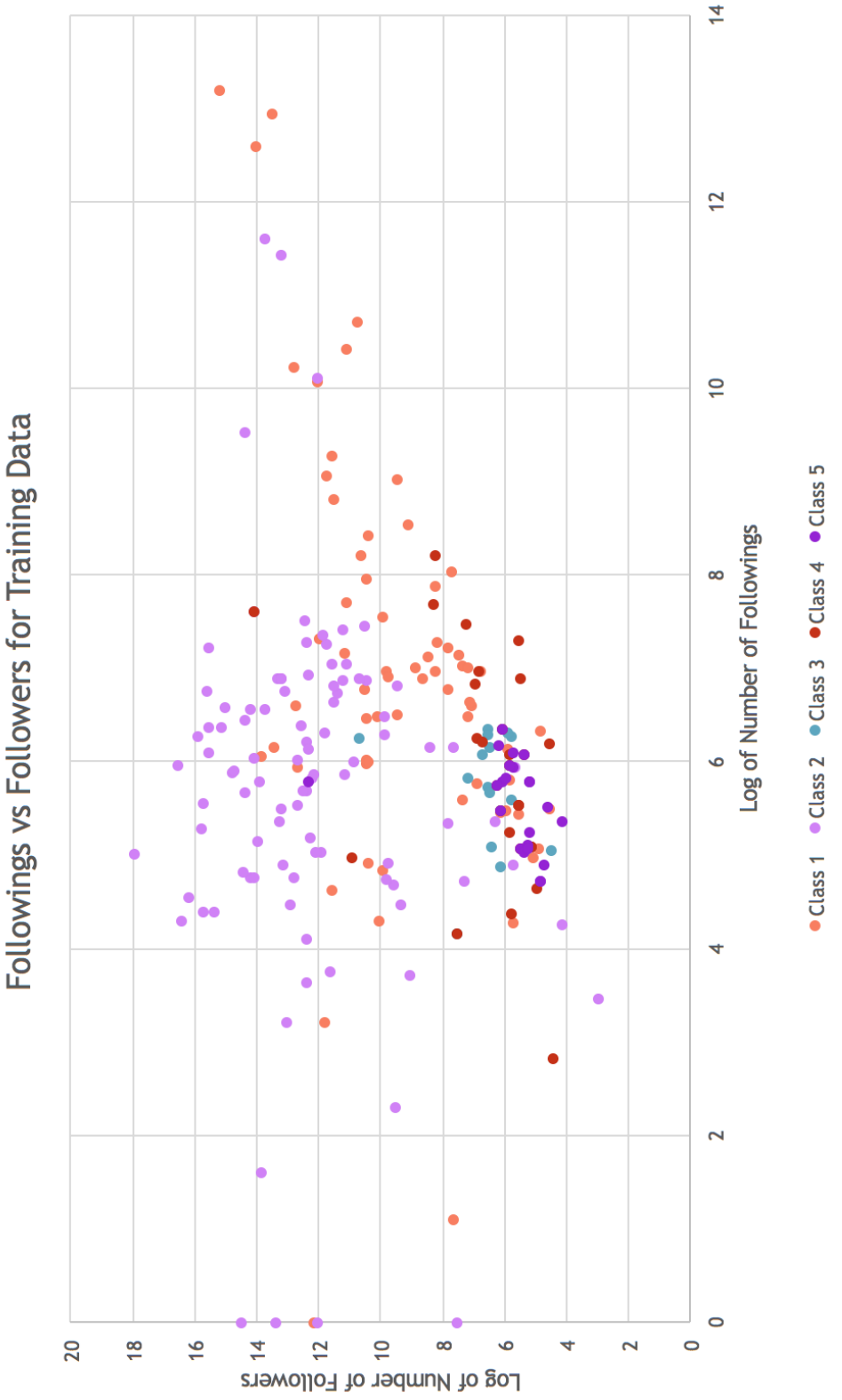


Figure 7.3: Classification of Users Plotted on Followings vs Followers

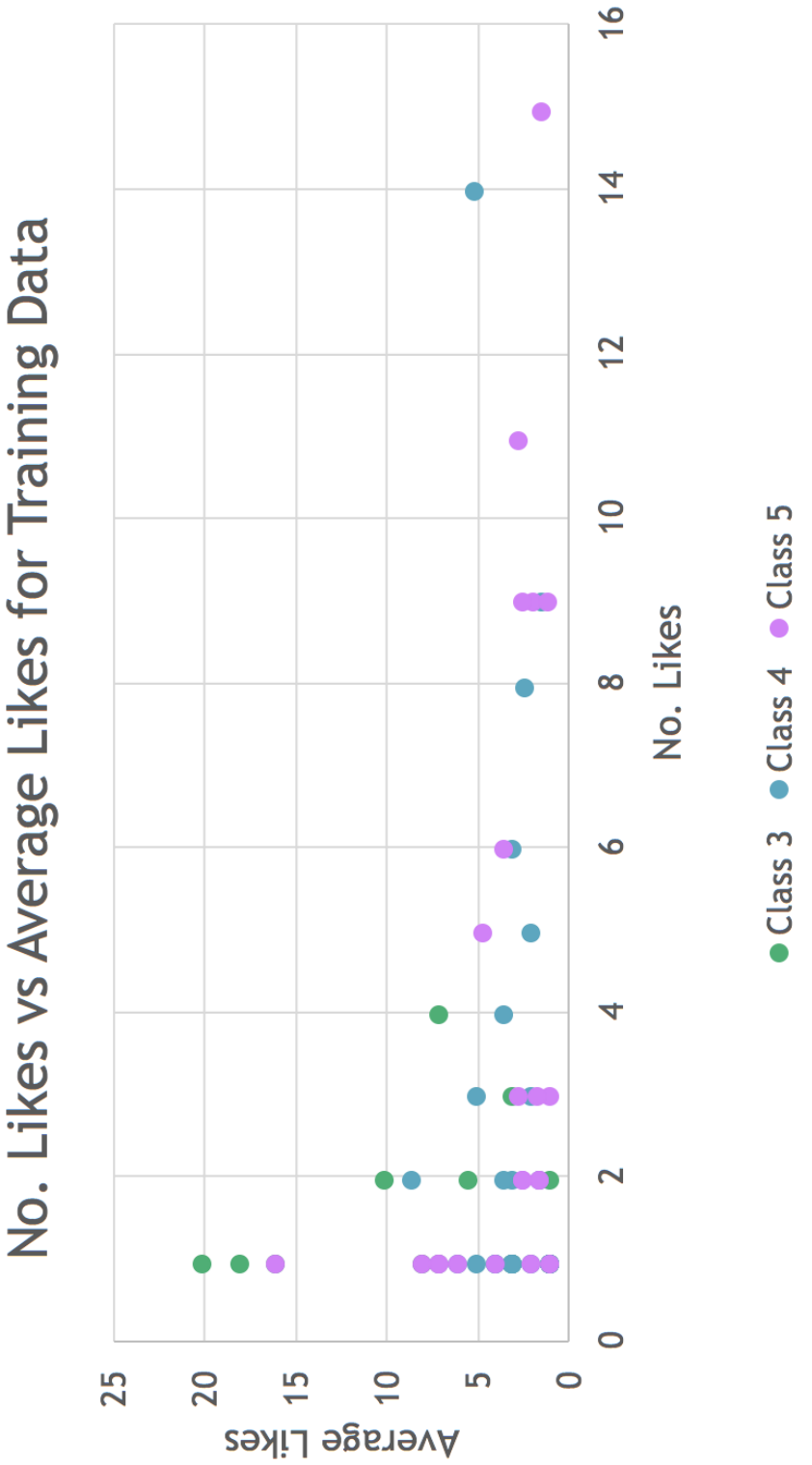


Figure 7.4: Classification of Users Plotted on No. Likes vs Average Likes

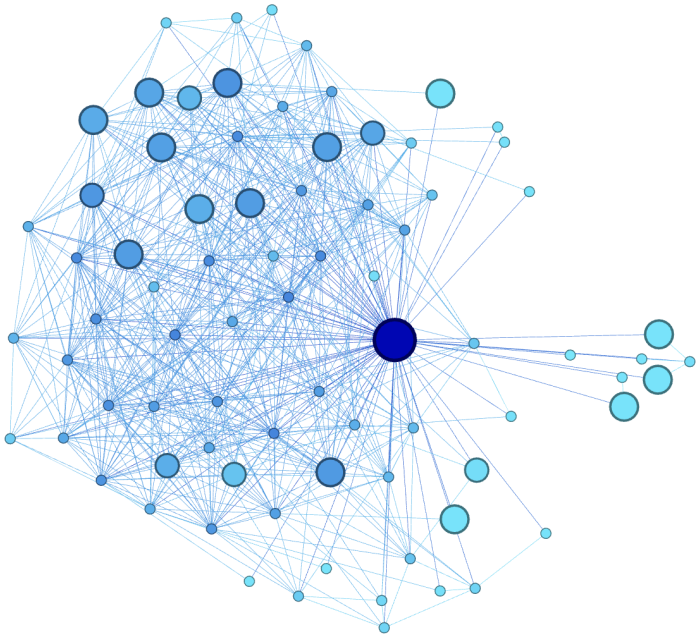


Figure 7.5: Twitter Graph for a User With Node Size Using Like Analysis

analysis was to find the Twitter profile of a person in a relationship discovered from Facebook. The target user's Twitter could not be located due to them having strong alias and information protection. Their boyfriend however could be found on Twitter, and with the expectation of the target user being classified as a close connection by like analysing the boyfriend's Twitter the target's Twitter was uncovered. Creating a system that combined techniques like this would be final the goal of the toolkit, but would require much work to be done to establish an architecture, and improvements to tools would be necessary, particularly using Google effectively.

7.3.3 Graphing Followers

The final, least polished, tool developed was the graphing of Twitter connections of the target user. Despite the graph being created successfully, it took a long time to process all of the followers, leading to development slowing down drastically. Some good results were uncovered though, particularly by using the features available on third-party programs that produce simple clusters to be analysed.

Figure 6.6 shows the unmodified Twitter graph of a user with anti-gravity causing clusters to form. In that graph, darker nodes indicate a better connected user. Intuitively, the more associated users would be the ones of interest considering they are structurally integral to the graph, yet it is the nodes with the smaller degrees which hold more interesting profiles. These are people unique to the target user, and are more important the larger the disconnected cluster. When applying the friend classification to the size of nodes the graph displays better results.

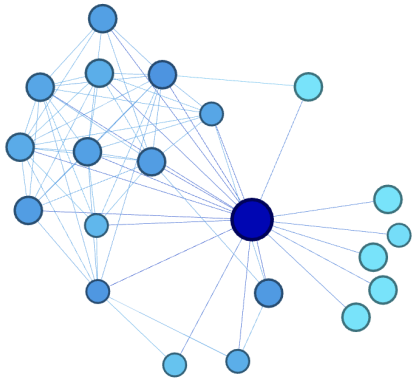


Figure 7.6: Twitter Graph with No Likes Users Removed

In figure 7.5 this improved graph is shown. The graph now appears to be able to be partitioned, with users in the groups to the top, bottom and right clearly being the most important due them being scored by the like analysis. Many of the users who are part of the largest clique do not score anything through like analysis, proving that being connected does not lead to importance. A possible reason for this is that having many connections may just be down to social media activity, as it is still likely these people are at least acquaintances for every other user. An updated graph with all users not scored in like analysis is shown in figure 7.6. This identifies the core clique to the user as well as starting links into the other friendship groups.

A possible extension to the graph and like combination would be analyse the cliques that contain high scoring users. The groundwork for this is already laid out with the Bron-Kerbosh algorithm featuring in the toolkit. By taking the maximal clique that contained the most users who score 4 or above a set of targets for future analysis could be made for any given user. From an attackers stand point, knowing which friendship group to infiltrate is core in impersonation.

7.4 Recommendations

As one of the primary purposes of the project is for educating the public on staying safe online, the main outcome from the results are a list of recommendations to both internet users and social media platforms into keeping their data safe. The following recommendations have been fully justified by the results outlined in this chapter, yet depending on the purpose of the platform or the intent of the user some may not apply to all.

7.4.1 User Recommendations

These recommendations are for users to keep their profiles separate as well as the likelihood of a successful attack on them, whether impersonation or coercion, minimal. They mainly focus on preventing the link being established between

different accounts on different social platforms:

Turn on the protected mode Twitter setting

Whilst in some people's eyes this would be hindering the purpose of Twitter, if you are using the website for personal reasons alone, that is to share with friends and family, then there is no reason for your Twitter to be unprotected. As shown via the Twitter analysis, the open API that Twitter provides allows for a multitude of ways to attack a user with relative ease. More so, the data provided comes from the less obvious sources on the platform, in the case of like analysis some people are even unaware that likes can be displayed to other users, let alone friendships be deduced from them.

Hide relations to other users as best as possible

Despite not being covered entirely in the report for privacy reasons, one of the main exploits discovered for finding new sources of data from privacy conscious targets was to use their closest connections. Through Twitter like analysis in combination with the Facebook relations page, accounts could be linked with any shared data between them bar a single friendship. Showing off relationships or family connections leads to the links being revealed, and should be avoided in nearly all cases. Additionally, there is no value in presenting the relations on profiles, since in most cases it will not be a defining enough feature for friends to use to find the profile.

Use a different alias per account

A simple Google search suffices when using common aliases among different profiles to locate new sources of data. By altering the alias or screen name by at least 5 characters the chances of Google linking the two accounts become drastically lower. On a similar note, using relatively common screen name or variant of such also helps mask the link between different profiles, at the expense of identity. The number one way of connecting people between their Twitter and Instagram account was using the alias of one and attempting a handful of variants, addition of number, full stop or underscore, on the other platform.

Avoid educational and work information

Some platforms, with the obvious example being LinkedIn, give great benefit to listing occupational information. There is some warrant to showing at least the current work placement on a Facebook profile, for instance to find other users from that institute, but little to no need to display this data publicly on other social medias. In a handful of cases, Twitter and Instagram connections were found using this data, along with other profiles from websites not covered by the scope of the project.

Each of the above recommendations have counter-arguments that have been mentioned. However, implementing these should give little chance that two profiles can be linked on different social platforms, if that was the users goal.

All are also within reason as they do not hinder the use of these social medias beyond the risk to reduce minor friendship connections.

7.4.2 Platform Recommendations

The following recommendations are directed to developers of various social medias, with a goal to be trying to protect users from the start. They use both the successes and the problems faced during the investigation, as well as the inconsistencies noticed between platforms:

Do not allow open APIs

Even though it's great for developers and researchers to have open access to user data, the level of potential exploitation is too high to justify. The only two platforms thoroughly analysed were Facebook and Twitter, due to the amount time it took to produce a useful scraper for the former. With the latter providing an open API and with the Java libraries already available it took little to no time to set up and begin analysis. Twitter do provide the protected mode for users to opt-out of this API, however this is not promoted enough nor is it the default.

Improve default security

All profiles on each platform are by default not at the best security. Both Twitter and Instagram have an opt-in protected mode which is both cases is not promoted strongly enough. Facebook allow for enough data to be accessed by default, i.e. relationship status and current education/occupation, to at the very least link to one other platform. Finally, all LinkedIn profiles are nearly fully accessible, and whilst they attempt to enforce a login, this can be bypassed. By enforcing stricter privacy settings by default, social medias allow users to make their own mistakes rather making oblivious users' mistakes for them.

Crack down on fake profiles

This recommendation serves two purposes. The first is to prevent data being harvested at all. Both Facebook and LinkedIn provide far greater information to a logged in user over an external one, an exploit used by the Facebook scraping tool. Preventing fake users allows this to continue to be acceptable rather than a pointless attempted block on general web crawlers. The second purpose is to stop impersonation once data has been collected. In over half of the users in the test set's case it could be said that a moderately successful impersonation could take place, at least enough to fool less aware friends or acquaintances.

Unlike the user recommendations, at least the last two of these recommendations are inexcusable. There is absolutely no need to not enforce better security on users, and the only setback for finding fake profiles is that it is difficult.

This chapter marks the end of the investigation section of the report. The final

three sections reflect upon the management and successes of the project, as well as a look forward into what can be done in the future.

CHAPTER

EIGHT

PROJECT MANAGEMENT

With a project not based entirely around development, it is crucial to manage time and resources efficiently to ensure that progress is being made since results might not be immediately apparent. The project aims changed during development causing initial scheduling to become irrelevant. Despite this, the project stayed on track through a series of smaller goals set regularly. In this chapter, the flexibility of the project, the development methodology, the final timeline of the project and the various tools and techniques deployed to keep the project moving forward are discussed.

8.1 Flexibility of Objectives

As mentioned in section 4.3 the objectives of the project changed drastically a small way into development. Originally set in the specification was the requirements for a fully-fledged system. That combined some of the tools created during implementation into one application. Whilst this may still have potential as a secondary venture following the current project, it became apparent that implementing such an ambitious system was difficult to achieve in the given time constraints. This caused a complete overhaul of not only the objectives but also the overarching aims of the project. Throughout development these objectives continued to be flexible and new discoveries led to more being added. The final set of objectives are those listed in chapter 4, however the original list was heavily focused on single users and their wider web footprint not only on social media. By gradually specialising the project into social media alone and by beginning to examine the effects of links between users, a greater understanding of the key issues was obtained, allowing recommendations to be thoroughly justified.

8.2 Development Methodology

Due to necessary flexibility of the objectives, development had to use an agile methodology. The focus was solely on one tool at any one time, however discoveries made by using that tool would typically decide the design of the next tool to be implemented. Regularly meetings between the developer and

the project supervisor to discuss findings also helped mould the direction of the project. In most cases, a rough plan of how each tool would function would be known before development, yet this was often subject to change due to unforeseen circumstances, particularly with the Facebook scraping tool due to the many barriers put in place by Facebook that had to be worked around. The developer appreciated the ability to be flexible in both design and development, considering this style is what they had already been exposed to in previous work. Attempts to use any other methodology, such as waterfall, would have been catastrophic for the project, due to the mass changes that were made after the initial specification.

8.3 Project Timeline

A final rough timeline of the project can be seen in figure 8.1. The original timeline from the specification is now completely redundant due to the changes to the project aims. It is difficult to be certain with the various dates of the project since the agile approach caused many areas to overlap. Also, the majority of time spent during development was researching and discussing possibilities over actual implementation. Along with this there was some time dedicated to the tinkering of parameters for the random decision forest, as well as waiting for users to process in overnight runs. This can be seen in the timeline during the Twitter section of development. A lot of time was also dedicated to prior research for the project, as this determined many of the objectives.

8.4 Management Tools and Techniques

The project depended on various tools for both development and the managing of documentation, communication and storage. Also, along with the aforementioned agile approach to the project, there were techniques used to prevent failure should an issue arise. This included weekly meetings with the supervisor of the project to discuss problems as they appeared. Ultimately it was these meetings that led to the shift in focus of the project, and the flexibile attitude of the investigator could also be seen as a managerial ability used. With regards to the technologies in the project, never had the developer used any of the third-party libraries implemented into the tools, which led to the need for a large amount of documentation and online forum reading to learn how to use each library to its maximum potential.

8.4.1 Development Tools

In figure 8.2 a visual list of tools used throughout development are shown through their logos. Whilst it is difficult to classify a programming language as a 'tool', Java was obviously a crucial dependency of the project since it was the language used for the majority of programming. This choice was due to the developer's comfort in using the language, so little extra learning was required. Maven was a great help during development as it allows for nearly any open-source Java library to be imported and used with ease [2]. Despite never using the tool before, the programmer found a handful of resources online that provided a strong enough understanding of Maven, at the very least for

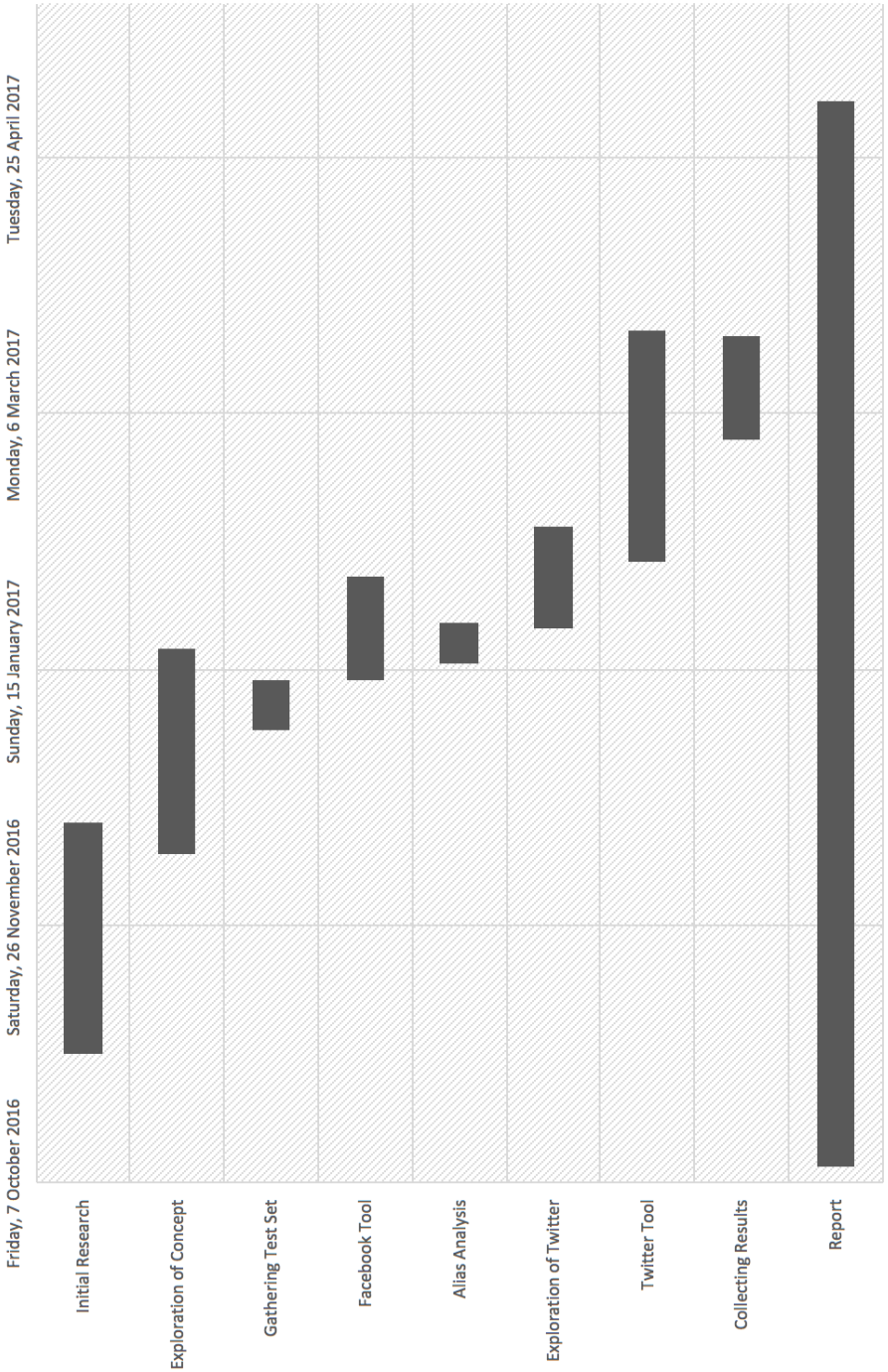


Figure 8.1: Final Rough Timeline of the Project

Figure 8.2: Logos of Development Tools Used

the purposes of this project, although the tool was never used to its maximum potential. In a larger software engineering project, the future potential system in mind, the organisational features of Maven may have to be used. Atom was the text editor used for development, and without the many extensions used progress would have been far slower [3]. The ability to handle nearly all editing, compilation and running within Atom cut down on all the time wasted switching between programs, which whilst minor, builds up after a long period of development.

8.4.2 Management Tools

Figure 8.3: Logos of Management Tools Used

Besides development there are major parts of the project that had tools to assist them; these are displayed in figure 8.3. This document is ultimately the highlight of the project, and by using the features in LaTeX the document produced is of a high standard. To edit LaTeX files Sublime Text was used, as this allowed for multiple tabs to be open and the interface was to the investigators comfort [69]. Whenever indirect communication was required with the supervisor, the online instant messaging tool provided by Facebook, Messenger, was used. As previously mentioned in chapter 6, GitHub was the online cloud service used to back up both the code and some of the data collected [21]. This prevented the project grinding to a halt should the main device used for development fail. For backing up the documentation of the project, Dropbox was used, as this is what the supervisor was comfortable with.

8.4.3 Risk Management

With any large project that spans over a long period of time there many risks that should they occur could be catostrophic for the results of the project. To prevent as much damage as possible should these dangers happen numerous measures were enforced at all stages of the project. The risk and their defences

are the following:

Complete failure of main device

The entire project, research, development and documentation, took place on a single machine. If this machine was to break and become unusable there was the potential for irreversible damage to the project. To prevent this all the developed code, some of the less private data collected and all documentation was backed up regularly on two cloud services, GitHub and Dropbox. Should the machine be damaged, all of the backed up data could be restored on any new device.

Facebook issuing cease and desist

There was the chance that during the use of the scraping tool that Facebook would ask that the tool no longer be used due to minor violations to their terms of service. While many counter measures were put in place to try and abide by the TOS, since it is a legal grey area and there could be a variety of interpretations, Facebook would have been within their rights to ask for progress to be ceased. At this point, all users of test set would have been required to provide their Facebook access token so Facebook's REST API could be used in place of a scraper.

Changes to third-party libraries

All of development relied on a variety of third-party libraries. At any point, there was the chance that one of these tools would stop being available or change the product would change significantly. If this was to occur, alternates, such as using the REST APIs of social medias or using a different library with a similar purpose, were available. No third-party software used had unique purpose, for instance there are many machine learning or graphing libraries available.

With the end of this chapter the entire details of what the project achieved and the methods used to do this have been discussed. The final two chapters examine the success of the project, what could have been improved and the future endeavours available.

CHAPTER
NINE

EVALUATION

All areas of the project, research, approach, implementation, results and management have now been discussed. By comparing what was achieved against the objectives laid out in chapter 4, the issues mentioned in chapter 3 and the overarching aims from chapter 1, a comprehensive analysis of the project's success can be completed.

9.1 Investigative Objectives Evaluation

In section 4.1, a list of objectives targeted at what the investigation was trying to achieve was given. These mainly relate to the results from chapter 7 and the initial approach to the investigation discussed in chapter 5. Tables 9.1 and 9.2 shows each objective in turn with their objective number, a success measure for the given task and a brief description to justify the rating given using references to sections of this document that go into further detail. Two of these objectives, I7 and I8, were deemed only moderately successful due to the groundwork being laid, yet further research being required to achieve exactly was stated.

9.2 Technical Objectives Evaluation

In section 4.2 a list of technical objectives were provided. This was a capturing of what technical achievements should occur during the project. Table 9.3 is a review of the functional objectives, which are mostly justified by the content in chapter 6, however where necessary notes are given. There was one failure from these objectives, where the Google search tool was not implemented to the standard desired. The non-functional objectives are reviewed in table 9.4 and full justification for their success is given.

I#	Objective	Successful
I1:	*Compile an unbiased test set of people of at least 30 who own profiles on a variety of social media accounts. Every member of it must of given their consent.*	**SUCCESS**
	Section 5.1 divulges the full details of how a test set of 50 people was obtained. All members gave explicit consent by responding to a social media post. Although everyone had some connection to the investigator, there was no bias when it came to race, age, gender or any other trait.	
I2:	*Examine user privacy on the following social platforms: Facebook, Twitter, LinkedIn and Instagram.*	**SUCCESS**
	Despite only Facebook and Twitter being at the centre of the investigation, throughout the investigation LinkedIn and Instagram were mentioned regularly.	
I3:	*Identify information which is made publicly available on each of the chosen platforms.*	**SUCCESS**
	The data possible from all four social platform was discussed in section 5.2, and Facebook and Twitter were examined in far more detail in chapter 7.	
I4:	*Find at least two methods of linking the same person's profiles across different social medias, and assess which platform is hardest to reach.*	**SUCCESS**
	Using similar aliases was the first method of linking profiles on separate platforms, with results discussed in section 7.1. Whilst not mentioned in that great detail, a second method by using friendships was used, reviewed briefly in 7.2.4. Finally, every platform had some way of easily reaching it, and while not explicitly said Twitter and Instagram had equal likelihood of being reached from Facebook.	
I5:	*Determine at least two methods of finding the connections between different people.*	**SUCCESS**
	The first method, talked about in section 7.2.4, was to use the data available on Facebook. The far more interesting approach was using the Twitter like analysis, thoroughly described in sections 6.4 and 7.3.2.	

Table 9.1: Investigative Objectives 1-5 Review

I#	Objective	Successful
I6:	*Make assumptions on the strength of connections between different people, and be able to group these connections into 'friendship' groups, using at least one customised technique.*	**SUCCESS**
	Using like analysis, the strength of specific connections between users could be found. By combining this with the graphing of Twitter follower relations, detailed in section 7.3.3, some core friendship groups could be found.	
I7:	*Estimate the chance of identifying or replicating a social platform's user's speech patterns based on the average amount of text mined from that platforms profiles.*	**MODERATE**
	Despite the amount of text possible to scrape from Twitter profiles being discovered, section 7.3.1, no further discussion into the likelihood of imitation was done. This could be achieved by further researching identification through text, such as the study by Zheng et. al. [81].	
I8:	*Estimate the chance of imitating a social platform's user based on the average number of images scraped from that platforms profiles.*	**MODERATE**
	Although the number of possible images from Facebook being looked at in some detail, section **??**, the impact of this on the likelihood of imitation was not mentioned. By researching more into image identification, potentially systems like that designed by De et. al discussed in section 2.1, this could be achieved [8].	
I9:	*Provide recommendations to users of social medias to keep their personal information secure.*	**SUCCESS**
	Several recommendations to users based on the results of the investigation were given at the end of chapter 7. These were thoroughly justified, mainly looking at preventing the link between different platforms being established.	
I10:	*Provide recommendations to social platforms to keep their users secure.*	**SUCCESS**
	Several recommendations to platforms were also provided at the end of chapter 7. Unlike the user recommendations, these were more pressing, with the note that no social platform should have any excuse not to be protecting their users via the recommendations listed.	

Table 9.2: Investigative Objectives 6-10 Review

T#	Objective	Successful
T1:	*Create an automated batch script that will apply all tools written to multiple users over time.*	**SUCCESS**
T2:	*Scrape all possible information from as many platforms as possible and store this information locally in a database for further examination.*	**SUCCESS**
	Even though data was only scraped from two platforms, the other two examined had issues surrounding and them, this has been justified in section 6.3.2.	
T3:	*Reduce the effect of any rate limited APIs by locally caching data, allowing for rapid development iterations.*	**SUCCESS**
T4:	*Create some way of using Google's search tool to perform wide search on a user, and pull these search results into a text file to be examined.*	**SUCCESS**
T5:	*Automatically detect when a search result has found a user's social media profile.*	**FAIL**
	Although some Google search API was created in tool kit, automatic detection was never achieved. This could have been done through extending the Google to check the results against some input parameter.	
T6:	*Use twitter4j or equivalent library to access the open API provided by Twitter and accumulate useful data from this.*	**SUCCESS**
T7:	*Implement at least one supervised machine learning technique, and analyse the consistency of this method.*	**SUCCESS**
	The results of using the random decision tree for like analysis, and the examination of the consistency, can be seen in section 7.3.2.	
T8:	*Produce social graphs for a given user, and either create software to display these or format the data to be used by third-party software.*	**SUCCESS**

Table 9.3: Functional Technical Objectives Review

T#	Objective	Successful
T9:	*Only harvest open data from social platforms that does not compromise user privacy.*	**SUCCESS**
	All data collected throughout the investigation was available on the public profiles of the test set users.	
T10:	*Ensure all third-party libraries used are open source.*	**SUCCESS**
	None of the libraries used during the implementation required a purchasing fee and were all available in the open source Maven directories.	
T11:	*Tools written should be extensible. They should be modular, and therefore expanding their purpose or repurposing them will be easier.*	**SUCCESS**
	Since tools could be turned on and off simple by commenting out sections of code in the main run script, they clearly are modular and do not depend on one another. This means they could be reused with ease in similar projects.	
T12:	*Tools written should be efficient. Whilst it is unfeasible to put a limit on their runtime, the longest tool should still be able to run overnight.*	**SUCCESS**
	No tool took longer than a single night to run for any single user, and the only tool that required a longer period of time was the graphing which after an initial long run would be much faster by using the local cached data.	
T13:	*Tools written should be well maintained. They should be well documented as to make the reusable for extension to the project in the future.*	**MODERATE**
	Whilst most code is lacking in comments, variable and function names are self-documenting, all code from external sources is referenced and chapter 6 describes the purpose of most of the codebase in detail.	

Table 9.4: Non-Functional Technical Objectives Review

9.3 Ethical, Social, Legal, and Professional Issues Evaluation

In chapter 3 a plethora of potential issues that could occur during the project were discussed. To ensure the results from the project are seen as legitimate these issues must be reviewed to see if they have all been dealt with correctly.

9.3.1 Ethical Review

The focus of the ethical issues was user consent to be part of the investigation. As mentioned in the section 5.1, all users gave their express permission to be part of the test set through responding to a social media post. Nobody asked to be removed from the test set during the investigation. The two users who were chosen as case studies for the project presentation both gave permission to be used as examples, however to maintain their privacy these results were omitted from the report. By avoiding data outside of public social medias profiles, there was no chance that a user's privacy was going to be compromised.

9.3.2 Social Review

Social issues are heavily related to the ethical issues, with the focus being on preventing an individual feeling victimised. When the test set was selected, there was no discrimination based on race, gender, age or any other demographic. Both race and disiblity were never used to classify or group individuals. Through the recommendations provided at the end of chapter 7, it is the investigators hope that the view on social media privacy will change.

9.3.3 Legal Review

A handful of legal issues were present in the project, particularly when it came to development. Every library used in the implementation was open-source, available in the Maven directories. When data was handled and stored the Data Protection Act was abided, with the data being stored on an encrypted drive. None of this data was made available online, and all personal details have been removed from this report. With regards to the sources of the data, since they were all social platforms, there is no doubt that they were legitimate and a right to own the data.

9.3.4 Professional Review

Most professional issues are related to the legal issues, for instance when it comes to reusing code or research. Whenever code was reused from an online source it was referenced through comments in the codebase. All research used through the project has been properly credited to their original founders. The results from the project have been thoroughly justified using a combination of statistical data collected from the tools created as well as referenced previous research and related work.

9.4 Project Management Evaluation

Discussed in chapter 8, it is believed that without using a flexible, agile approach the project would not have been remotely successful. Being able to adapt the aims and objectives of the project as problems were encountered or new discoveries were allowed the project to continue moving forward at a steady pace. This agile methodology would have been difficult to portray was in not for the structed documentation created via the assistance of the tools mentioned in section 8.4.2. The regular meetings with the project supervisor also greatly aided the flow of project, preventing issues encounted remaining unresolved for longer a week. Whilst the project did shift in focus, this benefited not only the results of the project but also the investigator who was more content with the new direction of the project as it was in their area of expertise.

9.5 Meeting the Aims

Ultimately the main measure of the success of the project comes down to whether or the not aims laid in chapter 1 were achieved. These aims were intentionally broad to allow the objectives of the project to remain flexible whilst the general direction was unaltered. Throughout this report, it is felt that each aim was eventually met. Many of the investigative objectives, namely I4, I5 and I6, focused on connecting different platforms as well as people, and each of these objectives was successfully completed. The majority of the results come from collecting and averaging data across a large set of users, and the most complex method used for aggregation was a well known supervised learning technique. None of the methods used were overly complex, yet still produced meaningful and interesting results. Finally, 7 recommendations were provided, 4 to users and 3 to social platforms, at the end of chapter 7. These recommendations encapsulate everything that was learnt from the results of the project, and are meant to be the final results.

This chapter is the final chapter to discuss the outcomes of the report. In the conclusion to follow the project will be summarised, and future work discussed.

CHAPTER
TEN

CONCLUSION

10.1 Summary

The investigation that took place during this project examined user privacy across four major platforms, with the main focus being on Facebook and Twitter, although Instagram and LinkedIn were also discussed. For the most part, the investigation examined the contents of a user's profile, and how this could be used to attack individuals. Several recommendations to users and platforms based on the discoveries of this project have been provided. Although the project did not meet the aims declared in the original specification, the new goals and objectives created are substantial enough to declare the project successful.

10.2 Future Work

The results from this project have a variety of purposes. While it is felt that at least some of the discoveries made are new in their field, more work is required to provide concrete evidence for these to be accepted.

10.2.1 Creation of System

The specification and the report that followed detailed a complex online social profile exploration system, which would use social media as starting point to collect data to form as user's complete online footprint. The tools created during the development stage of this project are a good starting point for this system, however a lot more work is required to make a fully-fledged system. This work would primarily be on a user interface as well as the addition of using powerful web crawlers such as Nutch to find data outside of social media.

10.2.2 Extension to Other Platforms

During the project the focus was mainly on Facebook and Twitter. Even though the ways to extend what has been created to new platforms such as Instagram and LinkedIn was mentioned frequently throughout this report, in order guarantee these methods apply on these platforms they would have to be physically

implemented. Additionally, to this these are not the only four platforms in existence, social media is an ever expanding market and it would be useful to create not only additional tools for individual platforms but also general purpose tools that could be used for any social media context. For example, like analysis could be altered such that parameters could vary dependant on the platform used.

10.2.3 Improving Statistical Data

For most of results the data given is statistical based on the test set used. However, even though 50 users were in the set, due to the variety of social platforms examined the actual number processed by each tool varies. Increasing the size of the test set would be one way of improving the reliability of the statistics produced. Also, two of the investigative objectives were only moderately successful, and this is due to estimates based on the statistics not being generated. Diving deeper into what each result means for individual users rather than simply stating them as an average would improve the use of this project as an educational work.

BIBLIOGRAPHY

[1] C. C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[2] Apache. Maven - introduction. *Available at:* `https://maven.apache.org/what-is-maven.html`.

[3] Atom. A hackable text editor for the 21st century. *Available at:* `https://atom.io/`.

[4] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[5] F. Celli, E. Bruni, and B. Lepri. Automatic personality and interaction style recognition from facebook profile pictures. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1101–1104. ACM, 2014.

[6] P. R. Center. Social media fact sheet. *Available from:* `http://www.pewinternet.org/fact-sheet/social-media/`, 2017.

[7] T. H. Centre. Protecting and unprotecting your tweets. *Available from:* `https://support.twitter.com/articles/20169886`.

[8] A. De, C. M. Bogart, and C. S. Collins. Detecting impersonation on a social network, July 9 2013. US Patent 8,484,744.

[9] M. Duggan, N. B. Ellison, C. Lampe, A. Lenhart, and M. Madden. Social media update 2014. pew research center. *Available from:* `http://www.pewinternet.org/2015/01/09/frequency-of-social-media-use-2/`, 2015.

[10] eBiz. Top 15 most popular social networking sites — april 2017. *Available from:* `http://www.ebizmba.com/articles/social-networking-websites`.

[11] Elasticsearch. Open source search & analytics. *Available at:* `https://www.elastic.co/`.

[12] P. Essiembre. How to crawl facebook. *Available at:* `https://www.norconex.com/how-to-crawl-facebook/`.

[13] Facebook. Company info — facebook newsroom. *Available from:* `https://newsroom.fb.com/company-info/`.

[14] Facebook. Graph api. *Available at:* https://developers.facebook.com/docs/graph-api.

[15] J. Fiaidhi, O. Mohammed, S. Mohammed, S. Fong, and T. hoon Kim. Mining twitterspace for information: classifying sentiments programmatically using java. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on*, pages 303–308. IEEE, 2012.

[16] FindFace. Findface.ru - . *Available at:* https://www.findface.ru.

[17] S. Fong, Y. Zhuang, and J. He. Not every friend on a social network can be trusted: Classifying imposters using decision trees. In *Future Generation Communication Technology (FGCT), 2012 International Conference on*, pages 58–63. IEEE, 2012.

[18] A. S. Foundation. What is apache nutch? *Available at:* https://wiki.apache.org/nutch/FrontPage\#WhatisApacheNutch.3F.

[19] J. Frankle. How russia's new facial recognition app could end anonymity. *The Atlantic*, 2016.

[20] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[21] Github. The world's leading software development platform. *Available at:* https://github.com/.

[22] Y. Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

[23] Google. Custom search. *Available at:* https://developers.google.com/custom-search/.

[24] GOV.UK. Data protection. *Available at:* https://www.gov.uk/data-protection/the-data-protection-act.

[25] M. M. Group. World internet users and 2016 population stats. *Available at:* http://www.internetworldstats.com/stats.htm.

[26] V. Gupta, G. S. Lehal, et al. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76, 2009.

[27] S. Hjelmqvist. Fast, memory efficient levenshtein algorithm. *Available at:* http://www.codeproject.com/Articles/13525/Fast-memory-efficient-Levenshtein-algorithm, 2014.

[28] T. K. Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[29] HtmlUnit. Htmlunit - welcome to htmlunit. *Available at:* http://htmlunit.sourceforge.net/.

[30] J. Hutchins. The history of machine translation in a nutshell. *Retrieved December*, 20:2009, 2005.

[31] O. S. Initiative. Licenses & standards. *Available at:* `https://opensource.org/licenses`.

[32] T. Innovation. Picking out one from a billion: the face recognition system from ntechlab. *Intel*, 2016.

[33] Instagram. Controlling your visibility. *Available from:* `https://help.instagram.com/116024195217477/`.

[34] Instagram. Faq. *Available from:* `https://www.instagram.com/about/faq/`.

[35] Instagram. Instagram developer documentation. *Available at:* `https://www.instagram.com/developer/`.

[36] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

[37] JGraphT. Welcome to jgrapht - a free java graph library. *Available at:* `http://jgrapht.org/`.

[38] JSoup. jsoup: Java html parser. *Available at:* `https://jsoup.org/`.

[39] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth. User interests identification on twitter using a hierarchical knowledge base. In *European Semantic Web Conference*, pages 99–113. Springer, 2014.

[40] M. Khan, Q. Ding, and W. Perrizo. k-nearest neighbor classification on spatial data streams using p-trees. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 517–528. Springer, 2002.

[41] T. J. Koetsier. How google searches 30 trillion web pages, 100 billion times a month. *Available at:* `https://venturebeat.com/how-google-searches-30-trillion-web-pages-100-billion-times-a-month/`.

[42] R. Kofler. Scraping the web with nutch for elasticsearch. *Available at:* `https://qbox.io/blog/scraping-the-web-with-nutch-for-elasticsearch`.

[43] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[44] D. T. Larose. k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, pages 90–106, 2005.

[45] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[46] Y. Liao and V. R. Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, 21(5):439–448, 2002.

[47] LinkedIn. About us. *Available from:* `https://press.linkedin.com/about-linkedin`.

[48] B. Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*, pages 627–666. Chapman and Hall/CRC, 2010.

[49] M. Madden, S. Fox, A. Smith, and J. Vitak. Digital footprints: online identity management and search in the age of transparency. pew internet and american life project, 2007.

[50] C. R. Magazine. Facebook & your privacy: Who sees the data you share on the biggest social network? *Available at:* `http://www.consumerreports.org/cro/magazine/2012/06/facebook-your-privacy/index.htm`.

[51] A. E. Marwick. The public domain: Social surveillance in everyday life. *Surveillance & Society*, 9(4):378, 2012.

[52] MathWorks. Classification using nearest neighbors. *Available from:* `https://uk.mathworks.com/help/stats/classification-using-nearest-neighbors.html`.

[53] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[54] M. M. Mostafa. More than words: Social networks? text mining for consumer brand sentiments. *Expert Systems with Applications*, 40(10):4241–4251, 2013.

[55] M. Ogneva. How companies can use sentiment analysis to improve their business. *Mashable*, 2010.

[56] S. Overflow. Join the stack overflow community. *Available at:* `https://stackoverflow.com/`.

[57] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.

[58] B. Pang and L. Lee. 4.1. 2 subjectivity detection and opinion identification. *Opinion Mining and Sentiment Analysis. Now Publishers Inc*, 2008.

[59] M. Pennise, R. Inscho, K. Herpin, J. Owens Jr, B. A. Bedard, A. C. Weimer, B. S. Kennedy, and M. Younge. Using smartphone apps in std interviews to find sexual partners. *Public health reports*, 130(3):245–252, 2015.

[60] QuickML. An easy-to-use but powerful and fast machine learning library for java. *Available at:* `http://quickml.org/`.

[61] restfb. restfb. *Available from:* `http://restfb.com/`.

[62] L. Rokach and O. Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487, 2005.

[63] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242. ACM, 2010.

[64] R. Samudrala and J. Moult. A graph-theoretic algorithm for comparative modeling of protein structure. *Journal of molecular biology*, 279(1):287–302, 1998.

[65] B. C. Society. Bcs code of conduct. *Available at:* `http://www.bcs.org/category/6030`.

[66] Statista. Global social network penetration rate as of january 2016. *Available at:* `https://www.statista.com/statistics/269615/social-network-penetration-by-region/`.

[67] F. Su and K. Markert. From words to senses: a case study of subjectivity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 825–832. Association for Computational Linguistics, 2008.

[68] A.-H. Tan et al. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, volume 8, pages 65–70, 1999.

[69] S. Text. Sublime text: The text editor you'll fall in love with. *Available at:* `https://www.sublimetext.com/`.

[70] D. Trottier. Policing social media. *Canadian Review of Sociology/Revue canadienne de sociologie*, 49(4):411–425, 2012.

[71] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10(1):178–185, 2010.

[72] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[73] Twitter. Company — about. *Available from:* `https://about.twitter.com/company`.

[74] Twitter. Getting started. *Available from:* `https://support.twitter.com/articles/215585`.

[75] Twitter. Welcome to the twitter platform. *Available at:* `https://dev.twitter.com/`.

[76] twitter4j. Twitter4j. *Available from:* `http://twitter4j.org/en/index.html`.

[77] P. Watson-Wailes. The ultimate guide to the google search parameters. *Available at:* `https://moz.com/blog/the-ultimate-guide-to-the-google-search-parameters`, 2008.

[78] K. Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18:1473, 2006.

[79] P. V. . K. Wilfrong. Scaling the facebook data warehouse to 300 pb. *Available at:* `https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/`.

[80] L. Wiskott, N. Krüger, N. Kuiger, and C. Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779, 1997.

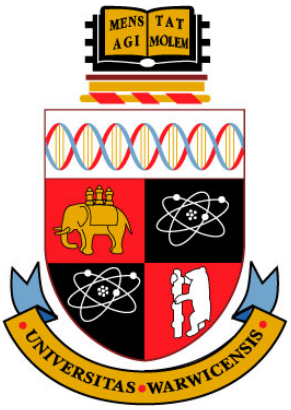
[81] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American society for information science and technology*, 57(3):378–393, 2006.

Appendices

APPENDIX

ONE

SPECIFICATION REPORT



Online User Privacy Investigation Using Social Profile Seeding

CS310 Computer Science Project

Project Specification

Adam Coles

Supervisor: Dr. Matthew Leeke

Department of Computer Science
University of Warwick

2016-17

Contents

1 Introduction

With access to the internet becoming increasingly vital in the modern world it is of no surprise that globally, as of June 2016, 49.2% of the population are considered to be active internet users [12]. Across all users social media is the most popular activity to engage in whilst online, and a staggering 31% of the global population own a profile on at least one social platform [22]. All this activity produces a vast amount of personal, potentially open data, with Facebook alone currently holding over 300 petabytes of information across their warehouses [25]. With an immense amount of information available on the internet your privacy becomes difficult to maintain.

1.1 Motivation

The term 'active digital footprint' refers to the personal data an internet user gives permission to be accessible online, and all users of social media will have one. As the number and purposes of social media expands this digital footprint becomes more detailed, allowing users to connect with new people who share similar interests or friendship circles. However, people are often unaware of just how much data they make available for the world to see, a fact utilised by the police to track down criminals and known associates [1] [24]. A larger social media presence gives more chance to find additional information about a person that they have not explicitly shared. This includes public records, which have transitioned from paper records to a digital format, and that have been criticised for giving away overly personal details [3]. Whilst the police use open data to benefit society, if they have that ability so do potential criminals or stalkers, and social media consumers must be aware just how much can be found out about them online [18].

Despite attempts to make people aware of the dangers of open online profiles, in general there is a blasé attitude towards online privacy. In 2012 a survey showed that 26% of American Facebook users shared their entire profile publicly, including all wall posts [17]. Without being scared into a realisation of how important privacy is, there will not be a change in attitude towards this issue.

1.2 Project Aims

The primary goal of the project is to create an online profiler that uses a Facebook profile as a seed in order to collect as many details as possible about a user from online open data sources. Whilst their will be some complex algorithmic design in order to speed up the process, the majority of aims focus on effective internet crawling. Results from the project should assist in a variety of areas, from law enforcement to social media awareness.

1.2.1 Locating Social Media

Even though some people will not have a wide selection of public profiles, using Facebook as a starting point assists in locating as many accounts as possible owned by an individual. All this data can be deemed to be personal and can be aggregated to create concrete assumptions by correlating truths between them. Depending on the number of accounts, and their privacy settings, limits the amount of data available for collection. For some extreme examples, personal truths can be drawn from the information given, for instance geo-tagged tweets on Twitter. Some websites

provide detailed API that should provide all the data required, although others will require the webpage to be parsed in more complex ways. Research must be done to ensure the most efficient and effective data harvesting method is used.

1.2.2 Connecting to New Sources

Once some social media has been examined more internet sources need to be found, such as the previously mentioned public records. A simple Google search with a handful of parameters can find obscure references about a person that may either solidify assumptions or create new theories that can be explored. This area of search is more likely to show unexpected results, as the target is in less control of the privacy of data displayed. The main drive of this aim is that the searches are recursive, that is if new information is found and confirmed the search should reoccur with additional parameters.

1.2.3 Receiving Real-Time Input

In some cases a human can confirm faster and with greater certainty specific details about data found. If, for instance, the profile picture of an individual is of them wearing a police uniform, and the system suspects they are police officer, a human can confirm the assumption, allowing the system to use this suspicion as fact. Not only does this speed up the process significantly, adding this feature reduces the complexity of algorithms used whilst improving the accuracy of the results. Decisions will have to made to restrict the amount of user input requested, as the system is meant to be for the most part automated.

1.2.4 Using Open Data and Open Software

One core value that must be upheld during the project is that all data used is open and any third-party software used is free to license. This not only ensures the privacy of test subjects it also ensures that there is no budget to development and production. End users of the system will typically not have the funds to pay for complex software or hardware, therefore it is essential that as little money is spent as possible.

1.3 Stakeholders

The primary stakeholders in the project are the developer and the project supervisor. Any people who give their permission for their data to be used whilst testing the system will have to have their privacy guaranteed, in case secret information is found. The opinion of a law enforcement officer on the final outcome is also desired, as they would be the primary user of a system similar to the one being developed.

2 Related Work

As mentioned, the use of someones online footprint to learn about them and their associates is already occurring within law enforcement [24]. This topic has also been a feature of many academic

studies, particular when looking at teaching others to manage their online image [6]. Existing systems will have a heavy influence throughout the project, as these are direct competition to the final developed product, and provide a good testing base line. They also assist when choosing third-party software and creating the complex algorithms that are necessary to find hidden information. Meanwhile scholarly work into this area will help when it comes to understanding the effect the results of this project will have on social and legal issues.

2.1 Collecting Information

It is likely that anybody who is interested in finding out anything about a person using the internet will turn to a search engine. In 2013 Google had approximately 30 trillion websites indexed, so even searching using detailed parameters produces thousands of results [16]. However, search engines are not the only way to scour the web. Customisable or specialised web crawlers such as Apache Nutch, a Java based modular crawler, can output more desirable results than Google.

Each social network has a different level of access when it comes to harvesting user data. Russell (2014) explored many techniques for mining certain sources, including friendship graphs on Facebook and Twitter frequency analysis [20]. Even when inside social media, crawling is still necessary to explore all possible links in a graph, as few transitions from page to page can be hundred percent guaranteed to be useful. For instance, just because two people are friends on Facebook does not necessarily mean they know each other, particularly in less educated areas. O'Dea and Campbell (2012) found that 13.5% of Australian adolescents accepted unknown friend requests [5].

For specific open source web crawlers, *Apache Nutch* is highly scalable, robust Java web crawler. As it is built upon the *Apache Lucene* framework, Nutch is modular and therefore can be combined with other packages to expand upon the features available, such as search or indexing with *ElasticSearch* [10]. *Norconex* is another Java based crawler with similar functionality to Nutch. It comes with a plethora of documentation and a handful of tutorials, including how to crawl with the assistance of Facebook API [8]. However, due to being less established than Nutch, *Norconex* has less support from other tools and is not maintained as well.

A web crawler locates websites and retrieves the HTML, yet to receive any meaningful results this data must be indexed and searched. *ElasticSearch* is the most popular search engine, currently ranking above its similarly Lucene based counterpart *Apache Solr* [7]. Both can be run within Java with HTTP and JSON interfaces, and for the most part are very similar, with notably *ElasticSearch* leaning more towards REST APIs. However *ElasticSearch* has a more robust aggregation feature, and therefore may be more suitable for the project [23].

Building on this further, when fed a large block of text the system may need to understand natural language to find key data. Many natural language processors exist for this function, but few are open source. *Semantria* for instance is primarily a sentiment analyser. A set of standard API is provided and a subscription is paid to have access to the *Semantria Cloud*, giving the benefit of high-speed, scalable performance. IBM's Watson also has in built sentiment analysis along with a range of other text manipulation modules. Whilst immensely powerful, some features of Watson have been replicated before to relative success within a university classroom [26], giving hope that as an extension these features can be simplified and added to the scope of this project.

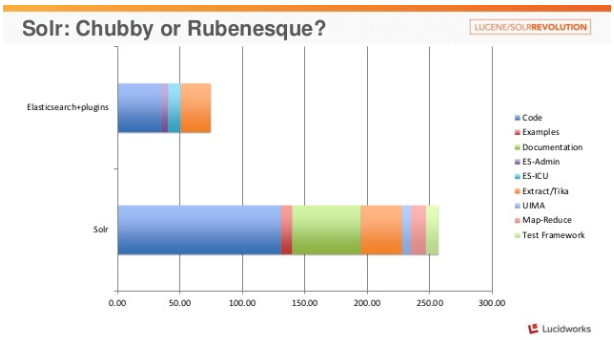


Figure 1: Comparing the Size of *ElasticSearch* Against *Apache Solr* [19]

2.2 Creating the Profile

Once data has been collected some means of aggregation is required to produce meaningful results. That is the data has to be compared to find correlations, the source of the data has to be examined for trust, and assumptions have to be made.

Within the majority of social networks, especially media platforms, there is a recommendation engine. While their outputs are not that desirable for the scope of this project, their means of gathering and processing information on a user could be helpful. As elegantly put by Finger (2014), a recommendation engine "reduces Big Data to small data", a definition that could easily fit with the aims of this project [9]. Netflix value their recommendation engine highly, with claims 75% of users watch from recommendations [2].

Table I: Recommendation Techniques

Technique	Background	Input	Process
Collaborative	Ratings from **U** of items in **I**.	Ratings from **u** of items in **I**.	Identify users in **U** similar to **u**, and extrapolate from their ratings of **i**.
Content-based	Features of items in **I**	**u**'s ratings of items in **I**	Generate a classifier that fits **u**'s rating behavior and use it on **i**.
Demographic	Demographic information about **U** and their ratings of items in **I**.	Demographic information about **u**.	Identify users that are demographically similar to **u**, and extrapolate from their ratings of **i**.
Utility-based	Features of items in **I**.	A utility function over items in **I** that describes **u**'s preferences.	Apply the function to the items and determine **i**'s rank.
Knowledge-based	Features of items in **I**. Knowledge of how these items meet a user's needs.	A description of **u**'s needs or interests.	Infer a match between **i** and **u**'s need.

Figure 2: Recommendation Engine Techniques [4]

Unlike websites with a recommendation engine, the proposed system has no access to private

information. Amazon, for example, store data on all products a user views, and from there can make educated decisions on other products of interest. As noted by Iskold (2007), Amazon relies on remembering what you've done "years and minutes ago", information not available for public viewing [15]. Despite this the aggregation techniques used are invaluable. Figure 2 shows Burke's (2002) descriptions of these techniques, which will provide a baseline for the assumptions made by the system [4].

There have been attempts at creating a social profile from crawled data in the past, unrelated to recommendation engines. Van Hinsbergh (2015) created a search engine which would attempt to create a profile of a person from a set of provided keywords and a useful starting seed [13]. Similarly to the proposed system, Van Hinsbergh's final application also took user feedback into account and used open data. Generously, Van Hinsbergh has given access to his report and some source code that will be referenced throughout development.

The proposed system is slightly different to other programs already available primarily due to the social media seed that will be the root of a person's profiling. This reduces the amount of manual searching required prior to utilising the application. User feedback will also be integral to the system, distinguishing it further from current applications.

3 Ethical, Social, Legal and Professional Issues

In every software development project certain standards must be maintained to ensure not only the end system is allowed to launch but also to solidify trust between developers and the end users. Stakeholders also require reassurance that their potential investment will not go to waste, or that they protected from any backlash due to improper procedures. The aim of this section is to discuss problems that may arise the ethical, social, legal and professional fields, and how preparations have been made to resolve them. For a base during development the British Computing Society Code of Conduct will be maintained, to protect customers and developer integrity [21].

3.1 Ethical Issues

Due to the highly personal nature of the intended system, a plethora of ethical issues emerge, the key issue being the privacy of the users testing the system. In order to keep the identities of the users safe, no personal data should be stored permanently after searching. This also eliminates the need for complex encryption when handling the information. Furthermore, any test users will have to give express permission for their profiles to be examined, and if they are selected to be an example for displaying the system's potential, they must give further permission for their details to be shown to stakeholders or other third-parties. At any point the user has the right to remove themselves as a test case if they feel uncomfortable with the information found.

3.2 Social Issues

Continuing from the protection of privacy of individuals using the system, there is the chance that some evidence found may lead to somebody feeling victimised. To prevent this as much as possible, the program will try to avoid presuming a persons sexual, religious or political preferences. Even though some of this data may not be private for all users, its best to abstain across the entire consumer range to avoid accidental discovery. On top of this, the system will avoid stereotypes regarding race or disability entirely, and there will be no bias assumptions made from these. Should the system be successful in its aims and locate personal data from inputted profiles the view on social media may change for some users, hopefully leading them to be more aware of profile security.

3.3 Legal Issues

Since the final program will inevitably use some third-party software or libraries the developers must ensure that they have the correct licensing. A major aim of the project it to only use open data and open software so all third-party sources should be covered by some open source license [14]. As personal data will be mined and stored at some time during the program operation, the system must comply to the terms of the Data Protection Act [11]. Following from this, the data collected must come from legitimate sources that have the right to own the data to begin with, although this may be hard to verify.

3.4 Professional Issues

It is critical for the platforms success that a user finds the data presented to be valuable and that they trust the system. As mentioned, data will only be temporarily stored to try to guarantee information found is only seen by the intended user. Also aforementioned developers will follow the BCS Code of Conduct which ensures they are working for the public's interest and for the profession [21].

4 System Requirements

Breaking down the main aims of the project into detailed requirements allows progress of software to be tracked throughout the course of development. These requirements may exactly represent the final system as they have room to change over time however they provide a strong base to work on.

4.1 Functional Requirements

The following requirements look at what the system can do; the inputs it receives, how these inputs are processed and the outputs it produces:

F1: *The system must allow users to search for information using a Facebook seed (profile).*

F2: *All available data should be mined from the Facebook input and then from other social media should a link be established.*

F3: *The system must recursively search for more data once a detail on the individual has been discovered, across the entire internet.*

F4: *The data mined by the system should be related only to the inputted profile in some way however tenuous*

F5: *The system should ask the user for feedback if an assumption has been made without full certainty and human decision would benefit.*

F6: *The system must avoid overly sensitive data when searching for information.*

F7: *The system should avoid elaborate inferences that both it and a human will have difficultly verifying.*

F8: *The system should have a clear interface to display the data, including a simple search bar for the Facebook link and some area to ask for user input during the search.*

F9: *The system should allow the user to define the depth of the search, allowing for a quick search to find the basics and more robust search to find as much as possible.*

F10: *The user should be apply to supply additional information to the system if it is already a known fact, giving the system a further head start when searching.*

4.2 Non-Functional Requirements

The following requirements look at the attributes of the system; how it will perform in a real-world scenario:

NF1: *The system should be extensible. Features should be implemented as modules, that can be easily added to data pipelines.*

NF2: *The system should be scalable. If moved onto a faster machine, the system should be able to locate and process information at a faster rate.*

NF3: *The system should be efficient. Even though search may be time consuming the user should be updated with information as soon as it is available.*

NF4: *The interface to the system should be easy to understand and provide information in a clear and unambiguous format.*

NF5: *The system should be maintainable, through both well documented and well presented code, so future development can be made if required.*

4.3 Hardware and Software Constraints

In order to keep the system accessible to its target users and to match the aims of the project some constraints must be made on the resources used. Then end program should be able to run on home computers as this typically the level used in law enforcement. There are no exact time restraints put on search response time, however it should be reasonable since the system is not fully automated, even on lower end machines. As mentioned previously, the software should use only open license sources, to add no extra costs to development and ownership.

4.4 Foreseeable Challenges

The internet is astronomically huge and is impossible to search fully. Deciding on the cut off point of search will be one of the main challenges of the project, hence some user input is allowed to give the system some idea of the level of detail sought after. The developer working on the project has no prior experience in web crawling, so there will be a learning curve to overcome before progress can be made on development.

5 Testing and Success Measurement

For any software project testing is critical in ensuring your end system meets the aims and requirements set out pre-development.

5.1 Testing Strategy

During development unit tests for each function and class will be produced to both make sure the outputs match what is expected and to catch the rare edge cases that may occur in a live run. Following this multiple related units will be combined together. A broader over-arching functionality of different areas of the code will be tested, for instance search or analysis. This is known as integration testing, attempting to find flaws in the information pipeline between units. All the modules will then be combined to create the final system tests, looking at the system as a whole. Ultimately theses tests will be directly related to the requirements set out in Section 3. Finally, and arguably most critical for this particular project, there will be user tests created to gauge the satisfaction levels of end users. Their feedback can be used to iteratively improve the system until everyone is happy with the results produced.

5.2 Success Criteria

Whilst in some projects simply hitting all the requirement is sufficient enough to claim it was a success, in many cases more details are needed. With this system user satisfaction is of the highest priority, and the user tests will be the main judge of the overall success, even though all requirements will still need to be met.

6 Project Management

To keep a software development project on track a plan is required to manage time effectively. This plan will be highly flexible to deal with sudden deadlines or potential unforeseen circumstances however will be a rough guide to follow during all stages of development.

6.1 Software Development Methodology

The software development methodology of choice is an agile approach, in documentation and the creation of the software itself. This easily allows for adaptations during development, should the aims or requirements of the project change. Also since there is a single developer allocating work is unnecessary so concrete decisions restrict rather than assist.

6.2 Design Approach

A top down approach to design will be used, that is each area of the system will be broken up into different modules, these modules designed and developed for the most part separately, and finally combined to create a larger system. This method makes the complete problem more manageable, although does mean it will take longer to have a working prototype. Despite this each module can still be tested using unit tests, a technique described in Section 5.1.

Before beginning work on any new section of the system a rough plan will be sketched, to avoid going out of scope or forgetting certain intricacies. To keep track of changes made, and to give room for features to developed in many ways, Git will be used as a version control software, with every commit being highly detailed. Comments will be used to explain complex parts of the code so a recall to memory is not required when it comes to writing the report.

6.3 Project Timeline

As the team developing this project consists of one person many agile methods are overly complex and would not be suitable, such as SCRUM. Instead, a simple adaptable Gantt chart (Figure 3) will be used to loosely track deadlines and give vague estimated times of the completion of certain modules in the system, as well as key project deadlines.

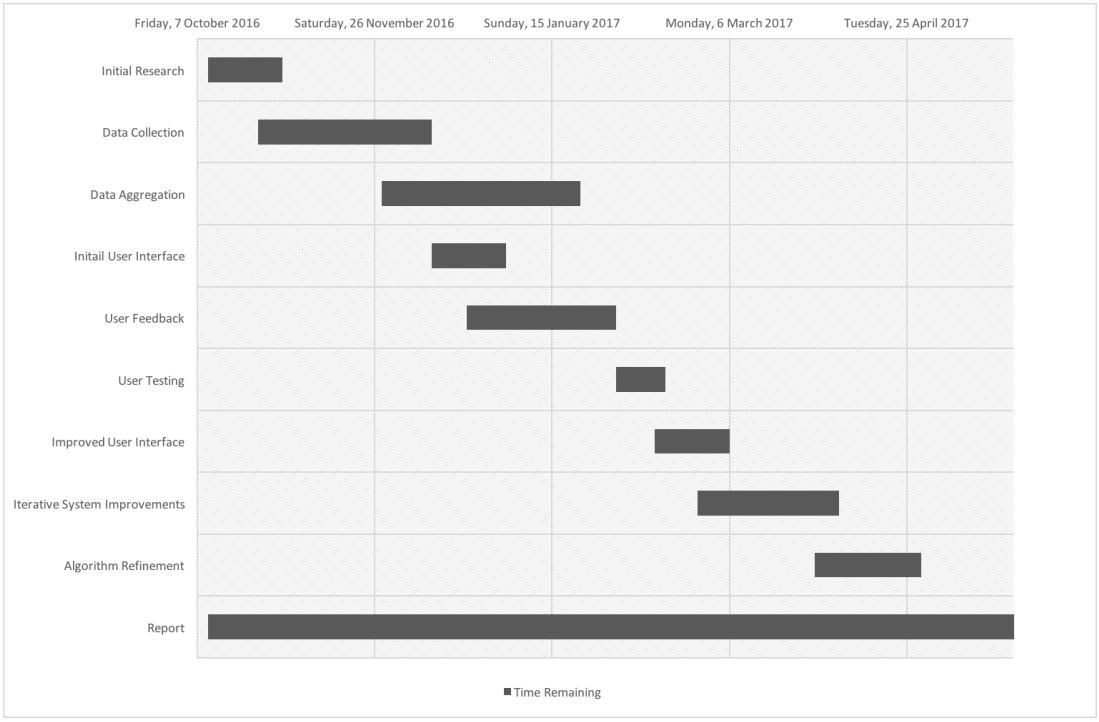


Figure 3: Gantt Chart Displaying Proposed Deadlines

6.4 Tools

During the project the following software tools will be used to aid in documentation, communication and development (Figure 4). Skype will be used so stakeholders can contact each other urgently when email will not suffice and a meeting cannot be made. LaTeX will be used to write any official

Figure 4: Selection of Development Tools' Logos

reports, as it produces a professional and easy to read document; TeXShop for Mac will hence be the word editor of choice. For code editing, a combination of Sublime Text and Vim will be used as this is the environment the developer feels most comfortable in. When sharing documents between multiple people is required DropBox will be used, since it provides a simple interface with notifications when changes are made.

7 Conclusion

To summarise the document, the core aim of the project is to create an online profile builder using social media as a seed. The main functions of the system have been outlined in Section 3.1, primarily the ability to locate different social media sources, mine them for information and to expand outwards to further sources on the internet. Over the next few months an application will go through design, development and testing, in hope of producing results that will change society's opinion on social media privacy.

References

[1] Susan B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2006.

[2] Xavier Amatriain & Justin Basilico. Netflix recommendations: Beyond the 5 stars (part 1). *Available at:* `http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html`.

[3] Kristen M. Blankley. Are public records too public? why personally identifying information should be removed from both online and print versions of court documents. *Ohio State Law Journal*, 65(2):413–450, 2004.

[4] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[5] Bridianne O'Dea & Andrew Campbell. Online social networking and the experience of cyber-bullying. *Annual Review of Cybertherapy and Telemedicine*, pages 212–217, 2012.

[6] Nicole Osborne & Louise Connelly. Managing your digital footprint: possible implications for teaching and learning. *European Conference on Social Media, Porto, Portugal*, 2015.

[7] db engines.com. Db-engines ranking of search engines. *Available at:* `http://db-engines.com/en/ranking/search+engine`.

[8] Pascal Essiembre. How to crawl facebook. *Available at:* `https://www.norconex.com/how-to-crawl-facebook/`.

[9] Lutz Finger. Recommendation engines: The reason why we love big data. *Available at:* `http://www.forbes.com/sites/lutzfinger/2014/09/02/recommendation-engines-the-reason-why-we-love-big-data/#7703e004218e`.

[10] Apache Software Foundation. What is apache nutch? *Available at:* `https://wiki.apache.org/nutch/FrontPage\#WhatisApacheNutch.3F`.

[11] GOV.UK. Data protection. *Available at:* `https://www.gov.uk/data-protection/the-data-protection-act`.

[12] Miniwatts Marketing Group. World internet users and 2016 population stats. *Available at:* `http://www.internetworldstats.com/stats.htm`.

[13] James Van Hinsbergh. Generating investigative profiles through open source data fusion. Technical report, University of Warwick, 2015.

[14] Open Source Initiative. Licenses & standards. *Available at:* `https://opensource.org/licenses`.

[15] Alex Iskold. The art, science and business of recommendation engines. *Available from:* **ReadWriteWeb**.

[16] Tune J. Koetsier. How google searches 30 trillion web pages, 100 billion times a month. *Available at:* `http://venturebeat.com/2013/03/01/how-google-searches-30-trillion-web-pages-100-billion-times-a-month/`.

[17] Consumer Reports Magazine. Facebook & your privacy: Who sees the data you share on the biggest social network? *Available at:* `http://www.consumerreports.org/cro/magazine/2012/06/facebook-your-privacy/index.htm`.

[18] Alice E. Marwick. The public domain: Social surveillance in everyday life. *Surveillance & Society*, 9(4):378–393, 2012.

[19] Alexandre Rafalovitch. Solr vs. elasticsearch - case by case. Lucene/Solr Revolution Conference, 2004.

[20] Matthew A. Russell. *Mining the Social Web*. O'Reilly Media, 2014.

[21] British Computing Society. Bcs code of conduct. *Available at:* `http://www.bcs.org/category/6030`.

[22] Statista. Global social network penetration rate as of january 2016. *Available at:* `https://www.statista.com/statistics/269615/social-network-penetration-by-region/`.

[23] Kelvin Tan. Apache solr vs elasticsearch. *Available at:* `http://solr-vs-elasticsearch.com/`.

[24] Daniel Trottier. Policing social media. *Canadian Review of Sociology*, 49(4):411–425, 2012.

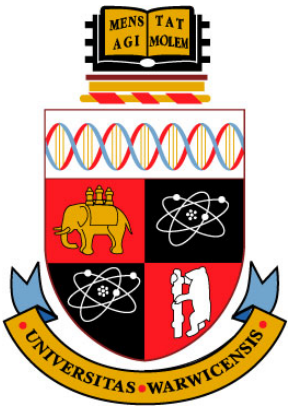
[25] Pamela Vagata & Kevin Wilfrong. Scaling the facebook data warehouse to 300 pb. *Available at:* `https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/`.

[26] Walid Shalaby & Adarsh Avadhani Wlodek W. Zadrozny, Sean Gallagher. Simulating ibm watson in the classroom. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 378–393, 2015.

APPENDIX

TWO

PROGRESS REPORT



Online User Privacy Investigation Using Social Profile Seeding

Progress Report

by

Adam Coles

Department of Computer Science

University of Warwick

2016–17

Contents

1 Introduction

With access to the internet becoming increasingly vital in the modern world it is of no surprise that globally, as of June 2016, 49.2% of the population are considered to be active internet users [13]. Across all users social media is the most popular activity to engage in whilst online, and a staggering 31% of the global population own a profile on at least one social platform [21]. All this activity produces a vast amount of personal, potentially open data, with Facebook alone currently holding over 300 petabytes of information across their warehouses [26]. The majority of internet users are unaware of how compromised their privacy becomes when sharing data online, and are therefore at risk of exploitation.

1.1 Revisiting Project Aims

The key aims of the project remain unchanged, with the system being composed of three main sections, targeting social platforms, wider online sources and human processing. Ultimately the primary goal is to create an online profiling application that uses a common social media seed as a start point and harvests data in an attempt to give the most accurate and detailed representation of an individual. The project strives to assist in two keys areas, law enforcement and online privacy education.

1.1.1 Locating Social Profiles

This aim remains largely the same from what was stated in the specification:

"Even though some people will not have a wide selection of public profiles, using Facebook as a starting point assists in locating as many accounts as possible owned by an individual. All this data can be deemed to be personal and can be aggregated to create concrete assumptions by correlating truths between them. Depending on the number of accounts, and their privacy settings, limits the amount of data available for collection. For some extreme examples, personal truths can be drawn from the information given, for instance geo-tagged tweets on Twitter. Some websites provide detailed API that should provide all the data required, although others will require the webpage to be parsed in more complex ways. Research must be done to ensure the most efficient and effective data harvesting method is used."

During the early stages of development it was decided that LinkedIn, a social network for maintaining professional connections and recruiting potential employees, was the next step from a Facebook profile. This should provide a solid launching pad for further searching by taking an individuals employment history.

1.1.2 Link to New Sources

Whilst the principle of this aim has been unchanged, certain aspects have been altered. After all possible social media profile have been visted different internet sources will need to be found. Specific organisations will be targeted for this crawling unique to an individual, such as previous

employers. Searches will be recursive, if a new potential target is discovered they will also be scanned with new parameters. It is this area of the system which is most likely to yield unexpected and crucial results.

1.1.3 Receiving Real-Time Input

This aim has not been altered from the specification:

"In some cases a human can confirm faster and with greater certainty specific details about data found. If, for instance, the profile picture of an individual is of them wearing a police uniform, and the system suspects they are police officer, a human can confirm the assumption, allowing the system to use this suspicion as fact. Not only does this speed up the process significantly, adding this feature reduces the complexity of algorithms used whilst improving the accuracy of the results. Decisions will have to made to restrict the amount of user input requested, as the system is meant to be for the most part automated."

1.1.4 Using Open Data and Open Software

Similarly this aim has not been altered from the specification:

"One core value that must be upheld during the project is that all data used is open and any third-party software used is free to license. This not only ensures the privacy of test subjects it also ensures that there is no budget to development and production. End users of the system will typically not have the funds to pay for complex software or hardware, therefore it is essential that as little money is spent as possible."

2 Research Progression

During the time since the specification was written most resources have been dedicated to programming instead of academic research. Despite this though there has been some research in the related fields to the elements of the system currently in development, as well the exploration of different open source libraries available to the developer.

2.1 Previous Research

The research displayed in the specification focused on two areas, collecting information and aggregating this information into meaningful results. This section will briefly summarise the conclusions of this research, and expand slightly based on what has been learnt since. Refer to the specification for greater detail in each area.

2.1.1 Collecting Information

It is likely that anybody who is interested in finding out anything about a person using the internet will turn to a search engine. In 2013 Google had approximately 30 trillion websites indexed, so

even searching using detailed parameters produces thousands of results [16]. An alternative to search engines are customisable, specialised web crawlers. A web crawler as defined by Dikaiakos et. al is a program that traverses the hypertext structure of the Web, starting from a 'seed' list and recursively adding more documents that are received from that list [17]. Of all web crawlers looked at, *Apache Nutch* was considered to be best in terms of scalability and robustness [8].

After data was collected via the crawler, it must indexed and searched to be of any use. *Apache Solr* and *Elastic Search* were the two search engines considered, with each having different advantages over the other [7] [3] [22]. The process of harvesting data from a certain website or set of websites is known as 'scraping'. More elegantly, scraping collects data from multiple locations and gives the possibility of repurposing the data for a new cause [25].

2.1.2 Creating the Profile

Scraping will produce a repository that must be analysed and aggregated to produce useful results. That is the data has to be compared to find correlations, the source of the data has to be examined for trust, and assumptions have to be made.

Recommendation engines are a prime example of data aggregation done correctly. They compress big data into small data [6]. A variety of techniques are used, as established by Burke (2002) (figure 1. However the proposed system has limited data access in comparison to recommendation engines, as each user will have a private digital footprint on every website they use that can be only accessed by that website provider. For instance, Amazon relies on remembering what you've done "years and minutes ago" when giving a purchase recommendation [15].

Table I: Recommendation Techniques

Technique	Background	Input	Process
Collaborative	Ratings from **U** of items in **I**.	Ratings from **u** of items in **I**.	Identify users in **U** similar to **u**, and extrapolate from their ratings of **i**.
Content-based	Features of items in **I**	**u**'s ratings of items in **I**	Generate a classifier that fits **u**'s rating behavior and use it on **i**.
Demographic	Demographic information about **U** and their ratings of items in **I**.	Demographic information about **u**.	Identify users that are demographically similar to **u**, and extrapolate from their ratings of **i**.
Utility-based	Features of items in **I**.	A utility function over items in **I** that describes **u**'s preferences.	Apply the function to the items and determine **i**'s rank.
Knowledge-based	Features of items in **I**. Knowledge of how these items meet a user's needs.	A description of **u**'s needs or interests.	Infer a match between **i** and **u**'s need.

Figure 1: Recommendation Engine Techniques [2]

The system outlined in section 1.1 will not suffer from one of the main problems faced with recommendation engines, the cold-start problem, which can be defined as 'recommenders cannot draw inferences for users or items for which it does not have sufficient information' [19]. Not dealing with the problem is disastrous for these applications, as they lose customer confidence. Since a social profile seed will be the input to the proposed system, there will always be a knowledge base to work off.

2.2 New Research

As development began to take place, more research was conducted when necessary.

2.2.1 Terms of Service

It became clear early on that some of the methods previously thought about to harvest data had the possibility of violating the terms of service of some websites. In particular, Facebook state "(Do not) collect users' content or information...using automated means...without our prior permission" [5]. Whilst not illegal to break said terms, the methods can be justified by saying the ideal end users, law enforcement, will have the access rights to anyone's data should they need it. To be safer crawling will not be performed on Facebook; only the users who give express permission to be accessed should be scraped. If, however, Facebook demand the activities to halt, alternate means must be prepared.

2.2.2 Social Media Interfaces

Many social media sites have application programming interface (API) that will allow easy access to some user data. In the case of Twitter the majority of data access does not require user permission, so libraries such as twitter4j can be used in the project to retrieve twitter information [23]. With Facebook, the user need to give a code to the application before most of their data be viewed. Should the original idea for implementation become unusable due to the issues described in the previous section, this code will be needed instead, and then the library restFB can be integrated into the system [18]. Other online resources such as Google also have developer API, but most of the time this is limited when used without cost. Only 100 searches are allocated per day using Google's service [10].

2.2.3 Finding Related Web Pages

Even though it is planned that Nutch will be used to crawl websites, an indexed search engine must first be used to locate websites on interest. Google allows many different parameters to be set to filter results down to precisely what the user needs [11]. One particularly effective way to reduce the size of search is by using statistically improbable phrases (SIPs). A term coined by Amazon to determine the key words of a book for their search algorithms, a SIP is a sequence of words that appear more frequently in one document than others in the set [1]. Adding an important phrase to the search query should significantly decrease the corpus of results.

Statistically improbable phrases can also be used to compare the relatedness of documents. Whilst Boulgakov and Stark were inconclusive in results of their study in this field, they strongly believed they could find a metric using SIPs that worked given more time [20]. Errami et. al (2010) did find that SIPs were a good way of identifying duplication between two documents, which could be used to distinguish whether or not two websites are fundamentally identical [4].

2.2.4 Selecting Relevant Data

Choosing which data to specifically target will be a critical part of the system. Personal data is an ambiguous term, Grant (2008) stating even legally "[since data protection act] 10 years on and the definition of personal data is more clouded now than it was in 1998" [12]. The aims of this project simple state 'creating an online profile' is the end target of the system. At the least a similar level of data to a complete Facebook profile should be reached, identified by Bonneau et. al as "a user's name, location and contact information, educational and employment history, personal preferences, interests, and photos" [14]. Following this a social graph could be simulated by looking at possible associates, either by connecting similar interests, observing any public friendships or by matching names in related pages [9] [24].

2.3 Future Research

Moving forward there is only one area of the system still not understood fully, making interferences and data aggregation. The developer however has some prior knowledge in simple artificial intelligence, giving them ability should they be short on time to reduce the complexity of this component. Otherwise, the techniques discussed in section 2.1.2 should be explored further, and knowledge that can be applied directly into production of the application will be researched.

3 System Progress

Following the development plan displayed in the specification, some progress has been made on the system, primarily in the gathering of data. On top of this, designs for the how the final application's data flow have been drawn up, and some technologies have begun to be implemented into the system.

3.1 Requirements Alterations

Currently there has been no change to the requirements outlined in the specification since the aims of the project have been altered only slightly from their original versions.

3.2 Development Technologies

There are various programming aids and code libraries that have been researched and tested. Notably integrated developer environments, for example Eclipse, have generally been decided upon against in favour of standard text editors, although this may change should the code base become unanticipatedly large.

3.2.1 Maven

Maven is a Java software management tool that allows for exterior dependancies to be imported and handled with ease. Despite a steep learning curve, the current iteration of the system has been built with the assistance of Maven and will continue to be built in this way throughout the entirety of development. When it comes to testing, Maven also has useful tools to perform repeatable unit tests with libraries such as JUnit.

3.2.2 Jsoup and HtmlUnit

Whilst typical internet crawlers are planned to be used for the later stages of the system they are bulky and have issues with terms of service as mentioned in section 2. When it comes to more fine tuned website parsing and traversal the combination of HtmlUnit and Jsoup provide an interesting alternative. HtmlUnit is a headless browser for Java, capable of simulating standard browsers including collecting cookies and dealing with Ajax requests. This allows for a plethora of additional bonuses, like the ability to login to websites and traverse the user only pages. Jsoup is a HTML parser which works well in combination with HtmlUnit as the latter can produce complete web pages with javascript resolved. Together they allow for scraping that appears similar to a humans interaction rather than an automated process.

3.3 Architectural Design

A similar style to the blackboard architecture design was chosen for the system. The blackboard design contains a data repository which as it expands activates different to the clients which use it. Typically programs that have some artificial intelligence or machine learning will use this design as updates to data trigger responses. In combination to the blackboard approach there will be a pipe and filter design for the start of the application, where data will flow from one component to the next using interfaces between them. This is to create a foundation knowledge base for the rest of the system to respond to. A more in depth display of the design is shown in figure 2.

Each module has a different purpose, described by it's name. The Nutch module is where a combination of Apache Nutch and Apache Solr, as mentioned in the specification, will crawl a given website for data related to the individual. In the search engine module queries will be optimised, searched on Google, and then their results will be refined and returned. Some specific social media platforms will have dedicated API for retrieving user information, such as Twitter, and the social profile module will be dedicated to harvesting these websites. The user response system will be used to retrieve user input as discussed in aim three.

The data repository in this design will act also as the inference engine, constantly processing the data it receives and making new requests to a specific module depending on the desired purpose. For instance if a related website is found via a Google search, the repository will send a request to the Nutch module to begin crawling that website. A final goal for the system will be to create an efficient and complex engine here that produces the near maximum amount of data possible,

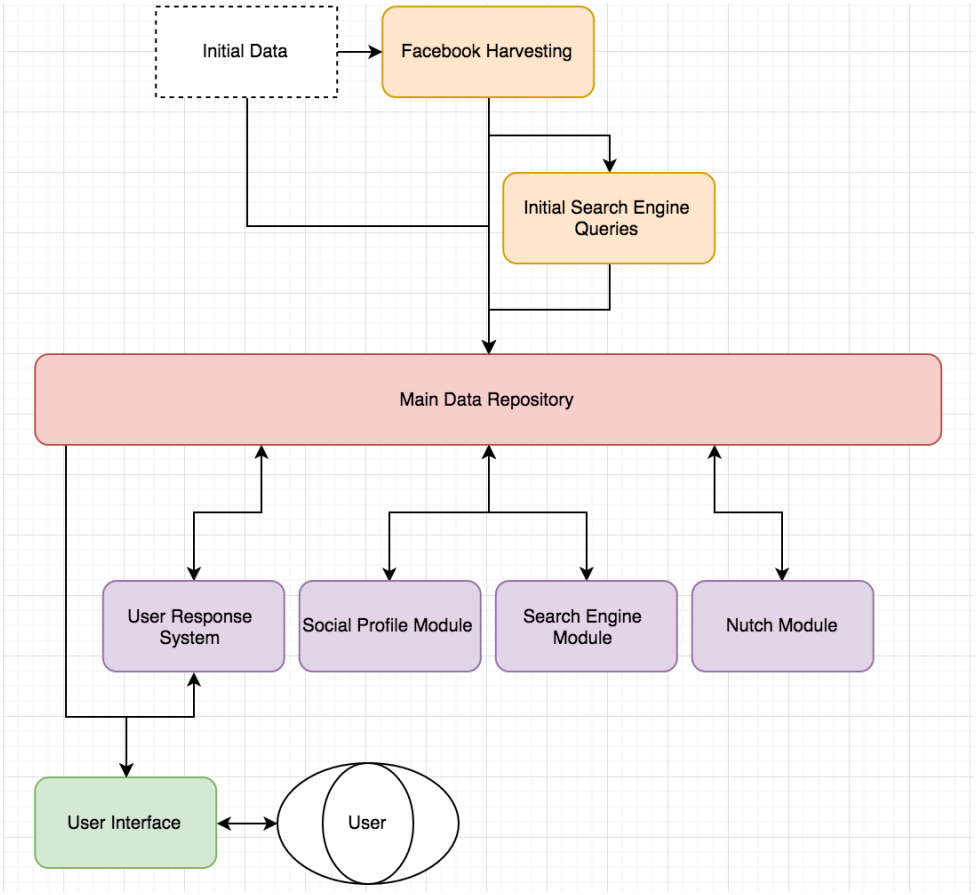


Figure 2: The Architectural Design of the System

however initially a simple rule based system will be built to target key aspects of users.

3.4 Implemented Data Collection

As the system currently stands the only progress to be made has been regarding the first few levels of data collection, as is expected from the gantt chart produced in the specification.

3.4.1 Facebook Harvesting

Since a Facebook profile has been chosen as the seed of the search, pulling as much information from there as possible must be the first step in data collection. A main issue encountered early on with accessing a user's information was that the majority is hidden without logging into Facebook. As previously mentioned though HtmlUnit allows the application to enter login details and retrieve the necessary cookie to browse Facebook from a user's perspective. Currently a parody account

is used to view the profile, and then each possible page where public data could be on display is scraped with the HtmlUnit/Jsoup combination. Manually the field names in the HTML which store required information were located and using a combination of Jsoup selection feature and conditional statements this data was collected. This leads into constructing queries for search engines to branch out to new sources.

3.4.2 Google Querying

A basic system has been created for submitting and handling Google search queries. Using HtmlUnit data can be inputted into the Google search bar and the results can be viewed in HTML format. Later in development this will be changed to constructing the resulting search URL manually to remove the overhead of submitting a form to Google and waiting for a response. A core principle behind the querying is that they occur concurrently to all other data collection, that is a module such as that processing the user's Facebook page can submit a query to the Google module and any other part of the system can access this result at a later data. However this also allows other system modules to bypass going through the main data repository if necessary, accessing Google directly to quickly retrieve search results should it be required.

4 Project Management

Reflection is crucial in an agile, multi-dimensional project such as this in order to ensure efficient use of the limited time available. Despite only a small amount of time so far being dedicated to design and development some insight has been made into the best way to approach the remainder of project.

4.1 Software Development Methodologies

In the specification it was stated that an agile development methodologies would be the most suitable for this project, due to it allowing adaptability should aims change and providing the single developer with greater flexibility. This method is working well however it is becoming apparent that the developer works best when small tasks are set, thought about in detail and then implemented once a clear idea has been constructed. For instance the initial Facebook profile scraping took longer than expected to develop as not enough prior research was done and no plan was created, leading to lots of trial and error.

4.2 Project Timeline

The timeline set out in the specification (figure 3) has not changed for the most part, although the developer is currently marginally behind schedule. Data collection is expected to continue throughout the major portion of development rather than ending after the first few weeks. With the overarching system design and the majority of initial research completed progress on the system can start occur more readily. The majority of the developers external commitments will also be coming to close in the next few weeks, freeing up more time to spent on the application.

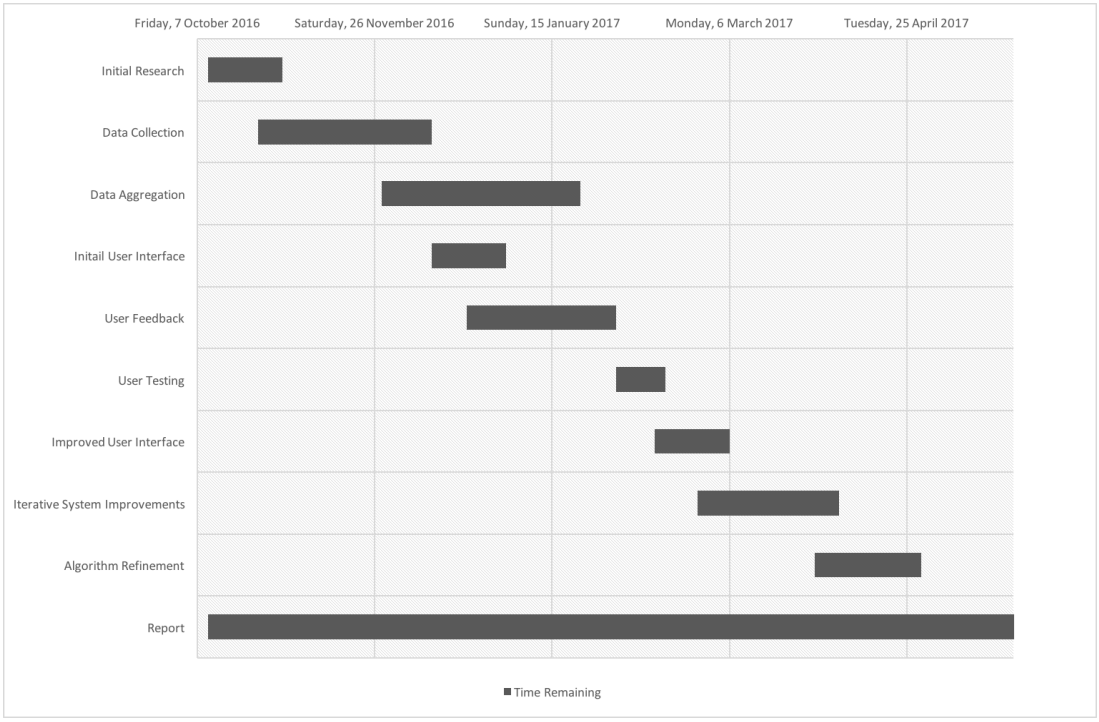


Figure 3: Gantt Chart Displaying Proposed Deadlines

4.3 Tools

A variety of tools were outlined in the specification, namely: LaTeX aided by TeXShop for documentation; Skype for instant communication between shareholders; Sublime Text and Vim for writing code and Dropbox for sharing files between multiple people. Furthering from this, more tools have been selected for both development and communication purposes.

4.3.1 Github & Git

Github is remote repository service that allows code to be backed up online and shared between multiple users. Git is already being used for version control in the system, where every few minor changes are 'committed' and you are able to roll backwards to different 'commits' should you need to revert to a previous version. The local repository is then stored online using Github, so should an expert be required to assist with coding giving them access will be straightforward.

4.3.2 Maven

As previously discussed in the implementation section, Maven will be used when handling dependancies and building of the codebase. Not only does this help with developing, it also allows

external modules to tracked and documented simply. Testing is also possible with Maven, and will be used for unit test throughout production.

4.3.3 Document Storing

Even though Dropbox can be used for backing up and sharing data, at times some documents require storage online only as they may be placeholders or work in progress. Google Drive will therefore be used to store these temporary files, such as diagrams and rough document plans, to keep them safe yet private.

5 Further Work

Although the initial seeds of the end goal system have been proposed, only one of the aims of the project have been started. Over the next few months comes the major development side of the project, which will focus on the following:

- Completion of the initial section of the system, involving harvest the seed profile and performing base queries for the system to work off.
- Begin creating the rules that the main data repository will use to further expand the data collection.
- Implement a data aggregator into the data repository to refine the results from the tasks it delegates to other modules.
- Build an are of the system dedicated to using Apache Nutch and Apache Solr that will crawl a given website from the main repository, as outlined in section 3.3.
- Expand the current search engine module to work more efficiently a produce more meaningful results.
- Design and create a basic user interface capable of both outputting data from and inputting data to the main data repository.

This is not a complete list as all areas of the system outlined in the architectural design, section 3.3, should be implemented into the final product, however these are the core tasks that the developer should be focusing on. On top of this final documentation needs to begin almost immediately to ensure it is kept up to date and is an accurate representation of how the system progressed.

6 Conclusion

To conclude, the progress made so far on the project as outlined in this report has been slow yet still roughly on track with the original timeline set out in the specification. For the most part despite a more concrete plan being laid out the key aims and requirements of the project have not changed.

In the next few months development will begin in full force, and since this is the most challenging part of the project problems are expected to arise. However, the developer feels comfortable that enough time has been allocated to create the required target application.

References

[1] Amazon. Amazon.com statistically improbable phrases. *Available from:* **https://www.amazon.com/gp/search-inside/sipshelp.html**.

[2] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[3] Elasticsearch. Open source search & analytics. *Available at:* `https://www.elastic.co/`.

[4] Mounir Errami et. al. Identifying duplicate content using statistically improbable phrases. *Bioinformatics*, 26(11):1453–1457, 2010.

[5] Facebook. Terms of service. *Available from:* **https://www.facebook.com/terms.php**.

[6] Lutz Finger. Recommendation engines: The reason why we love big data. *Available at:* `http://www.forbes.com/sites/lutzfinger/2014/09/02/recommendation-engines-the-reason-why-we-love-big-data/#7703e004218e`.

[7] Apache Software Foundation. Apache solr. *Available at:* `http://lucene.apache.org/solr/`.

[8] Apache Software Foundation. What is apache nutch? *Available at:* `https://wiki.apache.org/nutch/FrontPage\#WhatisApacheNutch.3F`.

[9] M. E. J. Newman & M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 2004.

[10] Google. Custom search api. *Available from:* **https://developers.google.com/custom-search/**.

[11] GoogleGuide. Part 1: Query input. *Available from:* **http://www.googleguide.com/category/query-input/**.

[12] Hazel Grant. Data protection 1998-2008. *Computer Law & Security Review*, 25(1):44–50, 2009.

[13] Miniwatts Marketing Group. World internet users and 2016 population stats. *Available at:* `http://www.internetworldstats.com/stats.htm`.

[14] Jonathan Anderson & George Danezis H Joseph Bonneau. Prying data out of a social network. In *2009 International Conference on Advances in Social Network Analysis and Mining*, 2009.

[15] Alex Iskold. The art, science and business of recommendation engines. *Available from:* **ReadWriteWeb**.

[16] Tune J. Koetsier. How google searches 30 trillion web pages, 100 billion times a month. *Available at:* `http://venturebeat.com/2013/03/01/how-google-searches-30-trillion-web-pages-100-billion-times-a-month/`.

[17] Athena Stassopoulou & Loizos Papageorgiou Marios D. Dikaiakos. An investigation of web crawler behavior: characterization and metrics. *Computer Communications*, 28(8):880–897, 2005.

[18] restfb. restfb. *Available from:* **http://restfb.com/**.

[19] Hridya Sobhanam and AK Mariappan. Addressing cold start problem in recommender systems using association rules and clustering technique. In *Computer Communication and Informatics (ICCCI), 2013 International Conference*, pages 1–5. IEEE, 2013.

[20] Alexandre Boulgakov & Giordon Stark. Sipping wikipedia. *Available at:* `http://courses.cms.caltech.edu/cs145/2011/wikipedia.pdf`.

[21] Statista. Global social network penetration rate as of january 2016. *Available at:* `https://www.statista.com/statistics/269615/social-network-penetration-by-region/`.

[22] Kelvin Tan. Apache solr vs elasticsearch. *Available at:* `http://solr-vs-elasticsearch.com/`.

[23] twitter4j. Twitter4j. *Available from:* **http://twitter4j.org/en/index.html**.

[24] X. Zhou & L. Li W. Xu. Inferring privacy information via social relations. In *International Conference on Data Engineering*, 2008.

[25] Noortje Marres & Esther Weltevrede. Scraping the social? *Journal of Cultural Economy*, 6(3):313–335, 2013.

[26] Pamela Vagata & Kevin Wilfrong. Scaling the facebook data warehouse to 300 pb. *Available at:* `https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/`.

APPENDIX

THREE

PROJECT PRESENTATION

Online User Privacy Investigation Using Social Profile Seeding

By Adam Coles
Supervised by Dr. Matthew Leeke

"Bosnia. They don't have roads, but they have Facebook."
- The Social Network (2010)

"The question isn't, 'What do we want to know about people?', It's, 'What do people want to tell about themselves?'"
- Mark Zuckerberg (2011)

65,000,000

32,000,000

15,800,000

Percentage of Users of Multiple Social Medias

%					
	-	29	39	36	33
	93	-	65	54	48
	95	49	-	48	54
	89	45	53	-	43
	92	38	57	41	-

Source: Survey conducted March 7-April 4, 2016.
"Social Media Update 2016"
PEW RESEARCH CENTER

Expanding Social Media Purpose

"Twitter is a service for friends, family, and coworkers to communicate and stay connected through the exchange of quick, frequent messages."

"Instagram is a fun and quirky way to share your life with friends through a series of pictures."

"Our mission is to connect the world's professionals to make them more productive and successful."

Potential Privacy Exploits

- Studies at University of Warwick
- Went to Cardinal Newman College
- Lives in Kirkham, Lancashire
- From Milton Keynes

Personal Information

- Name
- Places lived
- Places worked
- Family members
- Contact details

Potential Privacy Exploits

Images

- 8.7% of all Facebook users are fake using other peoples images (1)

- Currently no automated checking method for these fake accounts

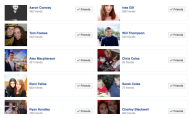
Potential Privacy Exploits

Speech Habits

- Can predict with 83% accuracy one author out of 20 from just 30 messages (2)

- Sentiment of tweets scan be generated with 91% accuracy using simple K* classifier (3)

Potential Privacy Exploits



Friendship Groups

Hobbies/Interests

Protecting Yourself

Which social media is least safe?

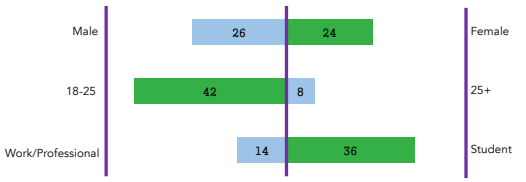
What information is necessary to link social medias?

What information should never be public?

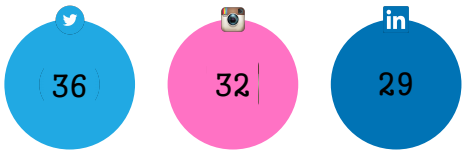
Data Collection

Test Data Set

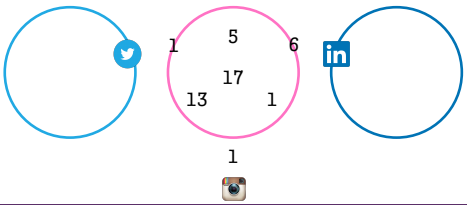
Test Data Set



Test Data Set



Test Data Set



Basic Alias Analysis

Platforms	Total	Same	One Diff	Similar
	30	11	3	8
	36	1	3	5
	32	0	5	8
	22	2	0	0

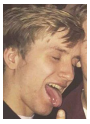
Facebook to LinkedIn

facebook		Linked in
Full name (shortened)	⟷	Full name
Most recent job history		Complete job history
Basic education history	University ⟷	Detailed education history
Multiple profile pictures		Single professional photo
All types of connections		Connections are worked related

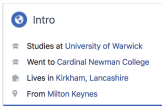
Facebook to LinkedIn

Scrape Basic Previous Education → University + Name → ~100% Accuracy

Preliminary User Data Mining



Target Face



Base Facts

Simple Rules to Locate New Sources

Examining Alex (omitted)

Diving Deeper With Twitter

Friendship Analysis Through Likes

Basic principle: The people you like the most are more likely to be your friends

Look at multiple factors:
- Do you follow them
- How many times do you like
- The average like ratio
- Their number of followers

Friendship Analysis Through Likes

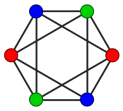
Classify each person as the following:
- 1 – Not known, celebrity you dislike
- 2 – Celebrity you like
- 3 – Acquaintance
- 4 – Friend
- 5 – Good friend

Apply machine learning in combination with previous factors to predict user relationship

Graphing Twitter to Cluster Connections

Creating a follower graph can serve many purposes:
- Clique friends
- Find small disconnected groups
- Locate outliers
- Find which groups contain top users

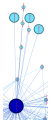
Combining like and cluster analysis provides meaningful results



Examining Inez (omitted)

Purposes of Friendship Analysis

Locating Profiles Through Friends



Finding Main Friendship Groups

Persistent Links to Users

Project Management

Concept and Methodology

Project has been altered significantly since conception, through both attempts at application and research made

Agile development approach so this did not hinder the schedule

Weekly meetings with supervisor to discussed the change of direction of the project

Issues Faced

Issue	Resolution
Facebook/Google do not have an open APIs	Use a browser emulator in combination with alias account and html parser
Twitter's API is rate limited	Batch run data collection over night for multiple users and store in local cache
It is ethically wrong to invade people's privacy	All users in the test data set gave their permission to be involved in the study
Image recognition hard and unreliable task for a computer to perform	Return links to photos to be manually checked by humans
Websites that rely heavily on AJAX are difficult to emulate and scrape	Avoid these websites for this study due to time constraints, for instance Instagram

Old	Data Collection	Data Aggregation	Interfac	User Test	Iterative	Refinement
	Nov	Dec	Jan		Feb	Mar
New	Research	Relations Discover	Data		Featu	Refinement

Constructing Results

Privacy Recommendations

Create Different Aliases

Make Twitter Private

Avoid Educational/Work Information

Keep Personal Websites Secure

Other Applications

Law enforcement can use the findings to both find criminals as well as impersonate them when captured

Researchers can use the findings to help them collect data on willing individuals

Informing others, particularly young people, on social media safety

Further Work

Continuing refinement of Twitter analysis

Adding similar analysis of likes and friendships to Instagram, although in need of heavy altering

Scraping and aggregating data with greater consistency across sources

Questions?

S. Fong, Y. Zhuang and J. He, "Not every friend on a social network can be trusted: Classifying imposters using decision trees," *The First International Conference on Future Generation Communication Technologies*, London, 2012, pp. 58-63.

Rong Zheng et al., "A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques", JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 57(3):378–393, 2006

J. Fiaidhi, O. Mohammed, S. Mohammed, S. Fong and T. h. Kim, "Mining twitterspace for information: Classifying sentiments programmatically using Java," *Seventh International Conference on Digital Information Management (ICDIM 2012)*, Macau, 2012, pp. 303-308.