Phishing detection using machine learning

MSc Research Project
MSc Cyber Security

Suraj Wandhare
Student ID: 18157432

School of Computing
National College of Ireland

Supervisor: Ben Fletcher

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Suraj Wandhare
Student ID:	18157432
Programme:	MSc Cyber Security
Year:	2019
Module:	MSc Research Project
Supervisor:	Ben Fletcher
Submission Due Date:	03/02/2020
Project Title:	Phishing detection using machine learning
Word Count:	3500
Page Count:	11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	3rd February 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Phishing detection using machine learning

Suraj Wandhare
18157432

Abstract

Phishing attacks cause a loss of millions of dollars every year. It involves social engineering which makes it much more effective. There are many proposed solutions to solve the problem of phishing using machine learning. This research is partly a study to compare different supervised machine learning algorithms to find the optimal algorithm for phishing detection using machine learning and partly to address the issue of URL shortening service exploitation. URL shortening is used on a daily basis to help reduce the size of the URL, but attackers use these services to hide the original URL of the phishing website. In this research we propose an application that takes in a URL, uses multiple feature extraction techniques to determine whether the URL is phishing or not. We propose a URL unshortener which will return the original URL, compliments the machine learning algorithm increasing its accuracy. The initial accuracy is 92.6% and the False Positive rate is 0.4% but when the URL unshortener is added the accuracy is increased to 96.6%.

1 Introduction

Over the last few years, the Web has seen a massive growth in the number and kinds of web services that include social networking, forums, blogs, and video sharing sites [1]. The more ways people are able to interact, the more ways to exploit these services. One-way criminals exploit this is by using Phishing attacks. Phishing can be described as an attack in which the attacker disguises as a trustworthy person to obtain information such as login credentials and credit card information. It is a combination of social engineering attack which is used to deceive and manipulate someone to give away their data. Phishing websites are crafted such that they are very similar to the original website which it tries to mimic. It contains at least one login page which captures the confidential information.

There are numerous ways to detect phishing sites, one of the most efficient way is to create a blacklist. PhishTank is an open source library which acts like a blacklist for phishing websites, controlled and operated by community members which ranks websites as phishing and not phishing based on votes. The important requirement for a blacklist is timely updates. Another efficient way to detect and prevent phishing is by combining previously built mechanisms to machine learning. Machine learning is the subset of artificial intelligence used by computer systems which uses algorithms, statistical models, patterns and inferences to complete a certain task without human intervention.

This study aims to compare the available machine learning algorithms for detecting phishing websites and create a system that include detection and prevention of 'phishing using URL shortening services'. The literature review carried out for this research revealed that there are numerous phishing detection techniques built upon machine learning but a only few address the problem with the exploitation of URL shortening services. A URL shortening service is third party service which takes a long URL and converts it into a long case sensitive alphanumeric code. Some of the most popular URL shortening services are

- Bit.ly
- adf.ly
- goo.gl
- tinyurl.com

This paper mainly focuses of different types of machine learning algorithms that can be used to detect phishing websites combined with the URL unshortening services. The dataset used for this research is taken from PhishTank.com and PhishLabs.com which are the two top reliable community based phish verification systems(more description is section 5). The research question proposed by this research is Can machine learning algorithms be used to prevent phishing by URL shortening techniques. Various algorithms and models have been researched and studied in the literature review to compare the already existing models and chosen considering the complexity of the problem.

2 Related Work

Phishing detection and prevention is a major step towards protection of internet and cybersecurity. Most phishing incidents occur due to the inability of the user to differentiate between a real and phishing webpage. Machine learning is a way to prevent phishing from reaching the user. One way to achieve a good defence against phishing is if phishing website is blocked before the user accesses it. Phishing detection using machine learning is mainly classified into three categories.

2.1 Blacklisting URLs and IP addresses

A phishing blacklist is a list of phishing URLs which have been reported by security experts or community members as harmful or dangerous. PhishTank is a open source community which contains a large database of websites which can be used as a blacklist. Google also provides an API called Google Safe Browsing API which comes built in with the Google Chrome browser. In this research the PhishTank database is used.

The study [2] demonstrates how a framework can be developed to create a blacklist for phishing websites. Authors propose a framework called SEAHound which works in four steps to differentiate between a normal and a phishing website. The first feature checks if the website has any bad commands or malicious commands. Second it checks if the webpage contains any message that denote urgency, which is fairly common in phishing attacks. Third is generic greeting and the last is link analysis which uses third party API to check the validity of the URL. The study promises a secure framework but as the use of scripts and frameworks to develop websites increases it becomes more difficult to analyse the HTML content of the webpage.

Another study [3] shows how machine learning based on multidimensional features driven by deep learning can help prevent phishing attacks. The author makes a very good argument that 47%-83% of phishing websites are added to blacklists after 12 hours, and 63% of phishing websites have a lifespan of only 2 hours. This makes the blacklisting approach obsolete. The author proposes a model consisting of CNN-LSTM(Convolutional Neural Networks-Long Short-Term Memory), DCDA and multidimensional feature detector. CNN is used to extract local features and LSTM is used to extract context dependency. As a counterargument rather than discarding blacklisting approach if it is combined with the new machine learning model produces more efficient and secure system.

Whitelisting URLs can also prevent phishing to a certain extent but as the internet increases and with new websites emerging everyday, it is not a feasible option to create a whitelist and update it regularly. This study [3] uses whitelist approach to create a browser extension that uses DNS servers to detect phishing and pharming attacks.

2.2 Domain name based phishing detection

The Uniform Resource Locator, also called as a web address consists of different parts from which information about it can be extracted. Consider the url `https://www.example.com/students/?id=1234#page2` The URL mainly consists of: [1]

- Protocol: Represents the technology being used to transfer the data. In this case https.
- Domain: The registered domain name of the website, www.example.com
- Path: The path associated with the page on the web server, /students/
- Hash: Shows the section on the page page2
- Query string: The data being transferred, ?id=1234

Machine learning can be used to extract features from a url. The study [4] demonstrates the feature extraction in depth.

- Length of the URL: The number of characters in the domain string. High number of phishing attacks have a long URL.
- Frequency of domain name: The number of times the domain name appears on the website.
- Page title and domain name match.

There are several more features that can be extracted from the URL to determine if the URL is used for phishing attacks. Authors in the study [5] propose phishstorm which is a phishing rating system which extracts URL features and ranks them on the basis of registered domain name. The author also suggests that URL obfuscation techniques are used to prevent phishing to be detected.

- URL obfuscation with other domain.

[1] `https://community.tealiumiq.com/t5/iQ-Tag-Management/URL-Components-Explained/ta-p/5573`

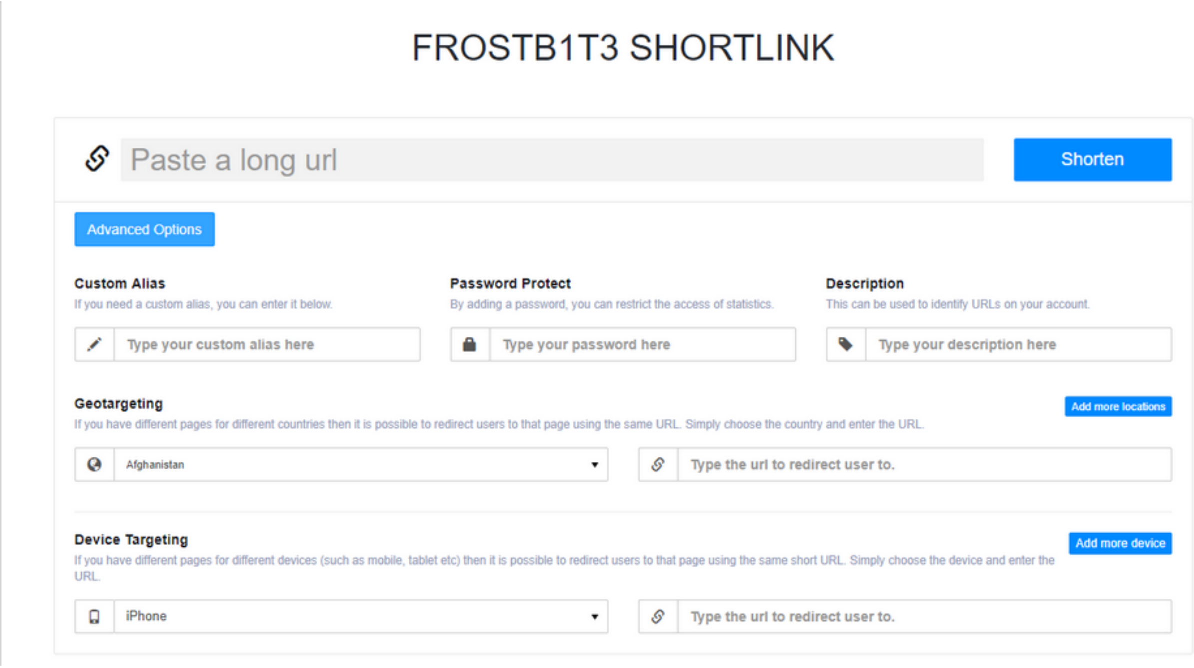
Phishing attacks		
Rank	**Category**	**% Clicks**
1	direct	61.93%
2	Social Media	12.14%
3	Social Networks	3.67%
4	Anonymization	3.64%
5	Webmail	2.31%

Table 1: Largest number of clicks

- URL obfuscation with other URL.
- Obfuscation with IP address.
- Obfuscation with URL shortener.

The research aims to address the problem of obfuscation with URL shortener. The study [6] explains how the URL shortening service is being exploited by attackers to conduct phisihing attacks. The data suggests that short URL that are used for phishing last more than 3 months while regular phisihing websites tend to expire within a month. These links are embedded in the advertisements present on website these days. The data below1 shows the percentage of clicks that occur from different sources

The URL shortening services used by attackers provide features like Geotargeting and Device targeting. An example of this is FROSTBITE spelled as FROSTB1T3.



FROSTB1T3 SHORTLINK

Custom Alias

If you need a custom alias, you can enter it below.

Password Protect

By adding a password, you can restrict the access of statistics.

Description

This can be used to identify URLs on your account.

Geotargeting

If you have different pages for different countries then it is possible to redirect users to that page using the same URL. Simply choose the country and enter the URL.

Afghanistan

Device Targeting

If you have different pages for different devices (such as mobile, tablet etc) then it is possible to redirect users to that page using the same short URL. Simply choose the device and enter the URL.

iPhone

Figure 1: FROSTB1T3 URL shortening service

2.3 Content Analysis based Phishing detection

Website HTML data can tell a lot about the data present on the website, and HTML of any website can be downloaded using a simple curl command. The suspicious advertisements and bogus offers are common on phishing websites.

The study [7] proposes a feature rich machine learning framework CANTINA+ for phishing detection. It is one of the best frameworks on paper for phishing detection. CANTINA+ uses 15 feature based phishing detection extracted from the URL, HTML DOM and other resources. It works as follows, in the training phase first a hash-based filter compares the values to a database of known phishing sites. Second the webpage is searched for login form, URL features are extracted, then online tools such as whois and page rank are determined, finally based on all these results the machine determines if the page is phishing. In the testing phase the machine makes a prediction based on the training data and goes through all the tests. But the study does not address the solution to the problem of phishing using URL shortener.

Based on the data available online it is evident that phishing detection using machine learning is a very popular topic and gets a lot of support from the community of researchers. This research tries to combine the work done on Phishing detection using machine learning and combine it with a URL unshortener, a module or extension that will unshorten the URL. In the next section we will discuss the approach used in this research to fill in the gaps that are present in the literature.

3 Methodology

This research was carried out according to the linear-sequential model or commonly known as waterfall model. The waterfall model is considered conventional but it is optimal in cases where the requirements are minimal. The application was divided into smaller components, which were built using frameworks and manual code. The components were GUI development, database, machine learning, URL unshortener. These were combined together to work as a collective unit.

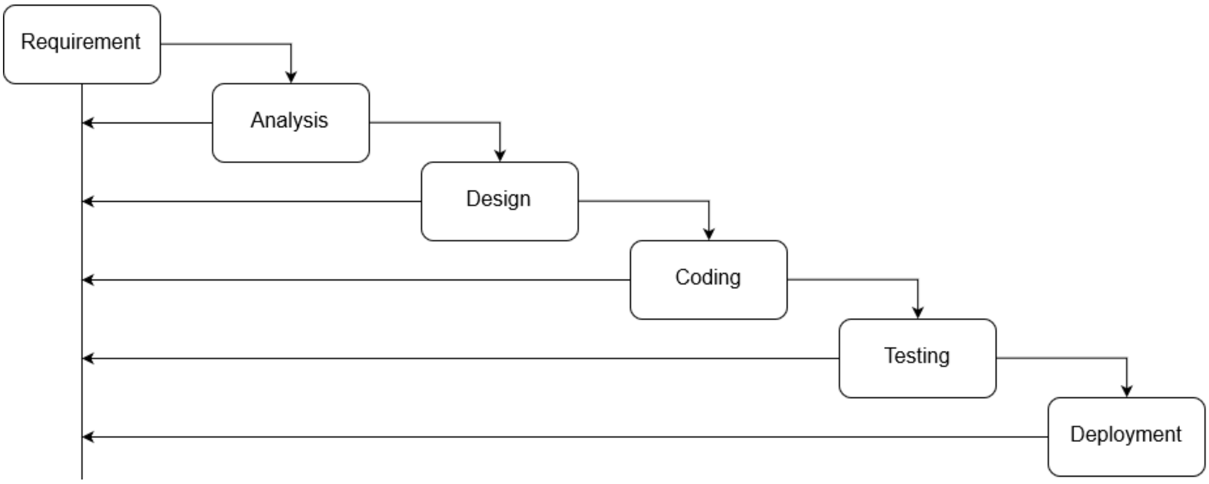


Figure 2: Waterfall model

The dataset contains 1700+ entries of URLs classified as phishing and non phishing URLs which are a combination of long and short URLs. The machine learning algorithm does not understand on its own if a URL is short or long and thus this paper proposes a solution to unshorten the URL and then pass into the machine learning. Different machine learning algorithms were tested and the optimal algorithm was chosen according to the results.

4 Design Specification

The system architecture is as follows,

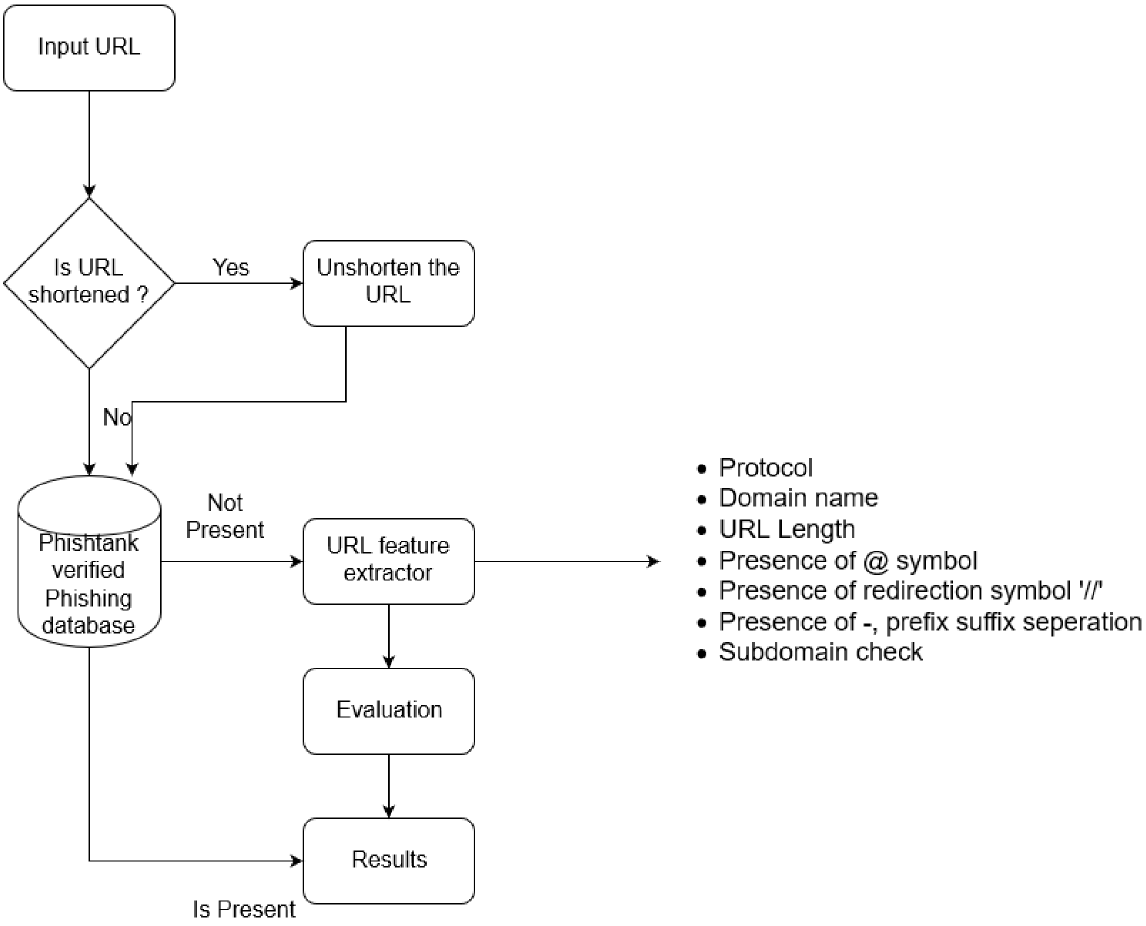


Figure 3: Architecture

Currently the working prototype has three features URL unshortener, Database checker and URL lexical analysis. The application requires an input URL from the user. The first component, URL unshortener compares the URL to a list of 400+ URL shortening services and returns the unshortened version of the URL. The second feature compares the URL with a phishing database of 30000+ phishing website URLs and checks if the entered URL is present in the database. If the database check returns false then the final component does a lexical analysis on the URL with the following checks.

- Protocol: The protocol used http/https.
- Domain name: Check if the website has legitimate domain name, example anazon.com instead of amazon.com.

- Long URL: The length of the URL.
- @ symbol in URL
- Redirection symbol '//'
- Separate prefix and suffix, if applicable, separated by -.
- Subdomain check: If the URL contains any sub-domains.

5 Implementation

This section describes the process adopted for the implementation of the proposed solution.

5.1 GUI Implementation

PyQT5 was used to develop the GUI using drag and drop [2]. PyQT is one of the most popular Python bindings for the QT cross platform library which itself is written in C++ but by building it in python the speed of application is much more as compared to traditional C++ GUI. PyQT comes with widgets, layouts, stylesheets, custom and built-in styles. The requirements for this research are less and it just requires an input URL field and a submit button. The GUI template was downloaded from github [3]

5.2 PhishTank Database

The database to check the URL against was downloaded from phishtank which provides a downloadable JSON data file which contains data entries of phishing websites. PhishTank provides developer support by signing up and registering an API key. [4] The application uses JSON parser to check the URL from the JSON file. The application checks the input URL in the phishtank database, if a match is found simply returns the PhishTank ID which can be used to further investigate on the URL.

5.3 Machine learning

Python libraries numpy, sklearn and pandas were used for machine learning. Numpy is a python package which provides support to multi-dimensional arrays . The application works on a confidence score basis, it means that the features extracted from the URL add up to a score, if the URL is very long and contains '@' symbols and '//' redirect symbols then there is more chance that the website is used for phishing.

Machine learning algorithm selection is one of the most important steps when deciding to build a application. There are mainly three categories of machine learning algorithms, supervised learning, unsupervised learning and semi-supervised learning. This research focuses on using the supervised learning algorithms as implementation is more efficient and accurate. Unsupervised learning requires more time and resources. Also labelled data is available online from verified sources such as PhishTank.com. The dataset used for this research contains:

[2] http://zetcode.com/gui/pyqt4/dragdrop/

[3] `https://github.com/chrschorn/pyqt-gui-template`

[4] `https://www.phishtank.com/developer_info.php`

	Phishing pages	**Legitimate pages**	**Total**
Training data	217	172	389
Testing data	1097	688	1785

Table 2: Dataset details

The optimal algorithm chosen for the implementation of this application was Random Forest because of the high rate of accuracy and precision. The following machine learning algorithms were used to predict the results,

- Logistic Regression is used to predict the probability of categorical dependent variable. It takes in binary input where data is coded as 1(true, success, yes) and 0(false, fail, no). This algorithm is simple to use and implement and also compatible with the dataset. The algorithm compliments to the dataset as the one of the drawback of this algorithm is that it performs poorly if complex non-linear relationships exist between the variables.

- Random Forests uses multiple learning algorithms for classification and regression. It constucts a decision tree at training time to predict possible consequences. This algorithm does not provide the optimal results as it lacks reproducibility.

- Neural Networks is a set of algorithms that mimics the working of the human brain, designed to recognize pattern. It also requires an experienced professional to use neural networks.

- Support Vector Machines commonly known as SVMs are used for both regression and classification purposes. It is based on the idea to divide the dataset into two classes on a hyperplane which is a line that linearly seperates and classifies a set of data. SVMs are difficult to work with as they require high computations to train the data. They are prone to overfitting.

6 Evaluation

The evaluation metrics used in this research were True Positive (TP), True Negative (TN) and False Positive (FP) and False Negative(TN). The term accuracy can be defined as ratio of correct predictions to total number of predictions.

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN}$$

Precision can be defined as the ratio of true positives to all positives i.e. correctly predicted positive observations to total positive observations.

$$Precision = \frac{TP}{TP + FP}$$

6.1 Results

The table 3 shows the True Positive (TP) and False Positive (FP) of the algorithms used during the testing phase.

	Type of phish in testing set			
	Unique		URL unshortener	
Algorithm	TP(%)	FP(%)	TP(%)	FP(%)
Logistic Regression	92.5	2.2	96.5	1.8
Random Forests	95.2	0.4	98.6	0.5
Support Vector Machines	93.3	0.8	94.7	0.4

Table 3: True Positive and False Positive rates

Algorithm used	**Accuracy(%)**	**Precision(%)**
Logistic Regression	89.2	81.96
Random Forests	92.6	96.47
Support Vector Machines	82.6	84.23

Table 4: Accuracy Chart of different algorithms

Table 4 shows the accuracy calculated during the testing phase.

The table e=below shows the results of the same test after using URL unshortener. The Accuracy of Logistic Regression and Random forest increases but decreases in the case of Support Vector Machines but there is a slight increase in precision for all the algorithms.

Algorithm used	**Accuracy(%)**	**Precision(%)**
Logistic Regression	91.5	82.96
Random Forests	96.6	97.15
Support Vector Machines	81.6	88.23

Table 5: Accuracy Chart after using URL unshortener

6.2 Discussion

The findings suggest that there is no optimal machine learning algorithm for the problem. However Random Forest algorithm was chosen because of consistent results and high accuracy. The study [8] explains how Random Forest was more precise and accurate as compared to other algorithms and the URL unshortener is complementing the machine learning algorithms and increases overall accuracy of the system. Another study [9] specifically designed a system that used Random Forest for phishing detection also support that Random Forest is a good solution for selection of machine learning algorithms.

7 Conclusion and Future Work

The first objective of this research was to compare the present supervised machine learning algorithms to find a optimal algorithm for phishing detection but the study was unable to prove that the algorithm used is optimal. The study promises a good accuracy and precision score with low False Positive rates. The second aim of the research was to implement a URL shortener bypass which will take any URL as input, check if it is short

URL and return the long URL if it is. The study also shows how the URL unshortener compliments the overall accuracy of the system. The limitation of the proposed study is time, the algorithms are efficient but they are not as fast as compared to the existing research.

The study was solely based on supervised machine learning algorithms but new emerging unsupervised and semi-supervised algorithms have promising results. The future work for this research is to present a similar comparative study for unsupervised machine learning algorithms. This is to address the problem of time, speed is a necessity in todays world and must be addressed.

8 Acknowledgements

I would like to thank the supervisor Ben Fletcher for his support and his guidance, technical expertise and motivation. I would like to thank National College of Ireland to provide me resources for this research. I am sincerely grateful to Dr. Arghir Moldovan for taking out time and helping me with the research. I would like to thank my family without whose support this would not be possible.

References

[1] M. N. Feroz and S. Mengel, “Phishing url detection using url ranking,” in *2015 ieee international congress on big data*. IEEE, 2015, pp. 635–638.

[2] T. Peng, I. Harris, and Y. Sawa, “Detecting phishing attacks using natural language processing and machine learning,” in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. IEEE, 2018, pp. 300–301.

[3] J. Kang and D. Lee, “Advanced white list approach for preventing access to phishing sites,” in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*. IEEE, 2007, pp. 491–496.

[4] H. Shirazi, B. Bezawada, and I. Ray, “Kn0w thy doma1n name: unbiased phishing detection using domain name based features,” in *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*. ACM, 2018, pp. 69–75.

[5] S. Marchal, J. François, R. State, and T. Engel, “Phishstorm: Detecting phishing with streaming analytics,” *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[6] S. Le Page, G.-V. Jourdan, G. v. Bochmann, J. Flood, and I.-V. Onut, “Using url shorteners to compare phishing and malware attacks,” in *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2018, pp. 1–13.

[7] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, “Cantina+: A feature-rich machine learning framework for detecting phishing web sites,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 2, p. 21, 2011.

[8] I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur, “A novel machine learning approach to detect phishing websites,” in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2018, pp. 425–430.

[9] V. Muppavarapu, A. Rajendran, and S. K. Vasudevan, "Phishing detection using rdf and random forests." *Int. Arab J. Inf. Technol.*, vol. 15, no. 5, pp. 817–824, 2018.