Faculty of Computer Science and Information Technology

Phishing Website Detection Using Website Logo

Chang Ee Hung

Master of Science
2019

Phishing Website Detection Using Website Logo

Chang Ee Hung

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Computer Science)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2019

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. It is original and is the result of my work, unless otherwise indicated or acknowledged as referenced work. The thesis has not been accepted for any degree and is not concurrently submitted in candidature at any other degree.

Name : Chang Ee Hung

Matric No. : 14020193

Faculty : Faculty of Computer Science and Information Technology, University Malaysia Sarawak

Dated : June, 2018

ACKNOWLEDGEMENT

I want like to express my gratitude and appreciation to several individuals who have been a great source of help and encouragement to me throughout my entire Master's research. First of all, I would like to thank my supervisor, Dr. Chiew Kang Leng for his kindness and patience in guiding me and encouraging me when I am in the midst of slack, chaos and discouragement. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. His constant presence and coaching keep me persevere and strive harder.

I also want to thank my beloved parents, Chang Nyong Ng and Thia Teek Inn for their uncountable love and caring in my life. I know that they always gives their constant support and prayers at my back. I cannot image if they were absent in my life.

I also wish to extend my appreciation to my wife, Janice U Shi En, for supporting me all the way in my Master's research. Although I need to sacrifice some precious moment with her for my research work, she is patiently bearing with it and continue supporting me. I am grateful for having her to accompany me through every sweet and sour moment in life.

Not forgetting my senior research mate, Colin Tan Choon Lin who helped me greatly in improving my work. Thousand thanks for his kindness and willingness to share his time and knowledge to me. Without his accompany and support, I believed I cannot reach to the altitude where I stood today.

Finally, I would like to express my highest gratitude to my God, Jesus Christ. A lot of challenges and hardships happened during my Master's research life, but by His grace and goodness, all difficult moments has been passed through as a grateful and wonderful experiences.

ABSTRACT

Phishing is an online security threat that combines social engineering and website deceiving technique to steal internet users' confidential credential. In order to protect internet users from phishing attacks, a hybrid phishing detection method has been proposed. The proposed method utilises logo image and search-engine to determine the identity consistency of a query website, where consistent identity indicates legitimate website and inconsistent identity indicates phishing website. The proposed method consists of two processes, namely logo extraction and identity verification. The first process will detect and extract the logo image from all the downloaded image resources of a webpage. Machine learning was integrated into the first process in order to ensure correct detection of the logo image. Based on the extracted logo image, the second process will employ the Google Image Search engine to retrieve the portrayed identity. Since the relationship of the logo and domain name is exclusive, the domain name is referred as the identity. A comparison will be performed between the domain names that are returned by Google with the one from the query website to verify the identity. Experiments were conducted over 1,000 samples with the true positive rate of 99.80% while the true negative rate is 87.00%. The promising results showed the reliability and capability of proposed method in detecting phishing websites. Benchmarking results also demonstrated the proposed method is superior than the existing similar method. In summary, the proposed method proved the effectiveness and feasibility of using a graphical element such as the logo in identity determination and phishing detection.

Keywords: Phishing detection, website logo, website identity, Google image search, identity consistency, logo extraction

Penggunaan Logo Laman Web untuk Pengesanan Laman Web Palsu

ABSTRAK

Phishing merupakan salah satu jenayah siber yang melibatkan kejuruteraan sosial dan teknik laman web palsu untuk mencuri maklumat sulit daripada pengguna Internet. Demi keselamatan para pengguna internet, satu kaedah hibrid yang boleh mengesan laman web palsu telah dicadangkan. Kaedah tersebut akan menentukan konsistensi identiti sesuatu laman web dengan menggunakan imej logo serta enjin carian. Jika identiti yang didapati adalah konsisten, maka ia merupakan laman web yang sah, manakala identiti yang tidak konsisten menunjukkan laman web tersebut adalah palsu. Kaedah tersebut terdiri daripada dua proses, iaitu proses pengesktrakan logo, dan proses pengesahan identiti. Pengesanan dan pengekstrakan imej logo akan dilakukan ke atas semua sumber gambar yang dimuat turun daripada laman web semasa proses pertama. Pembelajaran mesin diintegrasikan ke dalam proses pertama untuk memastikan ketepatan pengesanan imej. Berdasarkan imej logo yang diperolehi, Google Image Search akan mendapatkan identiti imej logo tersebut semasa proses kedua. Disebabkan hubungan logo dan nama domain adalah eksklusif, maka nama domain digunakan untuk menentukan identiti sebenar. Untuk menentukan identiti laman web, perbandingan dibuat antara nama domain yang dikembalikan oleh Google dengan laman web yang diuji. Eksperimen dijalankan ke atas 1000 lebih sampel dan mendapatkan keputusan 99.80% dalam pengesanan laman web palsu, serta 87.00% dalam pengesanan laman web tulen. Keputusan perbandingan juga menunjukan kaedah cadangan adalah lebih cekap daripada kaedah lain yang serupa. Kesimpulannya, kaedah cadangan membuktikan keberkesanan dan kebolehlaksanaan elemen grafik seperti logo dalam pengesahan identiti serta pengesanan laman web palsu.

Kata kunci: *Pengesanan laman web palsu, logo laman web, identiti laman web, Google image search, konsistensi identiti, pengekstrakan logo*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Accuracy
APG	Anti-Phishing Gateway
API	Application Programming Interface
APWG	Anti-Phishing Working Group
BUPT	Beijing University of Posts and Telecommunications
CANTINA	Carnegie Mellon Anti-phishing and Network Analysis Tool
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DNS	Domain Name System
DOM	Document Object Model
EMD	Earth Movers Distance
FLD	Fisher Linear Discriminant
FN	False Negative
FP	False Positive
GSI	Google Search by Image
HTML	HyperText Markup Language
NER	Named Entity Recognition
NN	Neural Network
OCR	Optical Character Recognition
QR	Quick Response
RAM	Random Access Memory
SEO	Search Engine Optimisation
SIFT	Scale-Invariant Feature Transform
SLD	Second-level Domain

SSL	Secure Sockets Layer
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TLD	Top-level Domain
TN	True Negative
TP	True Positive
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 Research Background

The advancement of information technology has lightened up our life as we are able to handle many daily works by using the Internet services. For example, instead of going to the respective service counter, people nowadays are able to pay their bills at any place they feel convenient with the availability of internet connection. A wide range of services are offered via multiple online platforms, ranging from casual information sharing (e.g., social networking) to a more monetary intensive related application (e.g., E-banking, E-commerce, online payment, etc.). Hence, most of the people would possess multiple login account for different purposes. However, the extension of this convenience comes along with some illegal profit-oriented threats known as online crime. The online criminals normally gained their profit by exploiting the vulnerability of internet account security. One of the most common online crimes is called the online phishing.

Online phishing is a security threat which combines social engineering and website spoofing techniques to deceive users into revealing their confidential information. Typically, phishers will try to harvest the online users' credential such as username, passwords and credit card details by masquerading as a trustworthy entity on the Internet. Usually the phishers will begin by sending a huge number of emails that appears as if it was sent from a genuine party. The email content is crafted to create a sense of urgency, worry, or offer some great incentive that compels the victims to take action. For example, the email will urge the victims to update their confidential information (e.g., login password) before his or her account is suspended. Once the victim innocently updates the confidential information, the

phishers will gain all necessary details, as the information of the user is sent to the phisher's counterfeit website rather than the genuine one. They will then use the victim's credential for illegal access on the genuine website.

The phishing threat did not show any signs of diminishing, but has grown considerably in the past decade. The main reason is that the phishers continuously improve their strategies to exploit the human factors. Many internet users do not have enough knowledge on the internet application. For example, they might not understand what is Uniform Resource Locators (URLs), or do not know how to utilise the security indicator such as Secure Sockets Layer (SSL) protocol or digital certificate which are nowadays available in most of the browser. Some of the users just rely on the webpage contents to determine the genuineness of a webpage rather than the URL (Mohammad et al., 2015a). However, with the advancement of web technology, the phishers are becoming more matured in exploiting visual deception. This has posed great risks to many unsuspecting users.

According to the statistic done by the Anti-Phishing Working Group (APWG), the total number of phishing attacks in 2016 was 1,220,523, which was a 65% increment over the previous year. Furthermore, APWG observed that the phishing activity in early 2016 was the highest ever recorded by the APWG since it began monitoring in 2004. During the fourth quarter of 2004, the APWG recorded 1,609 phishing attacks per month, but in the fourth quarter of 2016, APWG recorded an average of 92,564 phishing attacks per month, which is an increase of 5,753% over the 12 years (Anti-Phishing Working Group, 2017). The escalating trend is believed to be driven by the high profitability of the financial and cloud storage/file hosting institutions. According to an analysis by a cyber-security firm, PhishLabs (2017), the financial institution is found to be the most phished industry, accounting for 23% of the total phishing attacks in 2016, followed by the cloud storage/file hosting at 22.6%. More than 50% of all phishing attacks documented in 2016 are focused on these two

industries.

Phishing is a very serious problem, which causes a multitude of pitfalls which include identity theft, stolen money, unauthorised account access, and credit card fraud. The impact is dreadful as it brings tremendous financial losses every year. Phishing has been recognised as one of the fully industrialised economy crimes since 2004 (Stoodley, 2004). According to a report by RSA in December 2014, the businesses around the world lost a total of US$453 million as a result of phishing attacks. RSA (2017) forecasted that the phishing threat will create an annual cost of US$9.1 billion to global organisations in 2017. Besides the tangible losses, phishing also causes long term damage (e.g. the reputation, credibility or confidence losses).

1.2 Problem Statement

Although various phishing detection methods have been introduced by security vendors and scholars throughout the years, there is still no method that can provide a complete bullet-proof protection against phishing, as the phishers are continuously developing their techniques to deceive victims.

The biggest problem in fighting phishing always comes from the human factors. As mentioned in the previous section, phishers continue exploit the human factors (e.g., computer illiteracy and carelessness) as the loophole to achieve their purpose. Lacking of computer knowledge and self-carelessness could blindfold the Internet users from noticing the warning message from security indicators or any other sign of phishing attack. Based on the human behaviour studies by Alsharnouby et al. (2015) and Dhamija et al. (2006), the success rate of phishing attack on a typical Internet user is always fairly high. Therefore, although improving the public awareness is important in fighting against phishing attacks, it

is even more crucial and necessary to equip the users with a more automated security mechanism.

Besides that, a formidable problem in phishing detection is the challenge of detecting the newly launched phishing webpages (zero-hour phishing attack). It is an open challenge faced by most of the phishing detection approach (Jain & Gupta, 2017). To detect a new phishing webpage is not an easy task, as it might be a new variation that has overcame the existing detection mechanism. Phishers continuously put a lot of hard work in evolving their phishing techniques. Hence, existing heuristic-based methods that capitalise on suspicious features might eventually become inconsistent and ineffective over the time. For example, phishers can overcome the detection methods that extract features via HTML analysis by altering the HTML code or avoid from the use of suspicious features.

Next, the visual similarity-based methods might become incapable if the appearance or the visual elements of a webpage have been intentionally modified by the phishers. The weakness is due to the reliance on the predefined database to perform the similarity comparison. If the database is not up to date, the classification results will be affected. However, to maintain an up-to-date database of legitimate images to act as the image similarity reference will be a costly action (Zeydan et al., 2014). Similar database maintenance problem occurs in the list-based method as well. Hence, the cost of maintaining a predefined database has become one of the bottlenecks in phishing detection work (Dudhe & Ramteke, 2015).

Lastly, the language limitation is often a critical weakness in some phishing detection techniques that uses textual analysis. For example, existing works done by Xiang & Hong (2009), Ramesh et al. (2014), and Verma & Hossain (2014) tend to rely on semantics features that only works for English environment, thus causing their methods to become vulnerable when encountering non-English webpages.

1.3 Research Objectives

To address the problems highlighted in previous section, the following research objectives are outlined:

(a) To propose an automated phishing detection method that does not require user interaction.

(b) To extend the phishing detection on non-English webpages.

(c) To overcome the need and reliance on a self-maintained database.

1.4 Research Scope

Phishing attack are not only presented in different techniques, (e.g. deceptive phishing, whaling, pharming, Quick Response (QR) code phishing and etc.), but also performed via different means, such as email, website, hand phone or even postage. However, this research is only focuses on website-based phishing, which fall under the deceptive phishing attack category, which specifically refer to the instance when the users encountered a phishing webpage.

1.5 Research Significance

This research carried the following significance:

(a) Overcome the limitation of HTML content analysis phishing detection method.

(b) Offer a robust phishing detection method by utilising the logo of the targeted legitimate entity as a constant phishing feature.

(c) Reduce the cost in maintaining an up-to-date predefined database by utilizing existing Google's database.

(d) Elevate the capability of phishing detection despite the languages used by the webpages.

1.6 Thesis Overview

This thesis is distributed into five distinct chapters. The contents of the chapters are summarised as follows:

Chapter One: Introduction, introduces the background of research and briefs about the latest trend in phishing. This chapter also addressed the phishing detection issues that need to be overcome, and also the objectives and scope of research. In the last part of this chapter, some significance of the research has been mentioned.

Chapter Two: Literature Review, provides the fundamental understanding on different types of phishing approaches and reviews a variety of common phishing detection techniques.

Chapter Three: Methodology, presents the overall framework and explains the whole implementation process of the proposed method. In this chapter, every component of the proposed method will be explained in detail.

Chapter Four: Results and Analysis, elaborates the experiment setup and dataset specifications. Results of the proposed method are presented and the performance comparison with another similar phishing detection method has been shown too. This chapter also includes the discussion about the limitations of the proposed method and the possible countermeasures.

Chapter Five: Conclusion, summarises the whole research, and verified the contribution attained from completing this research. Presents some future plans that may be explored to improve the proposed method.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter introduces various types of phishing techniques and reviews on existing phishing detection approaches. Related evaluation metric will be introduced as well in the final section.

2.2 Phishing

"Phishing" is metaphorically similar to the traditional word ‘Fishing’. However, instead of using worms as bait to catch the fish, ‘phisher’ use counterfeit webpage as bait to lure internet users and get their credential information (Gupta et al., 2017). The phishing attacks were historically started by stealing America Online (AOL) system accounts in early 1990s, and over the years it has moved into attacking more profitable areas, such as online banking or e-commerce services (Khonji et al., 2013). As the internet technology is continuously advancing, the phishing techniques presented by phishers are also correlatively becoming more matured and complex. Today, there is a diverse range of phishing techniques employed by phishers, some of the major phishing attacks will be presented in the following sections.

2.2.1 Deceptive Phishing

Deceptive phishing is the most common and typical phishing technique (Chaudhry et al., 2016). As mentioned in the previous chapter, it usually begins by spreading out mass emails which pretended to be from a trustworthy party, such as service providers, financial organisations, or government agencies (Nirmal et al., 2015). Those emails always come with a warning message that will cause the readers to feel anxiety at the very first place, or a message that exploit human greed by offering free benefits. Normally, the users will be asked to access a provided link at the end of message for further action. Users will need to input their credentials at the redirected website to solve the mentioned critical issues or obtain the free benefits (IDG Consumer & SMB, 2007). Once the users reached the phishing website, most of them will be convinced by the appearance and the content of the fake website, thus keying in their credentials or clicking the download button without hesitation. In the end, the phishers successfully obtain users' credentials and use it to login the legitimate.

It is important to mention that the downloaded items from phishing website are highly probable of containing some malicious programs such as Trojan or keylogger. These types of malicious software will masquerade as harmless system's background processes, and secretly recording user credential information (i.e., keyboard input). The recorded credential will be sent to phisher for further malicious action. If the downloaded files contain more harmful programs like ransomware, the phishing case will be escalated to a more serious form of cybercrime which could incur higher damages.

Figure 2.1 and Figure 2.2 are examples of the phishing website targeting PayPal and its corresponding legitimate website, respectively. Figure 2.3 shows the overview of the deceptive phishing mechanism which is described at the first paragraph of Section 2.2.1.

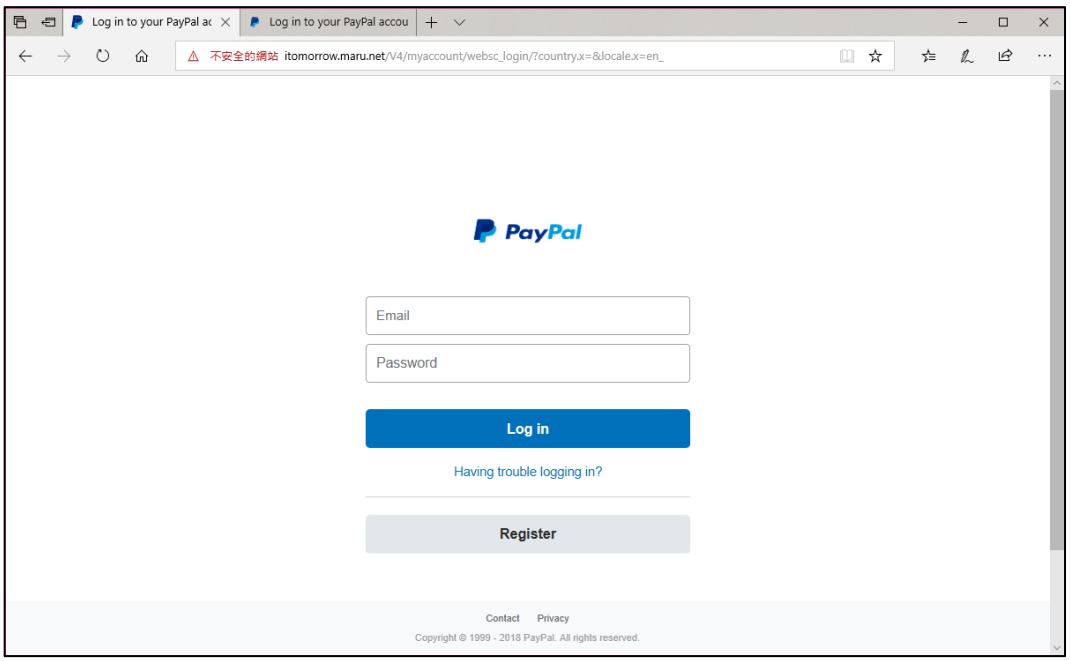


Figure 2.1: A sample of phishing webpage targeting PayPal

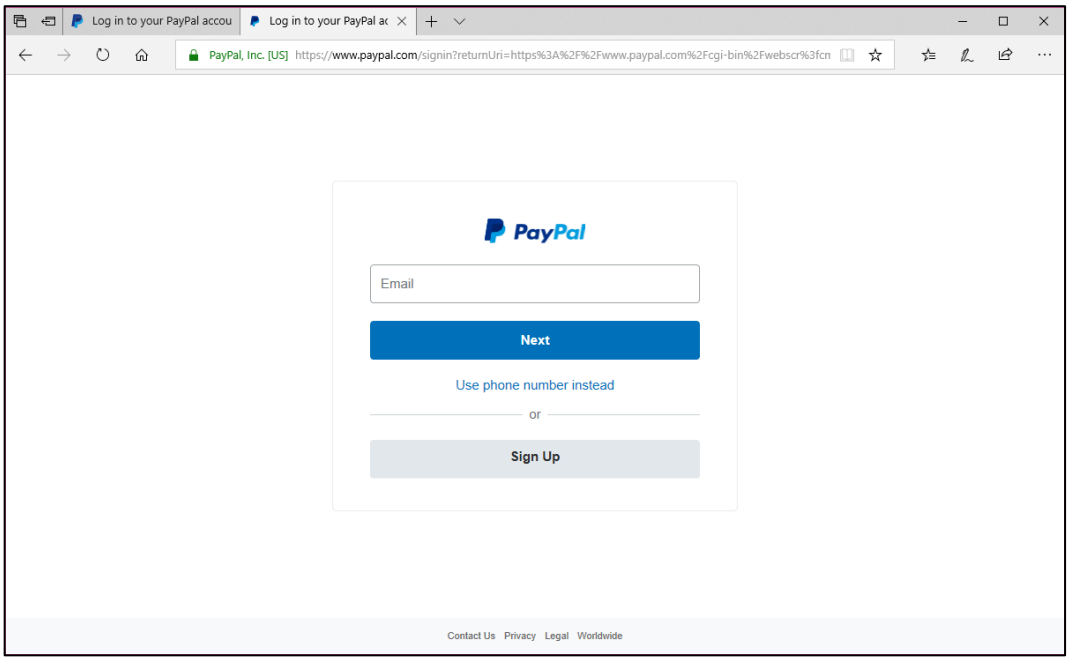


Figure 2.2: A sample of actual PayPal webpage (Paypal, 2018)

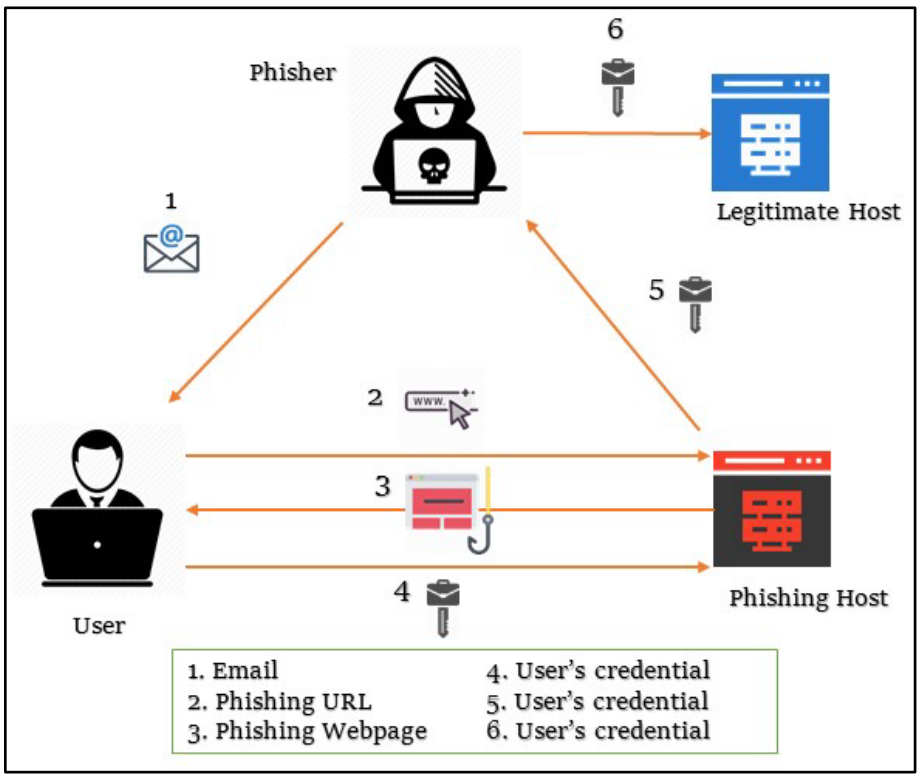


Figure 2.3: Deceptive phishing mechanism

2.2.2 Whaling

Whaling is another kind of phishing attack and it happens when phishers managed to obtained the CEO or other top executives login credentials (e.g. company email account) via deceptive phishing or malware-based phishing (Gupta et al., 2017). With the compromised email account, phishers will send emails to the lower-ranking employees and request some kind of financial transaction to be performed.

2.2.3 Pharming

Pharming attack redirects users to a fraudulent website even if the victims had input the correct and legitimate URL. This is possible because the phishers probably have successfully altered the content of the victims' local DNS file or the connected DNS server (Cisco, 2017).

2.2.4 QR Codes Phishing

Due to the mobile device popularity, phisher have extended their focus into mobile device territory. One of the attacks is QR codes phishing (Kharraz et al., 2014). Users will be redirected to a phishing or malicious website that host malware by just scanning the QR code, without the need to type the URL. Due to the smaller size of mobile device screen, it is difficult for users to notice the suspicious URL. Furthermore, depending on various condition, the decoding of QR codes can result in an uncontrolled process (i.e., download or send text) which might cause more serious damages (Mavroeidis & Nicho, 2017).

2.3 Anti-Phishing

In order to protect the internet users from falling into phishing scam, a wide range of anti-phishing systems have been introduced over the years. Although there are numerous anti-phishing techniques, it can be divided into two classes. Namely, user education and software automated. User education is to educate the internet users about online security, especially on how to utilise the security indicator and how to differentiate phishing webpage from legitimate. Software automated refers to anti-phishing techniques which protect internet users from phishing websites by using automated software. Software automated class can be further divided into phishing prevention and phishing detection approaches.

The phishing prevention approaches try to prevent phishing by enhancing the security features. The idea is to increase the difficulties for phisher to exploit the system vulnerability. For example, Siddique et al. (2017) has proposed an anti-phishing framework based on visual cryptography. Visual cryptography retains the secrecy of an image captcha by

breaking down the original image captcha into two pieces that are stored in two different servers. To reveal the original image captcha, both decomposed image captcha must simultaneously be available. Users can use the original image captcha as password once it has been revealed. The website thus can cross verify and proves its identity. Na et al. (2014) also proposed similar two server authentication schemes that are based on SSL/TSL. Both of this method has increased the difficulties for phisher to exploit the system vulnerability.

The second software automated approach is phishing detection. This approach is used to detect and warn users when they encounter a phishing attempt. This approach is important when prevention is ineffective or when it is impossible to have a prevention strategy. Our research will focus on this scope. The general categorisation of anti-phishing approaches is shown in Figure 2.4.

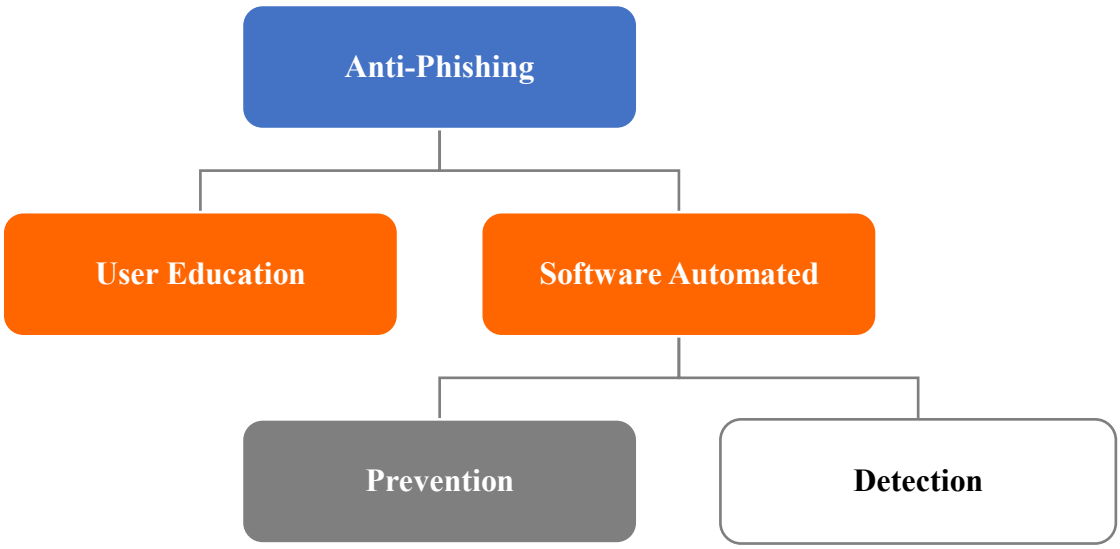


Figure 2.4: Anti-phishing overview

2.4 Phishing Detection

In general, phishing detection can be divided into list-based and heuristic-based approaches. List-based approach is used to check the existence of a query website in a predefined list of URLs. The list can be a black list, a white list, or both. On the other hand, the heuristic-based is a series of mechanism extracting some distinctive features from a query webpage to initiate the phishing detection. Indeed, heuristic-based approach is considered more prominent and received greater attention because it has overcome the issue of over-reliance on a predefined list to detect a phishing website (Gastellier-Prevost et al., 2011).

The heuristic-based approach utilises discriminative properties extracted from the query website to decide whether it is a phishing website or not. Usually, the discriminative properties are extracted from the HTML content, the structure of the URL, and third party information (e.g., WHOIS lookup and Alexa). Due to its flexibility and ability to detect new phishing websites, there exists a variety of combined techniques in this approach.

In general, heuristic-based approach can be divided to identity-mediated and unmediated approaches. In identity-mediated approach, the method will firstly determine the portrayed identity of the query website (Gowtham et al., 2017). If the portrayed identity has been proven to be different from the query website identity, the query website will be labelled as a phishing website. For example, if a website portrayed it as the PayPal website, but after the evaluation on the URL of the website, the results showed its URL did not match to the PayPal domain, hence the website will be classified as a phishing website that is masquerading as PayPal.

In the second category, the proposed method will start from feature extraction process, followed by the detection process. Machine learning and similarity measurement are some of the common approaches used in the detection process (Abu-Nimeh et al., 2007). It is

worth noting that the method in this category does not require determining the portrayed identity prior to the detection, and the portrayed identity is only used indirectly.

The overview and derivation of phishing detection approaches are shown in Figure 2.5.

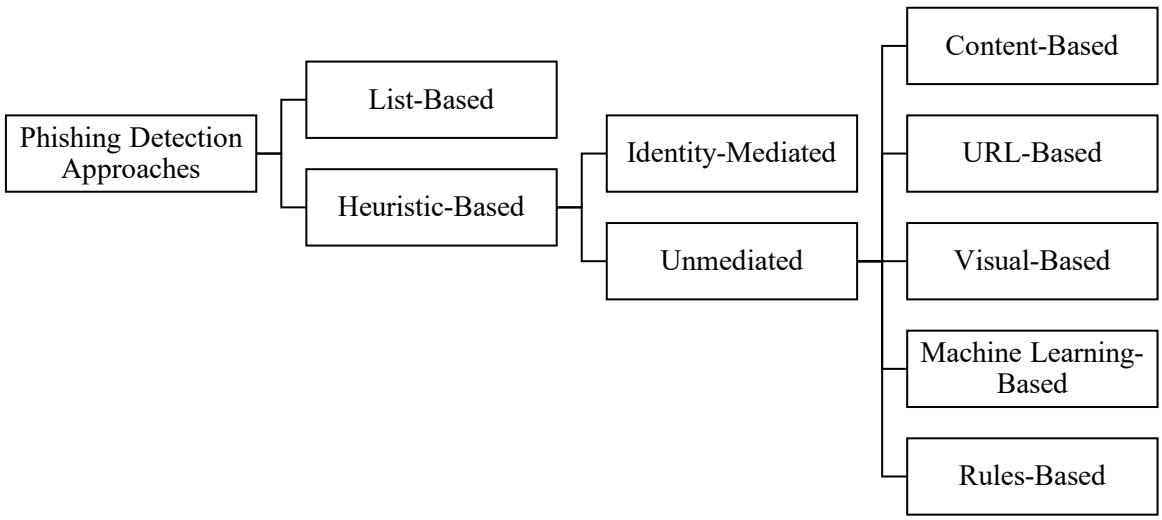


Figure 2.5: Overview and derivation of phishing detection approaches

2.4.1 List-Based

According to Jianyi Zhang et al. (2011), blacklisting appears to be one of the popular techniques in the anti-phishing community. Many popular web browsers have integrated this technique to detect the phishing websites (Schneider et al., 2008; Abrams et al., 2013). In this technique, a query website is checked with a list (i.e., a list of known phishing URLs), which is compiled and maintained by some consortium or organisation. If the checking returns a match, then the website will be labelled as phishing. On the contrary, instead of maintaining the blacklist, one can compile a list of legitimate URLs. This technique is known as whitelisting, which is also a type of list-based approach.

A method proposed by Cao et al. (2008) is one of the whitelisting examples that will maintain and store a whitelist at the client side automatically. Prakash et al. (2010) later introduce a more dynamic and flexible list-based approach, called PhishNet. This method will generate multiple variations of URLs based on the existing blacklist, and the generated URLs will be served as a predictive blacklist. Their dataset can be concluded as URL oriented, as this type of method only utilises the webpage URLs.

Although list-based approach provides the simplicity in designing and implementation, great effort is needed to keep the list complete and up-to-date. Hence, list-based approaches are now commonly combined with heuristic-based approach as a pre-processing step.

2.4.2 Identity-Mediated Phishing Detection

One of the effective ways to detect phishing website is by finding the identity consistency of a webpage. A notable example of identity-mediated phishing detection is the one proposed by Xiang & Hong (2009). Their methodology included two components, the first one is identity recognition and the following is keyword retrieval detection. In the identity recognition component, two textual objects from the Document Object Model (DOM) (i.e., the title and copyright field) will be used to determine the identity. However, if the title and copyright field are absent, it will employ a Named Entity Recognition (NER) technique to find the identity. NER is the task of identifying various types of entity names in free text, such as persons, organizations, etc. On the other hand, the keyword retrieval detection component used the Term Frequency-Inverse Document Frequency (TF-IDF) to identify keywords. After that, both of the components utilise search engine from Google and Yahoo

to determine whether a query website is a phishing website or not.

Another identity-mediated heuristic approach is the one proposed by Liu et al. (2010). This method will determine the targeted identity of a phishing webpage when the phishing webpage has been detected. The methodology utilised the idea of a self-organised semantic data model known as Semantic Link Network. This model is often used for organising web resources. Although this method involved different types of detection mechanisms, its groundwork are still based on textual element (i.e., the extraction of hyperlinks, keywords and textual contents for the operation of link relations, search relations and text relations, respectively).

GoldPhish is another prominent method related to the identity-mediated phishing detection approach (Dunlop et al., 2010). The main idea is using the Google search engine to determine the identity of a query webpage via the extracted textual content. In this method, optical character recognition (OCR) will be performed on the captured webpage screenshot to extract all possible textual contents (including the text within the logo image). Later, the extracted text will be fed into Google search engine and the search result is evaluated. Since Google search engine always return the most relevant website, the domain name of the returned relevant websites list is very likely to match with the query website. If no matching occurs throughout the list, the query website will be concluded as phishing webpage. This is because the portrayed identity (relevant website) is different from the real identity (query website).

Ramesh et al. (2014) proposed another identity-mediated heuristic approach. This method begins with seven keywords extraction from query webpage content using TF-IDF technique. The extracted keywords are later fed into search engine to get the search result. Three sets of domain names will be formed to determine the legitimacy of the query webpage. The first set of domain names is extracted from the URLs in the search results while the

second set is obtained from the hyperlinks in the query webpage. The third set of domain names list which contains the target domain name will be formed from the intersection of the first two sets of domain names. Later, the proposed method will deduct the final target domain name from the third set of domain names list.

It is worth mentioning that most of the identity-mediated approaches are based on the textual content analysis. The main limitation of using textual elements to determine the website's identity is the constraint by the language dependency. For example, an English-based detection method is only effective for website written in the English language, but not for the Spanish language. To the best of our knowledge, there are very few research studies from the identity-mediated heuristic approach in the literature which use graphical elements to determine the identity of a website. In particular, there is no research employing the logo to determine the website's identity as the one proposed in this research. The closest method to ours is GoldPhish (Dunlop et al., 2010). However, GoldPhish is only using the text within the logo and not the logo as a whole.

2.4.3 URL-Based

One of the simple yet effective ways for phisher to deceive users is by obfuscating the phishing URL. Many unsuspecting users easily overlook the tiny differences showed by the phishing URL. Hence, there are many anti-phishing researchers that put their attention on the URL-based approach (Maurer & Höfer, 2012). Generally, URL-based approach utilises lexical and host features from URL to determine if a query webpage is a phishing website. Study has found that phishing URL usually contained the targeted brand name and highly to have vowels and different character (McGrath & Gupta, 2008). Long URL and short domain

name are also a good indicator for phishing sign.

Works by Maurer & Höfer (2012) is another interesting URL-based approach. They utilised spelling recommendations from a search engine and a string similarity algorithm to determine whether a query page is a phishing website. Since every website has unique URL, phishers can only alter a little bit on spelling to imitate the targeted legitimate URL and lead the careless users to overlook it. The spelling recommendation is used for predicting the targeted legitimate URL, and the string similarity algorithm is used for finding the similarity rate between the legitimate and the suspected URL. If the similarity rate is high, it is a sign of a phishing website. As acknowledged by the authors, if the phishing URL does not contain any spelling mistake, the proposed method may fail.

Tan et al. (2014) use brand name in URL as a weighting feature to detect phishing website. In order to convince users to believe they are accessing a correct legitimate website, phisher normally will add the targeted brand name in somewhere of the phishing URL. The authors begin by assigning weights to the extracted words from HTML content. After that, based on the co-appearance at hostname, path and filename of a URL, several keywords will be chosen as the brand name candidate from the extracted words. The chosen brand name candidates will be fed into Yahoo Search engine to retrieve highest frequency domain name among the top 30 returned results. A WHOIS lookup is performed to find the owner of the selected domain name. If the owner of query domain name differs from the owner of domain name returned by the search engine, it is likely to be a phishing website. Some other URL-based works include Gupta & Singhal (2018), Bahnsen et al. (2017), Xue et al. (2016) and Nguyen et al. (2013).

2.4.4 Content-Based

A popular example of content-based method is called CANTINA (Zhang et al., 2007). This method calculates the TF-IDF from the content of a website and generates a lexical signature, which are used later as the keywords list for search engine query. Based on the returned result, CANTINA will determine the legitimacy of the query website. Xiang & Hong (2009) later enhanced the CANTINA keywords-retrieval methodology by implementing the identity-based detection algorithm. This method will first utilise two textual objects from the DOM (i.e., the title and copyright fields) and employ a technique called Named Entity Recognition (NER) to determine the identity. If the attempt failed, the second step will utilise the CANTINA keywords-retrieval methodology. The main features from these methods (i.e., TF-IDF, DOM and hyperlinks) are retrieved via HTML analysis; hence the HTML file will be the major component for their dataset.

Zhang et al. (2010) from Beijing University of Posts and Telecommunications (BUPT) introduce another content-based phishing detection technique that based on Anti-Phishing Gateway (APG), which called BUPT-APG. This method utilises an adaptive cosine similarity to calculate the similarity between the generated template in repository and the query webpage. The similarity results will determine the legitimacy of the query webpage.

Recently, Corona et al. (2017) has proposed a content-based phishing detection method called DeltaPhish. The main idea is to determine phishing by comparing the HTML code and some visual elements between the query webpage and the targeted legitimate homepage. 11 features have been extracted from webpage HTML codes, while the Histograms of Oriented Gradients (HOGs) and color histogram were obtained from webpage snapshot. The comparison of HTML content and visual element will be done by classifier. The classifier will output a dissimilarity score, in where the higher score indicate the higher probability of

being as phishing. In addition, the works from Soman et al. (2008), Mohammad et al. (2014), and Abdelhamid (2015) also contributed to the content–based method.

2.4.5 Visual-Based

Another interesting method that belonged to heuristic approach is visual-based method, where it uses the visual similarity measurement to detect phishing webpages. Generally, this approach consists of two steps: (i) visual feature extraction, and (ii) similarity measurement. In the feature extraction process, a set of features will be extracted from the query webpage. The examples of the feature extracted could be webpage layout (Rosiello et al., 2007), wavelet (Medvet et al., 2008), or Scale-Invariant Feature Transform (SIFT) (Huang et al., 2010). Based on the extracted features, similarity measurement process will compute the similarity value between the query webpage and the webpages in database. If the similarity value exceeded a certain threshold (means highly similar), yet the URL of query webpage is not matched with any domain name recorded in the database, it will be classified as a phishing webpage. For example, Liu et al. (2006) proposed a series of visual approaches in phishing classification. This method analyses the HTML webpages and decomposes them into salient blocks, and then calculates the similarities indicated by three metrics: block-level (detail), layout (global), and style (overall). Fu et al. (2006) also proposed a visual-based method that utilises the Earth Movers Distance (EMD) to calculate the webpage visual similarity for phishing detection. The authors will snapshot the query webpages and convert them into low resolution images and then use the colour and coordinate features to represent the image signatures. After that, the EMD algorithm is employed to calculate the signature distances of the converted images. If the similarity value of a webpage exceeded the

predefined threshold, the webpage will be classified as a phishing website.

The visual-based methods require more raw elements (e.g., screenshot, favicon, image and complete HTML page) in their dataset for feature extraction. Research done by Haruta et al. (2018), Mao et al. (2013), Zhang et al. (2013), and Hara et al. (2009) are all belonged to the visual-based method.

2.4.6 Machine Learning-Based

Another popular method is to employ a machine learning technique. For example, Garera et al. (2007) analysed the structure of URLs to determine various patterns which are usually exploited by phishers. From the analysis, the authors proposed several features which include page, domain, type, and word based features. With these features, the authors used logistic regression to distinguish a phishing from a legitimate webpage. Recently, Jain & Gupta (2018) utilise Support Vector Machine (SVM) and Naïve Bayes classifier on 14 extracted feature from URL to detect phishing, and their results are rather promising. Moreover, Jain & Gupta (2015) also proposed another machine learning technique which employ logistic regression classifier on 12 categories of hyperlinks features. This technique does not require third party services and is language independent. Some other similar methods which used a machine learning technique can be found in Sorio et al. (2013) and Chu et al. (2013).

2.4.7 Rules-Based

In a rule-based approach, the classification of phishing or legitimate webpage is taken control by a set of rules. Mohammad et al. (2014) proposed a method that combines 16 rules and utilises a classifier to do the classification. Those rules act as the transformation agents that transform multiple features into three states, which are "legitimate", "suspicious", and "phishy". The features are extracted from the webpage and can be categorised as address bar-based, abnormal-based, HTML-Javacript-based, and domain-based features. Instead of using the raw features directly, the method will feed the states to the classifier and C4.5 classifier is used in the classification. Clearly, this method is better in terms of speed, as it has less dimensionality (i.e., only the three states as opposed to the raw features set). However, the effectiveness towards more complicated phishing webpages such as those impersonating multiple targets is not guaranteed, since the transformation will definitely result in information loss. Similar rule-based approaches include Cook et al. (2009) and Abdelhamid (2015).

2.4.8 Discussion on the Strengths and Weakness in Existing Phishing Detection Techniques

This section discusses the strengths and weaknesses of existing phishing detection approaches, and summarise it into a table at the end of the section.

List-based approach is the most simple and easiest phishing detection solution, however to maintain an up-to-date and complete list requires great cost, and always suffers from the incompleteness.

For the identity-mediated approach, the significant advantage will be the capability of discover the portrayed identity of the phishing website. With the discovered portrayed identity, a more reliable decision could be made in the phishing determination process upon the query webpage. Nevertheless, as mentioned earlier, most of the identity-mediated approaches are based on textual elements and is limited by language dependency.

URL-based approach is another simple yet effective approach which is popular among the anti-phishing researcher. The drawback of this type of approach is, the phisher can easily evade from the detection by avoiding the use of suspicious elements, such as symbol or spelling mistake in URL.

Content-based approach has great potential in extracting various type of feature from webpage contents (e.g. HTML codes) for utilisation in phishing detection. However, the results are highly dependent on the HTML content. If the phishers manipulate the HTML code carefully to avoid the detection, the effectiveness and outcome will probably be affected.

Visual-based approach is good in detection when the phisher is using the images to replace the textual content in their phishing webpage. Nevertheless, it is vulnerable when the webpage layout is altered or the targeted webpage appearance is updated. Moreover, most of the proposed method in visual-based approach is depending on a predefined database. Hence, similar to list-based approach issue, it needs tremendous effort to keep the database maintained and up-to-date.

The results of machine learning-based approach are usually influenced by the extracted or selected feature. The analysis process is dissimilar in different type of machine learning, and some of it might have higher cost in computational power. Meanwhile, rule-based approach has better advantages in lowering the computational cost, yet the efficiency is not guaranteed when faced with complicated phishing webpage, such as multiple identities phishing webpage.

The overview on above discussion has motivated us to propose an identity-mediated phishing detection which uses visual element, but not depending on a predefined database, and robust to the language constraint issue.

Table 2.1 summarises the strengths and weaknesses of existing phishing detection approaches in different categories.

Table 2.1: Strengths and weaknesses of existing phishing detection approaches

Category	Strengths	Weaknesses
List-based	• Simple and easy to implement	• Hard to maintain a complete and up-to-date list
Identity-mediated	• Discovery of portrayed identity increase accuracy in phishing determination	• Dependence on the feature selection
URL-based	• Robust against visual and textual content manipulation	• Can be avoided with less suspicious features (true positive rate drops)
Content-based	• Multiple features can be selected for extraction and utilised.	• Language dependency (TF-IDF limitation)
Visual-based	• Overcome textual contents-based limitation • Free from language limitation	• Can be avoided by altering the webpage layout • Difficult to maintain an up-to-date images database
Machine Learning-based	• Easy to utilise	• Reliance on the selected features and machine learning type.
Rule-based	• Better in speed, less dimensionality	• Ineffective against complicated webpage.

2.5 Logo Detection

Million of images are created and published every day on the internet. Finding the logos among the online images has become one of the interest of image processing researchers and commercial organizations nowadays. It is because the logos are unique visual object which access to the identity of something or someone. From commercial and industrial view, logos played an important role in the customers' expectations associated with some particular product or service. Hence, logo detection mechanism could help an organisation to analyse information about the popularity of their brand by counting how often its logo appears on press or social media (Glagoļevs & Freivalds., 2017).

To handle the logo or trademark images, Jain & Vailaya (1998) proposed a method to retrieved logo or trademark images from images database based on shape information. This method consists of two stages. In the first stage, they utilise histogram of edge direction of a query images as a shape features to generate a moderate of plausible retrieval from database. In the second stage, the candidates from first stages will be screened through using a deformable template to find the best matches. Mehtre et al. (1998) proposed another method which combines the shape and colour features of an image based on a clustering technique to find logo or trademark images. This method applied clustering algorithm to find the colour clusters and shape clusters of images as the features set Then, the similarity of colour cluster and shape clusters will be measured between query image and database. The best match of images will be the pair with the highest similarity results.

Another interesting logo detection method using machine learning has been proposed by Baratis et al. (2008). The focus of their works is to extract the most characteristic logo or trademark images from website. First, all images including logo and non-logo of the website will be converted into grayscale images. After that, the intensity histogram, radial histogram

and angle histogram will be obtained from all the converted grayscale images. Seven features will be extracted from the respective three histogram of an image, and form into a features set of 23 dimensions. Based on these features, decision tree classifier will be trained to distinguish between logo and non-logo images. When a query image is presented, the trained decision tree will classify it as a logo or non-logo.

Some of the most recent works on logo detection are based on deep learning. For example, Iandola et al. (2015) proposed a method that uses modified GoogLeNet and AlexNet neural networks to detect logo. Their results are good, having a 74% of precision, but it has difficulties in detecting small logos and also the logos that have many different modifications. Recently, Bianco et al. (2017) has proposed a new logo detection method that utilises region proposals and Convolutional Neural Network. The proposed method achieves good results of 97% precision.

2.6 Machine Learning

Anti-phishing researchers always facing challenges in decision making when based on multiple values from features extraction. Hence, when dealing with features analytics, machine learning has become an approach used to create models for decision making. Since phishing detection research only deal with binary classification scenario (e.g. phishing or non-phishing), this section will focus on the supervised machine learning models which are commonly used for classification in phishing detection works.

2.6.1 Decision Tree

Decision trees is one of a common classification method which classify instances by sorting them based on feature values in the form of a tree. It was constructed in a top-down recursive divide-and-conquer manner. Each node represents a feature in an instance to be classified, and by utilising "if-then" rule set, each branch will represent a value that the node can assume. The higher the attribute position on the tree, the more impact it brings to the classification. Decision trees has the characteristic of high comprehensibility and performs better when dealing with discrete or categorical features. One of the most well-known algorithms for building decision tree is the C4.5 (Quinlan, 1993).

2.6.2 Naıve Bayesian

Naıve Bayesian (NB) is one of the most common machine learning algorithms in solving the text categorization. This classification method will try to estimate the conditional probabilities of classes given an observation, based on the Bayes rule which assumes conditional independence between classes. Due to the conditional independence assumption, the joint probabilities of sample observations and classes are always been simplified. Although the Bayes rule assumption always violated in practice, it still showes good performance in many applications scenarios (Rish, 2001).

2.6.3 K-Nearest Neighbour

K-Nearest Neighbour (KNN) is one of a non-parametric classification algorithm in machine learning. It has shown great performance in various information retrieval problems. KNN

uses an integer parameter K and usually applies Euclidean distances as the distance metric if there are no prior information inputs. The performance of KNN is greatly depends on the choice of K, as well as the distance metric applied. However, choosing a suitable K value will become difficult if the points are not uniformly distributed. This method is considered simple, yet still can perform and give a competitive result when compared to other sophisticated machine learning methods.

2.6.4 Logistic Regression

Another machine learning technique that has greatly been implemented is the logistic regression. Logistic regression is used to predict a dependent dichotomous or binary variable based on one or more independent feature variables. The input for logistic regression can be any value in nominal, ordinal, interval or ratio-level, whereas the result is showed as a probability value strictly ranges from 0 to 1.

2.6.5 Support Vector Machine

Support Vector Machine (SVM) was first proposed by Vapnik in 1992. This method will find the optimal separating hyperplanes that classify data by maximising the geometric margin space between the classes' closest points. SVM has advantages in solving small sample, nonlinear and high dimensional feature space classification problems. This method receives great attention after achieved excellent performance in some of the real-world applications. (Hearst et al., 1998)

2.6.6 Neural Network

Neural Network (NN), or sometime referred as Artificial Neural Network (ANN) is a machine learning model which is stimulated by how the biological neural network processes information. It has the adaptability to change the structure, based on the information that flow through network during the learning phase. This method always helps in exploiting patterns that are too difficult to be observed by human (Witten & Frank, 2002). Normally it is used to solve classification problems where the output values are not directly related to its input (Muhammad et al., 2013).

2.6.7 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a useful algorithm for the dimensionality reduction and classification. It is able to find a linear transformation of the training samples after a set of search results has been fed in. LDA has excellent performance if the features of the dataset are linearly independent.

2.7 Evaluation Metrics

This section discusses several evaluation metrics, which are commonly uses to benchmark the performance between different types of phishing detection techniques. The same evaluation metrics has been demonstrated by Khonji et al. (2013) and Ramesh et al. (2014).

Phishing detection can actually be treated as a type of binary classification problem with two possible outcomes. Namely, positive label is used to indicate phishing sample and

negative label is used to indicate legitimate sample. The detection results in relation to the known actual values are visualised in a 2 × 2 confusion matrix as shown in Table 2.2.

Table 2.2: Confusion matrix for phishing classification

	Classified as Phishing	Classified as Legitimate
Phishing sample	True Positive (TP)	False Negative (FN)
Legitimate sample	False Positive (FP)	True Negative (TN)

If a phishing webpage is correctly classified as phishing, this result will be referred as true positive (TP). But if the phishing webpage is wrongly classified as legitimate, then it will be marked as false negative (FN). On the contrary, true positive (TN) refers to the actual legitimate webpage has been correctly classified as legitimate, and false positive (FP) refers to the legitimate webpage has been wrongly classified as phishing. In order to get a clearer and better understanding of the detection performance, we consider Accuracy (ACC) as an additional performance metric. It measures the percentage of test set tuples that are correctly classified.

Using data from the confusion matrix, the performance of phishing detection is usually measured using the following evaluation metrics:

- True positive rate (TPr) — percentage of correctly detected phishing instances over the actual total phishing instances. It is calculated as

$$TPr = \frac{TP}{FN + TP} \tag{2.1}$$

- False positive rate (FPr) — percentage of legitimate instances that are incorrectly detected as phishing over the actual total legitimate instances. It is calculated as

$$FPr = \frac{FP}{TN + FP} \tag{2.2}$$

- True negative rate (TNr) — percentage of correctly detected legitimate instances over the actual total legitimate instances. It is calculated as

$$TNr = \frac{TN}{TN + FP} \tag{2.3}$$

- False negative rate (FNr) — percentage of phishing instances that are incorrectly detected as legitimate over the actual total phishing instances. It is calculated as

$$FNr = \frac{FN}{FN + TP} \tag{2.4}$$

- Accuracy (ACC) — overall percentage of correctly detected phishing and legitimate instances over the total instances. It is calculated as

$$ACC = \frac{TN + TP}{TP + FP + TN + FN} \tag{2.5}$$

2.8 Summary

This chapter introduces several types of major phishing attack, followed by reviewing on the existing phishing detection techniques. This chapter also explains the working mechanism and possible shortages of phishing detection techniques from different categories. It is important to address that the list-based approach takes substantial amount of time to add in new phishing website, while heuristic based detection can be vulnerable if phisher manage to avoid the respective predefine list of features. Furthermore, most detection methods which

relied on TF-IDF are still incapable for detecting non-English phishing website. Lastly, the typical evaluation metrics which are used to examine phishing detection has been introduced. The extensive reviews in this chapter have provided the necessary foundation and inspiration for designing a better phishing detection technique. We will discuss the proposed system in detail in the next chapter.

CHAPTER 3

METHODOLOGY

3.1 Motivation

The purpose of the phishers is to deceive the users to believe that the phishing website that they are visiting is a legitimate website. Therefore, the phishers will create or even clone out a website that visually resembles the legitimate website. Most of the time, visual components such as the logo, emblem, or trademark from the legitimate website are reused by the phishers on their phishing website. This leads us to the following question: How to differentiate a phishing website and a legitimate website if they appear to be visually similar or even identical?

A number of conventional phishing detection techniques have attempted to address this question by leveraging an inherent characteristic that exist in every website, namely, the website identity (Zhang et al., 2007; Ramesh et al., 2014; Tan et al., 2016; Marchal et al., 2017). By examining the identity of a query website, it is possible to distinguish between phishing and legitimate websites.

The aforementioned anti-phishing techniques that are based on textual elements, although more popular, has some limitations. For example, using the TF-IDF to choose the correct terms is challenging. In other words, it is difficult to extract the correct words which can accurately represent the identity of the website. Phishers can easily avoid or jeopardise the detection mechanism by applying some unrelated but statistically significant terms to the content. Furthermore, identity detection based on textual information is sometimes vague and noisy.

To overcome the limitations of conventional text-based phishing detection techniques, a new phishing detection model based on visual elements is introduced.

Motivated by the knowledge that phishers tend to utilise the visual components (especially the logo) that are ripped off from the legitimate website, we propose a new anti-phishing method based on identification of the website identity, where our novelty lies in exploiting the website logo as the main feature. To the best of our knowledge, there is no existing phishing detection technique that utilises website logo to find the website identity. Figure 3.1 illustrates the concept of using the website logo to identify the portrayed identity of the query website.

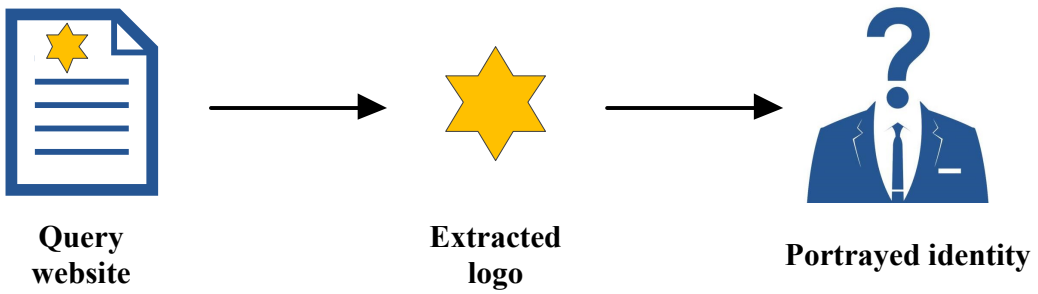


Figure 3.1: Leveraging website logo to identify the portrayed identity

Since the logo is equivalent to the visual identity of a legitimate website, it is valid and reasonable to utilise it as a clue in the proposed phishing detection technique. In addition, we believe that using graphical elements, especially the logo, is important to compensate for the limitations faced in textual-based methods, thus increasing the robustness of the phishing detection system.

3.2 Preliminary Experiment

This section presents our preliminary experiment using a basic model of the proposed phishing detection technique. The preliminary experiment is intended to validate the concept of utilising visual components (i.e., website logo) to pinpoint the legitimacy of a query website. The actual experimental results and the comparison with similar methods will be presented in Chapter 4. Figure 3.2 depicts the flow of the preliminary experiment.

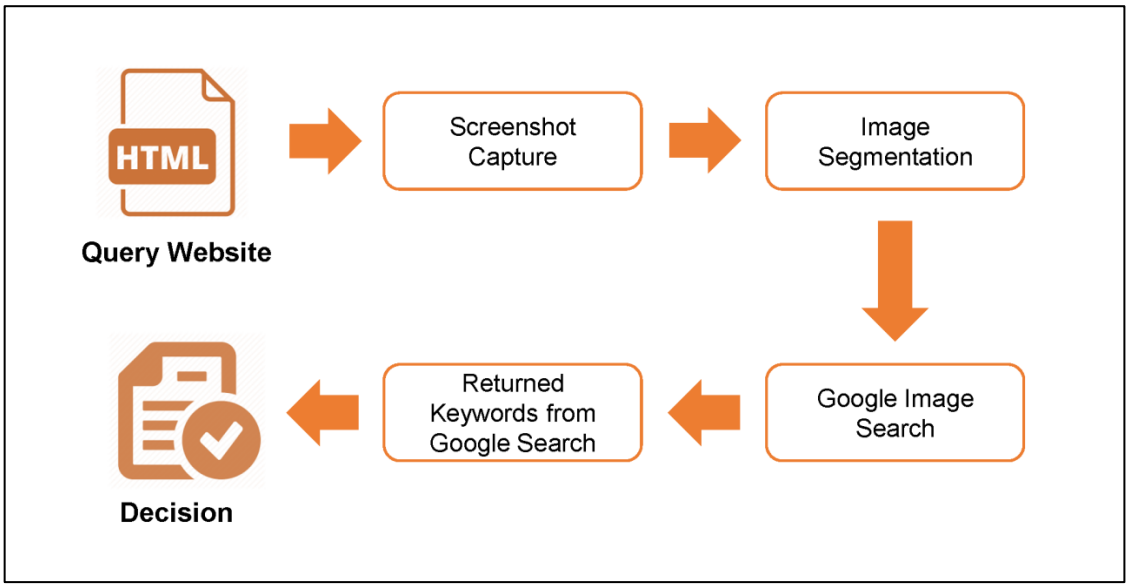


Figure 3.2: General flow of the preliminary experiment

The preliminary experiment involves two main processes: logo segmentation and website identity identification, which are the simplified version of the actual proposed method discussed subsequently in Section 3.3. To assess the effectiveness of our visual-based phishing detection model, the *fixed segmentation* technique is used for logo extraction in this preliminary experiment, as shown in Figure 3.3.

Based on our preliminary study which has been published (Chang et al., 2013), we found that 92% of the website logo is located around the top region of a webpage, mainly towards the top left side. Hence, only the top region of an image (the shaded regions shown in Figure 3.3) will be segmented from each screenshot.

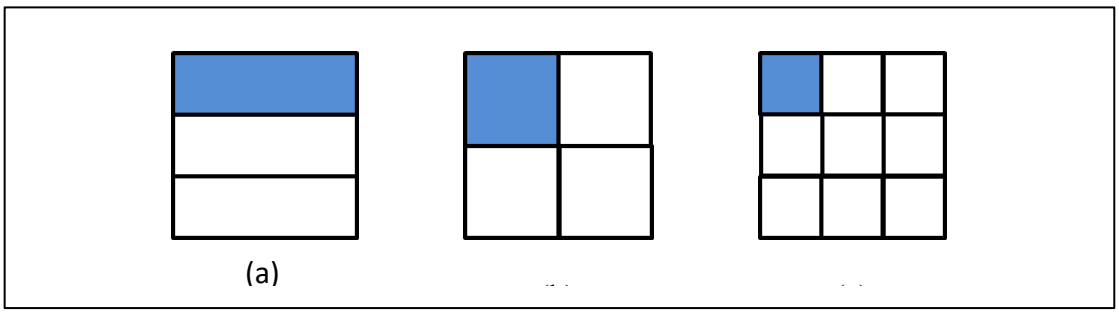


Figure 3.3: Fixed segmentation of logo extraction. (a) 1 × 3 segmentation. (b) 2 × 2 segmentation. (c) 3 × 3 segmentation

To evaluate the performance of fixed segmentation technique, we begin with the 1 × 3 segmentation mode on a screenshot image as shown in Figure 3.3(a). The segmented image will be uploaded to Google Image Search engine. After that, the returned keywords from Google Image Search results will be retrieved and used for the second search using the regular Google text search. We followed the recommendation by Zhang et al. (2007) to retrieve the top 30 search results for the second search. On top of this, it is also to fulfill the minimum statistically significance (Hogg et al., 2015). If the domain name of the query website does not match with any of the domain names returned from the top 30 results of the second search, the query website is considered as a phishing website. If no keywords are returned from the Google Image Search, it is considered as an unknown result.

In the remaining iterations, we attempt to achieve a tighter fit for the logo by progressively reducing the non-related area of the webpage screenshot, as shown in Figure 3.3(b) and Figure 3.3(c). The variation in the phishing detection rate will provide an

indication as to which segmentation mode performs the best.

Additionally, we have also tried the best fit logo segmentation by manually cropping the image as shown in Figure 3.4. The cropped best fit logo image will contain only the logo itself with minimum non-related area.

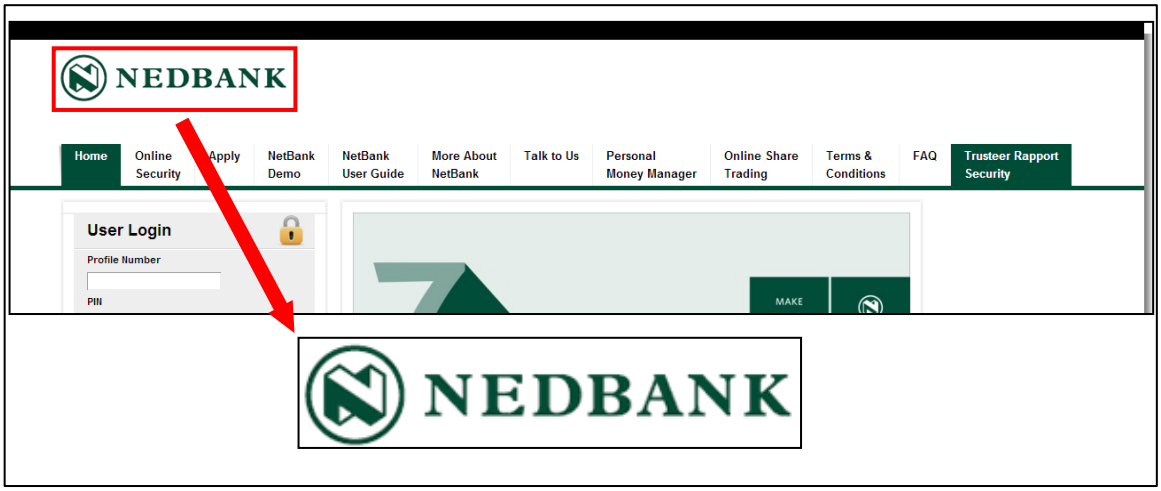


Figure 3.4: Best fit of logo extraction

The preliminary experiment results (available in Chapter 4) have verified the effectiveness of our basic phishing detection model that uses the website logo. Undoubtedly, placing website logo in upper regions of a webpage may indeed provide greater visibility to the visitors and strengthen the visual identity of the website. However, the preliminary experiment results also seemed to suggest that the location of the logo in a webpage is not consistent. In other words, the websites designers may not necessarily place the website logo on the upper regions of the webpage. Hence, there are some cases where the fixed segmentation modes are unable to locate any logo within the shaded regions shown in Figure 3.3. The fixed segmentation logo detection can be described as a "hit or miss" method, which is not flexible and robust against variations in position of the website logo. For example, when the website logo in placed at an uncommon area (i.e., in the middle region towards the right), the fixed segmentation technique will fail to extract any logo. It is crucial that the

logo detection technique is able to adapt and detect the website logo from various positions of the webpage. Therefore, an improved logo detection technique is introduced in Section 3.4 as part of our proposed phishing detection framework.

3.3 Proposed Framework

In this section, the framework and the mechanism of the proposed phishing detection technique are introduced. To facilitate discussion, the following definitions are used:

- **Query website:** The website which is under scrutiny.
- **Portrayed identity:** The brand or entity which predicated by the legitimate website. For example, a legitimate website with the domain name *https://www.paypal.com*, the portrayed identity is *PayPal*. Likewise, for a phishing website which impersonates the *PayPal* website with the domain name *http://www.it-initiative.org*, the portrayed identity is *PayPal*.
- **Real identity:** The actual identity of a query website. For example, *PayPal* is the real identity for the website with domain name *https://www.paypal.com*. Whereas for a phishing website with the domain name *http://www.it-initiative.org* which impersonates the *PayPal* website, its real identity is *it-initiative*.
- **Phishing target:** The legitimate website which is targeted by the phisher. In other words, it is the website which a phishing website is trying to impersonate.

Figure 3.5 and Figure 3.6 illustrates the portrayed identity and real identity for legitimate and phishing website, respectively. Figure 3.7 shows the mechanism of the proposed method. The proposed method can be divided into two main phases: (i) logo extraction, and; (ii) identity verification. A logo extraction process will extract the logo from

the query website. Based on the extracted logo, the identity verification process will evaluate the consistency between the real identity and the portrayed identity of the query website via the results from web search engine. If the identity is consistent, the query website is deemed legitimate, and vice versa. The decision logic is shown in Table 3.1.

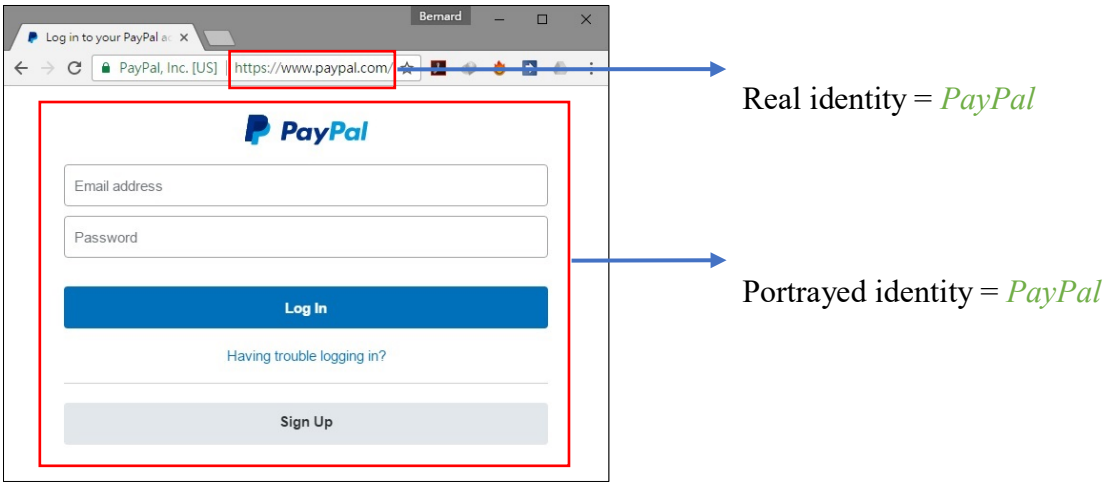


Figure 3.5: Portrayed identity and real identity in legitimate website (Paypal, 2018)

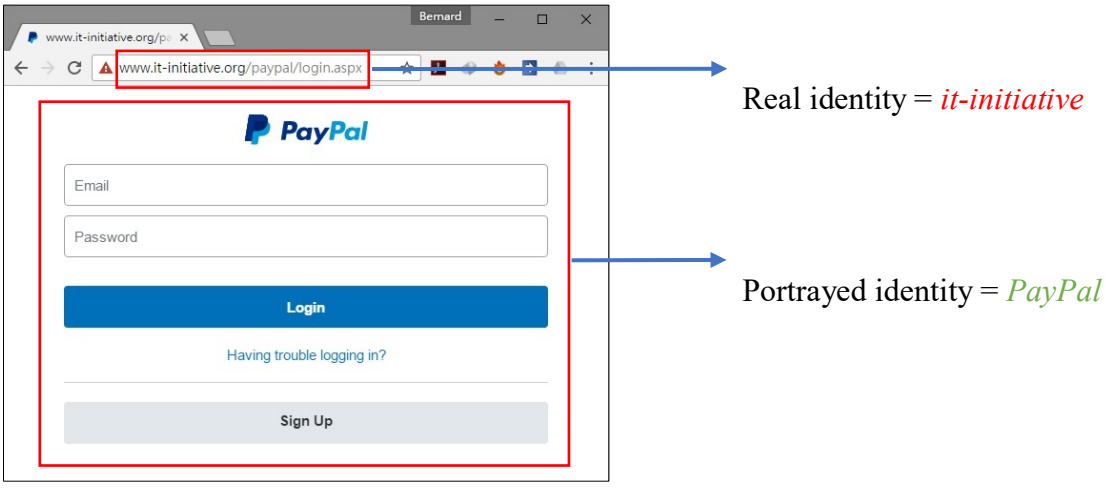


Figure 3.6: Portrayed identity and real identity in phishing website

Table 3.1: Decision logics of proposed method

Condition	Decision
Portrayed identity = Real identity	Legitimate
Portrayed identity ≠ Real identity	Phishing

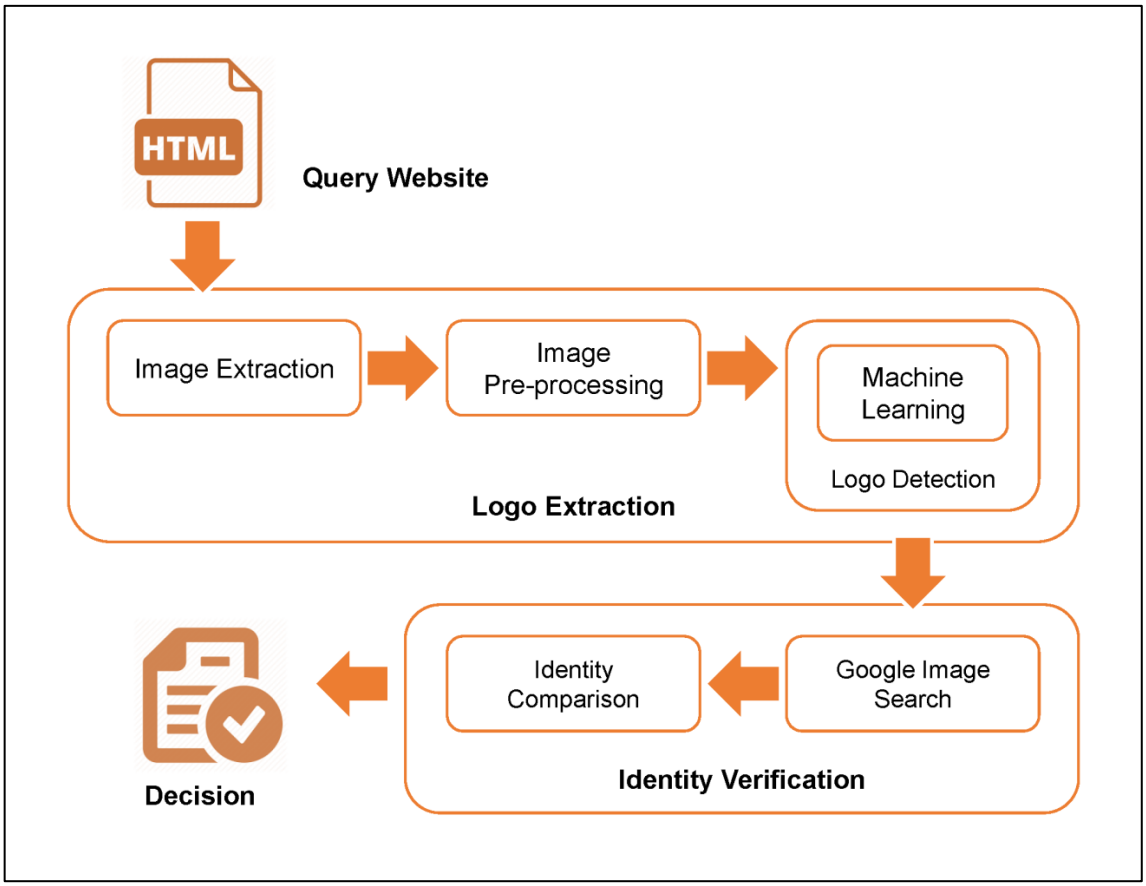


Figure 3.7: The mechanism of proposed method

3.4 Logo Extraction

The task of identifying and extracting the intended logo from a webpage is rather challenging. Therefore, we begin with a more realistic and modest way, by retrieving all the images from a webpage regardless of whether they are logo or not. Thus, the first step of the logo extraction process is to acquire all the images that are used in a query webpage.

3.4.1 Image Extraction

To download all the related images for further analysis, we utilised a MATLAB (Mathworks, 2015) script as a wrapper program to direct the download operations. The actual downloading is performed by the Wget (GNU, 2015) tool, an open-source software for downloading contents automatically from web servers. For each webpage, Wget will download the HTML file and its associated resources as listed in Table 3.2.

Table 3.2: Webpage elements and their corresponding file extensions

No	Element	File Extension
1	URL	.TXT
2	HTML files	.HTM/.HTML
3	CSS files	.CSS
4	Favicon	.ICO
5	Image resources	.BMP/.GIF/.JPG/.JPEG/.PNG
6	SVG files	.SVG
7	JavaScript files	.JS
8	Web font files	.EOT/.TTF
9	Screenshot of webpage	.BMP/.GIF/.JPG/.JPEG/.PNG
10	WHOIS information	.TXT
11	Other uncommon file types	-

After the webpages are downloaded, filtering is performed to remove multimedia files (e.g., MP3, MP4, AVI, WMV, WAV, MID, etc.), which can consume additional storage space. Furthermore, to the best of our knowledge, video and music files are not useful elements for phishing detection. In addition, some unknown files with over-long file names may get downloaded occasionally. These files cannot be opened, renamed, or even deleted using the regular Windows Explorer program. Hence, we used a third party file manager software to remove these unknown files.

Each downloaded webpage is stored in a folder, along with its associated resources, as shown by an example in Figure 3.8. Within each aforementioned folder, a MATLAB script is invoked to retrieve 5 types of image files by scanning for filenames with the extension .BMP, .GIF, .JPG, .JPEG, and .PNG.

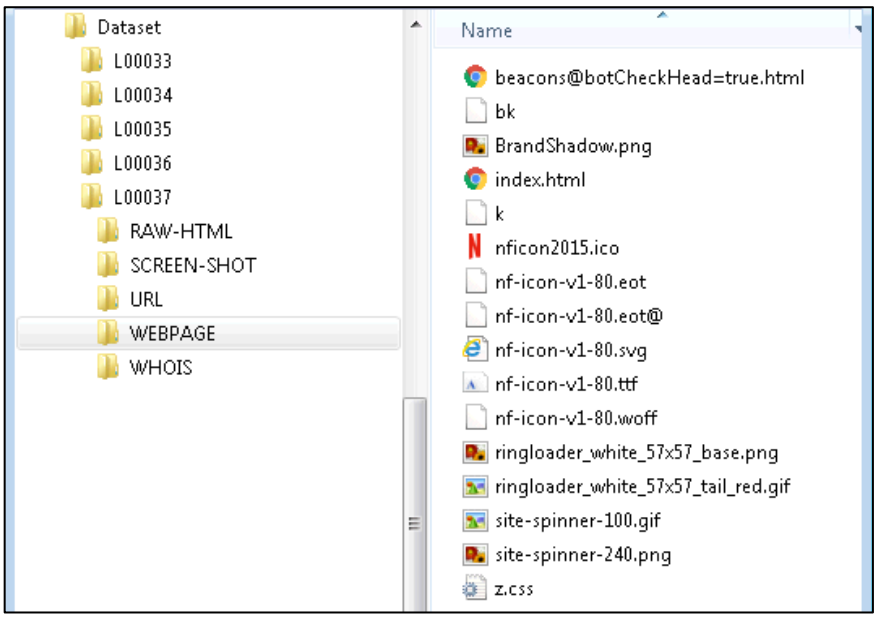


Figure 3.8: Example of folder contents for each downloaded webpage

3.4.2 Image Pre-processing

In this sub-process, pre-processing is applied to discard images which are unlikely to be logo. Namely, we exclude any image with a width or height which is less than 10 pixels, or an image which is monochrome (pixel intensity of one). Note that the main purpose of a website logo is to provide brand identification. Thus, images with a width or height of less than 10 pixels are considered too small and impractical for identification purposes. On the other hand, monochrome images contain insufficient colors, which is also unfavorable for brand identification. This filtering process is important to verify that the selected images have some basic characteristics of logo image.

3.4.3 Machine Learning-Based Logo Detection

This subsection introduces the features employed in the logo detection technique based on machine learning, followed by the selection of classification algorithm.

According to Baratis et al. (2008), a logo image has some common characteristics which differentiate it from a non-logo image. In their research, a machine learning technique is employed to classify the logo and non-logo images. In order to statistically represent the characteristics of the images, they defined a feature set of 23 dimensions (as discussed in section 2.5, the 23 features are come from the seven core features of each one of the three histograms, plus image file size and number of significant pixel intensity), and fed it into a decision tree for classification.

Motivated by Baratis et al.'s (2008) work, we employed a similar machine learning technique to find the correct logo image in our logo detection sub-process. However, we only apply the mentioned seven core features on intensity histogram, together with the

number of significant pixel intensity. We exclude the other two types of histograms in order to achieve a balance between performance and computational time, which is highly crucial for a real time phishing detection. It is worth to mention that, from our observation on Baratis's results, the features extracted from the radial histogram and angle histogram are less significant than intensity histogram, as their feature values between logo and non-logo images are almost the same, which contribute less in the machine learning training stage. It is because the more distinct the feature value between logo and non-logo images, the higher the accuracy in machine learning classification.

To supplement and enhance the existing feature vector set, two additional features are proposed. We conducted an empirical observation by randomly pick 100 logos from the dataset, and the results showed that the ratio of height to width for a logo image is most likely to be one. Differing from a logo image, the ratio for most non-logo images has a large deviation (i.e., significantly greater or lesser than one). Thus, the image resolution ratio is selected as the first additional feature. Besides, we also considered a common function in image processing called the Discrete Cosine Transform (DCT). It has the ability to pack input data into as few coefficients as possible, which is also known as energy compactness. Different image content will result in a different energy compactness level. Hence, energy compactness is chosen as the second additional feature. We added these two newly proposed features together with the 8 selected features from Baratis et al.'s (2008) work to yield a final feature set.

Most of the proposed features are derived from the pixels intensity distribution. Given an image, we first compute the pixel intensity histogram $H = [h_0, h_1, h_2, \ldots, h_{255}]$, where h_i indicates the histogram bin at *i*-th pixel intensity. After that, we normalise the histogram

to get the pixels intensity distribution as $h'_i = \frac{h_i}{\sum_{j=0}^{255} h_j}$. The proposed feature set is listed as follows:

- **Mean:** Mean of pixels intensity distribution, $\frac{\sum_{i=0}^{255} h'_i}{256}$.
- **Standard deviation:** Standard deviation of pixels intensity distribution, $\sqrt{\sum_{i=0}^{255} (i-\mu)^2 h'_i}$, where μ is the mean of pixels intensity distribution.
- **Skewness:** Skewness of pixels intensity distribution, $\sum_{i=0}^{255} \left(\frac{i-\mu}{\sigma} \right)^3 h'_i$, where σ is the standard deviation of pixels intensity distribution.
- **Kurtosis:** Kurtosis of pixels intensity distribution, $\sum_{i=0}^{255} \left(\frac{i-\mu}{\sigma} \right)^4 h'_i - 3$.
- **Energy:** Energy measures the homogeneousness of image content, $\sum_{i=0}^{255} (h'_i)^2$.
- **Significant pixel intensity:** The number of significant pixel intensity. We count the number of bins of pixels intensity distribution with a value greater than 0.02 (0.02 is determined empirically). Namely, the number of bin $h'_i > 0.02$ for $i \in [0, \ldots, 255]$.
- **Entropy:** Entropy measures the average bits per pixel, $\sum_{i=0}^{255} h'_i \log_2 h'_i$.
- **Otsu threshold:** A threshold that separates pixels into dark and light regions (Otsu, 1979).
- **Image resolution ratio:** Image height and width ratio.
- **Energy compactness:** Discrete cosine transform the image and compute the entropy of the DCT coefficients, $\sum h_{dct} \log_2 h_{dct}$, where h_{dct} is the histogram of DCT coefficients with 256 bins.

Note that a webpage may contain different types of images such as logo, scenery, clipart (e.g., button and icon), portrait, etc. It is difficult to consistently and accurately identify a logo from these images because the characteristics of some non-logo images are very similar to the logo. For instance, some clipart images highly resemble a logo. We acknowledge that the current implementation of the logo extraction process may sometimes return more than one image, which includes logo and non-logo images (i.e., clipart). However, this limitation is not critical as these non-logo images are usually obtained from the legitimate website and carries only a minimal level of portrayed identity information. Thus, these non-logo images can be filtered out in the next sub-process. The behavior of returning more than one logo image may prove to be useful sometimes, especially in detecting phishing websites that targets multiple legitimate websites. This issue will be further discussed in Chapter 4.

In the next stage, the proposed method will perform classification based on the extracted feature sets. As discussed in Section 2.6, many classification algorithms are available. Since the focus of this study is not finding the best optimized classification algorithm, we deliberately selected SVM as our classifier due to its simplicity, and promising performance demonstrated by other anti-phishing researches (Huh & Kim, 2011; Zouina & Outtaj, 2017; Jain & Gupta, 2018). We acknowledge that the use of SVM may be suboptimal; however, changing to an optimal classifier later is effortless. We used the SVM library that was implemented in Chang and Lin (2011) with the default setting (i.e., radial basis function is used as the kernel function, and the values for parameter γ and C is set to $\frac{1}{n}$ and 1.0, respectively, where n is the number of features used).

3.5 Identity Verification

Consistent identity means the real identity and the portrayed identity is identical. The real identity is obtained from the domain name of the query website, while the portrayed identity is retrieved from the entry of a logo database which matches to the extracted logo. Since the relationship between the logo and domain name of a website is exclusive, any mismatch is a sign of phishing attack. As such, a complete and up-to-date database that contains different website logos with the corresponding domain names is necessary. In practice, however, it is not feasible to maintain this database by ourselves. Hence, we utilise Google Images database as our source of logos database.

3.5.1 Google Image Search

In this module, the selected logo image will be fed into the image search engine to find the corresponding portrayed identity. To fully utilise the Google Images database, we employed the content-based image retrieval feature called Google Search by Image (GSI), which allowed us to retrieve the portrayed identity of a query website from the vast image database. This step is described as the Google Image Search sub-process in Figure 3.7.

Beside GSI, there are few others image search engine with their corresponding image databases. We have made comparison between the top five image search engine suggested by Berify (2018), which are GSI (Google, 2018), Bing Image (Bing, 2018), Tineye (2018), Pixsy (2018), and Berify (2018). From our observation, we found that GSI is the most accurate image search engine with larger image database. There are two major reasons

differentiate GSI from others:

(i) The returned results of GSI has higher accuracy and more informative compared to Bing Image and Tineye. We fed Public Bank and Paypal logo into these three image search engines, and only GSI is able to acquire the correct targeted URL in the first ten search results.

(ii) Berify and Pixsy require user to go through registration and subscription of service to use the search engine and database, while GSI is free with the required API for integration (Google, 2018). This is important for the final actual tool implementation.

The returned result from the GSI consists of three elements, namely: (i) the best guess for the query image; (ii) a list of visually similar images, and; (iii) pages that include matching images. Unlike GSI, Bing Image and Tineye returned result is only consisted a list of visually similar image with the corresponding URL. Figure 3.9 shows an example of the returned results from GSI using PayPal logo image as the query input. Our current setting is to use only the first page of the search result.

As shown in Figure 3.10, there are two options of using the GSI service, namely: (i) paste image URL, where this option allows the user to directly use the image's URL found on the Internet, and; (ii) upload an image, where this option allows the user to upload images from the local drives of computers. In our implementation, we feed the logo images to GSI using the first option with a simple workaround described as follows:

i) First, the selected logo images (obtained from Subsection 3.4.3) will be uploaded to a hosting web server.

ii) We employed a program called "Simple Soft" (shown in Figure 3.11) which utilises the GSI application programming interface (API) to run the image search query using the first option. We obtained the API from Google Developer (2018), while the program is contributed by Choo (2014).

iii) The URLs of all uploaded images from (i) (selected logo images) will be fed into that software. The software will return the keywords (from the first element) and related URLs (from the first element and third element).

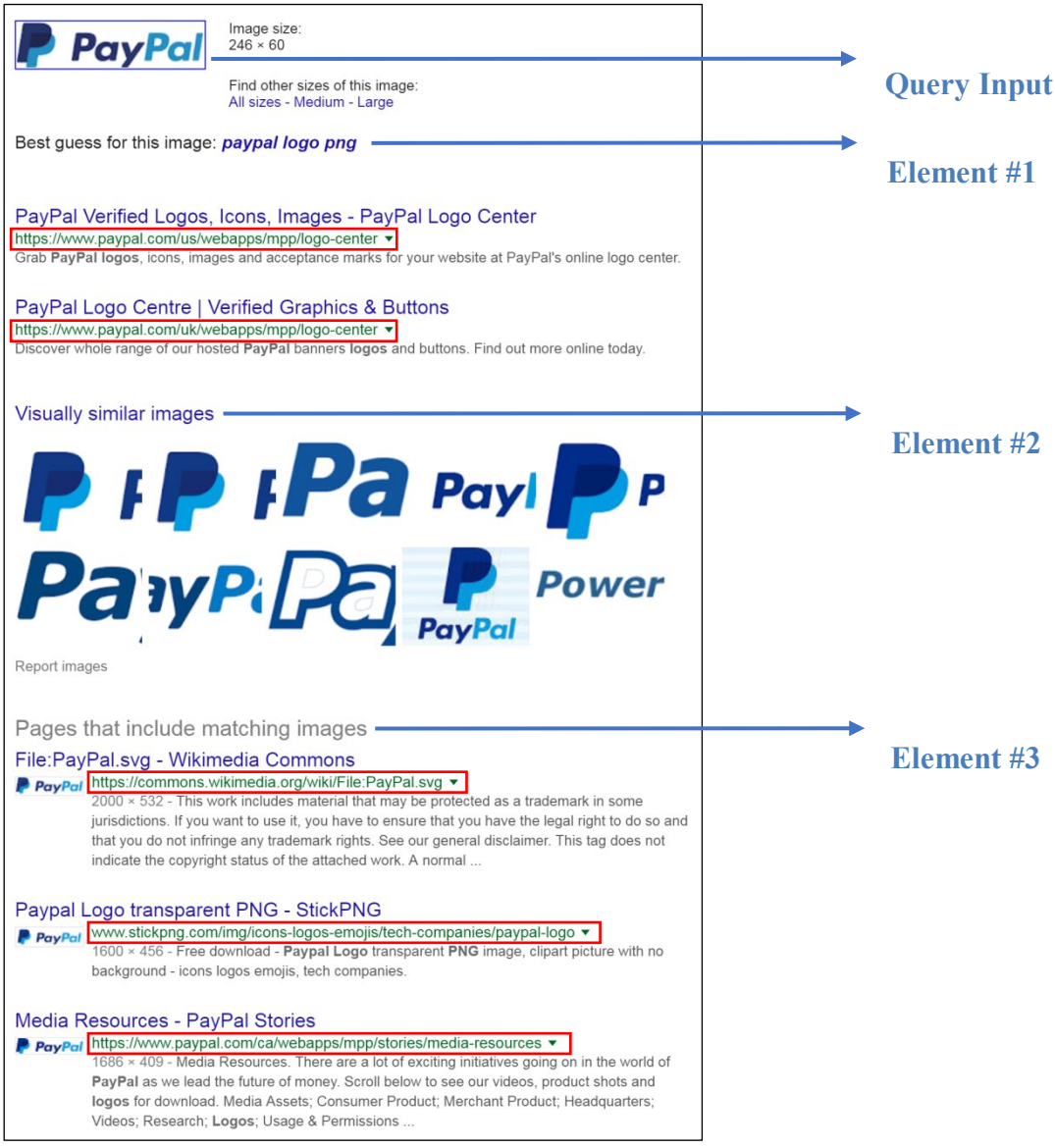


Figure 3.9: Search results returned by GSI using PayPal logo image as the query input

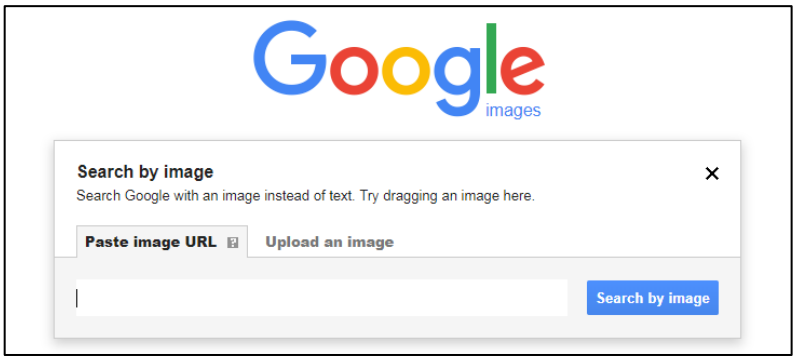


Figure 3.10: Example of a GSI interface (Google, 2018)

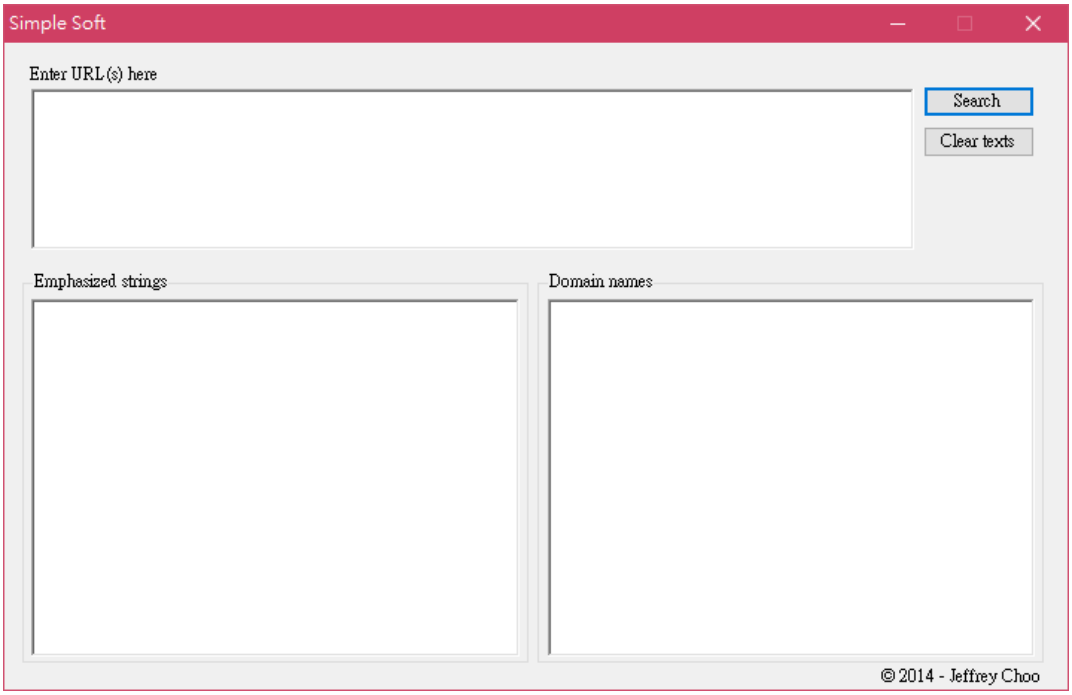


Figure 3.11: Interface of the Simple Soft program

3.5.2 Identity Comparison

From Subsection 3.5.1, all possible portrayed identities of the query webpage are retrieved from GSI result. In this identity comparison sub-process, the real identity of the query webpage is cross-checked with the entries of GSI results (i.e., possible portrayed identities). We begin by parsing the URLs (as marked with the red outlines shown in Figure 3.9) from the first and third elements of the search result.

Most anti-phishing techniques perform website identity comparison using domain names. However, comparing domain names alone may introduce false positives for highly established brands that have multiple country-specific websites. Table 3.3 shows an example for the eBay website, where it can be accessible through different URLs such as *https://www.ebay.com*, *https://www.ebay.com.my*, or *https://www.ebay.ph*. All of these websites belong to eBay. The only difference between them is the Top-level Domain (TLD) portion *.com.my* and *.ph* for the Malaysian and Philippines version of the eBay website, respectively. This identity mismatch issue usually happens on legitimate query webpages with a less popular TLD, where its matching domain name may only appear in the second or third page of the GSI results.

Table 3.3: Example of TLD mismatch issue for the eBay websites

Case	Portrayed identity	Real identity	Domain name matched	Detection result
1	ebay.com	ebay.com.my	No	Phishing
2	ebay.com	ebay.ph	No	Phishing

To cater for this TLD mismatch issue, the website identities can be compared using only the Second-level Domain (SLD). SLD is a specific URL substring that appears in front of the TLD, as shown in Figure 3.12. For example, the SLD for *https://www.mydomain.com* is *mydomain*. Table 3.4 shows the reduced false positives detection result using SLD comparison. Based on the SLD comparison result, case-1 and case-2 can now be correctly labelled as legitimate.

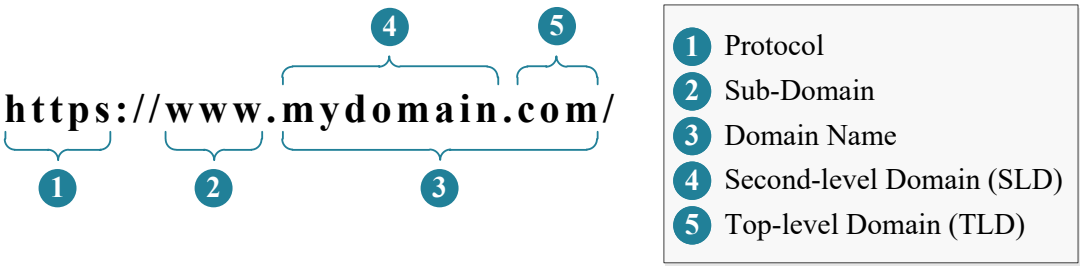


Figure 3.12: Structure of a URL

Table 3.4: Reduced false positives using SLD comparison

Case	Portrayed identity	Real identity	SLD matched	Detection result
1	ebay.com	ebay.com.my	Yes	Legitimate
2	ebay.com	ebay.ph	Yes	Legitimate

Hence, in the identity comparison sub-process, we extract only the SLD from each of the parsed URLs from Google Image Search results and compare them to the SLD of the query webpage. If the comparisons return at least one match, the proposed method will classify the query website as legitimate. Otherwise, it is classified as a phishing website.

As for the limitation in the logo detection sub-process mentioned in Subsection 3.4.3 (i.e., when multiple images of logo and non-logo are returned), the identity verification process is repeated for each image, and the comparison results are aggregated. Similarly, the proposed method will classify the query website as legitimate if the aggregated comparisons return at least one match.

3.6 Summary

This chapter describes the proposed phishing detection technique in detail. The initial phase of the proposed system focuses on identifying and extracting the logo from a query webpage. After that, during the second phase, the focus moved to verifying the real identity of a query webpage, by utilising the search results of Google Search Image that are retrieved using the logo obtained from first phase. The next chapter will discuss the details on experiment setup, benchmarking results and analysis.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Introduction

This chapter presents the evaluation results of the preliminary experiment, followed by the experimental results of benchmarking the proposed method against another related anti-phishing method. Analysis and findings are also reported and discussed. The dataset collection process and the prototype implementation are thoroughly explained.

4.2 Preliminary Experimental Results and Analysis

This preliminary experiment, as introduced in Chapter 3, is designed to verify the effectiveness of our basic phishing detection model that uses the website logo. For this initial experiment, 400 phishing websites and 50 legitimate websites are collected based on URLs from PhishTank (2017) and Alexa (2017), respectively. Screenshot of each website is captured by using a Google Chrome plugins function, and saved as JPG image file.

Four datasets are created from the screenshot images using the fixed segmentation and the best fit extraction as shown previously in Figure 3.3 and Figure 3.4, respectively. Namely, the first three datasets are the segmented images based on the fixed segmentation of 1 × 3, 2 × 2 and 3 × 3. The fourth dataset consists of the best fit logo images (i.e., image containing only the logo itself with minimum non-related area). The purpose of the fourth dataset is to serve as the ideal case condition (i.e., most accurate segmentation), which may lead to the best phishing detection performance. Table 4.1 shows the detection results for the four datasets. We abbreviate the dataset as Dataset 1, Dataset 2, Dataset 3 and Dataset BF for the

first three datasets and the best fit logo image dataset, respectively. On the other hand, evaluation metrics such as true positive, true negative, false positive and false negative are abbreviated as TP, TN, FP and FN, respectively.

Table 4.1: Evaluation results of preliminary experiment

Experiment	TP		FN		FP		TN	
Dataset 1	156	(39%)	244	(61%)	21	(42%)	29	(58%)
Dataset 2	300	(75%)	100	(25%)	22	(44%)	28	(56%)
Dataset 3	348	(87%)	52	(13%)	15	(30%)	35	(70%)
Dataset BF	370	(92.5%)	30	(7.5%)	0	(0%)	50	(100%)

The results in Table 4.1 suggest that when the non-related area on the segmented image is reduced, the detection performance increased. In particular, the TP rate has increased from 39% to 87%, while the TN rate has also increased from 58% to 70%. This is because the ratio of logo size that occupies the segmented image has become larger, which also means the non-related area now occupies a smaller region. Hence, a tight fit of the logo image is important to obtain an accurate result from Google Image Search. This is verified by the highest detection results obtained for Dataset BF, where the segmented image used to query Google Image Search is wholly based on the logo itself.

In summary, the preliminary results have proven that the proposed phishing detection model using logo is a feasible and effective technique. However, the effectiveness of the proposed method is highly dependent on the right input (i.e., the segmented image which consists of a logo) being fed to the Google Image Search. In our empirical observations, we found that most of the logo is located on the top left region of a webpage. Nevertheless, there are exceptions. For example, if a webpage has a logo located on the top right side while our method uses the fixed segmentation of 3 × 3, the segmented image will not contain the logo.

As a result, we will query Google Image Search with the wrong information, thus leading to a rise in false positive detections. In view of the limitations faced by the fixed segmentation technique, an improved logo detection method has been introduced in Section 3.3, where the evaluation results are given in Section 4.5.

4.3 Dataset Description

This section describes the processes in constructing a reliable dataset, which includes the selection of dataset sources, implementation of the webpage crawler and allocation of the samples to different datasets.

4.3.1 Influencing Factors on Dataset Design

Normally, a phishing website will be taken down and become inaccessible after a short period of time. The unavailability and inaccessibility of the phishing websites have often become a bottleneck for the research community in conducting experiments. In addition, the experimental results cannot be reproduced and validated without the original phishing websites, thus making it less reliable. Hence, as a practical solution, we have chosen to construct our offline dataset for experiment purposes.

Many anti-phishing researchers have created offline datasets to be used in their own experiment, but most of them were not publicly shared or downloadable. The downloadable sources for offline anti-phishing dataset are rather limited, leading to the increase of time and cost from having to create a new dataset every time. Although several downloadable offline anti-phishing datasets are available, they are not compatible and practical to be used

for our proposed method. For example, Mohammad et al. (2015b) have provided a downloadable phishing website dataset. However, this dataset contains only processed data, namely the feature values which have undergone transformation and the corresponding class labels. In other words, it is a complete processed dataset which is limited for machine learning purposes. It is not suitable to be used for other types of anti-phishing techniques. To ensure that a dataset can be widely adopted, it must be kept in the form of raw data (i.e., the actual webpage and its related resources).

Some researchers are only using the top ranking legitimate websites for their experiments (e.g., the dataset from Cao et al. (2008), Liu et al. (2010), Dunlop et al. (2010), Choo et al. (2014), Hara et al. (2009)). This shows that the less popular legitimate websites have been marginalised and overlooked. Consequently, their experimental results are biased, as it is generally easier to classify top ranked legitimates websites due to the abundant features available in them.

Some datasets may contain a high volume of specific brands which could also cause bias to the experimental result. Such datasets will lower the credibility of the experimental results by drawing advantages to certain methods. For example, if a certain targeted brand covers a major distribution in the dataset, it will eventually reduce the difficulty and technical challenges in the detection process, and hence increase the accuracy rate.

4.3.2 Choosing the Data Source

The dataset consists of phishing and legitimate webpages. Phishing webpages are downloaded based on URLs from PhishTank (2017), whereas legitimate webpages are obtained based on URLs from Alexa (2017). As mentioned in Subsection 4.3.1, low popularity legitimate websites are as important as popular legitimate websites. Therefore,

we have included DMOZ (2017) and BOTW (2017) as the source for low popularity legitimate websites.

Undoubtedly, PhishTank by far is the most complete phishing websites repository and has become our preferred source for the phishing dataset. It is a free community website where anyone can submit, verify, track and share phishing data. Differing from OpenPhish and Millersmiles, PhishTank does not require a fee to access the full database. In addition, PhishTank contains more phishing sources as other phishing repositories are also contributing to the PhishTank database. For example, the Clean-Mx is one of the phish reporters for PhishTank.

Alexa is a commercial website which provides traffic data, global rankings and other information on millions of websites. Several useful functions are available in Alexa, which make the dataset construction process efficient. These functions include the downloadable one million top URLs list in CSV format with well-organised content that makes filtering by region, category and ranking easier. The legitimate webpages from Alexa consist of diverse categories such as banking, social networking, news, e-commerce, forums and blogging. In short, most legitimate websites that Internet users tend to visit are already covered here.

Similar to Alexa, DMOZ is a multilingual open-content directory of World Wide Web links, while BOTW is a commercial web directory that provides websites in different topical and regional categorisation. During the data crawling process, we notice that BOTW and DMOZ were indeed able to provide the URLs which were not included in the Alexa top one million websites.

4.3.3 Implementation of the Webpage Crawler Program

We implemented the automated webpage wrapper program using MATLAB R2015b, and executed it on a Windows 10 computer with Intel E3-1230v3 processor and 8GB RAM. The program will read the URLs input from a text file and invoke Wget to download the webpages automatically. Each webpage sample consists of the HTML document and the related resources (e.g., images, CSS, JavaScript) needed for proper webpage display purpose. Besides that, the wrapper also utilised WebShot (2013) to take screenshots of webpages and saved them as a full-sized images or thumbnails.

4.3.4 Constructing the Datasets

For the actual experiment, two non-overlapping datasets are constructed from a total of 1140 webpages downloaded based on URLs from PhishTank and Alexa. A total of 140 webpages were allocated for Dataset 1, which consists of a total of 3894 images (i.e., 1947 logo images and 1947 non-logo images). Dataset 1 is used for the logo extraction process as discussed previously in Section 3.4. To ensure that the proposed method is able to detect all variations of logo images, we use the logo and non-logo images from both phishing and legitimate websites image resources. This is important, as sometimes the phisher will try to evade the detection by slightly altering the logo image (e.g., scaling down a logo image).

The remaining 1000 webpages (i.e., 500 phishing and 500 legitimate webpages) were assigned to Dataset 2. For Dataset 2, it is used for the identity verification process as discussed in Section 3.5. The final detection performance of the whole proposed method is measured using Dataset 2.

Additionally, we also constructed Dataset 3, which solely consists of 500 other less popular legitimate webpages obtained from DMOZ and BOTW. The purpose of Dataset 3 is to evaluate the effectiveness of the proposed method in certain challenging situations, namely the less popular legitimate webpages. They are ranked between 1001 to 5150 in Alexa and 16.4% of them (82 webpages) are not ranked in the Alexa top one million websites. These webpages usually have one or more of the following characteristics: (a) Newly launched; (b) low in quality or not compliant to W3C; (c) not optimised for Search Engine Optimisation (SEO), and; (d) excessive use of frames. Note that we do not include additional phishing webpages to Dataset 3 because the lifespan for phishing webpages is usually short. In addition, phishing webpages do not have popular and unpopular categories. Hence, it does not pose any additional complications to the proposed detection mechanism. Table 4.2 provides an overview to these three datasets.

Table 4.2: Description of Datasets

Dataset	Number of phishing pages	Phishing Source	Number of legitimate pages	Legitimate Source
Dataset 1	70	PhishTank	70	Alexa
Dataset 2	500	PhishTank	500	Alexa
Dataset 3	0		500	DMOZ and BOTW

4.4 Experiment Setup

The experiment is designed to evaluate the effectiveness of the proposed method, as well as comparing its performance with another anti-phishing method — GoldPhish (Dunlop et al., 2010). This is important to ensure that the results are comparable and insightful. We selected

GoldPhish as the comparison candidate due to the similarity of both methods, where the main idea is to leverage the Google search engine to determine the identity of a webpage via the extracted contents.

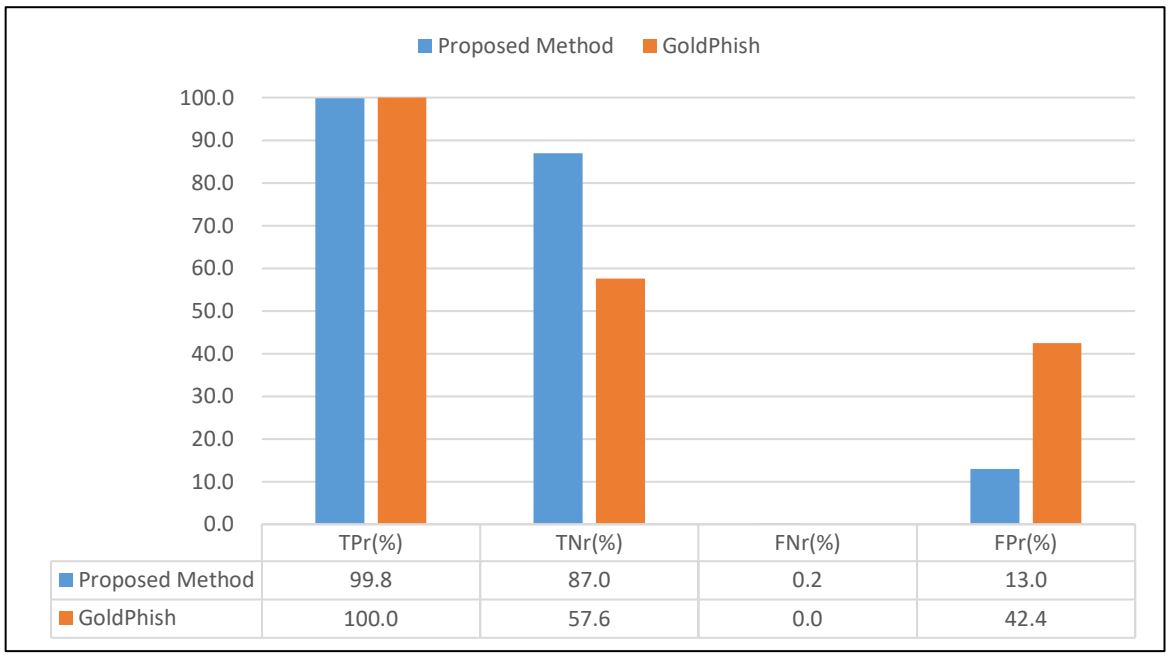
GoldPhish employed a different approach for feature extraction, focusing on extracting the textual contents. GoldPhish starts by capturing a screenshot of the webpage, and performs OCR to extract the textual contents (including the text within a logo). Then the authors will feed the extracted text to the Google search engine and evaluate the search results. Differing from our proposed method, GoldPhish depended solely on Google textual search. The domain name of each entry in the search result is cross-checked with the domain name of the query website. A mismatch means the portrayed identity is different from the real identity, therefore concluding the query website as phishing.

We implemented GoldPhish in C# programming language and evaluated our proposed method with GoldPhish by using the same datasets. It is worth to mention that the same datasets are used in the evaluation for both anti-phishing methods to ensure fairness in the benchmarking results. All experiments were conducted on a desktop computer equipped with an Intel Xeon E3-1230v3 3.3GHz CPU, 8GB RAM and Windows 10 Professional 64-bit operating system. The software required for implementing the experiments are listed in Appendix A.

4.5 Performance Results

The results for the experiment are shown in Figure 4.1. Based on Figure 4.1, both methods achieve comparable true positive detection rate, with our proposed method having a slightly lower true positive rate at 99.8% as compared to GoldPhish which scored 100% true positive rate. For the true negative detections, GoldPhish and our proposed method scored the rate of

57.6% and 87.0%, respectively. The results show no significant difference between our proposed method and GoldPhish in true positive detection. However, our proposed method has a notable leading of 29.4% higher than GoldPhish in terms of true negative detection. In order to get a clearer and better understanding of the overall detection performance, we considered Accuracy as an additional performance metric. The proposed method achieved an Accuracy of 93.4%, outperforming GoldPhish which only achieved 78.8%. In summary, the proposed method achieves significant improvement in overall accuracy when compared to the existing anti-phishing method considered.



	TPr(%)	TNr(%)	FNr(%)	FPr(%)
Proposed Method	99.8	87.0	0.2	13.0
GoldPhish	100.0	57.6	0.0	42.4

Figure 4.1: Comparison of performance between the proposed method and GoldPhish

From the experiment carried out using Dataset 3, the result (as shown in Figure 4.2) indicates a performance drop for both the proposed method and GoldPhish. This is not surprising, as Dataset 3 is purposely constructed with some uncommon characteristics (more details are discussed in the next section). Comparatively, the proposed method only dropped 7.6% in true negative rate, whereas GoldPhish dropped 8.2% in true negative rate. The

proposed method achieves 79.4% of a true negative rate and 20.6% of a false positive rate. These rates are still considered acceptable, given the nature of the dataset, while GoldPhish only manage to achieve 49.4% and 50.6% for the true negative and false positive rate, respectively. This result suggests that the proposed method is robust and capable of maintaining a good performance even on challenging datasets.

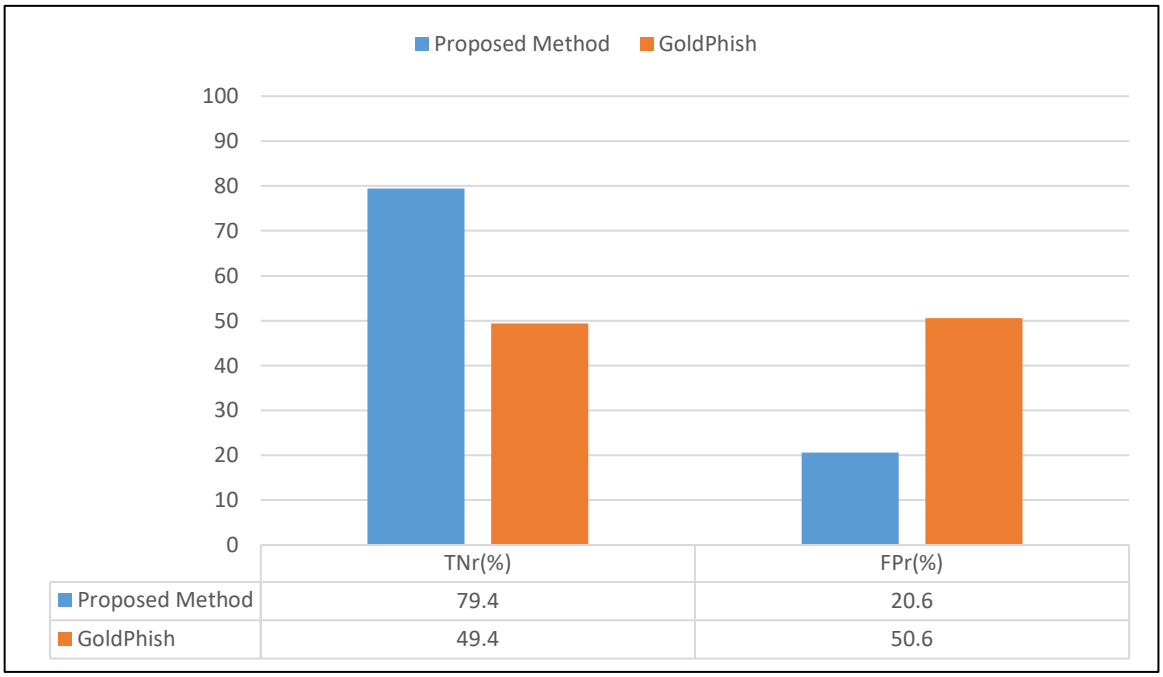


Figure 4.2: Results comparison using Dataset 3

4.6 Results Analysis

From Figure 4.1, we can clearly see that the proposed method showed promising results. This justified the effectiveness and feasibility of our proposed method to detect phishing websites using logo images. This advantage can be credited to the language independent characteristic of the proposed method. Although GoldPhish, an image content-based phishing detection method, performed very well in detecting the phishing websites (scored 100.0% of true positive rate) in this experiment, it has incorrectly labelled nearly half of the

legitimate websites as phishing. The low true negative detection is likely due to the fact that GoldPhish relies on the extracted textual contents. Thus, its performance is largely dependent on the accuracy of the OCR extracted texts. Due to the English-based OCR tool used in GoldPhish, its detection ability is only limited to websites with English content (Dunlop et al., 2010). Therefore, it may fail to extract significant texts from our dataset which consists of multiple language websites, such as Arabic, Chinese, and others. On the contrary, our proposed method does not suffer this limitation, due to the fact that we are using the logo image as the search query. However, the effectiveness of our proposed method will be affected by the input being fed to the Google Image Search sub-process. This limitation will be discussed further in Section 4.7.

In summary, our proposed method is proven to be an effective anti-phishing system, with an outstanding true positive rate of up to 99.8%. This type of key advantage can be considered as very desirable in anti-phishing applications. Overall, our proposed method delivered a comparable overall accuracy with the conventional anti-phishing method.

4.7 Limitations

In this section, some limitations of our proposed method and the possible countermeasure steps are discussed. There exist two complications:

i) The logo extraction process fails to extract and return the right logo image (note that the logo image actually exists).

ii) The webpage does not contain any logo image. These complications are the reasons for the relatively lower true negative rate (TNr = 87.0%).

Further analysis reveals that 56 out of the 500 legitimate webpage samples (these are the false positive samples) belonged to the first complication.

4.7.1 Complication in extracting the right logo

From the experiment, we observed that the first complication contributes to two situations. The first situation happened when our proposed method mistakenly treated a non-logo image as the logo, while removing the actual logo image. In the second situation, all images (logo and non-logo) are completely removed. In other words, these two situations leave no logo image to be used in the Google Image Search sub-process. From the 56 webpage samples, half of the samples belonged to the first situation and the other half belonged to the second situation.

To remedy some of these problems, we crop the top 200 pixels from the full-width screenshot of query webpage, as an attempt to obtain the logo within this region. From the experiments, we found that most legitimate websites have their logo located on the left or right side of the upper region. We acknowledge that this remedy is rather an ad hoc solution, and that the long term solution is to devise a better logo extraction approach. The proposed method will apply this remedy when the logo extraction process fails to extract any image. Hence, we only apply this remedy for the second situation (recall that in the second situation, the logo extraction process fails to extract any image). With this remedy, the proposed method managed to reduce the false positive from 56 to 42 webpage samples. On the other hand, there exists only one out of the 500 webpage samples that faces the second complication. This particular webpage is found to contain only textual content.

4.7.2 The challenge from low visual properties images

Besides these complications, 22 false positive samples came from the identity verification process. From our observation on most samples, we noticed that the logo

extraction process has indeed extracted the right logo image, but the Google Image Search sub-process returned the wrong results. Further analysis showed that the logo images of the 22 false positive samples had at least one of the following properties:

i) The image file has less foreground color and a transparent background.

ii) The logo has only a textual element (i.e., the brand name). For example, HAGERTY.

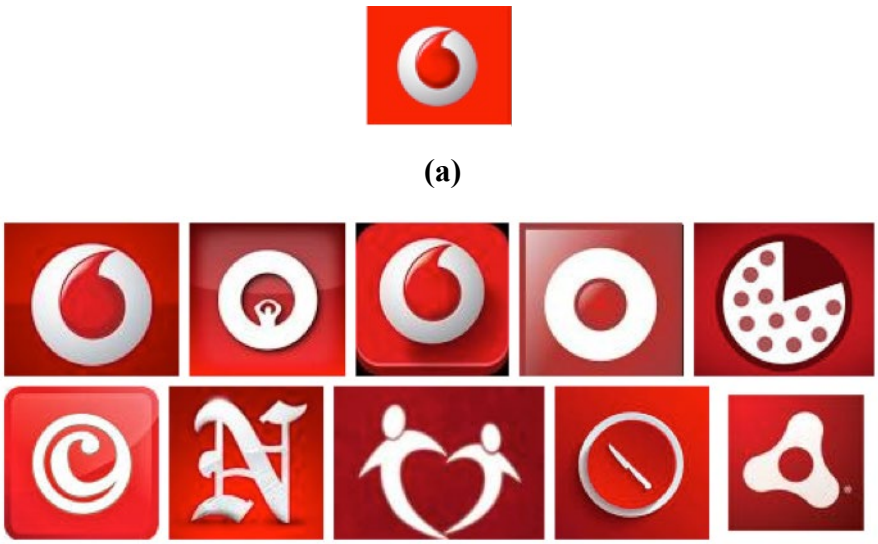
The lack of visual properties has caused the logo to become too general and similar to other images (i.e., cliparts). This scenario in turn has caused the Google Image Search to return unrelated results. In total, the proposed method wrongly classified 65 out of 500 legitimate webpages as phishing webpages (FPr = 13.0%). Namely, 43 and 22 false positive samples came from errors in logo extraction and identity verification processes, respectively. Besides the two properties of the logo image mentioned above, there are other factors which have confused our proposed method. They are:

i) The logo existed within a banner image.

ii) The logo existed within a sprite type of image, which is usually used to optimise webpage loading speed.

iii) The logo image is in a vector graphic file format (e.g., SVG format).

Besides, there are some logos which may highly resemble other logos. As illustrated in Figure 4.3, several returned images by Google Image Search are found to highly resemble the query images, but yet they are actually unrelated. This will cause an undesired detection result as well. Nevertheless, this is not surprising as even humans cannot effectively differentiate them.



(a)

(b)

Figure 4.3: Highly similar images. (a) Query image. (b) Similar images returned by Google image search

4.7.3 Impacts from limitation

Comparatively, the proposed method performs better in detecting phishing websites (TPr = 99.8%) as compared to legitimate websites (TNr = 87.0%). Although the above-mentioned complications and issues also happened in detecting phishing websites, the impact is less. The reason is because the proposed identity verification process is based on an identity consistency mechanism. Recall that in Section 3.5, consistent identity means that the real identity (i.e., SLD of the query webpage) is identical to the portrayed identity (i.e., SLD extracted from the Google search results). If the identity is found to be inconsistent, the proposed method will classify the webpage as a phishing webpage. Proving an inconsistent identity is easier than otherwise. Thus, the above mentioned complications and issues which have increased the false positive rate do not affect the detection of phishing webpage in the same way.

As for the experiment on Dataset 3, Table 4.2 shows the breakdown of the 103 false positive webpages (i.e., FPr = 20.6%). To simplify the discussion, we used the following abbreviations for different complications (note that the complications are those discussed above for Dataset 2):

C1: proposed logo extraction process fails to extract and return the right logo image;

C2: the original webpage does not contain any logo image;

C3: proposed logo extraction process had extracted the right logo image, but Google Image Search sub-process returned the wrong results.

Table 4.3: Comparison between Dataset 2 and Dataset 3 for each complication

Complication	Dataset 2 (No. of webpages)	Dataset 3 (No. of webpages)
C1	42	49
C2	1	23
C3	22	31
Total	65	103

Table 4.2 shows the comparison between Dataset 2 and Dataset 3 for each complication. Clearly, we notice that complication C2 has increased the most, while the increments for C1 and C3 are relatively marginal. These results reflect that the performance drop is significantly caused by the complication C2 (i.e., the original webpage which does not contain any logo image). While C1 and C3 also contributed a small portion to the performance drop, it is not critical. The results show that the proposed method can achieve consistent performance even for the less popular websites. This can be seen from the low increment for C1 and C3, where C1 increases from 42 to 49 webpages and C3 increases from 22 to 31 webpages.

Although C2 has a serious increment (increased from 1 to 23) at the first glance, it is expected and non-dangerous. This is because phishers are very unlikely to put their efforts

onto a less visited website, since it will be more lucrative to focus on popular websites. In addition, 17 out of the 23 webpages from C2 are webpages which are very unlikely to be phished. Note that these webpages have no login feature, non-financial websites, personal blogs or informative websites.

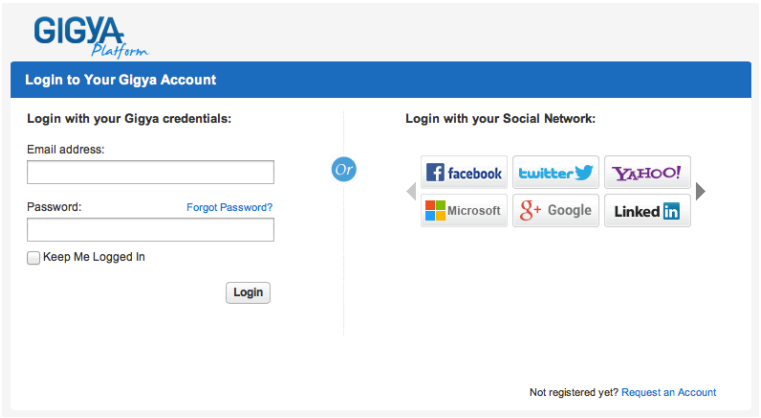
With such challenging webpage characteristics constructed for Dataset 3, the performance drop is inevitable. Hence, a drop of only 7.6% in the true negative rate is considered moderately good. Low quality logo design is usually found in the less popular websites, and it is the main reason for the complications C1 and C3. This type of logo is not unique, very small, and highly resembles a text. Besides, when the related information of a less popular website is being listed in other popular websites, the less popular website will be listed much lower in rank than the popular website in the search result. For example, when the proposed method is examining *http://www.arte.tv* website, the returned result is the Wikipedia page describing *http://www.arte.tv*. Obviously, Wikipedia is more popular and ranked higher than *http://www.arte.tv*, hence the returned result is from Wikipedia. However, we noticed that Google actually managed to return the right keyword in the search results, which is "arte". This has motivated us to look into refining the returned keywords, instead of only using the URLs for future work.

Another notable observation is that the proposed method does not necessarily fail when a logo image is absent. Occasionally, using non-logo images (images which were ripped off from legitimate website by phisher) could still lead to correct detection of the phishing webpage. This is reasonable, as very often there are other images which are also unique and representative of the legitimate website. Therefore, a search on these images will lead to the correct legitimate website.

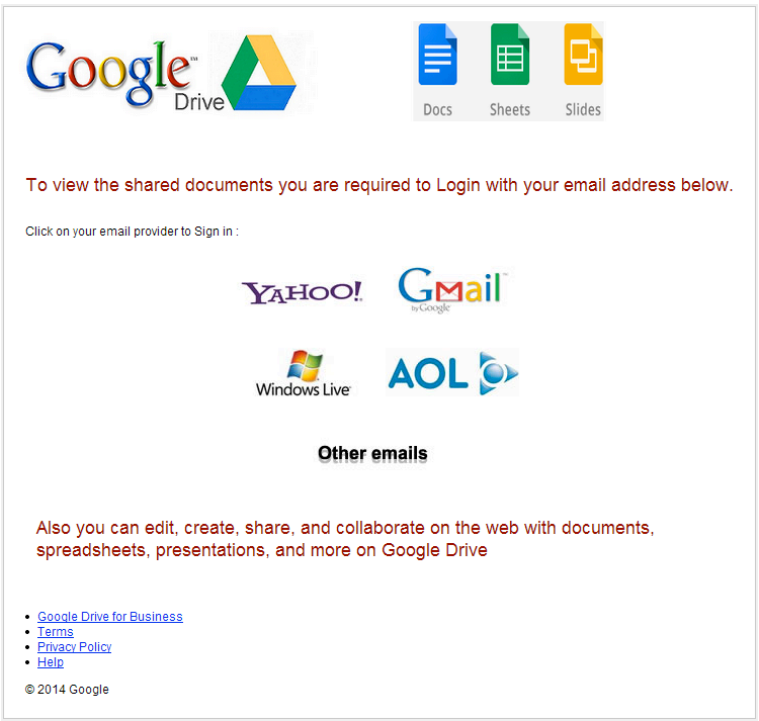
In order to improve the effectiveness of the proposed method, we believe that the first complication discussed above can be improved by enhancing the logo extraction process with a more effective logo detection algorithm. For example, using an object segmentation technique to extract logo image from the screenshot of webpage is a suitable approach. As for the second complication, we can integrate other heuristic approaches such as a TF-IDF technique.

4.8 Capability in handling multiple logos webpage

It is also noteworthy to mention that our proposed method is capable of handling a webpage with multiple logos, which often appears to be a challenge for anti-phishing techniques using the textual element. For instance, some websites allow users to login with multiple social network IDs, as shown in Figure 4.4. In this case, the website contains multiple logos on the webpage (i.e., logo of the website and logos of the social networks). The proposed method works because the identity verification process will aggregate the comparison results based on each logo. As discussed in Section 3.5, our method will classify the query website as legitimate if the aggregated comparisons return at least one match.



(a)



(b)

Figure 4.4: Example of websites which allow users to login with multiple social networks IDs. (a) Legitimate website. (b) Phishing website.

4.9 Summary

In this chapter, a conventional anti-phishing method (i.e., GoldPhish) is benchmarked to evaluate the effectiveness of our proposed method. The process of preliminary implementations and dataset construction are explained too. From the experimental results, our proposed method appears to be highly effective in terms of true positive rate, and also surpassed GoldPhish in terms of the true negative rate. Strengths and limitations of each anti-phishing method are discussed based on the results. Further analysis on the phishing samples and the experimental results suggested that the proposed method is capable of handling multiple language websites and the less popular websites. The limitation of proposed method and the countermeasure steps are extensively discussed.

CHAPTER 5

CONCLUSION

5.1 Research Summary

In this research, we propose a heuristic-based approach to detect phishing webpages. We use logo images to determine the consistency between the real identity and the portrayed identity of a website. Consistent identity indicates a legitimate website and inconsistent identity indicates a phishing website. The proposed method consists of logo extraction and identity verification processes. The logo extraction process uses a machine learning technique to detect and extract the right logo image. On the other hand, the identity verification process uses the Google Image Search to retrieve the portrayed identity and perform the verification. The experimental results show a promising outcome. This justifies the effectiveness and feasibility of using logos to detect phishing websites. We observed that using a graphical element such as logo is more advantageous compared to a textual element in determining the website identity.

5.2 Research Contribution

Three research objectives have been outlined in Chapter 1. First, this research intends to propose an automated phishing detection method that does not require user interaction. The second objective is to extend the phishing detection on non-English webpages. Lastly, this research intends to reduce the cost and reliance on the maintenance of a predefined database.

Throughout this phishing detection research, the following research contributions are achieved:

(a) Attained robustness in distinguishing non-English phishing webpages.

Results have proven the significance of using logo image as a language independent feature. It overcomes the limitation of textual analysis approaches and is capable of extracting the feature despite the variation of webpage languages. By evaluating the proposed method on a dataset with multiple languages, a better overall accuracy is achieved as compared to the benchmarked method.

(b) Reduce the cost of maintaining an up-to-date predefined database.

Image-based anti-phishing approaches which involve the visual similarity calculation will usually incur a high cost in maintaining an up-to-date database for image similarity comparison. However, the use of Google Image Search database in the proposed method has circumvented the necessity of the aforementioned database. Moreover, the Google Image Search facility has also taken the role of calculating the visual similarity properties.

(c) Improved the detection accuracy on phishing attack that is targeting newly launched legitimate website.

Google is known to crawl new and updated webpages continuously and add them into the Google search index. It is reasonable to believe that a newly created legitimate website will be indexed by Google in a short time. Hence, when a phisher is targeting a newly launched legitimate website, the proposed method has a relatively higher chance to detect the phishing webpage.

(d) Offers long-term effectiveness by leveraging on a stable phishing characteristic.

The proposed method is based on a stable phishing characteristic that stays intact over time, namely the logo images. Logo image of the targeted website will always be included in the

phishing webpages, as it is used to increase the similarity and reliability of the phishing webpage. As such, the proposed method that utilises logo images to find the actual identity in determining the webpage legitimacy is robust against evolving phishing strategies.

5.3 Future Works

Clearly, an effective logo extraction process will improve the overall phishing detection accuracy. We intend to explore an object segmentation approach as our future work. For example, instead of locating the logo image from a pool of downloaded images (image resources of a query webpage), we will capture the screenshot and perform object segmentation directly to extract the logo. This approach has a few advantages. Namely, the captured screenshot is the actual rendered web content, which means there is no other hidden image. Furthermore, this approach can avoid getting a logo within a sprite type of image which is usually used to optimise website loading speed. Using a sprite image as a query will cause Google Image Search to return an undesired result even through the logo existed within the sprite image. In addition, it will be more precise to perform logo extraction from the banner image of a website. In other words, the extracted images are less likely to contain other non-logo images.

REFERENCES

Abdelhamid, N. (2015). Multi-label rules for phishing classification. *Applied Computing and Informatics*, *11*(1), 29–46.

Abrams, R., Barrera, O., & Pathak, J. (2013). Browser security comparative analysis - phishing protection. Retrieved May 27, 2018, from https://www.nsslabs.com/index.cfm/_api/render/file/?method=inline&fileID=A02950BF-5056-9046-93D93A5D61314F1D

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A Comparison of Machine Learning Techniques for Phishing Detection. In *Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit,* 60-69.

Alexa Internet Inc. (2017). Keyword Research, Competitive Analysis, & Website Ranking. Retrieved January 10, 2017, from https://www.alexa.com/

Alsharnouby, M., Alaca, F., & Chiasson, S. (2015). Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, *82*, 69–82.

Anti-Phishing Working Group. (2017). Phishing Activity Trends Report, 4th Quarter 2016. Retrieved Febuary 14, 2017, from http://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf

AOL Inc. (2017). DMOZ - The Directory of the Web. Retrieved June 8, 2017, from https://www.dmoz.org

Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J., & Gonzalez, F. A. (2017). Classifying phishing URLs using recurrent neural networks. In *Proceedings of the APWG Symposium on Electronic Crime Research*, 1–8.

Baratis, E., Petrakis, E. G. M., & Milios, E. (2008). Automatic website summarization by image content: A case study with logo and trademark images. *IEEE Transactions on Knowledge and Data Engineering*, *20*(9), 1195–1204.

Berify. (2018). Reverse Image Search: The Best Image Search Engines of 2018. Retrieved October 20, 2018, from https://berify.com/blog/best-reverse-image-search-engines/

Befiry. (2018). Berify Image Search page. Retrieved October 20, 2018, from https://www.berify.com/

Best of the Web. (2017). Best of the Web Directory. Retrieved June 8, 2017, from https://botw.org/

Bianco, S., Buzzelli, M., Mazzini, D., & Schettini, R. (2017) Deep learning for logo recognition. *Neurocomputing, 245*, 23-30.

Bing. (2018). Bing Image Search page. Retrieved October 20, 2018, from https://www.bing.com/?scope=images&FORM=Z9LH1

Cao, Y., Han, W., & Le, Y. (2008). Anti-phishing based on automated individual white-list. In *Proceedings of the 4th ACM Workshop on Digital Identity Management,* 51-60.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27.

Chang, E. H., Chiew, K. L., Sze, S. N., & Tiong, W. K. (2013). Phishing detection via identification of website identity. In *Proceedings of the International Conference on IT Convergence and Security, Macao, China* , 1–4.

Chaudhry, J. A., Chaudhry, S. A., & Rittenhouse, R. G. (2016). Phishing attacks and defenses. *International Journal of Security and Its Applications*, *10*(1), 247–256.

Choo, J. S. F. (2014). Simple Soft [Software]. Sarawak: Kuching.

Choo, J. S. F., Chiew, K. L., & Sze, S. S. (2014). Phishdentity: Leverage Website Favicon to Offset Polymorphic Phishing Website. In *Proceedings of the 9th International Conference on Availability, Reliability and Security*, 114–119.

Chu, W., Zhu, B. B., Xue, F., Guan, X., & Cai, Z. (2013). Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In *Proceedings of the IEEE International Conference on Communications*, 1990–1994.

Cisco. (2017). What is Phishing? Retrieved May 27, 2018, from https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html

Cook, D. L., Gurbani, V. K., & Daniluk, M. (2009). Phishwish: a simple and stateless phishing filter. *Security and Communication Networks*, *2*(1), 29–43.

Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Meredu, G., Ariu, D., & Roli, F. (2017). DeltaPhish: Detecting phishing webpages in compromised websites. In *Proceedings of the 22nd European Symposium on Research in Computer Security,* 370–388.

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 581–590.

Dudhe, P. D., & Ramteke, P. L., (2015). A Review On Phishing Detection Approaches, In *International Journal of Computer Science and Mobile Computing, 4*(2), 166-170.

Dunlop, M., Groat, S., & Shelly, D. (2010). GoldPhish: Using Images for Content-Based Phishing Analysis. In *Proceedings of the Fifth International Conference on Internet Monitoring and Protection,* 123–128.

Fu, A. Y., Liu, W., & Deng, X. (2006). Detecting phishing web pages with visual similarity assessment based on Earth Mover's Distance (EMD). *IEEE Transactions on Dependable and Secure Computing*, *3*(4), 301–311.

Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007). A Framework for Detection and Measurement of Phishing Attacks. In *Proceedings of the ACM Workshop on Recurring Malcode*, 1–8.

Gastellier-Prevost, S., Granadillo, G., G., & Laurent, M. (2011). Decisive Heuristics to Differentiate Legitimate from Phishing Sites. In *Proceedings SAR-SSI 2011: 6th Conference on Network Architectures and Information Systems Security*, 1-9.

Glagolevs, J., & Freivalds, K. (2017). Logo detection in images using HOG and SIFT. In *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering,* 1-5.

GNU. (2015). GNU Wget (Version 1.16.3) [Software]. Available from https://www.gnu.org/software/wget/

Google. (2018). Google Image Search pages. Retrieved May 23, 2018, from https://www.google.com/imghp?hl=EN

Google Developer. (2018). Google Image Search API. Retrieved May 23, 2018, from https://developers.google.com/image-search/

Google Developer. (2018). Overview on Custom Search JSAON API. Retrieved May 23, 2018, from https://developers.google.com/custom-search/docs/overview

Gowtham, R., Gupta, J., & Gamya, P. G. (2017). Identification of phishing webpages and its target domains by analyzing the feign relationship. In *Journal of Information Security and Applications*, *35*, 75-84.

Gupta, B. B., Arachchilage, N. A. G., & Psannis, K. E. (2017). Defending against phishing attacks: taxonomy of methods, current issues and future directions. *Telecommunication Systems*, *67*(2), 247-267.

Gupta, S., & Singhal, A. (2018). Dynamic Classification Mining Techniques for Predicting Phishing URL. In *Proceedings of the International Conference on Soft Computing: Theories and Applications,* 537–546.

Hara, M., Yamada, A., & Miyake, Y. (2009). Visual Similarity-based Phishing Detection without Victim Site Information. In *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security,* 30–36.

Haruta, S., Asahina, H., & Sasase, I. (2018). Visual Similarity-Based Phishing Detection Scheme Using Image and CSS with Target Website Finder. In *Proceedings of the IEEE Global Communications Conference,* 1–6.

Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. In *IEEE Intelligent Systems and their Applications, 13*(4), 18–28.

Hogg, R. V., Tanis, E., & Zimmerman, D. (2015). *Probability and Statistical Inference, 9th Edition.* New York, NY: Pearson.

Huang, C., Y., Ma, S., P., Yeh, W., L., Lin, C., Y., & Liu, C., T. (2010). Mitigate web phishing using site signatures. In *TENCON 2010-2010 IEEE Region 10 Conference,* 803–808.

Iandola, F. N., Shen, A., Gao, P., & Keutzer, K. (2015) DeepLogo: Hitting Logo Recognition with the Deep Neural Network Hammer. In *arXiv:1510.02131*

IDG Consumer & SMB. (2007). Types of Phishing Attacks. Retrieved May 16, 2015, from http://www.pcworld.com/article/135293/article.html

Jain, A. K., & Gupta, B. B. (2015). PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning. In *Proceedings of the Computer Society of India*, 467–474.

Jain, A. K., & Gupta, B. B. (2017). Phishing Detection: Analysis of Visual Similarity Based Approaches. In *Security and Communication Networks, 2017*(4)*,* 1-20.

Jain, A. K., & Gupta, B. B. (2018). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, 1–14.

Jain, A., K., & Vailaya, A. (1998) Shape-Based Retrieval: A Case Study with Trademark Image Databases. In *Pattern Recognition, 31*(9), 1369-1399.

Kharraz, A., Kirda, E., Robertson, W., Balzarotti, D., & Francillon, A. (2014). Optical delusions: A study of malicious QR codes in the wild. In *Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks,* 192–203.

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091–2121.

Liu, W., Deng, X., Huang, G., & Fu, A. Y. (2006). An antiphishing strategy based on visual similarity assessment. *IEEE Internet Computing*, *10*(2), 58–65.

Liu, W., Fang, N., Quan, X., Qiu, B., & Liu, G. (2010). Discovering phishing target based on semantic link network. *Future Generation Computer Systems*, *26*(3), 381–388.

Mao, J., Li, P., Li, K., Wei, T., & Liang, Z. (2013). BaitAlarm: Detecting Phishing Sites Using Similarity in Fundamental Visual Features. In *Proceedings of the 5th International Conference on Intelligent Networking and Collaborative Systems,* 790–795.

Marchal, S., Armano, G., Grondahl, T., Saari, K., Singh, N., & Asokan, N. (2017). Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Transactions on Computers*, *66*(10), 1717–1733.

Mathworks. (2015). MATLAB (Version R2015b) [Software]. Available from https://www.mathworks.com/products/new_products/release2015b.html

Maurer, M. E., & Höfer, L. (2012). Sophisticated phishers make more spelling mistakes: Using URL similarity against phishing. In *Proceedings of the 4th International Symposium on Cyberspace Safety and Security,* 414–426.

Mavroeidis, V., & Nicho, M. (2017). Quick response code secure: A cryptographically secure anti-phishing tool for QR code attacks. In *Proceedings of the International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security,* 313–324.

McGrath, D. K., & Gupta, M. (2008). Behind phishing: an examination of phisher modi operandi. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats,* 1-8.

Mehtre, B. M., Kankanhalli, M. S., & Lee, W. F. (1998). Content-Based Image Retrieval Using a Composite Color-Shape Approach. In *Information Processing and Management, 34*(1), 109-120.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2013). Predicting Phishing Websites using Neural Network trained with Back-Propagation. In *Proceedings of the 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing*, 682-686.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Intelligent rule-based phishing websites classification. *Information Security*, *8*(3), 153–160.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015a). Tutorial and critical analysis of phishing websites methods. *Computer Science Review*, *17*, 1–24.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015b). Phishing Website Features. Retrieved December 24, 2017, from http://eprints.hud.ac.uk/24330/

Moinvaziri, N. (2013). WebShot (Version 1.9.3.0) [Software]. Available from https://nmoinvaz.gitlab.io/webshot/#/home

Na, S. Y., Kim, H., & Lee, D. H. (2014). Prevention Schemes Against Phishing Attacks on Internet Banking Systems. *International Journal of Advances in Soft Computing and Its Application*, *6*(1), 1–15.

Nguyen, L. A. T., To, B. L., Nguyen, H. K., & Nguyen, M. H. (2013). Detecting phishing web sites: A heuristic URL-based approach. In *Proceedings of the International Conference on Advanced Technologies for Communications,* 597–602.

Nirmal, K., Janet, B., & Kumar, R. (2015). Phishing - the threat that still exists. In *Proceedings of the International Conference on Computing and Communications Technologies,* 139–143.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66.

PayPal. (2018). Login Portal – Paypal. Retrieved May 23, 2018 , from https://www.paypal.com/my/signin

PhishLabs. (2017). 2017 Phishing Trends and Intelligence Report. Retrieved January 10, 2017, from https://pages.phishlabs.com/rs/130-BFB-942/images/2017 PhishLabs Phishing and Threat Intelligence Report.pdf

PhishTank. (2017). Join the fight against phishing. Retrieved January 10, 2017, from https://www.phishtank.com/

Prakash, P., Kumar, M., Rao Kompella, R., & Gupta, M. (2010). PhishNet: Predictive blacklisting to detect phishing attacks. In *Proceedings of the 30th IEEE International Conference on Computer Communications,* 1–5.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.

Ramesh, G., Krishnamurthi, I., & Kumar, K. S. S. (2014). An efficacious method for detecting phishing webpages through target domain identification. *Decision Support Systems*, *61*, 12–22.

Rish, I. (2001). An empirical study of the naive Bayes classifier. In *Proceedings of IJCAI-2001 Workshop on Empirical Methods in Artificial Intelligence IBM, 3,* 41–46.

Rosiello, A., P., E., Kirda, E,., Kruegel, C., & Ferrandi, F.(2007). A layout-similarity-based approach for detecting phishing pages. In *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops*, 454–463.

RSA. (2017). 2017 Global Fraud and Cybercrime Forecast. Retrieved January 10, 2017, from https://www.rsa.com/content/dam/en/infographic/2017-global-fraud-forecast.pdf

Schneider, F., Provos, N., Moll, R., Chew, M., & Rakowski, B. (2008). Phishing Protection: Design Documentation. Retrieved May 27, 2018, from https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation

Siddique, B., Malekar, U., & Kashyap, M. (2017). An Anti-phishing Framework Based on Visual Cryptography. *International Research Journal of Engineering and Technology*, *4*(3), 2186–2190.

Soman, C., Pathak, H., Shah, V., Padhye, A., & Inamdar, A. (2008). An Intelligent System for Phish Detection, using Dynamic Analysis and Template Matching. *International Journal of Computer and Information Engineering*, *2*(6), 1927–1933.

Sorio, E., Bartoli, A., & Medvet, E. (2013). Detection of hidden fraudulent URLs within trusted sites using lexical features. In *Proceedings of the 8th International Conference onAvailability, Reliability and Security,* 242–247.

Stoodley, K. (2004). Organized crime will back phishers. Retrieved May 27, 2018, from https://www.esecurityplanet.com/trends/article.php/11164_3451501_2/In-2005-Organized-Crime-Will-Back-Phishers.htm

Tan, C. L., Chiew, K. L., Wong, K. S., & Sze, S. N. (2016). PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decision Support Systems*, *88*, 18–27.

Verma, R., & Hossain, N. (2014). Semantic Feature Selection for Text with Application to Phishing Email Detection. In *Proceedings of the 16th International Conference on Information Security and Cryptology,* 455–468.

Witten, I. H., & Frank, E. (2002). *Data mining: practical machine learning tools and techniques with Java implementations*. New York, NY: ACM.

Xiang, G., & Hong, J. I. (2009). A hybrid phish detection approach by identity discovery and keywords retrieval. In *Proceedings of the 18th International Conference on World Wide Web,* 571–580.

Xue, Y., Li, Y., Yao, Y., Zhao, X., Liu, J., & Zhang, R. (2016). Phishing sites detection based on Url Correlation. In *Proceedings of the 4th International Conference on Cloud Computing and Intelligence Systems,* 244–248.

Zeydan, H. Z., Selamat, A., & Salleh, M. (2014). Current State Of Anti-Phishing Approaches And Revealing Competencies, In *Journal of Theoretical and Applied Information Technology, 70*(3), 507-515.

Zhang, J., Luo, S., Gong, Z., Ouyang, X., Wu, C., & Xin, Y. (2011). Protection Against Phishing Attacks: A Survey. *International Journal of Advancements in Computing Technology*, *3*(9), 155–164.

Zhang, J., Wu, C., Guan, H., Wang, Q., Zhang, L., Ou, Y., Xin, Y, & Chen, L. (2010). An content-analysis based large scale Anti-Phishing Gateway. In *Proceedings of the 12th IEEE International Conference on Communication Technology,* 979–982.

Zhang, W., Lu, H., Xu, B., & Yang, H. (2013). Web phishing detection based on page spatial layout similarity. *Informatica*, *37*(3), 231–244.

Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. In Proceedings of the 16th International World Wide Web Conference, 639–648.

APPENDICES

Appendix A: Software used

1. Title: MITLAB (R2015b)
   Version: 8.6
   Date release: September 2015
   Author: MathWorks® (*https://www.mathworks.com/*)
2. Title: GNU Wget for Windows.
   Version: 1.11.4
   Date release: December 2008
   Author: GNU (*https://www.gnu.org/software/wget/*)
3. Title: WebShot
   Version: 1.9.3.0
   Date release: June 2013
   Author: Nathan Moinvaziri (*https://nmoinvaz.gitlab.io/webshot/#/home*)
4. Title: GoldPhish
   Version: 1.0.0
   Date release: 2014
   Author: Jeffery Choo
5. Title: Simple Soft
   Version: 1.0.0
   Date release: 2014
   Author: Jeffery Choo

Appendix B: List of Publication

1. Chiew, K. L., Chang, E. H., Sze, S. N., & Tiong, W. K. (2015). Utilisation of website logo for phishing detection. *Computers & Security, 54,* 16–26.
2. Chang, E. H., Chiew, K. L., Sze, S. N., & Tiong, W. K. (2013). Phishing detection via identification of website identity. In *Proceedings of the International Conference on IT Convergence and Security*, 1–4.