Department of Computer Science

Automatically Identifying Employment Relationships in Social Media

Sonam Dossani

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2019 | CSMSC-19

Declaration:

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Sonam Dossani, September 2019

Executive Summary

Social engineering attacks are a form of cyber-attack in which human interactions are exploited using malicious strategies. They take advantage of psychological weaknesses in humans to manipulate them into giving away data or access to an organisation's system. The first step in any form of these attacks is detecting an employee target to trick, and so preventing their detection could avert the onset of the attack. The aim of this project was to develop a system that could identify who the employees of organisations are, from a dataset containing all of the Twitter affiliates of these organisations. Then, the features which the system used to differentiate between the employees and non-employees were used to develop countermeasures that could be implemented as a prevention strategy against identification.

The organisations studied in this project were newspapers (employers) and their journalists (employees). A dataset containing 200 employees and 450,000+ non-employee Twitter affiliates of these newspapers was composed to test the aim, and the machine learning (ML) models with the best performances were used to calculate each feature's importance and develop countermeasures.

This project drew together ideas of features from previous work, and also executed its own additional forms of data exploration and TF-IDF analysis of individuals' text posts to identify further features that were novel to this area of research. The implementation involved a much larger dataset than had been used in the past, a novel set of features, and a comparison of the performance of multiple ML classifiers with the same dataset.

The contributions of this project are as follows:

1) A script produced for this project automatically scanned 150,000 webpages for verified employees (ground truth), producing a curated dataset representative of local newspapers across the country.
2) This project has explored a novel hypothesis, that tweets can be more informative than other aspects of a profile in identifying employees. It has implemented, from scratch in python, a TF-IDF analysis of over 45,000 tweets to inform tweet-related features to test this hypothesis.
3) This project has produced methodical Python scripts for every step including: scraping websites to collect employees, extracting affiliate data from Twitter's API, TF-IDF analysis, feature extraction from each affiliate's profile data, data pre-processing, cleaning noise from the dataset, ML classifier training and testing, feature importance evaluation, correlational analysis of features, and producing graphs to visualise results.
4) This project produced a detailed workflow of data collection and training and testing of four ML classifiers, for evaluating their efficacy in identifying the employees from the overall data pool, and to find the classifier best suited to this particular problem.
5) This project has provided a collated list of realistic countermeasures which were developed from the features which were most useful to the classifiers in identifying who the employees were from the affiliate pool. Their applicability has been assessed by a thorough evaluation of each measure.
6) This project has improved upon previous work in this field in three ways: a) an implementation including a much bigger test group of verified accounts b) a novel list of features including textual analysis of tweets for new features c) comparing results from multiple classifiers.
7) This project has been able to produce a precision of 0.52 for this system, indicating its real-life applicability for cyber-attack screening.

Acknowledgments

Throughout the entire process of research, carrying out the work, and writing this thesis, I have received a great deal of support and encouragement.

The first person I would like to thank is my supervisor Matthew Edwards, who made himself available to answer my many questions and provided me with help for every step of the process. You supported me greatly and I have been able to learn a variety of new skills thanks to your help.

I would also like to acknowledge my co-supervisor Miranda Mowbray who was there to help me at the start of this project with my research proposal, which really helped me to determine a clear direction to go in. I will always be fond of our off-topic cybersecurity conversations after our meetings.

Next, I would particularly like to thank my personal tutor Steven Ramsay for moral support throughout the entire academic year right until the end, with which I was able to have the enthusiasm to keep working my hardest.

In addition, I would like to thank my friends and peers from my course. We were all in the same boat, and it was lovely having such a friendly, hard-working group of individuals to inspire each other throughout the project process.

Finally, I would like to thank my family, for their constant support and advice throughout this year, and for always being available to distract me when the work became overwhelming.

Contents

Chapter 1

Introduction & Project Aims

1.1 Motivation

Social engineering attacks encompass a form of malicious tactics where users of a system are tricked using psychological devices, to help the attacker obtain information that grants them access to that system [27], [35]. As well as breaking into a system using technical expertise, data or access to a system can be acquired through the use of trickery and manipulation of somebody authorised to access that system [6], [27]. Organisations store valuable data on their computer systems, whether it be an intranet or private database. Although access to this data is often protected by strong firewalls and other security measures, there is usually some human intervention that can be used to access the data e.g. a password. This can lead to attackers victimising the weakest part in the security chain: the employees with access to these organisations' data. Attackers can scour the internet and social media networks to deduce information about who the most vulnerable employees of a company are and form an attack plan. Here, vulnerability refers to those employees that might be the easiest to impersonate or manipulate based on how much of their personal information is available online. For this reason, countermeasures need to be implemented to prevent employee identification.

Once an attacker obtains personal information about an employee, access to company resources can be gained quite easily. A well-known example of this is the Ubiquiti Networks hack of 2015, where the attackers used the personal information of multiple executive employees to impersonate them: they targeted the finance department of the company while undercover, and sent emails requesting wire transfers be made to alleged company accounts, which were actually controlled by the attackers. In total they were able to steal over 45 million dollars from the company, by making orders while impersonating a member higher up in the company hierarchy [37]. Cyber-attacks are costly and troublesome for companies to deal with, so identifying sources of vulnerability and taking measures to limit these can greatly reduce the resources used to investigate these attacks.

Existing countermeasures used by companies include training programmes which inform employees about how much information, if any, is safe to share online [11]. Additionally, attack detection models are used in some situations such as call centres which help individuals identify when they are being victimised [32]. These measures are useful for dealing with ongoing attack threats, after the employees have been identified by the attacker.

A crucial factor that needs to be addressed is that employees make themselves detectible from non-employees based on the amount of personal information they willingly share online e.g. placing their company's name in their social media biographies. The first step is to identify the aspects of people's personal information found online that increase the ease of their identification as an employee of a target company.

The more employment relationships the attacker is able to deduce about a particular company, the more vulnerable its employees would be to manipulation. For example, by gaining access to an employee's ID, a phishing attack can be delivered in the form of an email or chat that impersonates that employee, asking a superior for sensitive information. If the employee data isn't readily available in the first place, this makes the attack much more troublesome to carry out.

By reducing the ease of obtaining personally identifying information about its employees from public social media, a company can make it significantly harder for an attacker to deduce email

accounts or sensitive pieces of information. Furthermore, these days planning attacks on social networks like Twitter are more difficult because the websites have security measures preventing spam-type messaging to hundreds of accounts. Instead, attackers need to carefully select their targets to be people that would be more likely to provide them with the information or access they're trying to obtain.

The aim of this research was to deal with the issue from the root i.e. prevent employees being identified as employees in the first place. It first developed a system that could automatically identify employee relationships from the social network Twitter, and then worked backwards from the effective identifiers to suggest protective measures that can be put into action. These countermeasures were formed using data from analysing the features that made it easier for machine learning classifiers to deduce employment relationships. Previous work [21], [43] has produced promising results with proof-of-concepts of this study, with samples sizes of 20 and 60 employees. This project aimed to expand the research into a wider sample that could provide further insights, using both a novel feature set inspired by tweet analysis and different classification approaches.

1.2 Research Questions

Following a background review within the field of social engineering attacks, and an evaluation of previous studies similar to this one, the gaps in the literature became evident and from this a clear pathway for this project was formed. With the aims and objectives in mind the following research question was identified, which this project endeavoured to answer through the methodology detailed in *Chapter 3*:

1) **Can a machine learning classifier be trained to use a group of features from the social media affiliate accounts of a company, to identify the individuals that are employed by that company?**

After researching and answering this question, this project worked towards the second goal which was to analyse the outputs of the classifier to identify which features, if any, added more weight to the classifiers ability to identify the employees from the pool of other affiliates. The second research question was as follows:

2) **Which features, or groups of features, are better predictors of employment relationships, and are these better to a substantial degree?**

Finally, the project aimed to answer the final research question, which was contingent on the two previous research questions having significant results. The final question was:

3) **Will exploring the results of the classifiers output and feature analysis aid in the generation of functional countermeasures? If not, what can we learn from the results of this project, when comparing the results to that of previous proof-of-concept research?**

1.3 Deliverables

The deliverables for this project are as follows:

1. A literature survey focusing on the current problems in the area of social engineering attacks using employee detection, and the success rate of measures that have been implemented and tested in the past.
2. A system created using machine learning approaches to identify the social media accounts of employees from a company, identified from the social media accounts affiliated with that company.
3. A discussion of the contribution each feature makes to the ability of the model to discern who is and who isn't an employee of the company being examined.
4. Realistic countermeasures based on the most important features to the model, that may protect against social engineering attacks where employees are the target.

1.4 Hypotheses

Hypotheses are useful for keeping the motivation of the project clear, and to compare the outcome of the project to, to observe if it has progressed in the manner that was expected.
The following hypotheses were tested:

1. Dependent on the quality of the ground-truth used to train the classifier, the model will be able to deduce, using a group of features and the social media affiliate accounts of a company, the individuals that are employed by that company. This will be tested by calculating an F1 score, which favours low false positive and low false negative results.
2. One or more of the features (or groups of features) used during testing of the model will be better predicters of employment relationships to a noteworthy degree.
3. Exploring the results of the classifiers output will aid in the generation of functional countermeasures. For each of most predictive features, this research will produce an idea of how protective measures might be used or decide that the feature not easy to protect against.
4. Features that are extracted from the most recent tweets of each user will provide supplementary insights into what makes an individual's profile easier to identify as being an employee profile vs non-employee profile.

1.5 Objectives

This project sought to explore whether employment relationships could be automatically inferred from social media accounts affiliated with a company. In particular, it aimed to use machine learning (ML) approaches to train a model to be able to deduce which affiliated individuals are employed by a company, based on a group of features. These features were extracted from profile data for each individual, and also from a set of their most recent tweets. After training the model with these features, it was used to categorise an unseen set of employee profiles, and its ability to do this accurately was assessed. Ground truth data about employment relationships was harvested from

business directories, and employee/non-employee profiles for training and testing the ML model was acquired from the Twitter API. After testing the employment identification abilities of the model, the second aim of this project was to analyse the specific features that made some employees more vulnerable to identification. These features were used to identify specific countermeasures that can be implemented to make it more difficult for these employment relationships to be deduced by an attacker. The core objectives were:

1. Develop a system that can take as input the total list of affiliates and associated profile features for a company, and as an output produce a list of Twitter accounts (from the affiliates) for people it believes are employees of that company.
2. Collate and discuss the features of a profile which facilitate the detection of who is employed by a certain company. These features will be subject to an analysis of importance to isolate those which were most useful to a classifier.
3. Use the feature analysis to identify countermeasures that a company can employ to protect against vulnerabilities that lead to social engineering attacks.

1.6 Project Novelty & Scope

In order to understand the novelty of this project, a brief description of some similar work in the literature is described here. Details are in *Chapter 2 - Background*. Two previous studies were part of the motivation for this research [21], [43], and they both trained ML classifiers to be able to recognise the employees from a pool of affiliates of the social media accounts of some companies. One study was by Edwards et al. [21] who produced a proof-of-concept with a sample size of 20 verified employees from the affiliate pool, and a range of features extracted from the profile data from Twitter. While their results were promising and informative, improvements could be made in terms of sample size for generalisability, and the types of features used for their research. Similarly, the second study that motivated this project was conducted by Shindarev et al. [43], with a sample size of 60 verified employees who use a popular Russian social networking site called VK. Neither of the two studies explored multiple features extracted from the sentiments of users, either in the form of tweets or written posts, which highlighted this as a potential gap for this project to explore.

The novelty of this project is three-fold:

1) This project created a larger dataset for machine learning classifier training and testing. Due to ethical and security reasons this dataset couldn't be one of the deliverables for the project, but the aim was to acquire a larger set of verified employees than previous work has done, with a target of approximately 200 individuals. These employees made up only a small percentage of the overall affiliate pool which contained over 450,000 non-employee affiliates.
2) This project included analysis of people's tweets as part of feature analysis. To train the classifiers a range of features were extracted and each individual in the dataset had a value for each feature. Previous work had not conducted a detailed analysis of tweets to acquire features from them, but this project integrated tweet analysis to attain greater feature variety.
3) This project trained and tested multiple classifiers on the dataset and not just one, because this allowed for the best classifier for the problem to be chosen. The importance of the outputs of the classifiers is described in *Chapter 4 - Results*.

The scope of this project was limited to data collected from the social networking service Twitter, consequently limiting the applications to Twitter attacks only. This was due in part to recent changes in privacy laws that have created difficulties in obtaining data from other social media platforms. It was also due to time constraints for this project.

Additionally, different personal relationships present themselves on social media, and they all have different styles of interaction e.g. between friends or between co-workers [15]. Since the model was trained using only employee data, the scope of the project was also limited to identifying employment relationships from social media interactions and no other forms of personal relationship.

1.7 Conclusion

This project addresses the problem of employee detection in social engineering attacks, particularly exploring the strategies of gathering sensitive personal information that are used by attackers. By training an ML model to be able to identify the employment relationships in a company only from social media data, this project provided insights into the difficulties, or lack of, that an attacker would face when trying to collect personal data. It focused on whether the ML model was able to correctly infer employment relationships, with as few false positive results as possible. Consequently, the project can be classified as being in the area of social engineering research, as existing ML approaches were applied when training each model to test out the hypotheses.

This project has supported previous findings [21] that employment relationships can be automatically deduced from publicly available data on Twitter, which has implications for company security in social engineering attacks. The system has the potential to be used in an analysis setting, where a company can assess their level of vulnerability in terms of the number of employees who are publicly exposed. The novelty of this project is found in its larger profile dataset, use of multiple classifiers, and use of different features than previous work in this area. Furthermore, this project analysed the most important features used in the model to help form viable countermeasures. They are discussed in relation to others that have been suggested in the literature for protection against social engineering attacks where employees are the target. Feature analysis has been able to provide some insights into why certain countermeasures are more protective than others, and may provide future research with the foundation to explore these automatic detection methods using other social media networks aside from Twitter.

Chapter 2

Background

2.1 Introduction

This section of the thesis discusses literature related to social engineering attacks, and particularly considers how identification as an employee is a key factor that affects a person's vulnerability. It dissects how the techniques to be used in this project may be useful for investigating whether specific features about a person can be useful in determining whether they are an employee of an organisation or not. The problem of exploiting employees is examined as well as why countermeasures are essential for prevention of such attacks. Moreover, this section discusses how the machine learning classifiers to be trained and tested are appropriate for investigating the research questions. Finally, the previous work which was the motivation this project is reviewed in order to understand why this project builds on their research and how it does so.

2.2 The Problem

A succession of social engineering attacks have targeted high profile companies such as Sony [16], Target [51], and Yahoo [36] in recent years, emphasising the imperative need for mitigation strategies. One reason for adopting social engineering attacks over more technical methods of exploiting a system is due to the attacker requiring less knowledge about security weaknesses in that system. It is much easier to exploit psychological vulnerabilities of individuals with an implied knowledge or access to confidential data [5]. Another reason is that some systems require human verification before allowing access, after which an attacker could use his technical expertise to cause difficulties [26].

As a common countermeasure, managers are reminded in security briefings to be wary of what they post on social media, because criminals trawl these platforms for employee data [1]. This data can then be used to target employees, such as those in the well-publicised Snapchat attack which took place in 2016. During this event, an HR employee was detected and targeted with a spear phishing email, a form of attack that targets a specific individual [8]. The attackers impersonated Snapchat CEO Evan Spiegel, and requested payroll information of over 700 people, which the employee provided [10]. The implications of this were enormous as social security data was leaked, which criminals often exploit for identify theft purposes [47]. There are many types of identity theft including Tax ID theft, which was the method used in the Snapchat attack to steal social security numbers, and Social ID theft, where data is extracted from social media for impersonation purposes [47].

The central issue at hand is the free availability of personal information on social media networks. The amount of information an attacker can obtain about a target individual or company is directly associated with the success of a social engineering attack [12]. Humans are largely susceptible to the forms of psychological manipulation practiced by these attackers, supporting the idea that employees are the weakest part of the security chain [7], [34], [41]. On a social media profile, a typical user may record their email address and offer details about their prior education, friendships, and professional history. Details such as these can be used in employee detection, which is where

certain features about an individual's online presence exposes them as an employee. In the case of attacks with employee detection, the professional history is screened, which is key in formulating a targeted social engineering attack [25], [29].

2.3 Relationship Detection Methods

An attacker's ability to detect a suitable target is a crucial part of formulating a successful social engineering attack [50]. This is because the vulnerability of a company is partly determined by the employees' responses to attack attempts [7], [13], [26]. When an organisation is targeted, a major criterion for a suitable target is that they are an employee, since it is employees that have access to systems or valuable knowledge. Distinguishing employee accounts from non-employee accounts in online social networks is not an easy task because people do not "label" their relationships to those they connect with e.g. friend, sibling. However, social ties can be inferred from the behaviour observed between users, and employee interactions have their own characteristics.

For example, Diehl et al. [15] created a ranking approach to confirm that one can identify manager-subordinate relationships by examining email interactions. One of their methods looked at traffic patterns from accounts that interact with each other i.e. the proportion of emails sent and received. The manager-subordinate relationship is comparable to the employer-employee relationship that this paper aims to identify. This paper will focus specifically on "follow behaviour" (among other features) instead of email interactions, but Diehl et al. [15] provide proof that inference of employee-type social ties is possible, which provides a basis for this study. Furthermore, though their work was based on email communications, the behaviour patterns are similar to how a user would interact on social networks such as Twitter, providing further motivation for this research.

The literature also references other types of relationship inference. Tang et al. [45] conducted a study where each user was modelled as a node, and the model estimated the probability of relations to other users within the social network. They built a framework that was able to automatically infer the types of social ties (e.g. colleague, friend, family) using a learning algorithm on partially-labelled datasets. To deduce a relationship, attributes such as 'frequent contact during work hours' was used as a determinant of a colleague relationship. Like the previous study [15], they also were able to identify manager-subordinate relationships from their email dataset, where accuracy and precision scores were 88.0% and 88.6% respectively. This similar result, despite using a different method, supports the feasibility of their technique for automatic inference of relationships from social network data.

Tang et al.'s study [45] is limited due to the partial labelling of relationships in the network, which is not always representative of a real online situation where there may be no labels present. For example, on the social network LinkedIn the partial labelling method is relevant because users can fill in labels on the site e.g. names of previous employers. But, for networks such as Twitter, labels are not used in any capacity. This provides a motive for unlabelled networks to be researched, to see if other factors such as profile features can be used to identify the employee accounts from a pool of the employer's Twitter affiliates. Therefore, to avoid this limitation the ML model in this research will only be probed with unlabelled test data after it has been trained i.e. none of the relationships in the test dataset will be known to the model prior to testing.

2.4 Automatic Detection of Employee Accounts

There has not been much work in the area of automated employee detection, however the research conducted by Edwards et al. [21] suggests that this kind of social media analysis is possible. They demonstrated an automated tool that could be used for data harvesting of employee information from social media sites using only data that is accessible to the general public. This indicative result was discovered through a method that considered specific features of an online presence to see what an indicator of an employee account is. Some noteworthy features included: bidirectional follow behaviour, mentions of the employees on the organisation's website, and accounts who are networked with each other as well as the employer account. Applying these features to a decision tree classifier they produced an F1 score of 0.65, which is a measure of the balance between the precision and recall of the classifier. This is encouraging in the motivation for this project both in that the score is > 0.5, indicating a positive result, and < 0.7, indicating room for improvement.

The outcome of the study was a proof-of-concept which was able to show the feasibility of the automatic detection method, but the limited sample of 20 verified employees can be developed in further research. This is one aspect that this research aims to take into account.

Furthermore, a similar study by Shindarev et al. [43] was able to come to a comparable conclusion (with an F1 score of 0.77) using another decision tree classifier on a popular Russian social network VK.com [3]. They demonstrated that company employee accounts can be separated from non-employee accounts using explicit features in an automated fashion, although the study was limited by the small list of features used by the classifier (5). Using a greater number of valid features would increase their accuracy of identification, and this improvement will be taken into account for the methodology of this research.

Although Shindarev et al. [43] used a different set of classifying features to Edwards et al. [21], it is evident that the common method used for the task is effective, and the features used may have been more suited to the social media presence in their country. Likewise, they were able to conduct their study on a completely different social media network (VK.com) and still came to the same results as Edwards et al. [21], showing the feasibility of research using analogous techniques. This also provides a scope for this research, and for exploring novel classifying features which may be used in combination with those used previously, [21], [43] to provide new results.

2.5 Understanding the Vulnerabilities

To construct countermeasures for a social engineering attack, it is important to first recognise the triggers which are used in such attacks to influence the victim's cognitive state to extract sensitive data. Bilge et al. [7] discuss seven psychological vulnerabilities first defined by Gragg [26], which they used to build a social engineering attack detection model (SEADM). The first vulnerability discussed is 'strong affect' [17], which describes how an individual's cognitive abilities are impeded when strong emotions are elicited e.g. the panic they may feel during a phishing attack where they are threatened to have their account frozen. Another interesting vulnerability is the 'diffusion of responsibility and moral duty'. This occurs when a person is coerced into thinking that their rule-breaking actions will have beneficial consequences e.g. revealing private information that they think will help the company. The individual typically believes that they will not be exclusively held responsible. A final, more common vulnerability is known as 'authority'. If the attacker can impersonate or lead the victim to believe they are higher up in the company hierarchy, then the victim is much more likely to comply with any requests. This vulnerability, combined with the fear of being punished for not complying, leads to a very robust social engineering attack [11], [50].

To combat the effects of these seven psychological vulnerabilities, [7] propose an attack detection model. It uses a decision tree to aid in deciding whether a particular request is suggestive of a social engineering attack. Literature has suggested that measures like these are a key asset to prevention of attacks, because at time sensitive moments during the attacks, people have difficulty making coherent choices. Using a decision tree model to identify which aspects of the situation might indicate an attack can be an objective tool that an individual can easily use under pressure [12].

It is essential to note that applying human behaviour to general-purpose models has its limitations: human behaviour is too complex to be categorised into these seven states, and under time-sensitive situations, we cannot predict the decisions they will make. The human-reasoning process is affected by many other factors unique to the individual.

2.6 Automating Social Engineering Attacks

The next step for constructing countermeasures involves having an understanding of how the actual attack is carried out once the attacker has identified their victim. Previous work has detected how social engineering attacks can be automated [45]. Despite this approach revealing the weaknesses that are exploited in a social engineering attack, the information gathered can be useful. Evaluating how an attacker can execute an attack can be vital to understanding how to prevent it. Additionally, knowing the demographic of the most vulnerable users can be valuable for targeting prevention strategies towards them.

This project aims to create a system to automate the identification of employee relationships. This could be used in an attack setting when trying to decipher which employees make themselves vulnerable to attack by publicising their work-related information. The following section discusses how automation has been structured in the past for similar purposes, which can be useful for advising countermeasures.

Automating attacks has been evaluated as being an easy way for an attacker to carry out identity theft or cloning attacks on peoples' online personalities to gather sensitive information. Bilge et al. [8] look at two different methods of automating attacks, the first of which involves exploiting a psychological vulnerability of the victim, as discussed earlier. This vulnerability is known as the 'deceptive relationship' [11], and is the tactic used to establish a relationship with the purpose of creating trust, as individuals who know one another are more lenient when it comes to sharing information. The authors [8], [11] describe how by creating a fake account of the victim and sending friend requests to their friends, the attacker can access personal information of the victim much more easily. After automating this type of attack using a crawler, they carried out real-world simulations of their tool on websites such as XING. The key result was that the acceptance rate of a friend request from a clone account was over 60%, despite participant being forewarned, indicating the high value of automated crawlers to an attacker.

This also supports the idea that attackers should impersonate real-life profiles rather than creating fake identities, as cloning appears to create a level of trust with the victim's friends. Linking this to social engineering attacks, a victim is more susceptible to a form of attack where they can be persuaded they are in contact with an authority figure e.g. a higher up employee in the company [50], such as with the Snapchat attack outlined earlier [10]. This provides motivation to construct countermeasures for employee detection, as an attacker first identifies someone to impersonate, which is a step that can be prevented with the right knowledge.

A caveat of creating and testing automated tools similar to Bilge et al.'s that actually interact with people online is that ethical approval is requested from participants whose accounts are being

subjected to the crawler [8]. This makes the situation less representative of a real-life attack, where the victim might notice the fake profile trying to befriend them and then block the account. In this research, informing test subjects that their social media profiles would be subject to an employee detection tool would affect the results because they could filter parts of their profiles that expose them as employees. However, studies like these add value by explaining how automated tools can be used easily by attackers on bigger social media networks such as Facebook and Twitter, and the methods that are employed.

2.7 Detecting Targets Using Twitter Features

With a similar methodology to this project, the work by Seymour and Tully [42] used specific features about users Twitter profiles to produce targeted spear-phishing tweets with a neural network, followed by automating attacks on those users. Their procedure made use of recurrent neural networks that they augmented using a clustering technique, to better identify the targets who would produce the best result. The Markov models they used produced targeted tweets by looking at co-instances of words in the training dataset. Their tool was built to tweet phishing posts targeted at certain individuals who they identified based on their account details and personal information provided by the users on their profiles. Only those users who were deemed high-value were targeted, highlighting the usefulness of countermeasures for a company to know who they need to protect the most.

Interestingly, their reasoning for choosing limited targets rather than simply targeting a long list of individuals lies in the fact that Twitters' security measures would have their accounts terminated. This provides further motivation for attackers to carefully select their victims using public information about them.

Some features they extracted from user timelines included frequency of posting, topics users post about, sentiment analysis of tweets, and the overall amount of information revealed on their profile. They also tried to extract patterns of Twitter behaviour. Using all of these parameters they were able to manually determine targets who satisfied a higher level of vulnerability, based on data collected from past spear phishing victims in the study. They found that after testing their model on 90 profiles, their automated spear phishing assessment had a success rate of between 30 and 66%.

The significance of their research for this project is that it provides positive reasoning for further investigating which features of an employees' Twitter profile will make them targets for social engineering attacks. Although the work by Seymour and Tully [42] examined some different features to what this project intends to look at, their successful outcome is motivating for this project.

Conversely, their small sample size of 90 profiles is much lower than what this project intends to use. A smaller dataset is favourable because it is easier to reduce any noise in the dataset that may contribute to errors in the machine learning stage. Scaling this research up to the much larger predicted size of this projects' dataset (> 100,000) is difficult, and so any positive results should be carefully considered. Furthermore, their tool used extracted features to place users into a cluster, and then only target those who were placed in a cluster that the investigators identified as being easier to phish. This technique can be critiqued due to the manual intervention necessary to decide which cluster should be targeted. Perhaps a more appropriate approach would have been to target all of the clusters separately and then compare success rates between them. To remove any such biases, this project will not pre-train any classifiers on which features lead to vulnerability and will instead allow the machine learning classifiers to produce an output detailing which features had higher weights in

their algorithms. Nevertheless, their success in using personal information from Twitter to train a model to generate spear-phishing tweets for each specific user is encouraging.

Furthermore, a well-known social engineering attack vector involves collating victim profiles across different social network sites to gather as much detailed information as possible [28]. The more information an attacker has, the more likely they are to have a successful outcome. Therefore, pinpointing the features which lead to identification can also suggest how social engineers identify accounts belonging to the same user across different networks. Goga et al. [24] looked at precisely this form of identification in their research, and they conducted an analysis of the features across three networks, including Twitter, to see what the most powerful features for identification are. They compared the following three features: geolocation, timing, and natural language of posts. By assessing the importance of each feature in their model's ability to collate profiles across three networks, they achieved a true positive rate of between 6% and 10%. This appears low, but if scaled up to assess the vulnerability of a company with 1000 employees, this would provide them with 60 potential victims. Although their methodology is different in the sense that they already have the victims identified, their results are still revealing because they show that people's posts on Twitter contain enough data to allow them to correlate accounts. If Twitter posts have been shown to be so informative, this leads to the hypothesis that an analysis of tweets will be able to provide additional features that can separate employee and non-employee groups.

Their results have been able to further inform the direction that this project should take in terms of including an analysis of tweets, because they have been demonstrated to contain rich feature information that can be revealing.

2.8 Choosing Machine Learning Classifiers

The goal of this project is to train and test multiple machine learning classifiers on their ability to discern who the employees are from a list of affiliates associated with an organisation. This next part of the literature review looks at various classifiers and what makes them appropriate for use in this project. It will take into account the type of data that previous research has used and the type of data this project intends to collect and use. It will also look at classifiers which have been used to investigate similar classification research questions to this project.

Knowledge discovery in databases (KDD) is an extremely valuable field of work in which techniques are developed to extract knowledge from large datasets. [22] describe the usefulness of analysing and extracting patterns from data using emerging tools rather than the out-dated ways of manual analysis. Data mining is just one step in the process of KDD which uses algorithms to discover patterns in the data. The steps from start to finish are: 1) selection of the data 2) pre-processing 3) transformation 4) data mining 5) interpretation and evaluation. This generalised process is what will be utilised in this project, and the data mining tools evaluated below will be used for step 4.

Decision Tree Classifier

The first classifier that was investigated was the decision tree classifier. It was an appropriate choice considering that the studies which motivated this project [21], [43] also made use of this classifier. As Edwards et al. [21] note, the decision tree classifier is valuable because of how easily its output can be understood. By using existing libraries which map the nodes throughout the tree, the

output can be displayed as a diagram which one can use to clearly identify which nodes are leaf nodes and which path will lead to a certain classification.

Friedl and Brodley [23] highlight the other advantages of the decision tree classifier such as the input data being suitable even if non-parametric (i.e. it is not required to fit a normal distribution), its ability to handle both continuous and categorical variables, and its explicit output with widely interpretable classification structure. In their research, three varying datasets of land cover mapping were used to evaluate the performance of the decision tree classifier for three different classification problems. For comparison purposes, two additional non-decision tree algorithms were tested. Their procedure even involved a ten-fold cross validation technique, which is also the basis for this project's methodology. Their investigation found that overall the decision tree algorithms produced consistently greater accuracy in classifying areas of land cover than either the linear discriminant function (LDF) or maximum likelihood classifier (MXL) algorithms. The accuracy difference was close to 9% when comparing the best decision tree's accuracy with the LDF and MXL. In conclusion, this result combined with the previous use of decision trees in the studies which motivated this project [21], [43] provide positive reasoning for including this classifier in this project.

A caveat of using this classifier is that it has been demonstrated to be affected by small variations in the training set [40]. To eradicate problems such as this, this project aims to compare multiple classifiers trained on the same dataset, to approach the research questions in a more unbiased manner.

Random Forests Classifier

The next classifier that was researched was the random forests classifier, which is another popular model for classification. Research has shown that the issue with decision tree classifiers being perturbed by small variations in the training set can be diminished by applying 'bagging', which is something that random forests utilise. Bagging is a means for producing "multiple versions of a predictor and using these to get an aggregated predictor" [9]. The aggregated predictor uses a plurality voting system when predicting a class, meaning that a particular class might be assigned in the test set regardless of if it has less than a majority of the vote in the aggregation.

Verikas et al. [48] carried out a survey style study to present the applications of random forests. They specifically researched the prediction accuracy as one of their variables. By surveying data from smaller and larger scale studies, they compared many different types of classification problems with differing parameters such as k-fold cross validations, multi-class, and binary classification. The type of input data varied from facial image recognition photos to credit-card fraud detection in banks using a mix of continuous and categorical variables [49], which is a similar input dataset to what this project will be producing. In these two studies, as well as many others included in the survey, the random forests classifier outperformed all other classification models by a sizeable margin. One key point is that although the results look extremely favourable, there does not exist a single model which will be best suited to every problem. This is known as the No Free Lunch Theorem and informs us to dissect each problem carefully to decide which classifier will be best for that particular dataset and research question [20]. Nevertheless, the positive and consistent results from this survey, which included a few classification problems similar to this project, provide good reason for random forests to be included in the methodology.

Furthermore, being that random forests consist of multiple decision trees, their performance has been demonstrated to be better overall. Ali et al.'s [4] research compared the classification outputs of 20 varying datasets on both the decision tree and random forest classifiers. After comparing the

precision, recall, accuracy, and f1-score within each dataset, they found that the random forest classifier is best suited to larger datasets with greater numbers of instances. The intention for this project is to collect a relatively large dataset of affiliates from the Twitter API, and this result supports the idea of using a random forest classifier.

Naïve Bayes

From previous studies that have motivated this project (see section 2.4), it is anticipated that the dataset will likely have imbalanced classes. Many real-world problems have datasets with imbalances simply because that is how the sampling instances occur naturally. This can sometimes be a problem for machine learning tasks because some algorithms are sensitive to a deficiency in one class, and techniques have even been developed to deal with this such as under sampling one class [31]. The predictive performance of the Naïve Bayes classifier has been demonstrated to not be affected significantly by stratified classes [19], and the same has been shown for the decision tree classifier, which supports their use for this particular project.

Further support for the Naïve Bayes classifier comes from the idea that because its assumptions are very different from the other classifiers being used in this research, it will be useful to observe its performance. The main difference is that the algorithm that is used assumes that all features used for training are dependent conditionally based on the class label [33]. Although this isn't an assumption that would naturally apply to most datasets, the classifier has still been demonstrated to compete well against other classifiers [39]. In their study, they investigated what characteristics in the dataset itself would affect the performance of the naïve Bayes model. Using Monte Carlo simulations, to understand the effect of risk and uncertainty of the model, they were able to envision the outcomes of the results produced. The findings were surprising because they were contradictory. They found that in cases where the data was fully independent, just how the model assumes the data to be, the accuracy of the model was very high. And, in the opposing case where the data was full of functional dependencies, the model also displayed a very good performance.

The dataset for this project will be extracting features from profile data, which will inevitably have dependencies e.g. individuals who have private profiles will always have '0' count values for certain features about their tweets, since they are not visible to be extracted. Because of the findings from Rish [39], and the fact that this classifier is not significantly affected by stratified classes, there is support for its use in this project.

Conversely, it is important to note that just because a classifier has been shown to perform well for certain similar types of data, it does not necessarily prove that it will perform in the same way on a new dataset. The noise, size, and dependencies of every dataset will affect the outcome from the same classifier [38]. The naïve Bayes classifier has the disadvantage of assuming that the data is entirely independent, because in nature that is very rarely the case. Additionally, if there is no training tuple for a particular class, then the model exhibits the zero-probability problem, where it is not able to classify that entry [33]. Still, this issue can be avoided using a stratified sampling method. Overall, the studies described in this subsection have provided support for the use of this classifier in this project, and even if its performance is sub-optimal, three other classifiers will also produce results for the same dataset.

Logistic Regression

In many medical diagnostic problems, logistic regression classifiers are the models of choice because of the way the model can handle imbalanced datasets. For example, datasets analysing a person's likelihood of getting diagnosed with cancer might have one case in one thousand where the individual is diagnosed [33]. Based on knowledge from previous research that this project was motivated by [21], [43], the dataset for this project is expected to be quite imbalanced, and so it was decided that this classifier should be explored.

In a review study by Dreiseitl and Ohno-Machado [18], a large-scale analysis was conducted of over 70 research studies that compared logistic regression and artificial neural networks for classification. These studies included binary classification which is what this project will use to identify employee vs non-employee profiles. They found that overall, logistic regression models tend to perform better then decision trees and other classifiers when the data is continuous. They also tend to have reduced generalisation errors than other classifiers. These factors show that this classifier could be useful for this project because the features that will be extracted for training the classifier will contain lots of continuous data. Also, a lower generalisation error is favourable because this refers to the model being more accurate for classifying previously unseen data.

One of the negative factors about this classifier is that it has many assumptions that need to be satisfied before training the model [18], such as the features having little multicollinearity, and having large sample sizes. It is estimated that the sample size for this project will be relatively large, and accounting for multicollinearity can be easily achieved by excluding some of the feature data. Therefore, overall there is suitable reason to test and observe what the outcomes from this classifier would be.

2.9 Previously Implemented Countermeasures

In this project, after training and testing the machine learning classifiers, the intention is to develop countermeasures that can be applied against specific factors that made the employees vulnerable in the first place. To gain a better understanding of the types of measures that have worked in the past, this next portion of the background research evaluates measures that have been tested previously. It is important to note that within the literature it is rare to find research that focuses specifically on Twitter or newspaper employees, so the research described below is based on similar ideas within a different field of work.

Airehrour et al. [1] created a user mitigation model that could be implemented in the New Zealand Banking system. Banking systems are particularly exposed to cyber-attacks because of the many-thousands of employees that not only work for them but are also in day-to-day contact with unfamiliar clients. Due to rapid advancements in financial technology, the sheer capacity of transactions taking place daily has grown exponentially, creating a viable target for social engineering attackers [46].

With relation to this project, the types of countermeasures they suggested that are most relevant are those which prevent identification of the employee to begin with. If this can be prevented, there is no target to be exploited. Airehrour et al. [1] describe the 5 stages of a common attack protocol, of which the first two stages include "researching the victim's personal information" and "planning and preparing attack based on this information".

One suggestion they make is for employees on social media sites such as Twitter who do not wish to reduce their online presence. Instead of modifying their online behaviour, measures such as blocking unknown users from contacting them could be a viable option. This suggestion was made broadly for social media in general, and it can be applied to Twitter since even public profiles have the

option of preventing contact from strangers. This way, even if the user is identified as an employee of an organisation, there is no way for the attacker to extract sensitive information from them. However, one issue with this approach is that an attacker could easily impersonate a friend and gain access by making a fresh account. The only way to be sure that this won't happen is to ensure the employee is not known as an employee on social media. The methods to prevent this from occurring are what this project aims to investigate and present.

2.10 This Investigation

Relationship detection is a field of work that has been investigated with quite some depth, with studies that have looked at the different types of relationship inference e.g. manager-subordinate, and friend-friend connections [15]. One relationship that has not been explored to the same degree is the employer-employee relationship which represents organisations and their workers. This is the first gap that this research aims to fill, by investigating which features from the Twitter profiles of the affiliates of an organisation can inform who the employees of that organisation are. This relationship is an important one for social engineering research, because many attacks that have been carried out in the past [16], [51], have exploited this relationship.

The first step of most social engineering attacks that take place online is target identification [7], so by averting this from happening, the attack may be prevented. There is the chance that by preventing identification of one employee, a different one can be chosen to attack instead. For this reason, the methods that will be developed in this project could be used as a screening tool for a company, in order to detect the features that make an employee vulnerable, in the same way an attacker would. Using Twitter features to differentiate between groups of people is not a new concept, and previous research [42] has been able to separate individuals into groups based on how likely they are to fall for a spear-phishing attack. This further supports the methodology of this research in which user's tweets from Twitter will be used as features to train and test the classifiers with.

The employee detection research by Edwards et al. [21] described earlier in this literature review is the most similar to what this research aims to accomplish. They were able to demonstrate the effectiveness of a decision tree classifier model in identifying employee relationships in an automated fashion, and this research will build on this by comparing a few classifiers to see if the result can be developed. This research will use similar techniques to those used by Edwards et al. [21] including building a features list to classify which accounts are employee-owned. One key difference is that their work resolved social media identities over four major social network sites, whereas this project is limited to Twitter, for reasons stated earlier in the scope section (*Chapter 1.6*).

To further the results from their work, this paper will aim to provide countermeasures based on a statistical analysis of the features that contributed most to the models output. Countermeasures that have been suggested and implemented in the past include reducing the online presence of the potential victim or even changing their account statuses to private. These measures are only effective for individuals who don't mind having to stay anonymous online, but for people who don't want to hide their online profiles, more innovative measures should be used. These measures are what will be researched in this project, in order to find suggestions that allow potential victims to keep their profiles public, but still have a safe experience on social media. The results from the classifiers will be compared to ground truth to evaluate the effectiveness of the model, in terms of how many true positive employee identifications the system can make from the pool of affiliate accounts for each company. The hypotheses stated in *Chapter 1 - Introduction* will be evaluated from these results.

Chapter 3

Methodology

3.1 Introduction

The following section details the procedure that was used for the research, including how the research was planned and prepared for and how the hypotheses were objectively tested. Each subsection of the experimental design starts with the aim of what needed to be achieved before moving onto the next stage, followed by the key steps in the process.
The methodology of this research project can be described in four main stages:

1) Data collection.
2) Data preparation & feature extraction.
3) Machine learning classifier training & testing.
4) Development of countermeasures.

Finally, the methods proposed for each of the three sections will be evaluated.

3.2 Testing the Hypotheses

Below are the hypotheses for this research, which were first shown in the introduction, and the methods which were used to test each one:

1. Dependent on the quality of the ground-truth used to train the classifier, the model will be able to deduce, using a group of features and the social media affiliate accounts of a company, the individuals that are employed by that company.
   How it was tested: An F1 score was calculated, which favours low false positive and low false negative results.
2. One or more of the features (or groups of features) used during testing of the model will be better predicters of employment relationships to a noteworthy degree.
   How it was tested: The feature importances for each classifier were calculated and compared to each other. Those features which were found to be important to multiple classifiers or had a high weight for one of the models, were classed as good predictors.
3. Exploring the results of the classifiers output will aid in the generation of functional countermeasures. For each of most predictive features, this research will produce an idea of how protective measures might be used or decide that the feature not easy to protect against.
   How it was tested: The features with high importances to the models were used to develop realistic measures that could be used against them. Those features which had unrealistic measures, or measures that would be likely not be useful were discussed in *Chapter 5*.
4. Features that are extracted from the most recent tweets of each user will provide supplementary insights into what makes an individual's profile easier to identify as being an employee profile vs non-employee profile.
   How it was tested: Once the feature importances were obtained, those features that were extracted from tweets were compared separately to those which came from other profile data, in the discussion.

3.3 Planning

Ethical Approval

In order to carry out a research study that involves using data from any human participants, it is imperative that the project is pre-approved by the relevant University of Bristol Research Ethics Committee. The committee needed to be informed of how the data would be collected/used, and also given details regarding storage measures of personal data. Although this project did not require any interaction or communication with any of the people whose Twitter accounts were under analysis, there were still some ethical considerations to be made. Firstly, the data that was collected needed to be fully anonymised once the necessary features were extracted from their profiles. This was to ensure that peoples personal data was not being stored for longer than required. Additionally, this would ensure that the results from the study could not be reported in reference to a particular company/individual. Secondly, the method of finding the individuals who could be checked for verified employee status needed to be ethical, in the sense that the data must already be of public nature. To abide by these considerations, Python scripting was used to fully anonymise the file containing the complete data for the machine learning analysis. Furthermore, the organisations were discovered through a public online business directory, and their employees were determined from public profile pages on these organisations' websites.

A full ethics application was submitted and approved by the Faculty of Engineering Research and Ethics Committee at the University of Bristol.

Requirements

For the purpose of this project the proposal set out some requirements that needed to be satisfied in order for this research to make a significant contribution to existing work in this area. The requirements for the data include having approximately 200 verified employee Twitter accounts within the larger dataset from which features were extracted for machine learning. The second requirement was that unique features would be extracted from profile and tweet information for each participant, as well as some features that have been used previously. The third requirement was that outputs from multiple machine classifiers would be compared, since previous similar work has looked primarily at the decision tree classifier.

Machine learning has its own set of requirements in order to make the research more effective. This includes cleaning and processing the data to be less "noisy" and extracting features that satisfy a set of assumptions for the particular classifier being used. Since the algorithms process large amounts of data in order to learn the patterns in the data, it was important that the dataset was cleaned thoroughly. Data mining research studies were reviewed in early stages of this project as vital tools ensure the data was cleaned and pre-processed to a high enough standard.

3.4 Experimental Design

Introduction

This section of the methodology will describe the specific aims and procedures that were carried out for each subsection of the methodology. It details how the participant data was first identified from online business directories, and how the Twitter API was then used to collect and collate further information about each business and their verified employees and all other affiliates.

Then, it discusses the steps taken to develop python scripts to clean and organise this affiliate data into a format that could be used for the next stage of machine learning. Next, the details regarding the machine learning classifier procedures are outlined. Finally, the development of countermeasures is discussed in context with the results from the classifiers' outputs.

Stage 1: Data identification & collection

In order to carry out training and testing of the four machine learning classifiers identified for use in the background section, a dataset was required. The dataset consists of affiliate data for 50 different local British newspapers which were identified from a business directory, where affiliates refers to all of the individual accounts who are either following or followers of the newspaper's Twitter account. For each newspaper, their website was scraped to acquire the names and Twitter accounts of as many employees as possible. This is the ground truth data. Then, from the list of affiliates for each newspaper, it could be identified whether or not the affiliate was an employee of the newspaper or not, by comparing their screennames to the ground truth. For each affiliate, regardless of whether they were employees or not, their profile information was acquired from the Twitter API, which included their name, biography information, and 200 most recent tweets. At this stage the dataset of over 450,000 affiliates needed to be cleaned of noise, to prevent the model from overfitting to the data. From this profile information, features were extracted, which is what the machine learning classifiers are trained and tested with, to observe whether or not they can identify who the employees are, based only on these extracted features.

It is important to note that there was the added possibility of extra noise in this dataset compared to that of the previous work that this project was motivated from. This is due to the dataset of newspaper affiliates being so large that there could be many past employees that were not captured during screening for noise, that could then have been classed as false positives. There was also the possibility of current employees who are unknown as verified employees in this project, adding further noise to the dataset. Because of this, one of the key requirements in data preparation was to ensure that the data was as free of noise as possible. The steps taken to achieve this noise reduction are described below in step 3.

Stage 1A: Identifying Employers

Aim:

The aim for this first part of data collection was to identify a source of ground truth data, which would consist of a list of employers and a reasonable number of their employees (approximately 200). The employers needed to be a range of different newspapers so that the data could generalise to a wider demographic. The employee names and Twitter handles collected from each newspaper formed part of the dataset, and the other affiliates of the newspapers formed the remaining part. It was necessary for the newspaper employers to have existing Twitter accounts that the identified employees were affiliated with on Twitter.

Procedure:

The procedure for identifying the ground truth data, which would be a list of employees who are confirmed to be working for their employer, involved using search engines to sort through online

business directories. The job category that was the focus of the search was newspaper journalists, because a preliminary search showed that there are many online directories containing lists of local newspapers. An online directory of local British newspapers was found containing an alphabetised list of the newspaper's websites. This website was deemed to be appropriate because the newspapers were not of the same scale as larger papers such as The New York Times, where the reporters are seen as celebrities. By using smaller scale employers, the employees would not be well known figures, thus allowing the results of the project to be more generalisable.

The employees and employers were required to have Twitter accounts because this was the platform that was chosen to conduct the research on. The first script that was written used the Python Beautiful Soup library to scrape the directory for the websites and names of each newspaper in the directory. From this, further scripts were written to scrape the websites for the Twitter handle of the company. If this did not exist, the newspaper was eliminated from the remaining steps. If it did exist, the newspaper website was scraped for news article links, where the employee names were more likely to be found. The explanation of identifying employee names and Twitter handles is found in Step 1C: labelling employees.

Stage 1B: Gathering Affiliates

Aim:

After identifying the newspapers which had active Twitter accounts, the affiliate data of each newspaper employer needed to be collected. This is because the machine learning stage of the project required information about each affiliate from the Twitter friends and followers of each employer, in the form of features. The raw data for each affiliate was first required from the Twitter API, then in a later stage the scripts were written for feature extraction.

Procedure:

For the feature extraction process detailed below, information about the affiliates of each employer needed to be collected. For each employer, the data for each of their affiliates, i.e. all of the individuals following them or being followed by them on Twitter, was required from the Twitter API. A Python library called Python-Twitter was used to more easily navigate the API, which also took charge of managing the data downloads within the enforced rate limits.

The first step in this process involved planning which features would be used later to train the machine learning classifiers, in order to decide which profile elements would provide these features. These profile elements had to be decided at this early stage to ensure that when affiliate data was downloaded from the API, all of the associated information was requested with it. After exploring the features used in previous work, as well as devising some novel features, a final feature list was collated.

The more profile elements required, the greater the time taken to collect all of the data, since the API prevents more than a certain amount of data requests per 15-minute time period. It was decided that the biography data and 200 most recent tweets of each affiliate would be necessary for some of the features that would be extracted later. It was also decided that general User data that is provided by the API, e.g. follower counts, friend counts, and follow relationship to employer, would also be requested for each affiliate. The final step was to write the necessary Python scripts to download and append this data to a csv file, in a format that would be useable at a later stage.

One problem that was discovered here was the feature that described the number of affiliates of a specific employer that were also affiliated with each other. While this could have made an

interesting result because employees also tend to run in social media circles together, it was an issue from a rate limiting standpoint. This is because a large proportion of affiliates also have large follower counts themselves, so in order to collect the affiliates of the affiliates the rate limits would have pushed the project timeline back considerably. Thus, it was decided that this feature would not be used for classifier training.

Furthermore, the rate limits of Twitter also substantially reduced the number of employers in the list from 79 to 50, due to large amount of time that it would have taken to download the affiliate data for all of these employers. To keep the project on schedule, it was decided that affiliate data would be collected for all employers with less than 22,000 followers. This reduced the estimated time of completion for the data collection from 36 days to 7 days, hence preventing delays for the feature extraction stage.

Stage 1c: Labelling Affiliates

Aim:

The aim of this final step of data collection was to label the affiliates of each employer as employees or non-employee affiliates. This defined the class that the classifiers had to make predictions about i.e. predicting whether an affiliate an employee or not. Furthermore, since the previous step caused a reduction of employers from 70 to 50 due to the rate limiting errors, further employees had to be identified, other than those that were found listed on the newspaper websites.

Procedure:

The employee names from each newspaper website were first found by parsing all of the news article links from each newspaper's homepage, to find the reporters name in the by-line. First these news article links had to be collected and this task was non-trivial because every newspaper had different ways of formatting their HTML. This meant that one script would work for one webpage but none of the others. The key to overcome this issue was to find patterns of similarity that could be generalised to a few different webpages, and then to expand the script gradually until most of the newspapers identified could be scraped effectively. The main pattern observed was that all news articles of a website started with the newspapers base URL followed by the substring "/news/", and so all of these links were first parsed from each webpage individually. From these news article links, after using URL validators to check them, the next scripts were written to identify the reporter's names and personal author pages.

After parsing the reporter's personal pages, those that contained Twitter links for the identified employee's Twitter account were added to a CSV database, along with their employer's name and Twitter handle. Over 100,000 news articles were scraped, and after removing duplicates and employees without social media links, 245 verified employee accounts were identified. From this list, only 102 of these individuals were affiliated with the employer's account on twitter, either as a friend or follower, making the list of employees much smaller than anticipated.

This number was much lower than the target of 200 and was directly the result of having to reduce the list of employers in the previous step, which affected the total possible number of employees. The subsequent motive was to use self-declared employment statuses from the non-employee affiliate's Twitter biographies as a source of secondary ground truth, with the purpose of cultivating a larger list of verified employee accounts. This method was used because it was predicted

that even the "non-employee" affiliates list might have some hidden employees that were not identified directly from the newspaper's websites, since the list of affiliates was so large (> 450,000).

Python scripting was utilised to loop through all of the affiliate accounts and identify those who had directly tagged the employer's twitter handle in their biography, along with a mention of the term's "reporter", "journalist", or "author". Due to the differences in the exact formatting of this declaration, it would have been too time consuming to identify which of these were verified employees or not. This is because a large number of affiliates mentioned working for their previous employers and then listed their current employer. So, it was decided that those accounts which had mentioned being a "former" employee were excluded from the affiliate pool. The remaining accounts were tagged as verified employees, and a random sampling of 20% of these accounts were manually examined to confirm that the script had accurately identified them.

Further noise reduction was the next step after acquiring this secondary ground truth data. Taking into account that the rows of affiliate data exceeded 450,000, it wouldn't have been possible to manually check each affiliate for whether they were employees or not. Instead, multiple scripts were written to analyse certain aspects of affiliate profiles and class them as additional employee accounts or not.

The first of the scripts to remove additional noise examined the accounts which had higher than average amounts of `employer_tweet_mentions`. A large proportion of affiliates had none, so a random sampling was conducted of 30 accounts that had more than 5, to see how many of them had mentioned being an employee, either in a tweet or in their biography. It was discovered that those accounts with > 10 `employer_tweet_mentions` were almost always employees, and since the number of accounts with this feature was low (< 75) the script validated and provided these employees' biographies for a rapid manual check. This script eliminated some noise and also provided an additional 11 verified employees.

Further scripts to eliminate noise looked at the `follow_relationship` of the affiliate to the employer. Since a mutual follow relationship was the least common, and also the most indicative of an employer-employee relationship, these profiles were the next ones to be screened.

Overall this method produced a collated list of 95 additional verified employees, bringing the total back to the projected aim of approximately 200 verified accounts. One final script was written to loop over every affiliate in the database of over 450,000 individuals, and set a Boolean value for whether they were a verified employee of the company or not, using the collected list of 196 names identified in this step.

Stage 2: Feature Extraction with TF-IDF (Implemented from Scratch)

Aim:

The aim of this stage was to write scripts to extract the necessary features from the dataset that would be used to later test which features were most useful in identifying employees from the affiliate pool. In order to provide ideas about which novel features could be extracted from tweets, a TF-IDF analysis was conducted on the tweets of a subset of employees.

Procedure:

The affiliate data from the previous stage was arranged in CSV format, with the columns containing the employers name, the affiliates name, a Boolean for whether they were an employee or

not, and the information collected from the Twitter API. This API information included their friend counts, follower counts, biography data, and 200 most recent tweets.

The initial intention of this project was to write a TF-IDF model from scratch to analyse the most recent 200 tweets from every single affiliate in the dataset (450,000+) with a public profile. This would be informative for testing the hypothesis that tweets will be more informative in identifying an account belonging to an employee compared to other features of the profiles. The TF-IDF would provide a numerical statistic that reveals how important a word is in each affiliates' set of tweets. It is useful because even if a word is present many times in one affiliate's set of tweets, it would be counterbalanced by the number of other affiliates' tweets that contain the word. This analysis would produce the top 200 words which could be used as features alongside other profile attributes, which would define the importance of the words to identifying an employee.

However, due to the immense size of the collection of tweets for all affiliates, this analysis would have been very costly in terms of time. Instead, the TF-IDF analysis was employed on the tweets from just the verified employees, in order to provide information about which words were relatively more frequent in the class of interest i.e. the employee affiliates. The employee sample was convenient because all of the newspapers were represented, since the computation was of a manageable size for the small number of employees in the sample. The script for the analysis was written by first researching the logistics of the algorithm and then writing a script to implement it from scratch. From this tweet analysis the top 20 most important words for this subset were used to decide some features that should be extracted for the entire sample. Further details of this are described in *Chapter 4 - Results*.

A secondary TF-IDF analysis was then completed on a comparative sample of employees and non-employee affiliates, but a script was written to run this analysis separately in a loop for each newspaper. The reason for this was to be able to compare specific differences between the top 100 words of the employees vs the non-employee affiliates. Comparing the outputs found that there were some key words that were relatively common for the employees to use but not for non-employees of the same newspaper. More details are described in *Chapter 4 - Results*.

In a similar fashion, further python scripts were written to extract other features from the data such as a count for the number of times the employer was mentioned in a tweet by a particular affiliate, the ratio of friends to followers the affiliate had, and the type of follow relationship between the affiliate and employer. Some features were also developed from a data exploration, described in *Chapter 4*. In total 14 features were extracted from the data collected from the Twitter API, and they were a combination of features used in previous work and novel features that were decided for this project (figure 1). The novel features are indicated with *.

Feature Name	Rationale
`follow_relationship`	Preliminary data exploration showed that a higher proportion of employees tend to have mutual and friend relationships with the employer compared to non-employee affiliates. There were also less employees with follower relationships to the employee than non-employees, making this feature a viable one.
`description_length*`	Further data exploration showed that a lot of non-employee affiliates had empty descriptions,

	but employees almost always had some text in their description, often listing their job title and previous/current employer's.
`screenname_length*`	This feature was chosen because observations made of employee Twitter handles showed that they often added abbreviations of their employer's newspaper name in their screenname e.g. Name_NYT.
`friend_count`	The employees were analysed to have a much different range of friend and follower counts than non-employee affiliates from preliminary data analysis (described in *Chapter 4 - Results*). For this reason, this feature was considered for machine learning training and testing.
`follower_count`	Same rationale as `friend_count`, above.
`fr2fo_ratio*`	The ratio of employee's friend to follower counts was assumed to be different than non-employees, simply because data analysis showed that there was a higher percentage of employees with relatively many more followers than friends.
`timeline_status*`	This feature was thought to be very important in differentiating between employees and non-employee affiliates, because employees were more likely to keep their timelines public so that they could share their news articles with their followers. Non-employees are more likely to keep privacy, and only show their tweets to their friends.
`rt_count*`	This feature was anticipated to be important because employees were found to retweet their own written news articles from the employer's Twitter accounts often, and even retweet other individuals who had posted about their news story.
`employer_tweet_mentions`	Non-employee affiliates were predicted to be much less likely to mention the employer's Twitter handle in their tweets, unless they were readers of the newspapers who were trying to engage in conversation. Employees were seen tweeting at the employer's when their articles were posted by them.
`bio_tags_count*`	Many employees would quote the Twitter handles of all of their current newspaper employer's as well as former ones. A few of them also had their fellow journalists listed in their bio as a form of promotion. Although non-

	employees also tagged some accounts, it was assumed that this would be less likely.
`rt2tweet_ratio*`	Some people on Twitter only ever retweet posts and don't tweet themselves, but when observing a random sample of employees, they were found to make their own tweets much more often than retweeting other people's posts, sometimes to share their views about news topics.
`own_tweet_count*`	This feature was thought to be interesting to include because of the opposite reason to the above feature: some employees in the data exploration sample almost never exclusively tweeted without retweets. So, it was thought that this might help to differentiate employees from non-employee affiliates.
`affiliate_mentions*`	Since employees of a company might run in the same social circles as each other, one idea was to count how many times other affiliates of the same employer were mentioned in a tweet or retweeted by each affiliate.
`link_count*`	Newspaper employees often retweet or post links to their own news articles to share them with their followers. For this reason, a link count was included to see there was a big difference between employees and non-employees posting links.

Figure 1: A table describing the rationale behind each feature that was chosen to be extracted from every affiliate in the data set, for classifier training and testing.

Stage 3: Classifier Training & Testing

Stage 3A: Pre-Processing Stage

Aim:

The pre-processing stage was for formatting the data into a setup that could be understood by the machine learning classifiers. This is because the current state of the data before this step would have caused a lot of errors (explained below). It also produced a setup for sampling the data so that a ten-fold cross validation could be implemented manually.

Procedure:

The initial step was to explore the data to understand its format. Using the tools available in sklearn and Pythons' pandas, the first few rows of the data could be displayed alongside their form and

description. From this it became clear that some of the variables were categorical and some continuous, which would be an issue for classifier training. This is because the algorithms require numerical inputs rather than arbitrary categories. To change this, some research was conducted on the several types of encoding that can be done. After assessing the pros and cons of a few types, one-hot encoding was conducted on the categorical variables. This ensures that all variables are given equal emphasis by the classifier's algorithm, as it creates extra features to separate the categories and codes them as 0's and 1's.

Next, the data needed to be pre-processed properly using sklearn's preprocessing package, which offered many transformation classes to modify the raw data into a representation that could be processed better by the classifiers. The data for each feature was scaled within a range of 0 and 1, so that the maximum absolute value would be standardised to the size of the unit. The MinMaxScaler was used to scale the data and was suitable because of its ability to preserve zero valued features in the data, and also due to its robustness against smaller standard deviations.

For the final part of pre-processing, the code to sort the data for a ten-fold cross validation was written. This is where 90% of the data would be used for training and 10% for testing, and this would be repeated for different sections of the data until all of it has been used for testing. Each affiliate was assigned a random value between 1 and 10 in a stratified manner between the verified employees and the remaining affiliates, to ensure that each training set had roughly the same number of employees. At this point, the data was ready, and the next scripts could be written for the actual classifier training and testing.

Stage 3B: Experimental Stage

Aim:

The experimental stage describes how the multiple classifiers identified in the background were actually trained and tested on the prepared dataset. It also produced confusion matrices and other analytics to allow visualisation of the performance of each classifier, making it easier to compare the outputs.

Procedure:

After pre-processing, the remaining code to train and test each classifier was written. Using the ten-fold cross validation technique described in the previous step, each classifier model was trained with 90% of the data, and the remaining 10% (i.e. those rows randomly assigned the number of the current fold) was used as test data. This was repeated until all of the data had been used to test. The 10 resulting test outputs were combined to produce an overall output, and from this the precision, recall, and f1-score was calculated. This overall test output was used to create a confusion matrix, of which one section showed the true positives i.e. how many individuals classified as employees were correctly classified as such.

Dependent on the current classifier being used, not all of the data could be used for training or testing. For example, logistic regression operates under the assumption that there will be little to no multi-collinearity within the data. After assessment, 4 of the features were found to be dependent on each other, such as 'private' profiles always having a '0' count for employer_tweet_mentions, since their tweets weren't visible to analyse. From these features, only one was included during training and testing of the classifier, and this process was repeated until all 4 had been used once. This allowed for the most favourable output to be selected.

After testing each classifier and producing a classification report, the datasets were balanced to investigate if that would improve any of the metrics. To do this, for every fold of the training data in the ten-fold cross validation, the quantity of employees in that fold was counted. Then, a matching quantity of non-employee affiliates was added to the dataset for that fold. This method allows the classifiers to be trained with a balanced dataset, that can aid in discovering patterns to better classify the minority class.

Stage 3C: Parameter Tuning

Aim:

After initial training and testing of each classifier, the parameters fed to each classifier were altered in a methodical way, to fine tune the outputs.

Procedure:

The final part of this experimental stage was to oversee parameter tuning. A script was written which systematically conducted a grid search. The reason for this is that during running of each model, the parameters are set before the learning process and so the initial or default values chosen do not always reflect what would produce the most accurate output.

Using a grid search allows the script to build a model for every single combination of parameters possible and then evaluate each model individually. Then, the overall best model was chosen along with the parameters which were used to achieve it. The caveat of this was that it was time consuming to run because a tenfold cross validation was used in every single iteration for every single combination of parameters. There were between 3 and 30 different values for each parameter. For example, for the random forests classifier the parameter for the number of trees in the forest (n_estimators) was tested with 9 different values, whereas the max depth of the tree was tested with 30 values within a range. In total there were 59 different values for the various parameters for the random forests classifier, which means there were 590 iterations in total when the ten-fold cross validation was taken into account. For the decision tree classifier there were 49 different values for the parameters in total, which means there were 490 iterations in total. The scripts stored the parameter values which produced the most favourable outputs, which is discussed further in *Chapter 4 - Results*.

Stage 4: Investigating countermeasures

Aim:

This final stage of the project involved analysing the outputs from each of the classifiers that were trained and tested, to see if there were any particular features that had higher predictive ability of an affiliate being an employee or not. From these features, countermeasures were developed.

Procedure:

Finally, in order to be able to develop countermeasures against specific features, those features that were most important to the classifier in correctly deciding if an individual was an employee or not had to be calculated. Using the sklearn library's feature importances method, the features which contributed most to the model's predictive power were assessed. For decision trees, this method

decides the importance of a node by using an algorithm that includes weights of each node and the number of node splits on a particular feature. Random forests calculate these separately for each tree and produce an average. After calculating feature importances the countermeasures could be developed based on the top most important features.

At this stage the precision of the classifiers was evaluated to decide which would be more informative, even if the classifier did not have a very good recall or overall f1 score. For example, if the precision of the logistic regression classifier was considerably higher than the others, then the features which had higher importances in classification of the affiliates would be the features that would produce the most informative countermeasures. Further details are described in the *Chapter 4 - Results* and *Chapter 5 – Implications for Countermeasures*.

3.5 Data Analysis

Data analysis was carried out on the affiliate feature data using python pandas and sklearn, which is a machine learning library for python. Within sklearn there are tools for the following four classifiers which were used in the analysis:

1) Decision Tree Classifier
2) Random Forests Classifier
3) Naïve Bayes Classifier
4) Logistic Regression Classifier

After analysing the data and producing outputs in the form of confusion matrices and classification reports, the process was repeated using parameter tuning techniques to find the most favourable output of the classifier.

Evaluating the efficacy and improvement upon previous work was assessed by comparing the precision, recall, and f1-score of each classifier with that of past work in the field. It was useful to compare the f1 scores as the harmonic average of precision and recall as well as the similarities and differences in the procedures of previous studies and this project. This provided an insight into what could have contributed to any differences in the results.

For this project it was important to have a plan for how to analyse classifier outputs if no significant f1 score improvements are found. Since this is a research-based project, there was always a possibility that the actual outcomes would be different to the expected outcomes. To prepare for this, research was undertaken to learn how to analyse confusion matrices and classification reports in the case of the classifier being unable to accurately identify the employees from the affiliates. From this we can learn how such a result came about.

3.6 Conclusion

This portion of the report has identified and described the step by step processes that were carried out in order to answer the research questions and test the hypotheses. Firstly, the project was planned out, and the ethics procedures were carried out to comply with the University of Bristol's ethics committee's standards. Then, the employee data to create a verified list of employees for ground truth was scraped from newspapers obtained from an online business directory. After this, the Twitter API was used to collect profile information for every affiliate associated with the employee's accounts. TF-IDF analysis was then conducted from scratch, to inform some of the features that could be extracted from the profile information, and further features were informed from a data exploration

exercise. After cleaning noise from the dataset, the machine learning classifiers were trained and tested with these features in a ten-fold cross validation, and the parameters were tuned using a grid search to find the most suitable parameters. The final classifier outputs were obtained from these refined models, and feature importances were calculated to inform the countermeasures against them. Finally, the results were evaluated for their success by comparing key parts of the results to that of previous similar work. The next section will step through each classifier's data analysis and explain the significance of the most important results.

Chapter 4

Results

4.1 Introduction

This section of the paper presents the results from each stage of the project. It begins by displaying results of the TF-IDF analysis which was informative for producing features for the machine learning training and testing. This analysis looked at the minority class of employees separately, before comparing a balanced dataset of some employees and non-employees, to find features that could be used to differentiate the classes. It then presents the results of correlational exploration of the data which took place at the start, to investigate which features were more important for the model. Then, the results of each classifier are shown, pre and post parameter tuning, to display how this affected the results. Analysis of the parameter tuning is shown, and the outputs of the classifiers with their best parameters will be revealed. Each classifier's classification report is discussed and compared to a benchmark. Finally, feature importance for each classifier is shown and discussed, and the precision-recall curves for the best performing models are displayed.

4.2 Output of TF-IDF Model

Below is the table showing the output of the 20 most important words for a sample of just employee affiliates discovered from the analysis. The output revealed that two of the top three most important words in each set of tweets for employees was a reference to a link e.g. "http, https". From this, the first feature, which was a count of the total number of links present in an affiliates most recent tweets, was selected. The next most important word for employees found from this analysis was the word "RT" which represents a retweet. In Twitter, a retweet is used where a user sees another users tweet and wants to share it on their page, which also provides credit to that user. Thus, the next feature selected for the feature extraction phase was a count of the retweets. The ratio of retweets to the user's own tweets was also decided to be a feature, as it was noted that the verified employees tended to retweet tweets about their articles often, with fewer producing their own written tweets.

Top 20 Most Important Words Discovered from TF-IDF Analysis of a Corpus of Tweets			
1	**https**	11	new
2	**http**	12	good
3	**rt**	13	day
4	thanks	14	news
5	great	15	time
6	today	16	know
7	just	17	people
8	hi	18	thank
9	ve	19	story
10	like	20	no

Figure 2: Table displaying the output of TF-IDF analysis of a sample of just the employee affiliate's tweets to show the most important words in the corpus.

Next, another script was written to conduct TF-IDF analysis on both a sample of verified employees and an equal sized sample of non-employee affiliates. This time the scripts produced outputs separated by employer, so that words specific to the employer-employee relationship could be explored. Once again, this sample was stratified to ensure each employer's affiliates were represented equally. This script produced an output of the top 100 words for each employer's employee and non-employee sample. The aim of this was to look for distinctive features in the text of the see if there was anything specific to employees that might be used to identify them, by comparing their outputs to the non-employee affiliates for the same newspaper. It was discovered that a proportion of the employees (> 13%) had their employer's name in their top 100 words. This was either due to directly 'mentioning' their employer in a tweet (35.6%), or retweeting them (64.4%), in which case the twitter API stated their name next to the "RT". This informed the next two features which were the number of own employer mentions in their tweets and also the number of affiliate mentions. For comparison purposes, none of the top 100 most important words for the non-employee affiliates were the employer's Twitter handle, highlighting that this feature might be a suitable way to differentiate between employees and non-employee affiliates.

All features involving tweet analysis are original to this project, and the remaining features were either based on previous work or decided based on the following analysis.

4.3 Data Exploration 1: Describing the Data

The next stage after deciding the features for feature extraction was to explore the data to see what the distribution of data is like across the sample domain. This is useful to not only provide early insights about the data, but also to help establish how representative the data is for the overall problem it is trying to resolve. It also motivated some additional features. The following findings were discovered:

- The local newspapers included in this study comprise 6% of the total number of newspapers included on LocalMediaWorks, a directory containing a large number of the UK's small local newspapers. The 94% not included in the study either did not have a Twitter account or did not provide any verified employees/had anonymous reporters.
- Verified_employee is the binary class that the classifier will being trained to predict and is a categorical feature with the values True or False.
- The percentage of verified employees in the sample is 0.055% which represents 196 employees out of 451,856 non-employee affiliates.
- The percentage of employees with public accounts is 97.1% whereas the percentage of employees with empty or private accounts is 2.9%. This is a contrast to the percentage of non-employee affiliates with public (83.6%) vs empty/private (16.4%) accounts. From this came the idea for the feature 'timeline_status'.
- The follow relationship of verified employees to their employers is 4.6% friend (employer follows them), 61.5% follower (they follow the employer), and 34.1% mutual. This is again a big contrast to non-employee affiliates of which 3.4% are friends, 90.8% are followers, and only 5.8% are mutual. From this came the idea for the feature 'follow_relationship'.

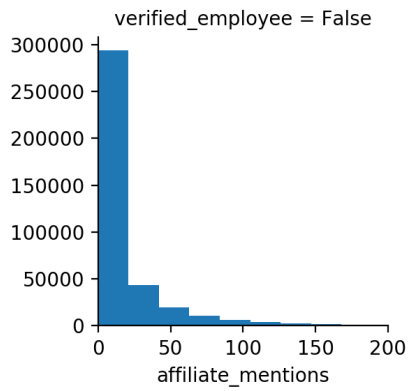
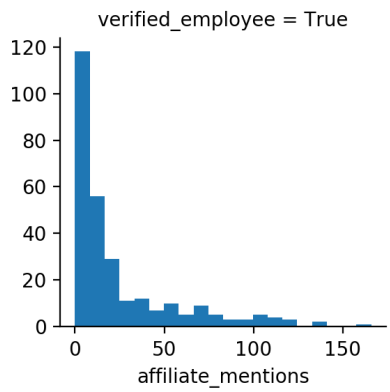
- The popularity of affiliates on Twitter varied from those having 0 friends or followers, to individuals with 4,526,331 friends and 107,072,037 followers. Once again, this is a big difference when compared to verified employees whose friend and follower counts reached a maximum of 3806 and 230,416, respectively.

From these descriptive findings it should be noted that the sample is quite representative for employers of small local newspapers that have an online presence and also display their employees on their websites. Although in this study there is the ground truth data which specifies who is and isn't an employee, the objective is for the results to provide insights into what features can identify employees from other employers who are not public about who works for them.

4.4 Data Exploration 2: Correlational Analysis

After feature selection and extraction, the final dataset could be analysed to assess how each feature is linked with being a verified employee or not. Features that have greater correlations with being an employee might have more feature importance for the classifier models. Early identification can be used to decide which features carry more importance. This was useful for training the classifier with the most important features first, and then cumulatively adding more features to see if/how much the model improves. A histogram displays the distribution of individuals in the sample using specified range sizes. This aids in analysing aspects of specific bands, such as whether those affiliates with a higher retweet count were more likely to be verified employees or not.

Below, figure 3 shows the correlations of the features that had the greatest discrepancies between being a verified employee or an affiliate.

Non-employee Affiliates	Verified Employees
a) 	b) 
c)	d)

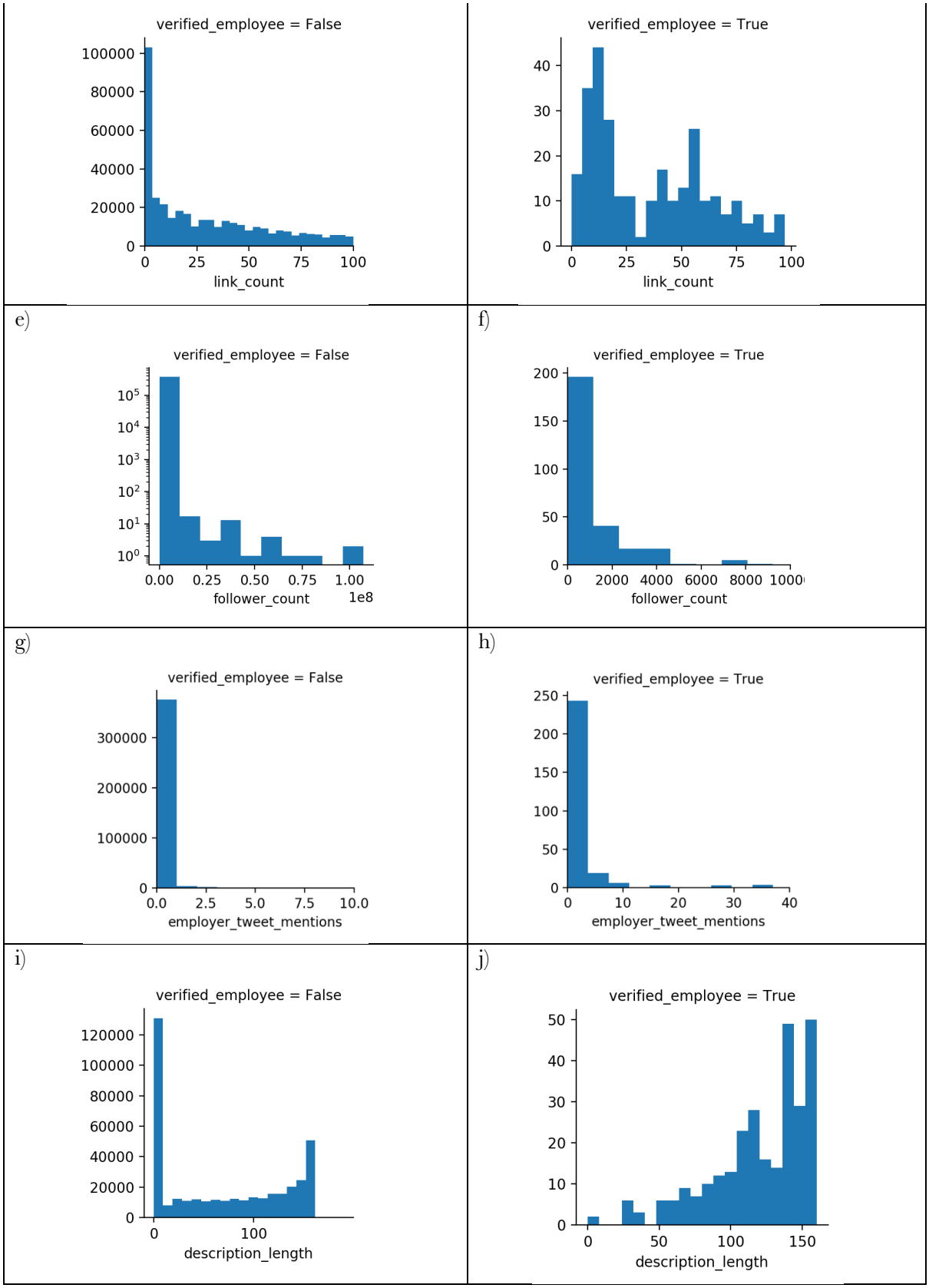


Figure 3: A table displaying the differences in the correlations of various continuous features with both the verified employee class and non-employee class.

From an initial observation of the above histograms it is clear that there are features that have varying ranges between the employees and non-employee affiliates. For example, the range of values for `employer_tweet_mentions`, (that is the count of how many times the employer was mentioned in their tweets), is four times larger for employees than non-employees. Specifically, non-employees typically had 0 to 2 mentions of the employer ranging up to a maximum of 10 which is a contrast to verified employees who had a range reaching up to 36 mentions. The majority of the employees had between 0 and 4 mentions, which is still notably greater than that of the other affiliates.

Further observations can be made from the `affiliate_mentions` graph. This is the number of mentions of the other affiliates of the same employer. Employees are often part of the same online social circles, which involves interactions with the other employees on twitter. From the above graph, the behaviour of both classes is comparable until the 100 `affiliate_mentions` mark, after which point the verified employees tend to have a greater number of mentions.

The `link_count` in tweets is also very different between the two classes, where the larger class of non-employee affiliates has a much greater proportion of the group with a lower link count. The employees have an overall greater number of individuals with more than 40 links in their most recent tweets, compared to those with less than 40. This could be explained by employees retweeting many of their own articles from their employer, which may also be supported by the fact that this class has a larger range of values for their `employer_tweet_mentions`.

The biggest discrepancy between the two classes is that the range of `follower_count` for the employee group is much smaller than that of the non-employee affiliates, whose range had to be displayed on a logarithmic scale to be visible on the histogram. The majority of employees have less than 1000 followers on Twitter and the employee with the most followers had under 8200 in total. In contrast, for the non_employees the range extended up to 800,000 followers, and many individuals in the class had over 10,000 followers. This data suggests that this feature will be important for the classifier in identifying the employees from the list of affiliates.

Moreover, the `description_length` was also found to be largely varying between the two classes. Whereas the majority of non-employee affiliates had mostly empty Twitter biographies, with a maximum length reaching up to 170 characters, the verified employees were on the opposite end of the scale. The proportion of individuals in the employee class with a biography of 0 characters was one of the smallest subsets, with the majority of the individuals having between 100 and 150 characters. This could be explained by the employees trying to market themselves in their biographies as journalists so that people who are interested in their articles can follow them. In addition, having an empty biography makes it more difficult for people to know what they will be tweeting about.

Other features were analysed and found to not be correlated with being in either class, and figure 4 displays those with the largest discrepancies. These features include `follow_relationship`, `timeline_status`, `friend_count`, `screenname_length`, and `rt_count`. This was an unexpected finding because of the differences observed previously in the `friend_count` feature, where the verified employees only went up to a maximum of a few thousand friends (~4000) and the non-employee affiliates class had some individuals with many millions of friends. The histogram in figure 4.c, which depicts friend count, displays a restricted range of up to 5000, because that's where 99.4% of the friend counts could be found, but the similarities between the two classes are apparent. It was expected that the non-employee affiliate class would have many outliers since the class size is so large, but even so, the general trend of graphs comparing both classes is almost the same, with the curves in figures 4.c and 4.d tapering off around the 2000 friends mark. On Twitter, 'friend' refers to the number of people an individual follows, and the employees may try

to limit this so that they are more likely to be able to interact with other employees or their employer. This is further backed up by the greater number of affiliate and employer mentions in the verified employee class compared to the non-employee affiliates.

The other feature which was found to be most similar between the two classes was the `screenname_length`, where the maximum length in both classes was 15 characters, and the minimum was 4 for the non-employees and 7 for the employees. The similarity can be explained by the regulations imposed by Twitter, where screennames cannot exceed 15 characters. The screennames can be as short as 1 character, but as the graph shows, there were no individuals with less than 4-character screennames in either class. The observation that both classes follow similar trends for `screenname_length`, which increase more or less proportionally until they reach 15 characters, is likely due to the Twitter guidelines and not due to a factor that applies only to individuals in one class.

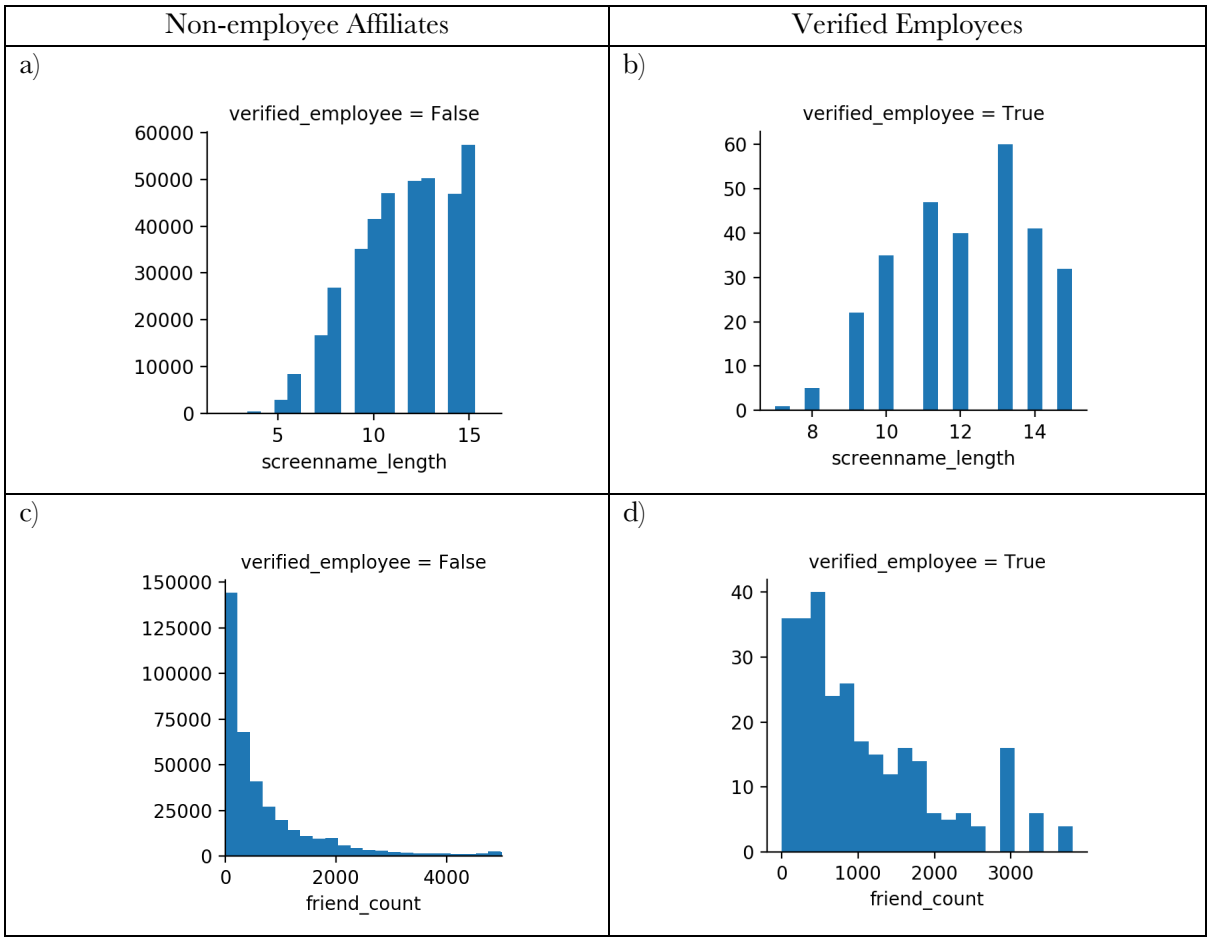


Figure 4: A table displaying a graphical analysis of some of the continuous features that were most similar between the verified employee and non-employee classes.

They key takeaway from this correlational analysis is a preliminary indication of which features are likely to be more informative for the classifiers going forward. Those include the features in figure 3, which were found to have bigger discrepancies in correlations when comparing the employee and non-employee classes. This implies that these are the features that will likely be more informative to the classifiers for deciding whether an affiliate is an employee or not. Preliminary

exploration of data is also useful for comparing to the feature importances which will be calculated from each model after running them.

Following data exploration, the data was pre-processed using techniques detailed in the methodology. The next step was to train and test the four machine learning classifiers chosen for this project.

4.5 Initial Training and Testing

Initially, after appropriately pre-processing the data, the four machine learning classifiers were trained and tested on the data, using a 10-fold cross validation. Each classifier had its own prerequisites which were taken into account during the process. The classifiers were first trained with all of their features to get a general sense of the output, and in a later phase the features were added in a cumulative manner, to make observations about which features had the highest importance and which, if any, hindered the performance of the model.

Decision Tree Classifier

The first classifier to be trained was the decision tree classifier. Below is the classification report and confusion matrix produced from this initial phase. The precision score for the minority class of verified employees was 0.50 and the recall 0.10, producing an f1-score of 0.16. If the f1-score from the motivation study for this project is taken as a benchmark, then the f1-score produced here is very low. Edwards et al. [21] were able to produce a score of 0.65, whereas the f1-score here is 0.16, which is significantly lower. By comparing the true positives with the false positives from the confusion matrix, it is clear that the classifier was categorising only 19 out of over 450,000 (< 0.004%) of the non-employee affiliates as employees, contributing to the good precision score.

The f1-score for the majority class of non-employee affiliates was 1.00 because the performance of the classifier in categorising non-employees as such was very good. This is likely due to the extreme size of the majority class. Additionally, only a small percentage of individuals (9.5%) were correctly classified as employees from the total pool of verified employees, yet the rate of misclassifying non-employees as employees was very low, contributing to the precision score of 0.50 for the minority class. Overall the decision tree classifier had a reasonable performance before the parameters had been tuned, or the features had been selected and added in order of importance to the model.

Initial Decision Tree Output

Confusion Matrix	Predicted: False	Predicted: True
Actual: False	TN = 451649	FP = 10
Actual: True	FN = 188	TP = 8

Figure 5: The confusion matrix produced from a ten-fold cross validation of the decision tree classifier showing the actual vs predicted values for the dataset.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	1.00	1.00	451659
True	0.44	0.04	0.07	196

Figure 6: The classification report produced from the initial output of the decision tree classifier, showing the precision, recall, and f1-scores for both classes.

For each classifier, the dataset was balanced after initial testing, using the methods described in *Chapter 3 - Methodology*. This means that the classifier was trained using an equal amount of employee and non-employee affiliates in each fold of the training stage. Balancing the dataset proved to cause no improvement to any of the classifiers' outputs, and surprisingly made the results much worse. The decision tree classifier had a very high rate of misclassification, where 25% of the non-employee affiliates were classified as employees. The precision was 0.00 and recall was also 0.00, leading to an f1-score of 0.00.

The parameters for this classifier included the minimum number of samples per leaf, the maximum depth of the tree, the minimum number of samples to split an internal node, and the number of features to look at when deciding the best split for the tree. These parameters were initially set to default values (*criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *max_features=None*), and after initially running the classifier parameter tuning adapted these values making use of the idea that certain parameters can increase the models' performance for a problem. The range of values to be used for each parameter was as follows: max_depth values ranged from 1 to 30, min_samples_split ranged from 1 to 15, min_samples_leaf ranged from 1 to 15, and max_features ranged from 1 to 14 (the total number of features). After training each parameter with each decided value, the best performance was found with the following parameters:

- Min_samples_split set at 3.
- Min_samples_leaf was set to 9, where lower values gave worse performance outputs and values greater than 9 produced the same or worse performance.
- Max_depth was set to 5, but the performance of the classifier did not change significantly when this was set to a value up to 2 on either side.
- Max_features was also set to 5 for the best output.

From the confusion matrix and classification report below, it was revealed that the precision had increased from 0.44 to 0.50, which is a notable increase. A further 11 verified employees were correctly classified as employees rather than non-employee affiliates, which is a noteworthy improvement. On the other hand, the model also classified 9 more of the non-employees as employees, but the improvement in classifying the minority class compensated for this setback in the overall precision. The f1-score for the employee class increased from 0.07 to 0.16, due to the overall improvement in both precision and recall. The precision, recall, and f1-score for the majority class were unchanged, due to the large class size and thus the comparatively small number of misclassified individuals in that class.

Decision Tree Output ***Post Parameter Tuning***

Confusion Matrix	Predicted: False	Predicted: True
Actual: False	TN = 451640	FP = **19**
Actual: True	FN = 177	TP = **19**

Figure 7: *Post parameter tuning* confusion matrix produced from a ten-fold cross validation of the decision tree classifier showing the actual vs predicted values for the dataset.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	1.00	1.00	451659
True	**0.50**	0.10	**0.16**	196

Figure 8: The classification report produced *post parameter tuning* from the output of the decision tree classifier, showing the precision, recall, and f1-scores for both classes.

Figure 9 below displays the portions of the visual decision tree output that lead to an affiliate being classified as a `verified_employee`. The values displayed are not true to the actual count values as they had been scaled during pre-processing transformations using the MinMaxScaler, and discussion of these values will use the unscaled values.

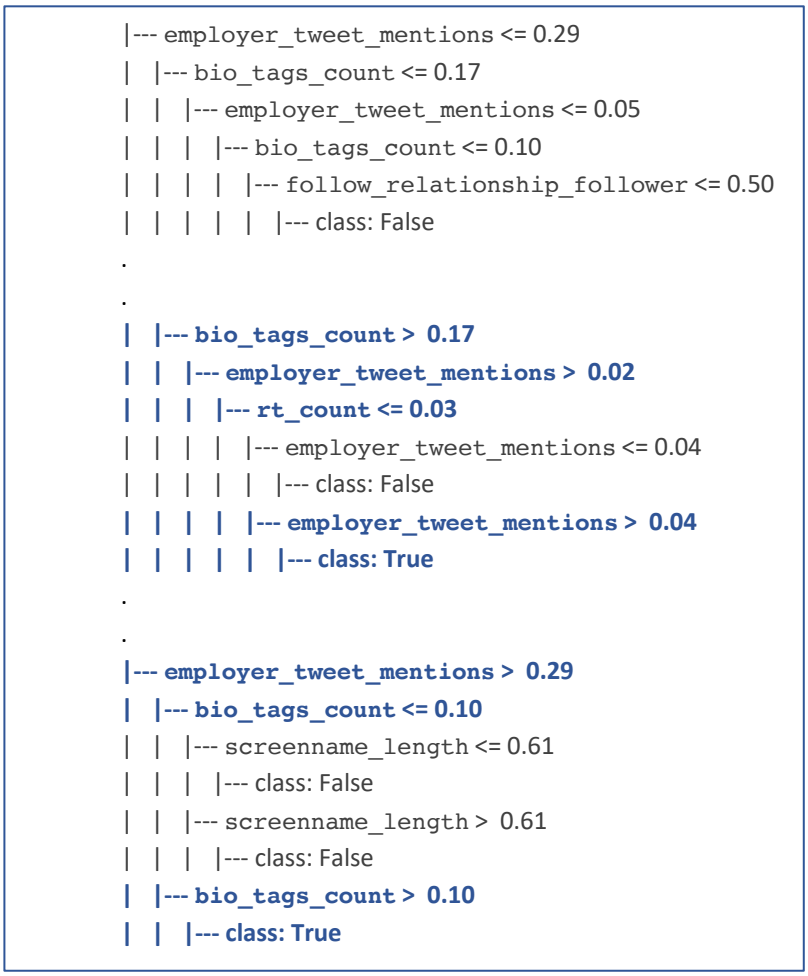


Figure 9: A figure showing the decision tree that was produced during the initial training and testing stage. Only the portions that led to classification as an employee have been shown, and values have been scaled and thus are not true to the original count values.

The output shows that to be classified as an employee the first split of nodes occurred with the feature `employer_tweet_mentions`, which could lead to classification in two ways. Firstly, if the tweets contained greater than 26.4 employer mentions and the number of account tags in the affiliate's biography was greater than 3.4, the affiliate would be placed into the verified employee class. Secondly, for those affiliates with fewer than 26.4 `employer_tweet_mentions`, their classification depended on their biographies containing over 5 account tags, less than 6 retweets in their most recent 200 tweets, and at least 7 mentions of their employer in their tweets.

This visual tree output provides an early indication of which features were the most important and which weren't as necessary for the classifier. The second phase of the results uses a cumulative addition of the features (*Chapter 4.7*) in order to investigate which were the most important and which might have hindered the model's ability to identify the employees from the affiliate pool.

Random Forests Classifier

The random forests classifier was the next to be trained and tested with the same data. Once again, using the same ten-fold cross validation technique the following results were produced:

Initial Random Forests Output

Confusion Matrix	Predicted: False	Predicted: True
Actual: False	TN = 451642	FP = 17
Actual: True	FN = 182	TP = **14**

Figure 10: The confusion matrix produced from a ten-fold cross validation of the random forests classifier showing the actual vs predicted values for the dataset.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	1.00	1.00	451659
True	**0.45**	0.07	**0.12**	196

Figure 11: The classification report produced from the initial output of the random forests classifier, showing the precision, recall, and f1-scores for both classes.

Once again, the performance of this classifier is not optimal, with only 14 out of the 196 total verified employees being identified as employees. Additionally, 182 of these employees were classified as non-employees, which lead to the recall for the minority class to be 0.07. On a more positive note, the proportion of non-employee affiliates being classified as employees is very low (< 0.004%), leading to a decent precision score of 0.45 for the minority class. Overall, the random forests classifier still had a better performance than the decision tree pre-parameter tuning.

Comparing the recall of the initial random forests output with that of the initial decision tree output, there is a small improvement from 0.04 (decision tree) to 0.07 (random forests). What this represents is the number of true positives that were recalled from the total correctly. The recall isn't the best measure in the case of this problem because of two reasons. Firstly, the classes are so imbalanced that a high recall for the majority class appears a lot more positive than it really is. Secondly, although this model and the decision tree model both had similarly low recall scores, the precision offsets this low response because of how it can be interpreted. A model with a low recall but high precision is good at classifying the minority class without misclassifying the majority class too often.

By comparing the f1-scores from this classification report to that of the benchmark study (0.65 f1-score), it is clear that the f1-scores for the employee class are very low. Since the f1-score takes into

account both the precision and recall, the low recall explains the low f1-score. However, if we compare the precision to that of the benchmark (0.58), we can see that this result is a promising one.

The parameters used in training the random forests classifier at first were the default parameters (*n_estimators=10*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *max_features='auto'*), and in this next parameter tuning step the values were tested for each parameter in a methodical way to select the best one for the overall classifier performance. The following parameters were altered: n_estimators (which represents the number of trees in the forest), max_depth of each tree (deeper trees have more splits and capture more information), min_samples to split an internal node, min_samples_leaf, and max_features to look at when deciding to split.

Using the procedure outlined for parameter tuning in the methodology, the model was trained multiple times with a different parameter value each time. The range of values to be used for each parameter was as follows: n_estimators ranged from 1 to 100, max_depth values ranged from 1 to 30, min_samples_split ranged from 1 to 15, min_samples_leaf ranged from 1 to 15, and max_features ranged from 1 to 14 (the total number of features). The following ideal parameters were identified for the random forest classifier:

- N_estimators was set at 30 trees, because increasing the number of trees above this had led to a decrease in performance of the model.
- Max_depth was set to 20 and increasing the value any higher than this led to overfitting where the model could not generalise findings for the new data.
- Min_samples split was not altered as performance was optimal using the default value.
- Min_samples leaf was also kept at the default value for highest performance. Increasing the value only led to an underfitting issue.
- Max_features was set to 5. Values below this caused performance to be worse, and values greater than this did not increase performance.

From the confusion matrix and classification report below, it was revealed that the precision had increased from 0.45 to 0.52, which is a significant increase. This is also an improvement upon the tuned output of the decision tree, which reached a precision of 0.50. The f1-score also showed an improvement from 0.12 to 0.14. While this may be a small improvement, it is still important for the following reason: Two more verified employees were recognised by the classifier, and 2 less of the non-employees wrongly classed as employees were correctly placed into the False category. Although this appears to be a very small improvement, it is significant in the sense that for an attacker this would provide an additional two targets that are likely to be vulnerable, and less resources would be wasted on the now smaller list of incorrectly identified employees.

Random Forests Output ***Post Parameter Tuning***

Confusion Matrix	Predicted: False	Predicted: True
Actual: False	TN = 451644	FP = **15**
Actual: True	FN = 180	TP = **16**

Figure 12: The confusion matrix produced from a ten-fold cross validation of the random forests classifier showing the actual vs predicted values for the dataset after *parameter tuning*.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	1.00	1.00	451659
True	**0.52**	0.08	**0.14**	196

Figure 13: The classification report produced from the output of the random forests classifier *post parameter tuning*, showing the precision, recall, and f1-scores for both classes.

The result for the random forests classifier is more promising than the decision tree because it did not classify more of the non-employees as employees. In total, the decision tree misclassified 15 individuals in the majority class, compared to the decision tree where 19 individuals were misclassified out of the total of over 450,000. In reference to this project's aims, to produce beneficial countermeasures they should be based on features that lead to correct classification, so a higher precision would be more useful in that sense.

The precision, recall, and f1-score for the majority class were unaffected once again, due to the large class size and thus the comparatively small number of misclassified individuals in that class.

Naïve Bayes Classifier

The Naïve Bayes classifier was the next model to be trained and tested with the affiliate data. From the confusion matrix and classification report it is apparent that the classifier failed to clearly recognise the employees from the pool of affiliates. The precision score of 0.00 and f1-score of 0.00 confirm this. Although the classifier did manage to correctly classify 132 of the employees out of the 196 total, this is not a positive result if the majority class is also taken into account. In the majority class, almost 24% of non-employee affiliates were incorrectly classed as employees. That is, over 108,000 were misclassified, leading to the extremely low overall precision for the employee class. This demonstrates the poor performance of this model for this particular problem.

Since the classes in the dataset are exceptionally imbalanced, the most important result from each classifier is the precision, because it depicts only the percentage of the results which were relevant. Therefore, the results of this particular model's output will not be useful for analysing feature importance or developing countermeasures, because the model did not execute the task to the same level of precision that was demonstrated with the previous two classifiers. Because of this, parameter tuning was not carried out to try and improve the precision of 0.00, as any improvement would still not outperform the previous classifiers.

Naïve Bayes Output

Confusion Matrix	Predicted: False	Predicted: True

Actual: False	TN = 343611	FP = **108048**
Actual: True	FN = 64	TP = **132**

Figure 14: The confusion matrix produced from a ten-fold cross validation of the Naïve Bayes classifier showing the actual vs predicted values for the dataset.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	0.76	0.86	451659
True	**0.00**	0.67	**0.00**	196

Figure 15: The classification report produced from the initial output of the Naïve Bayes classifier, showing the precision, recall, and f1-scores for both classes.

Logistic Regression Classifier

The final classifier to be trained and tested on the affiliate dataset was the logistic regression classifier. The output performance was found to be worse than that of the previous classifier. From the classification report below, it is shown that the precision and f1-score are once again 0.00 and 0.00 respectively. Additionally, 416,042 out of the total 451,659 entries of the data that were non-employee affiliates were classified as employees. The frequency of incorrect classification is 92% for the majority class, which is a negative result considering that the data available for this class was so large. Furthermore, although the recall of 0.73 for the employee class is higher than any other classifier, the result is not meaningful because that is simply a measure of how many of the true positives were correctly identified. Being that the rate of false positives was also so high, the overall f1-score for both classes was very low.

Logistic Regression Output

Confusion Matrix	Predicted: False	Predicted: True
Actual: False	TN = 35617	FP = **416042**
Actual: True	FN = 64	TP = **143**

Figure 16: The confusion matrix produced from a ten-fold cross validation of the Naïve Bayes classifier showing the actual vs predicted values for the dataset.

Classification Report	*Precision*	*Recall*	*f1-score*	*Support*
False	1.00	0.08	0.15	451659
True	**0.00**	0.73	**0.00**	196

Figure 17: The classification report produced from the initial output of the Naïve Bayes classifier, showing the precision, recall, and f1-scores for both classes.

Due to the poor outcome from this classifier and the Naïve Bayes classifier for this particular problem, the remainder of the results section will focus on the analysis and development of the decision tree and random forests models, which had the most favourable outcomes.

4.6 Analysing Feature Importance

Analysing feature importance is necessary to understand which features contributed most to the ability of the various classifiers to identify the employees. From this information, the countermeasures in the discussion section could be developed. As stated in the aims for this project, the countermeasures need to be based on the important features, so that they would realistically be helpful in concealing the identity of vulnerable employees.

To assess model performance using just a subset of the most important features, the feature importances were produced and analysed. These features were calculated using the feature_importances measure in sklearn's library for each classifier, which revealed which features had the greatest effects on the outputs. The values range between 0 and 1, and the higher the value, the greater the effect of the feature on the outputs. Using these features, a subset was created from which the model was retrained and retested, and more features were added cumulatively until model performance was optimal. In some cases, the added features had to be removed if they hindered performance.

In figures 18 and 19 below, the feature importance for the top ten features in the decision tree and random forests classifier are displayed. In the top 5 most important features, the two that are common between the two classifiers in question are `employer_tweet_mentions`, and `affiliate_mentions`. The most important feature for the decision tree was `employer_tweet_mentions`, which had an importance of 0.38, which represents the effect of the feature on the model's outcome. The most important feature for the random forest classifier was `friend_count`. Overall, the random forests classifier had over 10 features that had an importance of over 0.05, whereas the decision tree only had 3. This might be explained by the behaviour of the individual models, since the random forest is an aggregate of multiple decision trees that have a plurality vote.

Moreover, the biggest difference between the two models is that the decision tree made use of fewer features overall to make a classification decision, compared to the random forest which displayed a noteworthy importance (> 0.05) for many of the features, such that some were not even included in the top 10.

Feature Importance: Decision Tree Classifier

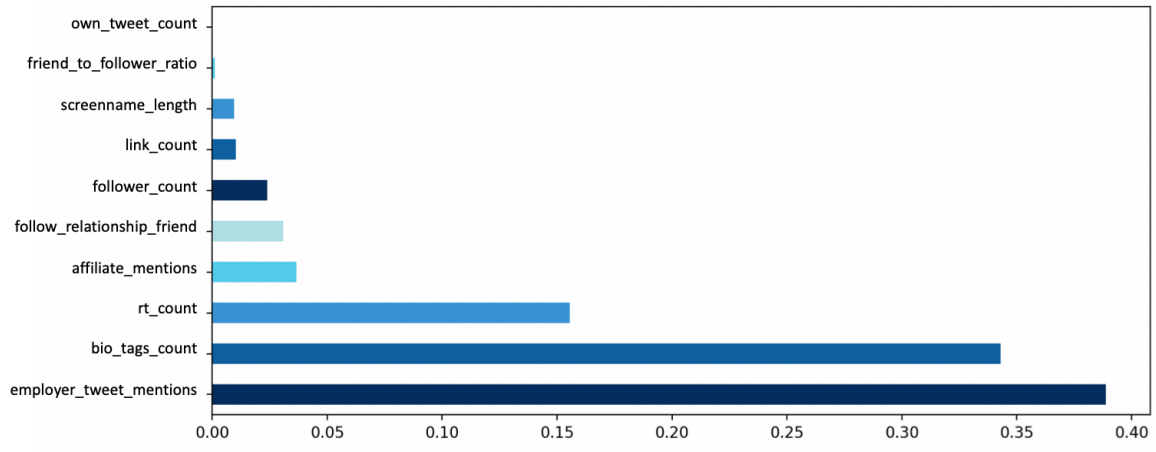


Figure 18: The feature importance analysis of the decision tree classifier.

Feature Importance: Random Forests Classifier

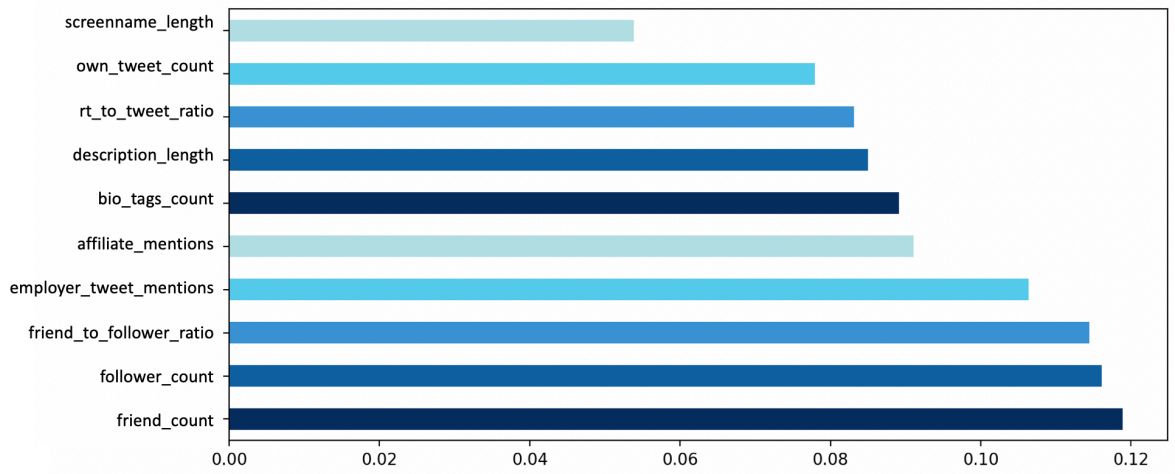


Figure 19: The feature importance analysis of the random forests classifier.

The precision recall curve is suggested over the receiver operating characteristic curve, or ROC curve, because the dataset is imbalanced between classes [14]. The precision recall curve

displays changes in precision against the recall on the same plot, which simplifies analysis. The average precision (AP) score has also been calculated, which represents the weighted average of precisions achieved at each threshold, where the recall increase from the preceding threshold provides the weight. The AP score for the decision tree (0.05) is higher than that of the random forests classifier (0.04), even though the latter peaked at a higher precision overall (0.50 vs 0.52).

From the below curves, the overall classifier performance can be assessed. Although relatively good precision scores have been achieved (0.50 and 0.52 for the decision tree and random forests, respectively), the precision recall curves highlight that the overall classifier performances were sub-optimal. In an ideal case, both the precision and recall would be high, indicating that the classifier could not only correctly identify the true positives, but also identify most of them. However, in this case the positives should still be noted, that in a dataset of over 450,000 individuals, 19 individuals from the very small quantity of verified employees in the dataset could still be identified based on their profile features.

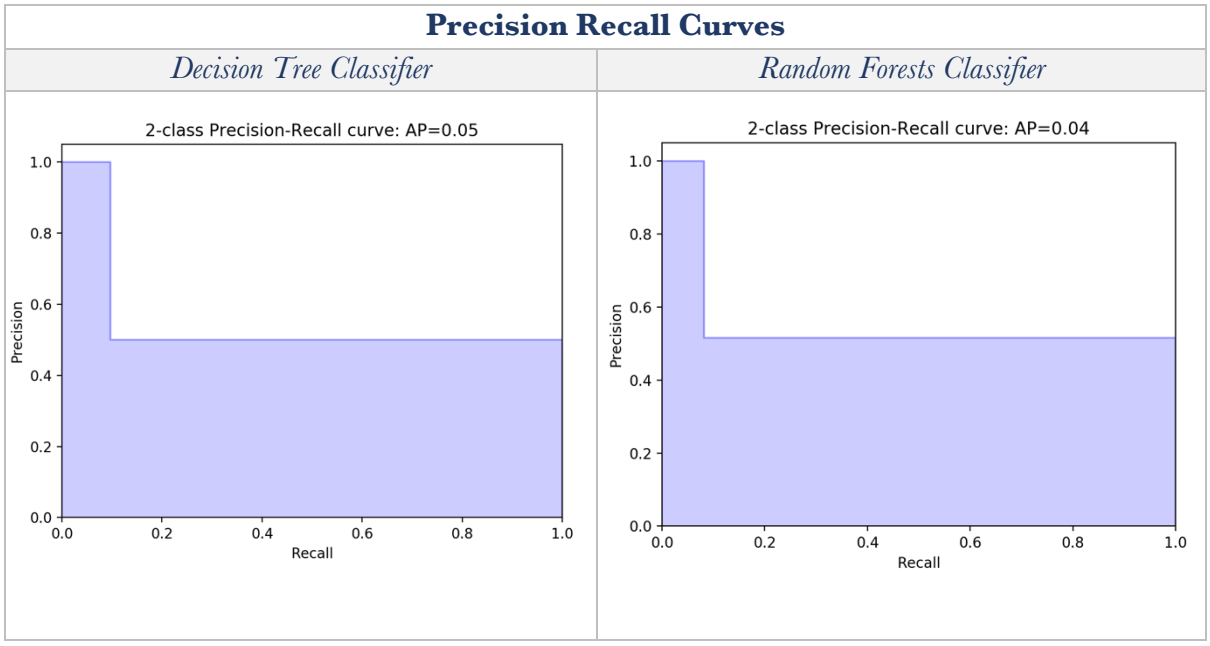


Figure 20: The precision recall curves for the two best performing classifiers: decision tree and random forests.

4.7 Cumulative Addition of Features

After producing the final models and assessing feature importance, the features were added in a cumulative manner to each model (decision tree and random forests), to see which ones had the biggest impact on each model, and at which point adding more features did not improve the model's performance. This method started by adding the most important feature and worked down the list towards the least important feature. For the decision tree classifier, the model continued to improve as the top four most important features were added one by one, but once the fifth most important feature (`friend_relationship`) was added, there was no difference in model performance before and after the addition. Therefore, for this model, the following features were most important for classification (in order): `employer_tweet_mentions`, `bio_tags_count`, `rt_count`, and

`affiliate_mentions`. With just these four features the model was able to achieve the precision score of 0.50.

By repeating the same procedure and adding features to the random forests model cumulatively, a very different result was found. Whereas for the decision tree there were few features that had high importance values, the features had much more equal weights for the random forests classifier. The results from this task showed that 7 out of the top 10 most important features had to be included in the model for the precision to reach the same value it had with all 14 of the features (0.52). Therefore, for this model, the following features were most important for classification (in order): `friend_count`, `follower_count`, `friend_to_follower_ratio`, `employer_tweet_mentions`, `bio_tags_count`, `affiliate_mentions`, and `description_length`.

Having extra features for either of the models did not hinder the score in any way, but by training the classifiers in this cumulative manner, the less important features could be excluded from further analysis.

4.8 Conclusion

To conclude, of the four classifiers trained and tested on the dataset collected for this research, only two of them were able to produce results of any significance: the decision tree classifier and the random forests classifier. Both of these models had similar outputs, but the features they were able to identify as being most important to classification decisions differed somewhat. Findings from TF-IDF tweet analysis include the discovery of terms that were found to have importance in the corpus of affiliate tweets. The two features that were common between the models, and so will be most useful for informing countermeasures were the `employer_tweet_mentions` and `affiliate_mentions`. Both of these features came from tweet analysis after the TF-IDF results were produced along with `retweet_count` which was found to be a top three important feature for the decision tree. Future work can definitely benefit from further use of the results from that analysis by incorporating a full bank of words style approach, where every word in the top 200 for the corpus is used as weighted feature for every affiliate. The next section of the thesis evaluates the feature importances from the models and discusses countermeasures which were developed from them.

Chapter 5

Implications for Countermeasures

5.1 Introduction

The following section begins by discussing the features which were unexpectedly found to not have an effect on the models' abilities to identify the employees from the dataset. Some reasons as to why are explored. Then, the countermeasures for identification are discussed, in direct reference to those features which influenced them. These features are the ones that were found to have the highest feature importance for the two models with the highest precision (decision tree and random forests), or to be common between the two models. Finally, this section concludes with an explanation of how realistically these measures can be implemented in conjunction with general measures that are currently in routine use by many organisations.

5.2 Features with Surprising Results

Follow_relationship

This feature represented whether an affiliate and the employer were mutually following each other or if it was a one way follow relationship in either direction. It was decided to extract this feature during the data exploration phase because there were significant differences between the verified employees and the non-employee affiliates in the percentage of individuals in each of the three groups (friend/follower/mutual). The employee affiliates had 34.1% of their group in the mutual follow category, whereas only 5.8% of the non-employee affiliates were in that group. This led to the theory that having a mutual follow would be indicative of an employee. However, during the training and testing stage of the models, this feature was not found to be in the top 10 most important features for either the decision tree or the random forests classifier. Conversely, the feature `friend_relationship` i.e. where the employer follows the affiliate, but it is not reciprocated, was actually more indicative of somebody being an employee. Interestingly, `friend_relationship` was the fifth most important feature for the decision tree classifier, but the random forests model did not even have this feature in the top ten.

Timeline_status

Another feature with a surprising outcome was the `timeline_status`, which referred to whether a user's tweet timeline was empty (no tweets), public (visible tweets for everyone), or private (tweets visible to only followers). One assumption was that users with private or empty timeline would be less likely to be classified as employees because there were 10% more users with public timelines in the employee affiliates compared to the non-employee affiliates. But it was discovered that this feature was not in the top ten most important features for any of the models. Other features were much more powerful predictors of employment, making `timeline_status` less informative than predicted. It is possible that one reason for this is that the number of private accounts in the non-employee class was not large enough to make a difference. Only 2.9% of employees had private accounts compared to 16.4% of the non-employee affiliates with private accounts, which is still low compared to the vast majority of accounts in both classes which were public.

5.3 Countermeasures Considering Specific Features

Employer_tweet_mentions

This feature, which refers to the number of times an employer has been mentioned in the tweets of an affiliate, was found to be the most important feature for the decision tree classifier (with an importance of 0.38), and in the top four most important for the random forests output. What this suggests is that the act of mentioning their employer's Twitter handle in their tweets makes an employee more vulnerable to identification. A simple solution to this would be to suggest employees to not directly @ the employer i.e. tag the employer's Twitter handle in the tweet too often. The number of times that made it obvious to the decision tree classifier that an individual was an employee was 26.4 times in the past 200 tweets, which means that these employees were mentioning their employer in over 13% of their tweets. By not directly tagging their employer's handle when discussing them, this could greatly reduce the vulnerability of an employee, and it is a simple measure to keep in mind when using social media.

Retweet_count

The `retweet_count` feature is similar to the `employer_tweet_mentions` feature above, because a lot of employees were found to retweet links or tweets from the employers account, and retweets include a "mention". Because of this, similar advice can be suggested: employees should be remined to stay mindful of how often they retweet their employer. Retweeting non-employer related posts from other accounts is not as risky as retweeting the employer, because by retweeting the employer their name is stated on the retweet, making for a much more identifiable employee. However, retweets are not as risky as direct mentions, where the individual specifically writes the employer's handle in their tweets, because many non-employee affiliates also retweet the employer. Especially in this case where the employers are newspapers, they are much more likely to be retweeted by random accounts because they post news articles which people might want to share with their followers.

Bio_tags_count

`Bio_tags_count` is a feature that refers to the number of other accounts' Twitter handles that are visible in a user's public biography. Even private accounts have a visible biography, making this an important feature to be aware of. Considering that the majority of newspapers in the employers list are small local ones, the reporters often work for multiple newspapers in the area and not just one. This means that many of these verified employees would list multiple Twitter handles of all of the newspapers they work for in their biography. This is especially true for the verified employees who were identified through secondary ground truth collection, where they had self-declared their employers in their biography. Since doing this has made these individuals more likely to be classified as employees from both the decision tree and random forests classifier, it should be noted that a relevant countermeasure would be to simply not declare multiple employers in their biographies. A good way to still allow the biography to be descriptive of the individual would be to write about being a reporter without specifically tagging multiple accounts in the biography. The `bio_tags_count` feature was found to be more significant in the decision tree classifier for people who had more than 3.6 tags in their biography. A caveat of this countermeasure is that having multiple employers it is specific to this dataset of journalists, because employees with other careers would most likely have just one. This reduces the generalisability of this measure.

Affiliate_mentions

This feature refers to the number of times other affiliates, of the same employer, had been mentioned in the tweets of a user. Since employees tend to run in similar social circles, it would also make sense that they mention each other in their tweets. However, individuals mentioning other affiliates (employee or not) of the newspaper in their tweets were more likely to be classified as employees. This suggests that limiting the number of mentions of other individuals that also follow/are followed by the employer would help to minimise how identifiable a person is. This suggestion is quite challenging because the newspapers have so many affiliates that it would be impossible to check who the person you are mentioning follows every time you make a tweet. However, based on the evidence from previous work [21] that employees affiliating with other employees makes them more vulnerable, employees should take into account that interacting with other employees online can make them a target. The countermeasure that should be suggested is to ensure that employees are aware of who other employees of their organisation are online, so that they can minimise public interactions with them. If the employee has their tweets in a protected status (private account) then this would minimise this risk fully, but for individuals with careers like journalists this is not a feasible suggestion. This is because journalists often interact with people on social media to spread the word about their articles, so having a private account could hinder their reach.

Follow_relationship_friend

`Follow_relationship_friend` refers to when the employer follows the affiliate but not vice versa. In this case, if an organisation with a large following, like most of the newspapers included in this research, are following a non-organisation such as their employee, it points towards a personal connection between them. This could easily be interpreted as an employer-employee relationship, and so the countermeasure would be to ensure that the follow relationship is not one sided. Furthermore, the friend lists are usually much shorter than following lists, making them much quicker to search through if an attacker was searching for employees. Mutual follow relationships were not found to be significant for the classifiers to identify individuals, so employees should ensure that they follow the employer back, to minimise any risk here. This feature was in the top five for the decision tree classifier, but had little impact for the random forests classifier, so its effect should not be emphasised as the most important feature to use a countermeasure for. However, if an organisation wants to keep its employees anonymous, the best course of action would be for them to not follow employees on public spaces such as Twitter.

5.4 Conclusion

The obvious solution to counteract all of these features would be to simply keep the Twitter account on private, for only approved users to see, but in the current age of social media that is not a very realistic solution. Having a private account for somebody working in journalism could prevent them being able to share their work with a wider range of people, because they would have to manually approve their followers which can be troublesome. By being more mindful of the features above that make identification more likely, an employee can have a safer online experience when it comes to social engineering attacks. Even employers should endeavour to get involved with screening their employees online social presences to make it a safer experience.

The most important takeaway from this analysis is that some features might be more indicative of a follow relationship, but countermeasures are not viable. Thus, to create a more well-

rounded approach to defence, multiple countermeasures should be implemented because their collective ability to prevent identification as an employee could be useful. For example, using the table summary below, a company could create a policy to advise their employees to implement the following five countermeasures. If the approach works, then they could expand the suggestions to other similar companies also. The single most important feature was the `employer_tweet_mentions` because of its high ranking to both classifiers and relatively simple countermeasure: by simply asking employees to be mindful of how often they tag the employer in their tweets, they can significantly reduce the classifier's abilities to identify them. This suggests that it could also protect against attackers who might have identified this feature to be a suggestive one for employee identification.

5.5 Summary Table of Countermeasures:

Feature	Classifier Importance Rank		Countermeasure	Other Information
	Decision Tree	Random Forests		
Employer_tweet_mentions	1	4	Less than 10% of tweets should mention employer.	Only 2 features were of high importance to decision tree, this was one of them.
Retweet_count	3	> 10	Do not retweet directly from the employer account. Retweet same tweet from another person to break the direct link.	Not of high importance to random forests, but relatively easy to defend against.
Bio_tags_count	2	6	Do not directly tag or mention employer in biography and keep other account tags to minimum.	This feature had relatively high importance for both classifiers.
Affiliate_mentions	4	5	Refrain from public interactions with other known employees of the company.	Easier to implement if the person knows who the other employees are, but this isn't always the case.
Follow_relationship_friend	5	> 10	Employers should avoid following employees, but the reverse relationship is fine.	Lower importance but easy to defend against.

Chapter 6

Evaluation

6.1 Comparison of Results

The aspects of previous work that this project was able to improve are three-fold. Firstly, this project included an analysis of tweets of all individuals included in the research, to test the hypothesis that features extracted from tweets would add important value to the classification step. Secondly, the size of the verified employee dataset was 10 times larger than that of previous work by Edwards et al. [21] allowing for the model to have improved accuracy for predictive evaluation. Thirdly, by using the same dataset to train and test multiple classifiers, this research was able to find the model most suited for this particular problem and evaluate feature importances based on that. Additionally, conducting TF-IDF analysis to assess the semantics of a portion of the tweets was useful in informing which features to extract.

There are some possible explanations for why this project was not as successful as previous work. The first reason is that due to the large dataset of over 450,000 individuals, the classes were extremely imbalanced. Only 196 entries in the dataset were of verified employees, which is less than 0.044% of the total. Even though the ten-fold cross validation ensured a stratified division of the affiliates into the ten folds, this still means that a very small amount of the training data was for verified employees. This would have made it much more difficult for the classifiers to detect the features that contribute more to identification as an employee, especially because recall scores were so low. If we compare this to the research by Edwards et al. [21], who had 20 verified employees out of the 3,753 affiliates, the employees make up 0.53% of the total. This means that their research had a 12 times greater ratio of employee to non-employee affiliates, which further suggests that the ratio is an important factor in classification. One reason for this could be the methods of interaction that people in differing careers use e.g. journalists are more likely to interact with the public more often because part of their online identity requires that. In contrast, lawyers would likely only be interacting with other lawyers and not their clients on social media, for confidentiality purposes.

Another issue with datasets that are this large is that it is much more likely to have multiple profiles that perform similarly across the different features. For example, an employee and non-employee might have very similar values for every feature just because the probabilities of having multiple similar profiles in a bigger dataset are increased. Nevertheless, there is a positive of this issue in that the real-world applicability is of the results are high. In a real-life setting, an attacker would have to look across thousands of profiles to identify who the employees are, and so identifying the features that make this easier are where the most realistic countermeasures can be applied.

One final difference between previous work and this project is that Edwards et al. [21] took the research one step further and then collated employee's identities across multiple social media websites, including Facebook and LinkedIn. The purpose of this was to evaluate how a social engineer would benefit from collecting more information about an individual from multiple sources. This would have been quite informative to include as part of this project's research, however due to recent changes in privacy laws the API's for these other social networks have become limited to developers who are willing to pay for the data. Still, even with only evaluating Twitter data, this project has been able to fulfil the aims and objectives previously stated in the introduction, to identify the features that

contribute most to identification and develop countermeasures against them. This will be the subject of the next subsection.

6.2 Evaluation of Aims vs Achievements

This section will consider the objectives stated at the start of this project and evaluate to what extent the project achieved what it aimed to.

Aim 1: Produce a literature survey focusing on the current problems in the area of social engineering attacks using employee detection, and the success rate of measures that have been implemented and tested in the past.

The background section of this report provided pertinent details surrounding social engineering attacks, and what the current standard of countermeasures are. It also introduced and evaluated some proof-of-concept research which was the motivation for this project, offering a gap in the research for a full-fledged version of their research. By learning from the procedures and techniques used in previous works, this literature survey was able to clearly inform the direction of this project, especially with regards to including analysis of tweets for feature selection.

Aim 2: Develop a system created using machine learning approaches to identify the social media accounts of employees from a company, identified from the social media accounts affiliated with that company.

The methods used in this research were able to successfully train two machine learning classifiers (decision tree and random forests) to be able to identify a portion of the small pool of verified employees as employees. Although the recall was less than desirable, and the f1-score did not prove to be an improvement upon previous work, the results can still be viewed as positive. Firstly, from such an immense pool of data, the very tiny percentage of employees present were still able to be identified with a notable precision (0.52) with the random forests classifier. This means that the features which were identified as contributing to people's discovery as employees were useful for developing countermeasures. Secondly, the shortcomings of the results can be attributed to certain aspects of the project which can inform future work to avoid such issues in the future. As mentioned in the following section for critical analysis of the project, some additional features which were not able to be extracted for this project could have provided insights and better overall f1-scores for this dataset. Furthermore, the source of the dataset (newspaper employer's affiliates) was very different to that of previous work which looked at law firm employees, and this can be useful information for a team wanting to research the differences here in the future.

Aim 3: Collate and discuss the features of a profile which facilitate the detection of who is employed by a certain company. These features will be subject to an analysis of importance to isolate those which were most useful to a classifier.

This project conducted an analysis of feature importance as part of the Results section. For the two classifiers which were able to produce a notable level of precision in identifying the employees from the affiliates, the features which aided this were collated and discussed in a top ten list. The features which were common between the lists, and also those which were found to be necessary when

training the classifiers cumulatively, were then used in the next step to aid in developing countermeasures. Those features which were of least importance were eliminated from a copy of the dataset, and the classifiers were then retrained on the subset of most important features. Because this led to an unhindered precision and recall score, we can be confident that the eliminated less-important features did not have any significant influence on the classifiers, and so should not be regarded as needing measures to counteract their effects.

Aim 4: Use the feature analysis to identify countermeasures that a company can employ to protect against vulnerabilities that lead to social engineering attacks, where employees are the target.

After feature importance was analysed, this aim was fulfilled by providing a detailed account of countermeasures that are both realistic to apply in a real-world scenario, and not complicated to screen for. The previous section, *Chapter 5 - Implications for Countermeasures*, mentions the measures in order of the features with the highest importance, and also discusses the caveats for some of the countermeasures. Overall, this project has been able to suggest potential countermeasures for the five most important features, excluding some which would have been unrealistic to develop countermeasures for, thus fulfilling the aim.

6.3 Critical Analysis of Project Execution

Research Design

The step by step design of the research was well planned out and executed with reference to setting aside the right amount of time for each step, and any contingency measures that had to be taken. For example, after collecting the ground truth of verified employee data, the affiliate data for these employers had to be collected from the Twitter API. The Twitter API's rate limits were extremely low (900 requests per 15 minutes) compared to the quantity of affiliates that needed to be collected for some of these newspapers with almost a million affiliates. Because the scripts were already partially completed when this became a problem, a contingency plan was put into action. For the remainder of the verified employees, only those whose employers had under 22,000 followers would be included, so as to be able to collect all affiliate data in a reasonable time frame. This meant that the project remained on schedule for the next stage. For future work a recommendation would be to attempt to obtain an enterprise account with Twitter, which would allow much less harsh rate limits, although it comes at a steep cost. This would only be necessary for a much larger scale project but bearing in mind that the data for almost half a million affiliates in this project took almost 12 days to collect, it would be a sensible suggestion, if feasible.

On reflection, one criticism of the research design is that it only used one large dataset to train and test the machine learning models. Perhaps using two different datasets and comparing the results might have provided additional insights. Previous work acquired ground truth data from law firms, and this research used newspaper reporters as the career of choice. However, collecting employee data for another job type as well could be informative as to why this research's results were less than favourable. Having said that, this projects data sample is believed to be representative of this particular career, with around 6% of all local newspapers in the UK included. This 6% is noteworthy since only those newspapers without a social media presence or without employees declared online were excluded. A good precision score was achieved, but the f1-score, representing overall

performance, was poor. Since previous research has been able to show findings of greater significance, perhaps the type of career of the individuals in the dataset could make a difference to the results. It would have been non-trivial due to the time restraints of this research, but in future work it could be suggested that multiple datasets could be collected and compared in the same manner using the same features, to test this theory.

Data Collection

The most time-consuming stage of this research, in terms of both acquiring the knowledge to implement it properly and the execution, was the collection of the ground truth and Twitter API data. In hindsight, if this was known prior to undertaking this task, some changes would have been made in the process. Collecting the ground truth data required producing a script that could successfully extract employee names and Twitter accounts from over 150,000 differently styled web pages. The actual run time for the final version of this script after sampling its success on a variety of webpages was a few days. This is because it initially had to obtain these web pages from a business directory which continuously blocked the computer running its IP address. Knowing this now, if this project was to be repeated one suggestion would be to acquire the use of a paid cloud-computing service such as Amazon Web Services (AWS) or Microsoft Azure. For example, one use could be to divide the web pages into separate groups and run the scripts separately on these groups so that they could have their data extracted simultaneously. Not only would this be a faster method because of the divide and conquer approach, but cloud computing services have much faster servers than the tools used for this project, providing much more efficient implementation.

Furthermore, the second part of data collection, where the affiliate data for each employer was obtained from the Twitter API, could also be improved in some ways. One issue was that a lot of Twitter users use their own URL shorteners when adding links to their tweets. When Twitter provides links from the API, there is the option of shortened or expanded URL's, but even the expanded URL's from users who have used URL shortening tools are not in the actual expanded format. This caused problems during feature extraction where the feature for how many links led to the employer's newspaper website could not be fulfilled. Online research showed the presence of many URL expanding tools, however they either had costs associated with them or the rate limits were too limited to the point where it would have delayed this project's progress. For future work the recommendation would be to allocate additional time or expenses to use these tools, since this feature might have been an informative one.

Training & Testing

During the training and testing of the four classifiers careful considerations were made to ensure that the data was cleaned of noise as meticulously as possible, and that assumptions for each classifier were satisfied. For example, after removing additional noise by finding the secondary ground truth data (individuals who had self-declared themselves in their Twitter biographies as employees for a specific newspaper), further data cleaning was done. By writing additional scripts to identify people who had called themselves "reporters" or "journalists" but hadn't specified a newspaper, further noise could be removed. This was because a lot of these account could very well have been employees of the employers in the verified list, but since there was no way to know, they could be contributing to the failings of the classifiers. Nevertheless, some changes can be considered for a duplication of this research. Even though time was taken to thoroughly inspect the data to search for noise, the

incredibly large size of the dataset (> 450,000 entries) means that further processing could have been done. In addition to further cleaning of the dataset, it should also be suggested that some further ground truth data be gathered for each newspaper, as this would verify additional affiliates as employees that may have unknowingly contributed to noise in the dataset. Finally, if this project had additional time set aside for data processing then some further manual verification would have improved noise-reduction. Manually verifying data entries is the most time consuming yet thorough method that could have been applied to a randomly selected subset of the data, to look for additional entries that disrupted the dataset.

Moreover, on assessment of the procedures followed for classifier training and testing, pre-processing was carried out competently. The assumptions that were required to be satisfied for certain models like Logistic Regression were checked and fulfilled, and the data was pre-processed accordingly. Even so, a wider range of features might have been beneficial at this point, providing further means for classification. Some features that were considered but were not included were a) number of links in tweets that lead to employer's newspaper website b) number of other affiliates in the friend/followers list of each affiliate.

Feature a) could not be included due to the URL shortening issue mentioned in the previous section. Feature b) could not be included due to Twitter rate limits which would have required a few extra weeks to collect data for affiliates of the affiliates. Both of these are issues that can be dealt with in future work if extra time or expenses can be allocated to deal with these rate limits.

Personal Development

On reflection of this project, a measurable amount of personal development has been achieved. The Python language was unfamiliar until the start of this project, which aided in understanding and executing a wide variety of tasks. One of the biggest hurdles to overcome was acquiring the ground truth data for employee verification. But, with time spent creating and adapting Python scripts, eventually a dataset that was acquired that was the targeted size. Another personal development was learning about implement TF-IDF analysis to weight the important words in tweets, and this was useful for extracting feature ideas to test one of the hypotheses about tweets. Thirdly, another task that was learned was how to pre-process and clean noise from data to adequately carry out machine learning analysis. Cleaning the noise was a non-trivial task, because a dataset with over 450,000 entries is likely to have a lot of noise. But by using additional feature extraction and writing exploratory scripts to assist with the process, this was another skill that was acquired in this project. Finally, the task of data analysis was studied using techniques from the literature and carried out effectively with help from the matplotlib and sklearn libraries. Aside from the use of libraries, many scripts were written without libraries such as those for exploring the data, cleaning the data and cross-validating it for classifier training and testing. Overall, this project was executed with efficient time management skills and well-structured plans for each part of the project, to ensure that each stage of the project was carried out effectively.

6.4 Challenges

Challenges in Data Collection

Some scripts that needed to be written for data collection were more troublesome than others. For example, a lot of websites were found to asynchronously load the webpages and the data separately. This meant that the html parsing script couldn't access the actual text on the webpages in order to search for employee information. This was fixed after researching and using a set of Python bindings called PyQt4 that could load data from async webpages. Then, the next challenge was tackling the many different styles of html that newspapers used, in order to find out the best way to write a universal script that could extract data from most of them. This was done by first starting out with a script that could get data from one website, and slowly altering parts of the script to account for more and more websites, until it was able to be applied to all of the 150,000 webpages in the collected list. As part of that task, the script had to deal with constantly being blocked as a crawler, but this was easily overcome by adding manual time constraints to the script. After that task was complete, there was the issue regarding the Twitter API's rate limits, for which the only solution was to reduce the number of employees in the list overall, so that the data could be collected within a reasonable time frame. Overall, data collection had a limited number of challenges which were able to be swiftly overcome in the allocated time for this stage of the project.

Challenges in Data Analysis

The data analysis stage, which included classifier training and testing, came with its own set of challenges. Firstly, the low precision and recall scores led to some further data exploration to try and reduce the noise further and see if there was another reason for the less than favourable results. One attempt included balancing the training dataset so that that it contained an equal number of verified employees and non-employee affiliates. The test dataset was unchanged. It was hypothesised that this would have improved the f1-score, but the result was quite the opposite. The precision dropped from 0.50 to 0.00 on the decision tree classifier, and the same effect was produced with the random forests classifier also. The model misclassified the majority of the non-employee affiliates as employees, which offered the idea that the large presence of the non-employees and the collective effects of their features were more important in separating the two classes. One additional challenge in the data analysis stage was finding the most efficient way to discover which parameters should be used for each classifier. The solution was to write a grid-search script that ran the classifier many times, trying every sensible combination of parameters in a loop and storing the results along the way. This took time to run each model multiple times, but the result was that they were trained with the best parameters for the task.

Challenges in Data Interpretation

Data interpretation was the most interesting stage of the project, because there was opportunity to explore what caused the precision, recall, and f1-scores obtained from each classifier. The results described the top ten feature importances of the features extracted to train the classifiers, but developing countermeasures was a challenge. One of the reasons for this was that not all of the most important features could realistically be countered online. For example, the top three most important features for the random forests classifier were `friend_count`, `follower_count`, and `friend_to_follower_ratio`. If this is considered for an employee with a social media presence, it is very difficult to manage these counts. Having a high friend count might point towards somebody being an employee but asking the employee to watch this value would significantly affect their online

user experience. Thus, it was decided that it would make more sense to develop measures for those features that could be realistically managed.

6.5 Conclusion

To conclude, this discussion has suggested possible reasons for why the results were not as expected, including the imbalanced class problem and type of dataset chosen for this research. Other possible reasons are open for exploration in further work, which will be detailed in the next section. Overall, this project has still been able to provide a range of applicable countermeasures that individuals can use to protect themselves. It has also been able to collate the specific features that make individuals more likely to be recognised as employees from a social engineer's perspective. This can be utilised by organisations that want to be mindful of this, where they can employ the procedures used in this research to screen their employees to see who the most vulnerable are. By using features that are not specific to the career of journalism, the applications of the results are present for a wider collection of organisations than just newspapers.

Chapter 7

Future Research Recommendations

The following section outlines some suggestions that could be taken into account for future research on this topic. The recommendations include ideas that were either thought to be beyond the scope of this project or could not be completed in the allocated time frame.

One of the main suggestions for this research, which was first mentioned in *Chapter 5 – Implications for Countermeasures*, would be to collect two datasets of individuals and their employers in different career paths. The reason for this is that results from previous work by Edwards et al. [21] were shown to be quite different from the results in this project, where the ability of the classifiers to identify employees was much higher, even though the classes were comparably imbalanced. One of the main differences between the two pieces of research was the type of employer and hence the affiliates. By comparing multiple datasets, the reasons for the differences could become clearer and the results could also become more generalisable.

Another suggestion that could be used to scrutinise why the models from this research underperformed is to research and extract a wider range of features for classifier training and testing. By gaining additional data, such as information about the types of links that are tweeted, analysing profile pictures, and conducting sentiment analysis on each affiliate's tweets, the models might be better able to classify the affiliates into the correct class. As described in the literature review, the performance of a classifier is directly contingent on the quality of the features extracted for training the models, hence making this suggestion an important one.

One specific implementation that would have been informative for this project would be to conduct a full bank-of-words feature extraction for sentiment analysis. At the start of the project a TF-IDF analysis was carried out to help inform the features based on the most important words in each affiliate's set of tweets. But, a more sophisticated technique would have been to include each of the 200 most important words as a column for each affiliate, where the weights could be calculated for each word per affiliate. This could then be used to further characterise each affiliate, since text sentiments greatly differ between types of individuals [2]. Unfortunately, this task was not computationally-manageable for the project, and the time it would have taken to run such a script would have extended much beyond the allocated time for feature extraction. But as a recommendation for future work, this would greatly expand the pool of features and make it more likely that some important findings could emerge.

Moreover, a suggestion for future work which would require more resources would be to expand the horizons to other social networks such as LinkedIn. LinkedIn is a website that is used for professional networking and so would likely be quite informative of the types of features that identify an employee, since it is made for employees to connect and share their progress in their field. Social engineering attacks have been carried out using LinkedIn in the past [30], [44] and comparing the results of classifiers for the same employers but across multiple social networks would advise about common features that make people easier to target.

A final way that future work could improve upon this research is by actually implementing the countermeasures on the employer's accounts for a range of the employers whose data was used for training and testing the classifiers. Then, a comparison could be carried out to observe how well the classifier performs after the changes to their profiles have been implemented. This would aim to

concretely define which features make employees the most identifiable, because those features that still make them recognisable by a classifier even after trying to alter their online behaviour would be the most important ones to be conscious about. Furthermore, this would provide support for the countermeasures that were found to work, which could be used to advise other companies in the same field about how best to protect their employees.

Chapter 8

Conclusion

In conclusion, while there is plenty of research concerning social engineering attacks and defence mechanisms, the threat will be a growing one until more robust defences are revealed [15]. This research endeavoured to make a contribution in the particular stage of the attack where the targets are first identified. A system that is able to identify employment relationships automatically can provide an insight into the methods an attacker uses to execute this task manually. This research has been able to demonstrate that there are key aspects of an individual's Twitter profile that can be used to identify them as being employees or non-employee affiliates. The importance of this lies in its applications to the field of social engineering defences. By discovering the features which make people vulnerable to identification, various countermeasures were developed which have been evaluated to be realistic to implement. Although the strength of these findings from the models are not as robust as expected, the outcomes were still informative because of the good precision scores, which suggest that the important features wouldn't lead to mis-identification of non-employees as employees.

A limiting factor of this research was that the social media analysis and training of the ML model only made use of public Twitter data, and not data from any other social media sites. This is a caveat because social engineering attacks involving employee detection are not limited to data collected from Twitter. An attacker will often collate information from multiple sources Edwards et al. [21]. Nevertheless, using only public Twitter data allowed this research to emulate the kinds of information an attacker would have easy access to, and provided a specific insight into the features an attacker might look for in Twitter based attacks. It is possible that limited data from Twitter alone is sufficient to plan a well-formed attack, which is significant as it would require less effort on part of the attacker.

Furthermore, recent changes in API laws have served as a deterrent for producing automated attack tools for attackers that aren't willing to invest large sums of money into identifying employees. But, on sites like Twitter that have API's where large amounts of data can be downloaded for free, the possibilities of automated crawlers are still there. An attacker could make a similar system such as the one produced for this research, if they had the right pool of data, and identify these specific features for themselves. Then the tool could be used against a dataset for an organisation where the employees are not known beforehand, to identify the victims. For this reason, a suggestion that was made in *Chapter 5 – Implications for Countermeasures* was to screen employees on a regular basis, to identify what the specific features are for the type of organisation in question. This would allow the more vulnerable employees to be identified, and relevant profile changes to be implemented.

The final findings of this research can be seen as limited in some respects, because the results did not make improvements to the f1-scores that had been achieved previously for very similar research [21], [43]. But this project has been able to provide insights into why that might have been the case, due to the great differences in sample sizes and sample job types. Nevertheless, the fact that two of the ML models were able to perform their intended relationship inference to a good level of precision has provided further support for investigating automatic inference of employment relationships as a tool for producing defence suggestions. The additions that this work was able to provide include a novel range of classifying features which incorporated tweet analysis, a larger dataset of verified employee profiles, and multiple classifier modelling. The larger dataset and novel range of features were particularly useful because greater variety in the features allowed for a new

hypothesis to be explored about the inclusion of tweet analysis in employee detection. Moreover, the investigation of countermeasures was another contribution of this research, especially because they were informed by an analysis of the features that were most useful for the classifier in deducing employment relationships.

This project underwent a critical evaluation of all aspects, including research, methodology, and analysis of the results. Overall, it has been evaluated to have fulfilled the aims that were set out at the start of the project and progressed through testing of each of the hypotheses. This project explored potential improvements that could be made in future research exploring similar research questions and has provided concepts that could be incorporated if a similar project is undertaken in the future.

Chapter 9

Bibliography

[1] David Airehrour, Nisha Vasudevan Nair, and Samaneh Madanian. 2018. Social engineering attacks and countermeasures in the new zealand banking system: Advancing a user-reflective mitigation model. *Information* 9, 5 (2018), 110.

[2] Hanaa A. Aldahawi and Stuart M. Allen. 2013. Twitter mining in the oil business: A sentiment analysis approach. In *2013 International Conference on Cloud and Green Computing*, 581–586.

[3] Alexa Internet Inc. 2019. Top Sites in Russia By Country.

[4] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. 2012. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)* 9, 5 (2012), 272.

[5] Sophia Alim, Ruquya Abdul-Rahman, Daniel Neagu, and Mick Ridley. 2009. Data retrieval from online social network profiles for social engineering applications. In *2009 International Conference for Internet Technology and Secured Transactions,(ICITST)*, 1–5.

[6] A. Berg. 1995. *Cracking a Social engineer,[Online]. LAN Times*.

[7] Monique Bezuidenhout, Francois Mouton, and Hein S. Venter. 2010. Social engineering attack detection model: Seadm. In *2010 Information Security for South Africa*, 1–8.

[8] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. 2009. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, 551–560.

[9] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[10] Ellie Burns. 2016. Snapchat falls hook, line & sinker in phishing attack: Employee data leaked after CEO email scam. *Computer Business Review*. Retrieved June 15, 2019 from https://www.cbronline.com/business/snapchat-falls-hook-line-sinker-in-phishing-attack-employee-data-leaked-after-ceo-email-scam-4824852/

[11] Alan N. Chantler and Roderic Broadhurst. 2006. Social engineering and crime prevention in cyberspace. (2006).

[12] Anubhav Chitrey, Dharmendra Singh, and Vrijendra Singh. 2012. A comprehensive study of social engineering based attacks in india to develop a conceptual model. *International Journal of Information and Network Security* 1, 2 (2012), 45.

[13] Dan Conway, Ronnie Taib, Mitch Harris, Kun Yu, Shlomo Berkovsky, and Fang Chen. 2017. A qualitative investigation of bank employee experiences of information security and phishing. In *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*, 115–129.

[14] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240.

[15] Christopher P. Diehl, Galileo Namata, and Lise Getoor. 2007. Relationship identification for social network discovery. In *AAAI*, 546–552.

[16] BOJANA DOBRAN. 2018. 7 Most Famous Social Engineering Attacks In History, Be Prepared. *PHOENIXNAP GLOBAL IT SERVICES BLOG*. Retrieved June 15, 2019 from https://phoenixnap.com/blog/famous-social-engineering-attacks

[17] Xun Dong, John A. Clark, and Jeremy L. Jacob. 2008. User behaviour based phishing websites detection. In *2008 International Multiconference on Computer Science and Information Technology*, 783–790.

[18] Stephan Dreiseitl and Lucila Ohno-Machado. 2002. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics* 35, 5–6 (October 2002), 352–359.

[19] Chris Drummond and Robert C. Holte. 2000. Exploiting the cost (in) sensitivity of decision tree splitting criteria. In *ICML*.

[20] Richard O. Duda, Peter E. Hart, and David G. Stork. 2001. Pattern classification 2nd edition. *New York, USA: John Wiley&Sons* (2001).

[21] Matthew Edwards, Robert Larson, Benjamin Green, Awais Rashid, and Alistair Baron. 2017. Panning for gold: automatically analysing online social engineering attack surfaces. *Computers & Security* 69, (2017), 18–34.

[22] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. From data mining to knowledge discovery in databases. *AI magazine* 17, 3 (1996), 37–37.

[23] Mark A. Friedl and Carla E. Brodley. 1997. Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment* 61, 3 (1997), 399–409.

[24] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. 2013. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22nd international conference on World Wide Web*, 447–458.

[25] Robin Gonzalez and Michael E. Locasto. An interdiscplinary study of phishing and spear-phishing attacks. *URL http://cups. cs. cmu. edu/soups/2015/papers/eduGonzales. pdf*.

[26] David Gragg. 2003. A multi-level defense against social engineering. *SANS Reading Room, March*13, (2003).

[27] Sarah Granger. 2001. Social engineering fundamentals, part I: hacker tactics. *Security Focus, December* 18, (2001).

[28] Ryan Heartfield and George Loukas. 2016. A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks. *ACM Computing Surveys (CSUR)* 48, 3 (2016), 37.

[29] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. 2007. Social phishing. *Communications of the ACM* 50, 10 (2007), 94–100.

[30] Sylvain LeJeune. 2018. Social Engineering and connection requests on LinkedIn. *Secplicity*. Retrieved from https://www.secplicity.org/2018/10/31/social-engineering-and-connection-requests-on-linkedin/

[31] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2008. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*39, 2 (2008), 539–550.

[32] Francois Mouton, Louise Leenen, and H. S. Venter. 2015. Social engineering attack detection model: Seadmv2. In *2015 International Conference on Cyberworlds (CW)*, 216–223.

[33] Kevin P. Murphy. 2006. Naive bayes classifiers. *University of British Columbia* 18, (2006), 60.

[34] Gregory L. Orgill, Gordon W. Romney, Michael G. Bailey, and Paul M. Orgill. 2004. The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems. In *Proceedings of the 5th conference on Information technology education*, 177–181.

[35] John Palumbo. 2000. Social Engineering: What is it, why is so little said about it and what can be done?,. *SANS Institute* (2000).

[36] Nicole Perlroth. 2017. All 3 Billion Yahoo Accounts Were Affected by 2013 Attack. *The New York Times*.

[37] Howard Poston. The Top Ten Most Famous Social Engineering Attacks. *Infosec Resources*. Retrieved June 15, 2019 from https://resources.infosecinstitute.com/the-top-ten-most-famous-social-engineering-attacks/#gref

[38] Foster Provost. Machine learning from imbalanced data sets 101. *AAAI Press* 68, 2000 , 1–2.

[39] Irina Rish. 2001. An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence* 3, 22 (August 2001), 41–46.

[40] Lior Rokach and Oded Z. Maimon. 2008. *Data mining with decision trees: theory and applications*. World scientific.

[41] Jamison W. Scheeres. 2008. *Establishing the human firewall: reducing an individual's vulnerability to social engineering attacks*. Air Force Inst Of Tech Wright-Patterson AFB OH

[42] John Seymour and Philip Tully. 2016. Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter. *Black Hat USA* 37, (2016).

[43] Nikita Shindarev, Georgiy Bagretsov, Maksim Abramov, Tatiana Tulupyeva, and Alena Suvorova. 2017. Approach to identifying of employees profiles in websites of social networks aimed to analyze social engineering vulnerabilities. In *International Conference on Intelligent Information Technologies for Industry*, 441–447.

[44] Chris Stephen. 2017. Social Engineers Target LinkedIn: How to Protect Your Organization. *Threat Vector*.

[45] Wenbin Tang, Honglei Zhuang, and Jie Tang. 2011. Learning to infer social ties in large networks. In *Joint european conference on machine learning and knowledge discovery in databases*, 381–397.

[46] Douglas P. Twitchell. 2009. Social engineering and its countermeasures. In *Handbook of research on social and organizational liabilities in information security*. IGI Global, 228–242.

[47] USA Government. 2019. Identity Theft. In Scams and Frauds. Retrieved from https://www.usa.gov/identity-theft

[48] Antanas Verikas, Adas Gelzinis, and Marija Bacauskiene. 2011. Mining data with random forests: A survey and results of new tests. *Pattern recognition* 44, 2 (2011), 330–349.

[49] Christopher Whitrow, David J. Hand, Piotr Juszczak, D. Weston, and Niall M. Adams. 2009. Transaction aggregation as a strategy for credit card fraud detection. *Data mining and knowledge discovery* 18, 1 (2009), 30–55.

[50] Emma J. WILLIAMS and Adam JOINSON. 2017. Understanding Employee Susceptibility to Phishing: A Systematic Approach to Phishing Simulations. *Naturalistic Decision Making and Uncertainty*. (2017), 265.

[51] Email Attack on Vendor Set Up Breach at Target. *KrebsOnSecurity*. 2014. Retrieved from https://krebsonsecurity.com/2014/02/email-attack-on-vendor-set-up-breach-at-target/