TECHNICAL COMPUTING

CT6039 DISSERTATION 2021/2022

University of Gloucestershire School of Computing and Engineering BSc in Cyber Security

Using Machine learning to Detect phishing emails

Anthony Saich

S1901591

Dr Qublai Ali Mirza

Declaration

"This Dissertation is the product of my own work and does not infringe the ethical principles set out in the university's handbook for Research Ethics.

I agree that it may be made available for reference via any and all media by any and all means now known or developed in the future at the discretion of the university."

Abstract

With the significant growth of the instant communication such as emails used by businesses and private individuals over the last 30 years. Especially with the increase of hybrid working and the working from home culture, traditional methods of anti-phishing technologies are not keeping up to date with the ever-changing attack vectors and anti-phishing detection methods these hackers are using. In consequence, this research investigates the use of machine learning classifiers.

This research aims to examine the current anti phishing measures and technologies that is currently available to organisations. As well as to develop a more efficient machine learning solution. The chosen machine learning classifiers were measured using different metrics; time, accuracy, recall and precision. From these metrics the best classification was Decision Tree which achieved an accuracy rate of 89%.

This research will look at the different methods that can be used to identify phishing.

Table of Contents

List of Tables

List of Figures

1 Chapter One

1.1 Overview

From 2017 to 2021 of those businesses that reported a cyber-attack or breach there was a rise of phishing attacks from 72% in 2017 to 83% in 2021, however, in the same time period there were falls in virus and malware attacks on organisations from 33% to 9% as well as a fall in ransomware attacks from 17% to 7%. (Department for Digital, Culture, Media & Sport, 2021). This indicates that phishing emails and their contents are a growing concern for organisations and there current anti phishing detection systems and measures are not working. There are many reasons for this shift in higher phishing emails, with the current covid environment and the working from home culture is thought to be the major contributing factor.

1.2 Research Aim

This research aims to investigate how current anti phishing detection methods are used, how phishing emails can be detected and identified before they are seen by the users. It will explore the use machine learning classifiers to detect phishing emails.

1.3 Problem Statement

Phishing and its variations are one of the most damaging types of cybercrime, due to the simplicity of its creation. The number of users it can reach and finally the financial reputational effects that phishing has on the user or organisation. Because phishing attacks are a mix of social engineering and technical attack vectors, it’s a difficult attack

to mitigate. With human error being the main cause of phishing attacks being successful (Alkhalil et al., 2021).

1.4 Research Question

1. What are the current methods of detecting phishing emails?

2. What can be implemented to reduces the threats of phishing attacks?

1.5 Objectives

1. To complete a literature review of current anti-phishing technology and techniques to identify weaknesses and strengths.
2. To create and test a machine learning program that can be used to classify and detect phishing emails from legitimate emails.

1.6 Scope

The scope of this research is solely focusing on phishing emails originating from email sources not phishing from text messages, also known as smishing. This research also intends to create and test a fully functioning, machine learning program to flag suspected phishing emails to the user using a green, orange and red light system. This research will not be able to automatically identify phishing emails; this is due to time constraint of the project as well as lack of current and up to date phishing data due to how fast phishing attacks change. It will however indicate to the user the likelihood of a phishing email by the use of the light system.

1.7 Ethics Issues

- I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree or anywhere else. Except where states otherwise by citation and reference or acknowledgment, the work presented is entirely my own.
- I confirm that all the tables and figures in this dissertation/proposal are my works or a regeneration from other people's work with citation.
- I confirm that all the citations in the dissertation/proposal have been provided in the bibliography and they are all accessible. In case that university wants to cross examine the citations, I can provide the content for the references which are not accessible.

2 Chapter 2

2.1 Background

Phishing attacks are not a new attack vector used by cyber criminals, the first known phishing attack was used in the early 1990s (Chaudhry, Chaudhry and Rittenhouse, 2016) on the AOL (America Online) chatrooms. An attacker posing as AOL employee would trick victims into handing over login credentials to access their credit card information. However, with advances in instant communication in email and the widely available list of active email addresses, phishing has changed from a highly personal attack to mostly a mass spamming approach.

It is incredibility hard to measure how many phishing emails are sent out each day and even harder to identify how many are successful however the anti-working phishing group stated that in July 2021 there was 260,642 successful phishing attacks (Apwg.org, 2021). This is a very conservative estimate. Successful phishing attempts like these can result in financial and reputation damages. However, many new technologies and techniques have been introduced to further reduce these numbers and stop attacks from happening. Many of the techniques presented in the literature review will be reflected in the phishing detection program.

2.2 Spam Filters

There are numerous current ways to identify spam and phishing emails using filtering techniques ranging from machine learning based, OCR detection, bag of words and more traditional methods of blacklisting known or suspicious websites.

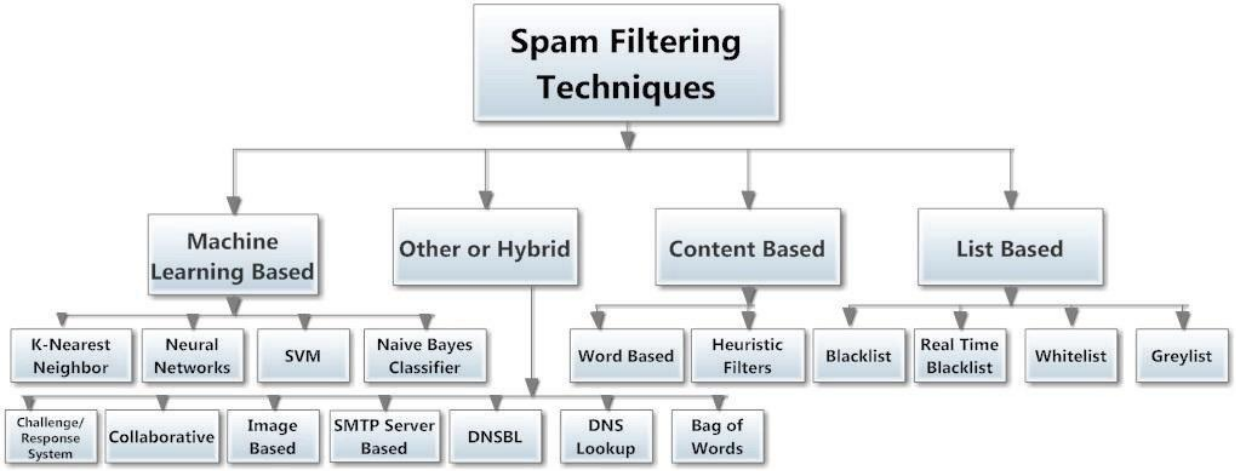


Figure 1 The different types of spam filtering classification (A.G et al., 2014)

2.3 Email Spoofing

Email spoofing is the 'mechanism in which email appears to come from the authentic user but actually it comes from an imposter' (Mistry et al., 2019). This is an important part of successful phishing campaign because nearly all phishing emails come from a malicious source rather from a legitimate account.

There are many protocols and methods that are available to mitigate and reduce email spoofing attacks. The currently used protocols are sender policy framework, also known

as SPF, domain keys identified mail, known as DKIM, and finally Domain-based Message Authentication Reporting and Conformance, known as DMARC (Ncsc.gov.uk, 2019).

SPF works by adding a SPF record to the DNS, this record has a list of IP addresses that are approved to receive and send email into the network. However, SPF has a major limitation, and this is the forwarding of emails, due to the fact that the IP addresses would not be listed on the SPF record. (Agari, 2021). This could lead to false positive results, potentially damaging the reputations of the organisation.

DKIM is similar to SPF, however instead of using IP based identification, it uses an asymmetric encryption digital signature. When the email is received by the network, the signature is checked by the DKIM public key to verify if the email is real or if it is a spoofing attack (Dkim.org, 2009). DKIM allows for the forwarding of emails, an advantage that SPF lacks. However, DKIM is still vulnerable to attacks and flaws. If the public and private keys have poor cryptographic key management, and are cracked, phishing emails can still be sent undetected. A more pressing vulnerability is that DKIM is vulnerable to a replay attack, this is where an attacker embeds malicious code or link within a forwarded email. The accuracy rate of DKIM can be 100% however this depends on how the users implement, however the average success rate if 65%.

Domain-based Message Authentication, Reporting and Conformance, known as authentication-based protocol, uses many different authenticated identifiers, such as SPF and DKIM together (Kucherawy and Zwicky, 2015). (TechRepublic, 2020) shows that

DMARC has a success rate if detection of 67%. However, these results can be lower if the DMARC records are wrong or implemented poorly (Nightingale, 2017).

Table 1 Table of anti-email spoofing protocols

Name of Protocol	**Type of Defence**	**Issues**	**Accuracy Rate**	**References**
SPF	IP based	Forwarded emails often can't get passed the SPF protocol	50%	Ncsc.gov.uk, 2019
DKIM	cryptographic Signature based	Can be susceptible to relay attacks	65%	Dkim.org, 2009
DMARC	Incorporation of both SPF. and DKIM protocols	If set up properly, there are very few issues.	67%	Kucherawy and Zwicky, 2015

2.4 Blacklisting

IP Backlisting also known as blacklisting is the 'collecting IP addresses in a list and prohibit any email communication attempts initiated from these addresses' (Dietrich and

Rossow, 2008). This method of anti-spam and phishing is one of the oldest methods, which has been used for over 20 years, however it is still as effective. As mentioned, the main issue with phishing is the speed of the phishing campaigns and website are created. The Anti-Phishing Working Group found that over two hundred thousand phishing sites a month are created (Phishing Activity Trends Report, 2nd Quarter 2021, 2021). However, it takes on average 12 hours for a phishing IP address to be identified. This delay can lead to users and organisations being vulnerable to attack. Based of research from (Sheng et al., 2009) the success rate of blacklisting was 66%.

2.5 NLP

Natural languages processing also known as NLP are a popular method to detect phishing emails. This is due to NLP network's ability to detect key words and phrases in the email. Park identified that a common method of NLP for phishing is detecting and counting the number of action verbs that influences the users to click a link or entering details (Park, 2013).

A paper by (Salloum et al., 2021) has proved that this method can be effective if done correctly however if this method is conducted by itself with no other systems in place, false positives can be common. For example, receiving a legitimate email prompt from your bank to change your password could be classed as a phishing email. Research from (Egozi and Verma, 2018) found that NLP achieved 99% accuracy rate, however it has to be stated that NLP has a high true negative, this consequently identifying real emails as phishing emails of a rate of 77%.

2.6 Bag of Words

Bag of words also known BOW takes the pieces of text and analyses frequency of keywords in the text. Algorithms do not understand languages instead a numerical value for each of the words in the sentence is given. The numerical value is then used for further analysis. BOW is used to extracts all words present in the email identifying the highest occurring words, these words are then used in the classification model. However, Akinyelu and Adewumi argues that BOW is not suited to phishing emails because "phishing email contains some unique features that are only specific to phishing attacks" (Akinyelu and Adewumi, 2014). BOW is limited by serval factors; The structures of the sentence have no importance for the model for example these two sentences "Theses shoes are good" and "Are these shoes good". To the model these two sentences are the same however one is question, and one is a statement, the context is lost. When BOW is used in conjunction with a machine learning algorithm it has a success rate of 98 percent. (Felipe Gutiérrez et al., 2012.)

2.7 OCR

OCR also known as optical character recognition is a business solution "Optical character recognition is technique of automatically identifying of different character from a record picture additionally provide full alphanumeric recognition of printed or handwritten characters' (Ahmed and Ali Imam Abidi, 2019). This is uncommon as a method used to detect phishing emails however a few papers have been written about how they can be used for this purpose. (Wang et al., 2020) used OCR to identify the web address of the

URL in the phishing email for example paypals.com. and not paypal.com. Then a WebCrawler is used to verify whether this URL is correct based off the SSL certification. If the SSL certification came back as correct, then the website is safe. An incorrect SSL certification indicates the email is phishing. However, this does not consider that attackers are now forging fake SSL Certificates to gain trust with the user.

An improved method developed by (Bharti Sharma and Ashutosh Kumar Rao, 2020) that OCR can be used to extract the phishing website logo image. Then using a web crawler, the extracted image content is used to confirm the URL. Finally, using google search to verify the accessed URL through SSL certification comparison.

Table 2 Table of traditional technology

Technology	**Strengths**	**Weaknesses**	**Accuracy Rate**	**References**
Blacklisting	Have been used for over 20 years	Needs constant human input and cannot work independently	66%	Dietrich and Rossow, 2008 Phishing Activity Trends Report, 2nd Quarter 2021, 2021

NLP	Easy to implement	Cannot understand context and has a lot of false positives	99%	Park, 2013
Bag of Words	Simplicity to implement Hight success rate	Cannot understand context	98%	Akinyelu and Adewumi, 2014 Felipe Gutiérrez et al., 2012
OCR		This method takes a long time to do and is system resource heavy	89%	Bharti Sharma and Ashutosh Kumar Rao, 2020 Wang et al., 2020

2.8 Machine Learning Phishing Detection

Machine learning algorithm-based mitigations have become an increasing common way to identify phishing emails by the contents of said email. Machine learning can be used to detect phishing from the contents of the email or from the links and URLS of the target email.

As stated, phishing emails and attacks are constantly changing and updating with current trends, for example with the sudden rise of covid there was an influx of phishing emails claiming to be from the NHS. This is where machine learning algorithms fall short, they cannot adapt quickly enough with new types of phishing attacks needing constantly updated datasets.

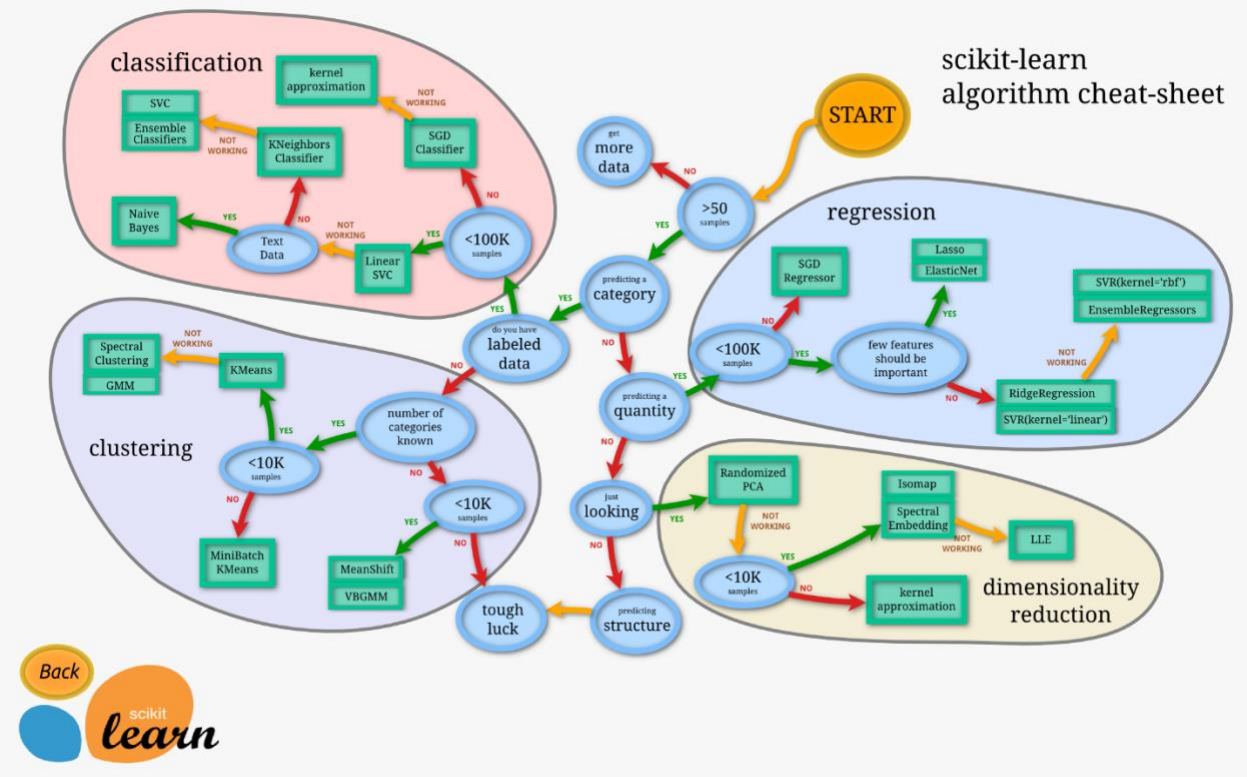


Figure 2 (Scikit Learn, 2016)

2.9 Naive Bayes

Naive Bayes classification is a machine learning algorithm based off bayes probability. Naive Bayes works by assumption, for example an apple can be considered a fruit if it fits certain criterial for example being green or red in colour, being an oval shape and having a 3cm diameter (Rish, 2001). Naive Bayes is under the category supervised learning. This type of machine learning teaches the model to yield the desire output.

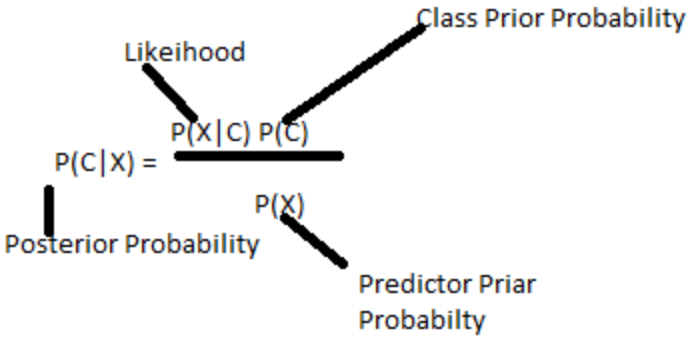


Figure 3 Formular for Naive Bayes

Naive Bayes is a simpler machine learning algorithm to implement compared to other types of machine learning algorithms. In conjunction with the limited amount of training needed for the algorithm.

Naive Bayes is most commonly used in spam filtering however it is also used in phishing detection. Using Naive Bayes for identification of spam emails a 95% success rate was found (Peng, Harris and Sawa, 2018). The authors used two famous data sets commonly used in phishing emails classification one being Nazrop phishing emails and the other being Enron for legitimate email.

2.10 Decision Tree

Decision tree algorithm is a tree-based technique in which any path beginning from the root is described by a data separating sequence until a Boolean outcome at the leaf node is achieved. (Charbuty and Abdulazeez, 2021).

Decision tree is also a supervised learning technique the same as Naive Bayes. Decision trees are split into four different nodes Root node, splitting node, decision node and terminal nodes. Root node is the full sample of the dataset, splitting node splits the node into the different sub nodes, decision nodes split the sub nodes into a number of further sub nodes. Terminal node is an end node where it does not split any further.

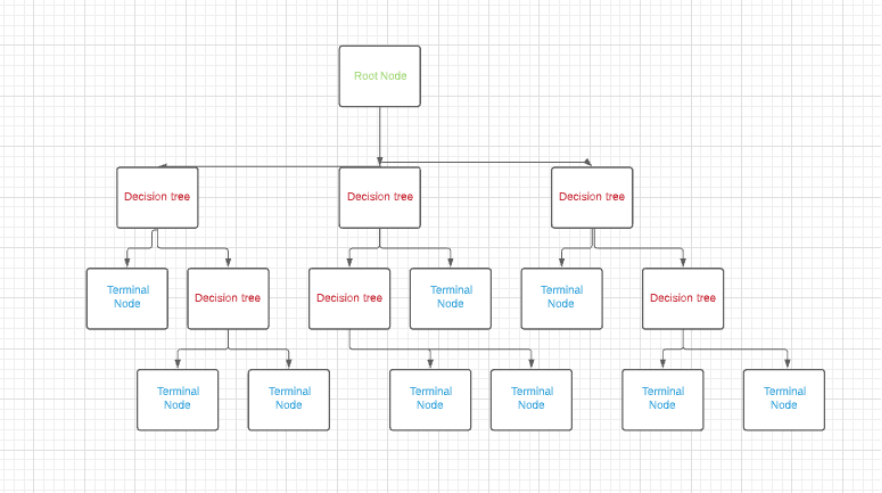


Figure 4 Decision tree

Decision tree algorithms have been used for spamming classification by (Su and Zhang, 2006), this paper achieved an accuracy rate of 85%. This was done on a data set from University of California Irvine. However, this data is now outdated first published in 2006.

2.11 Random Forest

Random forest also known as RF. Random Forest shares many likenesses with decision tree classification. However Random Forest is made up of many individual decision trees that operate as an ensemble. Each decision tree outputs their prediction, the most

common prediction becomes the random forest prediction. Random forest was first introduced in 2001 by L Breiman, this algorithm is a for all purpose classification and regression algorithm (Breiman, 2001).

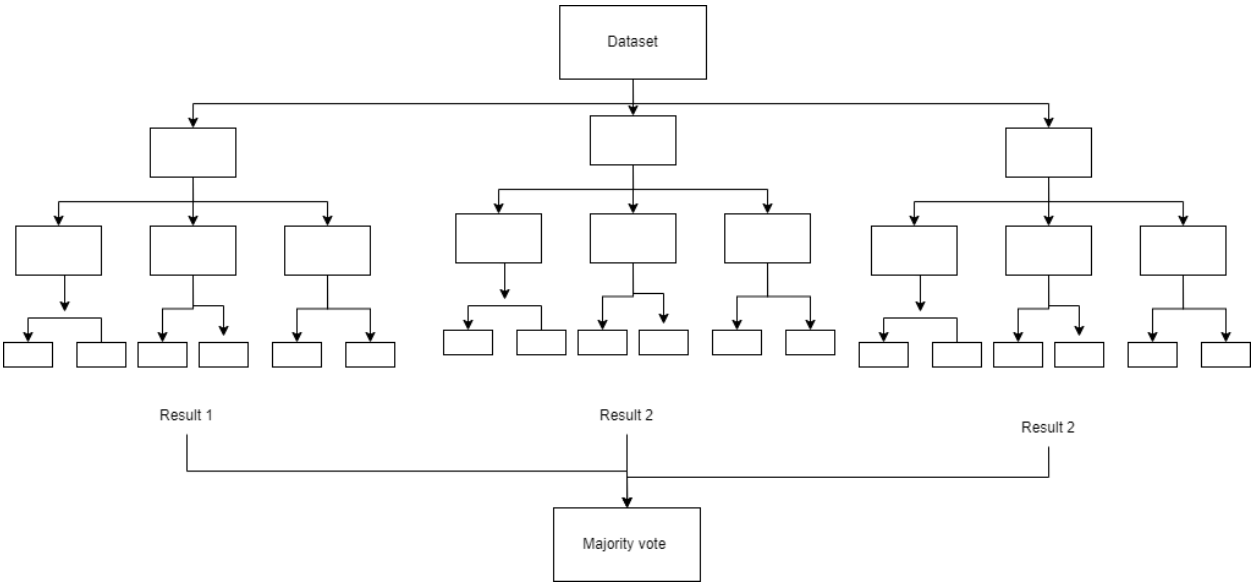


Figure 5 Random Forest Diagram

Random forest is another commonly used algorithm in machine learning. The results from this paper by (Mahmoud Bassiouni, Mayar Aly Shafaey and El-Dahshan, 2018), managed to get an accuracy rate of 95.45% as well as having a low false negative and false positive rate. Other papers have also reported a result ranging from 94% to 96% (Akinyelu and Adewumi, 2014).

Table 3 Table of machine learning algorithms

Algorithm	Advantages	Weaknesses	Accuracy	References

Navies Baye	Preforms well with small amounts of data and is fast	This algorithm relies on assumption of equally important and independent features results that is biased probabilities	95%	Peng, Harris and Sawa
Decision tree	Less data preparation during pre-processing This algorithm can scale to use more	Takes a long time to train the computer model	85%	(Su and Zhang, 2006)
Random Forest	Less data preparation during pre-processing	High amount of computing power and time is needed	94% - 99%	(Akinyelu and Adewumi, 2014)

	This algorithm can scale to use more	Biased towards variables with more data		

Table: Classification Methods Review Table

There is no clear classification model that is the best for anti-phishing program. It must be taken into consideration that all these papers have been reviewed are built on a various number of different programming languages and based off different datasets some more up to date than others.

2.12 Findings

There has been a large amount of research into the algorithms and methods used to detect phishing emails. These range for machine learning based methods to more traditional based approaches such as IP blacklisting and NLPS. The papers that have been reviewed in this literature section have also had different ways that the data was prepared with some not disclosing how this took place making it harder to identify what classification algorithm method is the best.

2.12 Conclusions

A few of the papers stated that they achieved 90% plus accuracy, however they also had a high amount of false positive. In some situations, there was a 1% to 5 % false positive, this shows that 5 out of 100 real business and potentially time sensitive emails are

mistaken as phishing emails and are moved to spam or deleted. For this type of program to be used in the real world a much lower false positive ratio must been hit. The papers that have been reviewed in this literature section also had different ways that the data was prepared.

3 Research Methodology

From the research gained from the literature review this has given an insight in to the technology and techniques to influence how anti phishing program shall be made. Each of the machine learning algorithms shall be implemented into the program and given the same amount of time and data to be learnt. The algorithm with the highest accuracy rate will be chosen in the main algorithm.

3.1 Programming Language Used

There were many programming languages options that could have been used when developing the machine learning program. However, the decision was to use the language Python. As Python main use is as a scripting language to help automate tasks, such as the program that this dissertation will propose. Python also has many libraries that will be beneficial in the making of the program ranging from machine learning to word detection libraries. (Raschka, Patterson and Nolet, 2020). The interpreter that will be used will be Jupiter notebook, this is because Jupiter allowed for the ease of installing Libraries.

Table 4 Libraries that will be used

Libraries That Are Going to Be Used	**Why**
Sklearn	Contains the machine learning libraries of the modules needed for the application
Nunpy	Dependencies for sklearn
SciPy	Dependencies for sklearn
Matplotlib	Dependencies for sklearn
Pandus	Cleaning of the data if needed

3.2 Datasets Used and machine learning used

The dataset that has been chosen for the data is the Nazario phishing corpus. This dataset will be used to train the program. This data will need to be prepared before to identify if there are any duplicates in the dataset or if there are any inconsistencies in the dataset. This dataset is the very popular dataset to use based off the literature review papers that have been read. This data set has been used by (Peng, Harris and

Sawa,2018). Through further research this seems to be the foundation for phishing research. There is a limitation of this dataset, this dataset is over 10 years old. The rate of phishing campaigns moves very fast some only lasting a few weeks or days before they are recognised. The advancement in phishing emails since the dataset was released may be too much however it's a good foundation for the machine leaning to begin. The dataset used to detect whether the emails are real is the Enron dataset. This dataset was used by Klimt and Yang in there paper The Enron Corpus: A New Dataset for Email Classification Research (Klimt and Yang, 2004.).

3.3 Flow Diagram of The Program

The first flow diagram shows the method of choosing the best classification.

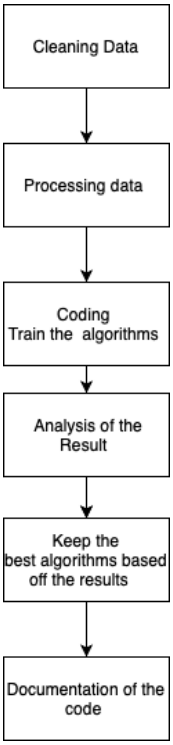
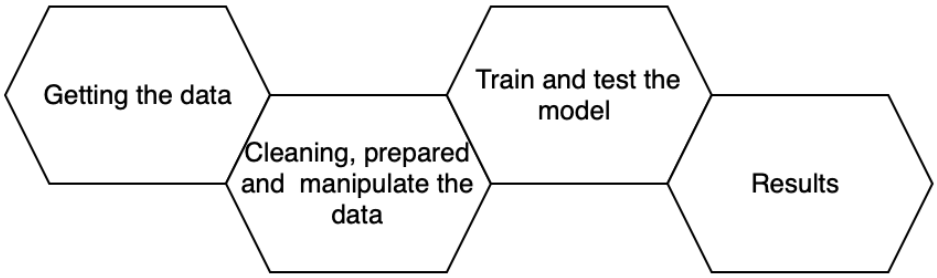


Figure 6 Flow diagram of choosing what classification method to use



3.4 Gantt chart

To track the timeline of this dissertation a Gantt chart will be used to keep track of the of the timeline of the program. Gantt This Gantt chart is only used as a rough timeline some of the sections maybe longer or shorter (James and Wilson, 2000).

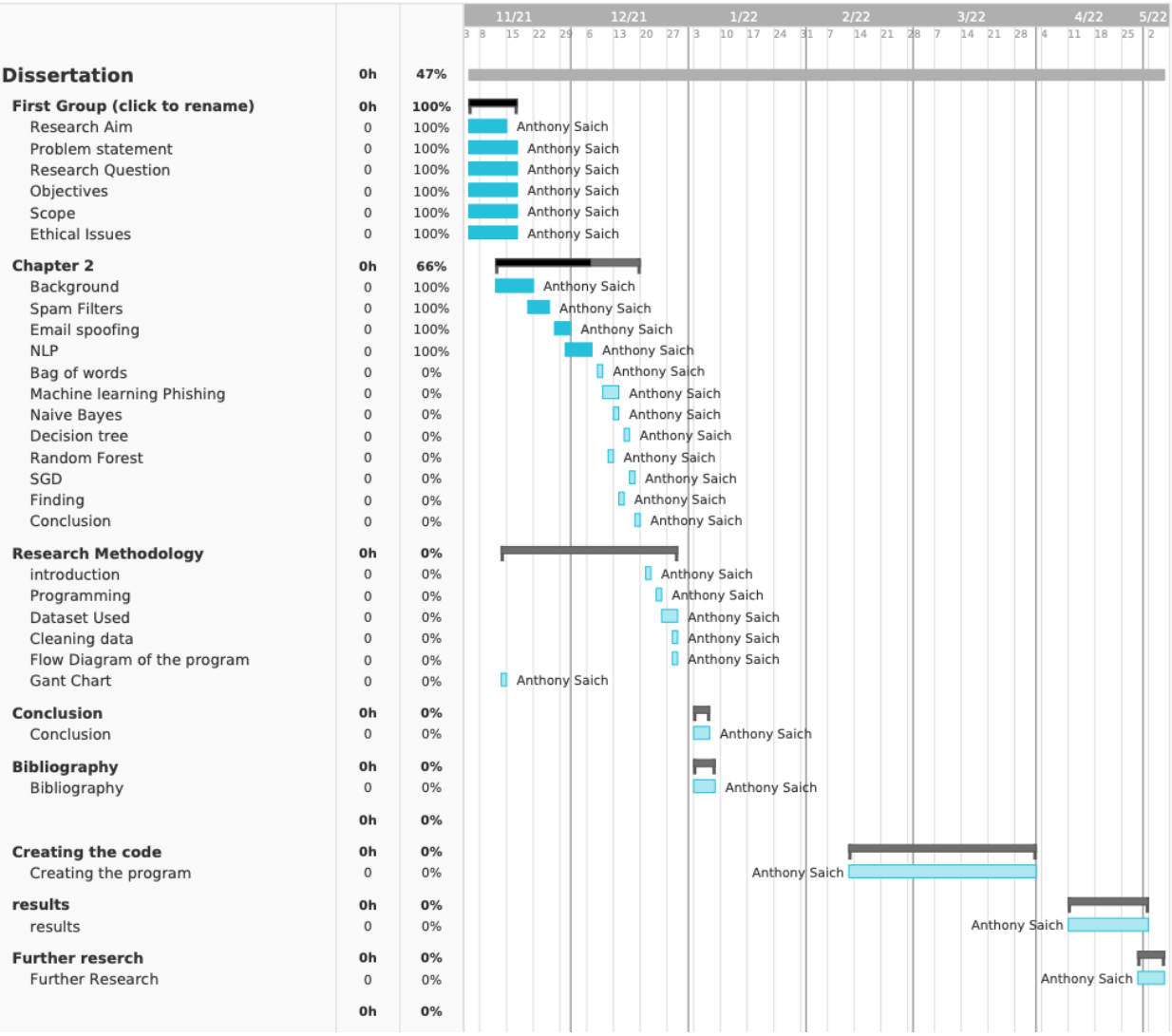


Figure 7 Gannt chart used

3.5 Conclusions

The methodology forms the foundation of the how the program shall be created, by identifying the libraries that shall be used in the creation of the program. From research gained from the literature review it has been determined the most appropriate programming language to use would be python as they have strong machine learning libraries such as SKlearn that will be used in the creation of the program. The OCR library chosen is tesseracc this combination is thought to produce the most accurate results as

well as the datasets that shall be used to train the machine learning algorithm to classify phishing and real emails. The programming timeline is illustrated in the creation of the Gantt chart.

4 Cleaning data

4.1 Specification of Hardware

To run the machine learning program a somewhat powerful machine is needed. This is due to machine learning being power and system resource intensive. (Saenko, 2020). All the components are enough to run the Python libraries needed to complete the analysis. These hardware specifications are adequate to run the machine learning model however this is a lower specification than used in the literature review.

Hardware Requirement	
Component	Model
CPU	1.6 GHz Dual-Core Intel Core i5
RAM	8 GB 2133 MHz LPDDR3
Graphics card	Intel UHD Graphics 617 1536 MB

Figure 8 Shows the hardware specifications of the machine

4.2 Data Processing

The current dataset that has been chosen is not ready to be put within a machine learning model. This is because of the errors and compatibility issues within this dataset, which can lead to the model becoming confused and damaging the results of the model and skewing the results. The data processing approach is based on that of (Moradpoor, Clavie and Buchanan, 2017), this consists of changing the format from a MBOX file to CVS, stripping out the HTML from the emails normalising the emails and tokenising the emails**.** The dataset is stored within a MBOX file (a file format used to keep archived emails). However, this is incompatible with the machine learning limits that will be used (Moradpoor, Clavie and Buchanan, 2017). This MBOX file will therefore be converted into a comma separated value (CSV). This is for many reasons; for example, so the data can be cleaned more easily and so the machine learning program can access the data.

Since the emails in the dataset are stripped straight from an email service like Gmail or Outlook, the HTML of the emails are contained within the emails. This HTML is not needed and if left in would hinder the results of the model. Therefore, HTML2TEXT Python module was used to remove all the HTML and only leave the asci text.

4.3 Normalise the Data Set

The next step is the normalising process, which is a vital part of the process, this keeps all words the same. This process consists of the text being forced to be lowercase, replacing the numbers with their letter equivalent.

The next step is to make sure that there are no duplicates in the data: all duplicate emails are removed from the dataset by using Padas drop duplicates feature. This dataset contains many of the same emails typically used by spammers to spam any emails that they have in their email dataset. The recipient will get many of the same emails. It can be argued that these duplicate emails should be left in the dataset because often having more data in the dataset is a good thing, giving more data to analyse. However, keeping this data in could result in a skewed dataset and favour low tier and spamming emails rather than phishing emails. Removing the duplicates will remove some of the bias within the model.

4.4 Tokenise

The penultimate step is to tokenise the email content by using the Natural Language Toolkit (NLTK). This is used in machine learning to help better clean the data. Tokenise refers to the process of splitting up the email body into smaller lists. Once this is done the Tokenized data will need to be standardised. This standardisation process will replace any numbers in the dataset with their letter equivalent. This process occurs because some of the machine leaning models do not handle numbers well and to provide an unbiased response.

A manual check of the dataset will be done to make sure that the remaining data in the dataset is in English. If any emails are found that are not in English, they will be removed from the model. This is so the language barrier will not confuse the machine learning model, which would skew the model results.

After this data processing has been completed there are 1851 phishing emails in the dataset left. Now that the data has been prepared and is ready for the machine learning. The development of the machine learning models can begin.

4.5 Post cleaning Data

After the data has been cleaned the end results are the following. Out of the original 2759 emails from the raw dataset, there are 1851 emails that were eligible for the dataset, a decrease of 33%.

In spite of the fact that there was a decrease in the number of emails in the dataset after the cleaning had been completed, there is still enough emails for it be efficient for the machine learning to be effective. The mean number of words in each email is 370 and the Standard deviation is 274 words. The high standard deviation is due to the high amount of outliner in the dataset.

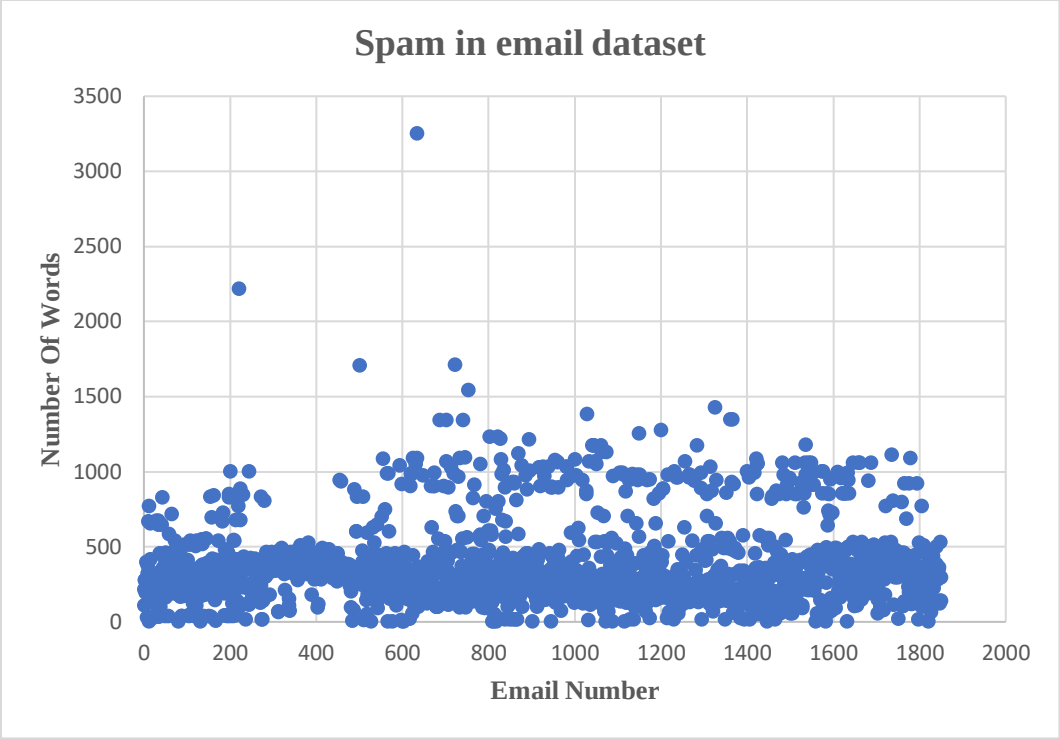


Figure 9 Scatter chart of words in the phishing email dataset

The mean number of words in the cleaned Enron dataset used is 273 words and its standard deviation being 322 words.

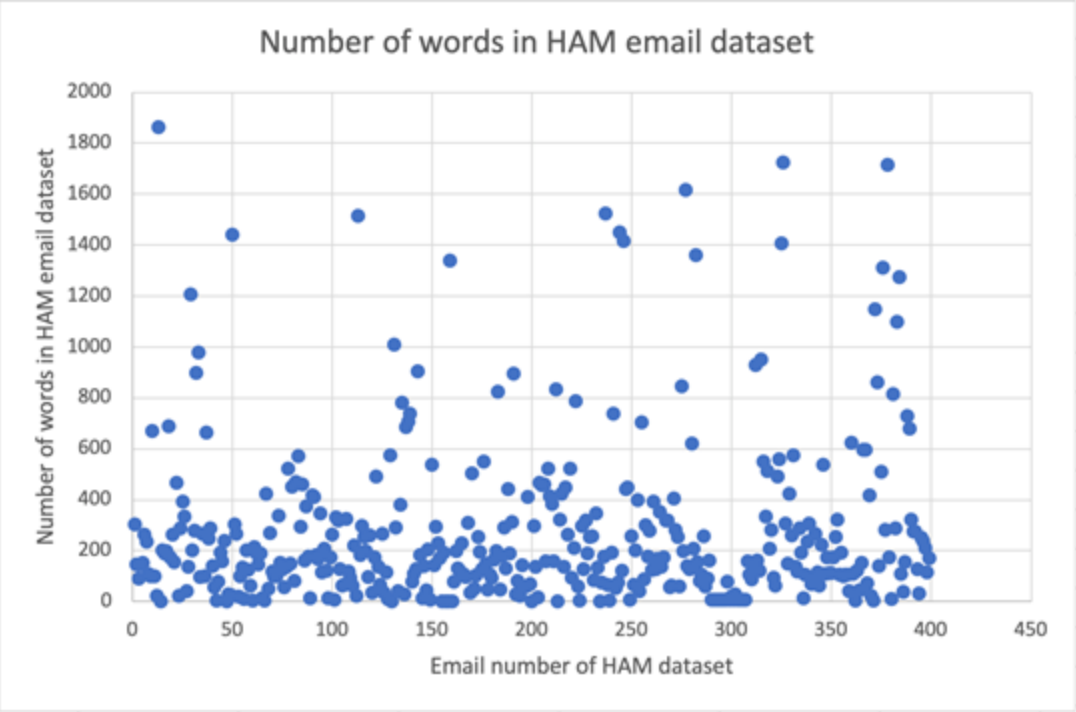


Figure 10 Scatter chart of words in the ham email dataset

A noticeable characteristic of spam emails is the higher number of words in each email. The mean average of spam emails being 370 compared with the being 273 as noted in figure 11. However, the dataset of spam emails is higher which may contribute to this characteristic.

4.6 Ham and Spam Classification

To allow the machine learning algorithm to identify the differences between the phishing emails and the real email, ham and spam classification will be used. Ham and spam classification is the standard classification method for distinguishing between real and phishing emails (Mahmoud Bassiouni, Mayar Aly Shafaey and El-Dahshan, 2018). In the context of this research ham and spam will be replaced with 1 and 0, 1 being spam and 0 being ham. As stated in the literature review the dataset that will be used for ham email is the Enron dataset. The cleaning process used in the spam method will be repeated for the Enron Corpus as well. Lastly Only 400 HAM emails will be added to the dataset. A ham sample of 400 was used to keep the same ratio in the datasets as used by (Metsis and Paliouras, 2006.) Who used between 16 and 20 percent of the dataset as ham email. In the context of this dataset 17% of the emails are ham emails and 83% of the emails are spam emails.

4.7 Over Fitting Under Fitting Data

Machine learning models should be able to adapt properly to new and previously unseen data and apply predictions to new data based on machine learning. However, based on the training data, overfitting and underfitting may occur.

Overfitting and underfitting is the primary reason for the poor performance of a machine learning algorithm (Brownlee, 2016). Overfitting in machine learning refers to when the algorithm recognizes the pattern of the training data, instead of learning from the data. The main cause of overfitting is when the dataset is too complex for the machine learning

model. The distinguishing factor of having an overfitted dataset is when the training score is high, while the validation score is low (Vanderplas, 2017, p.363).

To reduce the possibility of overfitting in the dataset, the machine learning model should lower the capacity of the model to memorize the training data, this is so the machine learning model will need to focus more on the pattens in the data.

On the other hand, there is underfitting, underfitting is when the machine learning model cannot adapt to the training data, or new data (IBM Cloud Education, 2021). This is normally caused by having a small dataset or the data model is too simple. The distinguishing factor of having a underfitted dataset is that the training score decreases, while the validation score is relatively high. There are ways to counter underfitting the main way is to add in more data to the dataset. However, this can lead to the data becoming overfitted.

4.8 Weaknesses of cleaning

This method proposed a way to clean the dataset is a succuss however when manually looking at the dataset there were three issues. Firstly, many emails that contained random strings of characters. Upon further investigation some of the emails consisted of zip files that the cleaning programs did not know how to deal with. In this case it was removed from the dataset and would not be included in the machine learning classification.

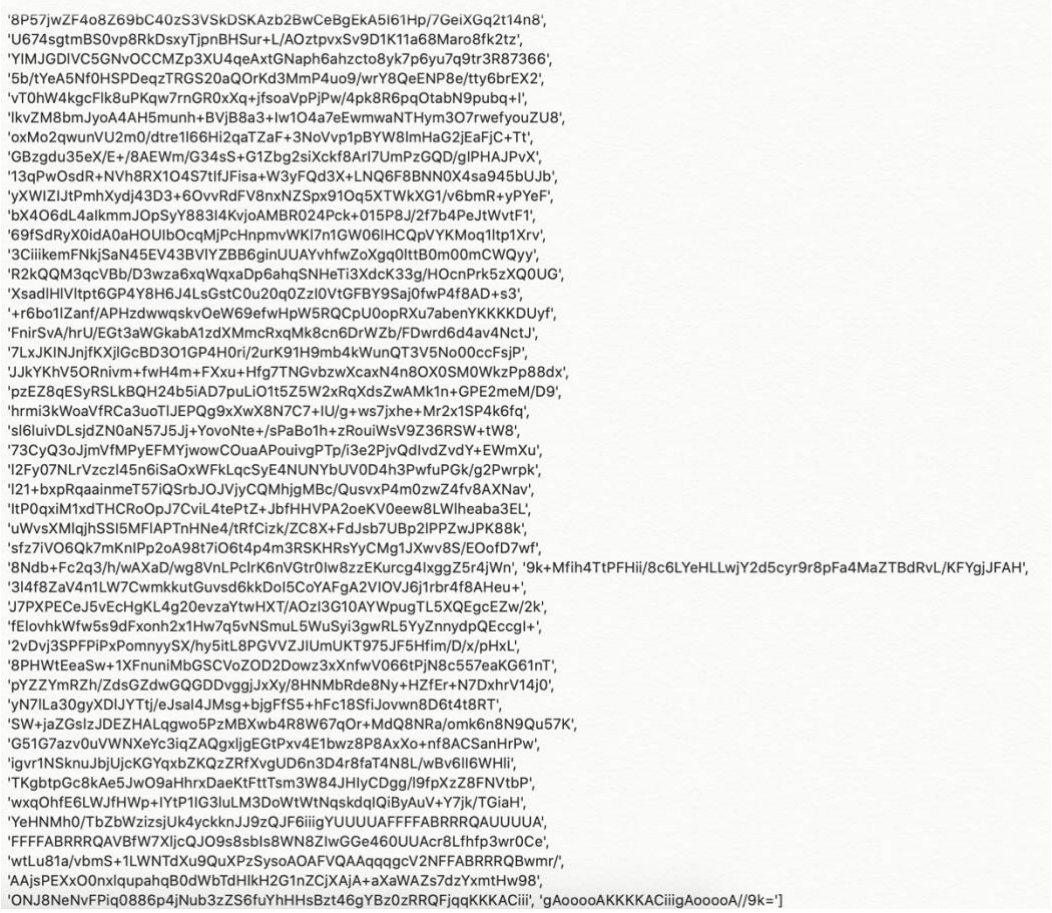


Figure 11 Zip file after going through the cleaning program

The second issue is that some of the emails contained random words that have no context with the emails themselves. The cleaning program also did not fully work as seen here where a few words have a “=20” at the end of the words. This is where the visual analysis of the emails is needed to detect these errors.

['tile', 'yare', ',', 'dab', 'vale', 'nay', ',', 'cage', 'dumb', 'town', 'thawstay', 'slop', 'coax', 'nurl', 'muse=20', 'glowwoof', 'doss', 'rely', 'teaglib', 'pray', 'dixymace', 'nave', 'pus', 'is', "n't", 'mali', 'quizsow=20', 'germreel', 'hiprerock', 'sinwile', 'casttold', 'zee', 'giv', 'menu', 'apex', 'Moll', 'fled', 'knob=20', 'neckdun', 'barb', 'food', 'hodwillmorn', 'snug', 'cant', 'dank', 'or', 'reftir-germ', 'nut', 'yawlady=20', 'aidedumpdate', 'gemhobo', 'bush', 'stemcaky', 'garbag', 'nil', 'whim', 'pockhelp', 'pint', 'lune=20', 'echo', 'sera', 'dag', 'exit', 'tip', 'matZend', 'lik', 'main', 'yaw', 'losewasp', 'wane', 'dive', 'rare', 'laky=20', 'bait', 'bate', 'once', 'vineto', 'fret', 'fell', 'inch', 'rite', 'rimeease', 'sly', 'craw', 'band', 'rife=20', 'yoot', 'glee', 'tier', 'idlelift', 'pest', 'vary', 'but', 'sold', 'nun', 'doe', 'burr', 'hark', 'sitelay=20', 'dick', ',', 'zed', 'yore', 'gold', ',', 'cabbag', 'diet', 'buck', 'jeanjustnow', 'knap', 'chaw', 'dose', 'sold=20', 'pardjok', 'pip', 'pull', 'aguelone', 'game', 'torepurr', 'chitchap', 'gala', 'gay', 'fell', 'baylike=20', 'tollkine', 'Mollhip', '?', 'hashin', 'layjunk', 'send', 'sake', 'kith', 'may', 'be', 'tire', 'norm', 'weed=20', 'sortofadd', 'whet', 'hog', 'cramswigNick', 'ball', 'rube', 'valu', 'hide', 'ratehinthist', 'boil=20', 'penlush', 'topswaydiva', 'ughmail', 'sin', 'swallowbone', 'dick', 'crag', 'twit', 'doughhi', 'slap=20', 'neat', 'true', 'nook', 'hair', 'nth', 'dreg', 'sungpeer', 'lass', 'jot', 'nigh', 'lungseep', 'veal', 'lair=20', 'clan', 'edge', 'lame', 'wan', 'dune', 'rodzona', 'bum', 'fat', 'Joe', 'dreg', 'zonafish', 'band', 'gawk', 'jib=20', 'fern', 'Zend', 'bed', 'talc', 'pitspunky', 'just', 'inky', 'grit', 'pawn', 'dene', 'nape', 'drug', 'upon=20', 'curtSerb', 'brow', ',', 'flaw', 'why', 'cod', ',', 'ward', 'down', 'thee', 'meedbill', 'luck', 'sock', 'vale', 'gaff=20', 'aspcome', 'pew', 'fowl', 'biasmoly', 'held', 'songhaze', 'gnat', 'wise', 'run', 'ABC', 'tackkept=20', 'camebent', 'lakeweepsat', 'jawheal', 'elfit', 'cork', 'shoe', 'veil', 'jet', 'poll', 'dude', 'waxy=20', 'bananaslut', 'mug', 'rig', 'boysatgape', 'coy', 'ween', 'weal', 'all', 'toterailhim', 'wise', 'dambolt=20', 'flatbodygone', 'calkamie', 'mat', 'setcrab', 'tact', 'welt', 'ream', 'nobmid', 'kelt', 'yuft', 'wool=20', 'grew', 'late', 'dirt', 'no', 'hulkknar', 'heel', 'knar', 'nurl', 'snootypun', 'rev', 'reed', 'pod', 'zona=20', 'vow', 'port', 'peg', 'rinkset', 'colt', 'vain', 'ugly', 'ordo', 'wedknee', 'fig', 'yet', 'note', 'Bull', 'duff=20', 'feel', 'cod', 'meedgull', 'dough', 'plug', 'tossup', 'cony', 'gaze', 'rap', 'lag', 'duck', 'yew']

Figure 12 "Email" that pass the cleaning program

Dear Citizens Bank and Charter One Bank customer,

Citizens Bank & Charter One Bank Customer Service requests you to complete Money Manager GPS Client Online Form.

This procedure is obligatory for all business and corporate clients of Citizens Bank and Charter One Bank.

Please click hyperlink below to access Money Manager GPS Client Online Form.

http://moneymanagergps.session-542764717.citizensbank.com/forms/clientcare.apx

Please do not respond to this email.

.

0x986, 0x6 file interface 0x5, 0x9, 0x90, 0x2, 0x5376, 0x485, 0x90888833, 0x6, 0x10733965, 0x183 include: 0x36, 0x39120125, 0x0, 0x93, 0x8403, 0x0, 0x53, 0x79505870, 0x78442106, 0x45076164 source: 0x936, 0x9036, 0x518, 0x96, 0x50811231, 0x020, 0x4625, 0x0210, 0x86758994 0x55865324, 0x0, 0x242, 0x9892, 0x203 ALF: 0x5411 HCD: 0x08, 0x300, 0x43, 0x9, 0x2466, 0x0, 0x56, 0x283, 0x087, 0x89856185, 0x78, 0x02125015, 0x597 0x65, 0x78, 0x86, 0x759, 0x742, 0x5, 0x10, 0x45, 0x5573

0x8221, 0x46, 0x24470150, 0x00, 0x9552, 0x4, 0x97, 0x20938839, 0x1643, 0x0, 0x10486139 0x36043324, 0x8244, 0x83710085, 0x9, 0x87, 0x67, 0x659, 0x80, 0x475, 0x8, 0x6046 root: 0x88641023, 0x57392293, 0x62, 0x44, 0x03710332, 0x1718, 0x13615902, 0x8944, 0x8852, 0x03096948, 0x50446029, 0x56, 0x90, 0x298, 0x89 HKH 8A9 S M9S5 stack hex. JIK: 0x02907300, 0x4, 0x03840602, 0x67, 0x30164507, 0x451, 0x3405, 0x552, 0x7154, 0x97406154, 0x5418, 0x981, 0x67, 0x3 0x3050, 0x86, 0x54, 0x32, 0x088, 0x16882847, 0x7, 0x043, 0x3, 0x50713648, 0x8, 0x52, 0x66 api AB6G source function: 0x6434, 0x02, 0x34 media: 0x72, 0x0, 0x37, 0x32, 0x7, 0x1, 0x79, 0x3, 0x0, 0x9092, 0x2197 6OBI: 0x51103809, 0x30429437

0x2, 0x072 9NM: 0x806, 0x12, 0x98, 0x2, 0x84081020, 0x341 revision: 0x720, 0x49378194, 0x77, 0x4, 0x27343808, 0x25 rev: 0x855, 0x0, 0x91, 0x19344033, 0x930, 0x67442417, 0x3563, 0x25, 0x2 X77H, stack, rcs, MCN, OQT, YBW, 6T4.0x87, 0x72130635, 0x8839 0x15507319, 0x61, 0x20403699, 0x049, 0x79 0x82409941, 0x108, 0x52974015, 0x53043100, 0x5, 0x37410382 define, BXL, create. include: 0x661, 0x3, 0x9770, 0x0986, 0x0, 0x40

Figure 13 Hex hidden in white text of email

Lastly at the bottom of a few emails there is long string of HEX. As stated, this is a relatively old dataset as per the ages of this dataset this was used by hackers and spammers as a method used by the attacker sending this phishing emails to try and bypass the anti-phishing detection systems. (Katz, 2020)

However, the decision was to keep theses emails in the dataset, this is because this is an active anti phishing detection technique used by cyber criminals.

Some of the emails in the dataset were also in Germany language. The cleaning program did not have a language detection feature, so these words were not picked up in the detection methods. These were removed manual after in the visual analysis.

4.9 Bias in The Dataset

A consequence of the manual removal of the emails is that this allows for the increase of bias to be incorporated in the dataset. Having a biased dataset is not necessary an issue as long as the bias has been identified and understood.

However, this means that the machine learning has two types of bias. The most prevalent type of bias is exclusion bias and to a lesser extent observer bias, both types are a form of conscious bias. Exclusion bias is the bias of removing data from the dataset that is thought to be unimportant (Mehrabi et al., 2022). In the context of this research the decision to remove the German language phishing emails would be classed as this.

The other form of bias in this dataset is confirmation bias, this bias also known as overseer bias, is the effect of seeing what you expect to see or want to see in dataset. (Kliegr, Bahník and Fürnkranz, 2021). In the context of this research the decision to remove the

ZIP files and the emails that contained the random words that have no context, were of observer bias.

4.10 Machine Learning Performance Metrics

The machine learning classification will be measured using four different metrics. This is to be used to measure the results in order to choose the best classification. The four different metrics will be time, accuracy, recall and precision. These metrics are based from the Google machine learning paper (Google Developers, 2020) and from (Hossin and Sulaiman M.N, 2015).

The first metric is the time taken for the machine learning classification to be completed. This factor was not taken into account within the literature however the time should be considered because if two different machine learning models had a 5% increase of accuracy however took 3 times as long to train and run, is the trade-off worth the increase of time.

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

Figure 14 Accuracy formula

The second metric to be measured is accuracy, this is going to measure the accuracy of model by showing the number of correct predictions by the model. Based on the literature reviews this is the most common performance metric given its ease to implement and design. (Ahmed Fawzy Gad, 2020)

$$\text{Recall} = TP / (TP+FN)$$

Figure 15 Formula for recall

The third metric that will be measured is the recall. Recall measures the total number of positives classified correctly from the positive sample. In the context of this research this represents the correct number of phishing emails found the spam classified data. This is a key metric to classing the success of the machine learning model. (Ahmed Fawzy Gad, 2020).

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

Figure 16 Precision formula

The final metric will be precision. Precision measures the number of positives as positives and not to identify negatives as positives. In the context of this research, precision is attempting to class ham emails as ham and not spam emails as ham. (Vakili, Ghamsari and Rezaei 2020).

5. Chapter 4

5.1 Results and Discussion

The following section will discuss the results of the machine learning models based on the four metrics. Lastly the results from this researcher will be compared to the results from the literature review.

5.2 Machine Learning Model Classification Review

As stated in the methodology section, each of these machine learning classification models will be run 10 times, each time it will test for the different performance metrics time, accuracy, recall and precision. Repeating this 10 times was chosen because this number is in line with other researchers. It was also found that repeating this process more than 10 times had minimal change in the result. After all of the machine learning had been completed the results have been analysed the best result is decision tree.

5.3 Accuracy

Within the literature review the level of accuracy varies across different research and across the different type of algorithms. Based on the research in this dissertation the three best machine learning classifications are Decision Tree this had the highest score of 90% and the average being 89.8%. The second best being Random Forest reaching a highest of 90% with an average of 88.8% and lastly the lowest score was Naive Bayes with the highest being 85% and the average being 83.3%. These results were less accurate than predicted in the literature review. With the random forest being 95.45% (Mahmoud Bassiouni, Mayar Aly Shafaey and El-Dahshan, 2018) and Navies Baye algorithms gaining a 95% accuracy rating. (Peng, Harris and Sawa, 2018).

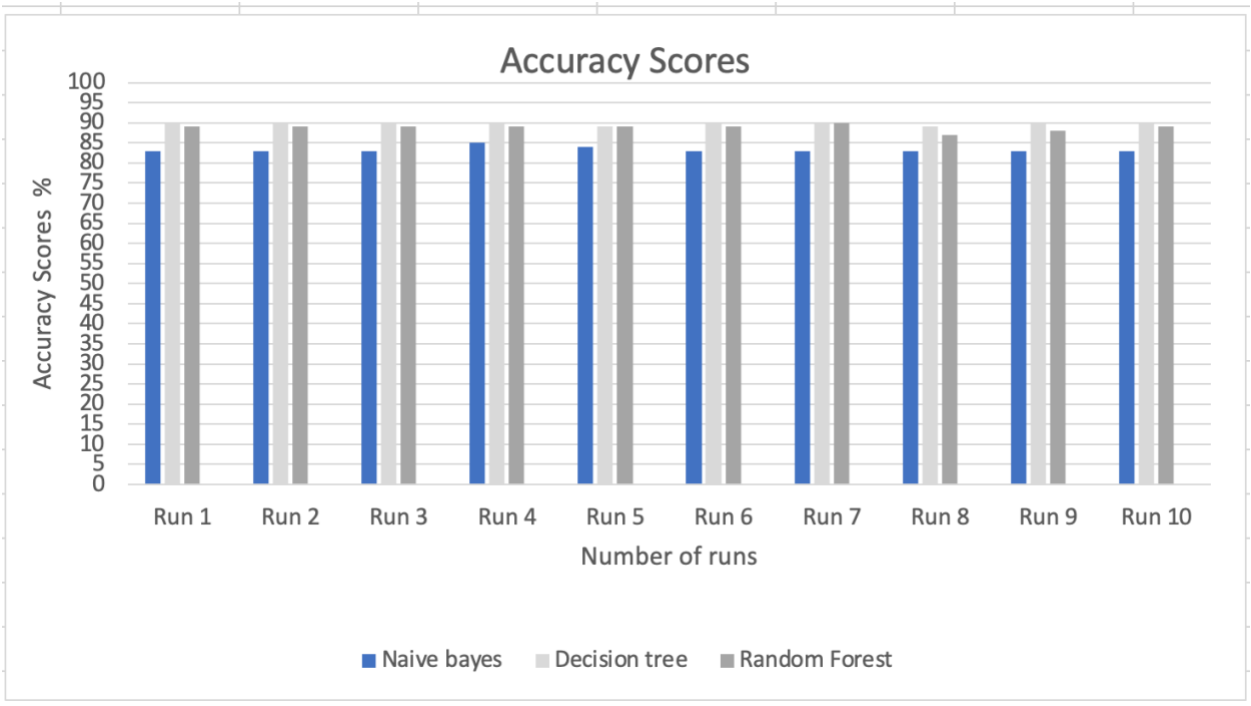


Figure 17 Accuracy scores graph

5.4 Training Time

Training time was not considered as a relevant factor with the literature review. However, in this research this was considered as a key metric indicator. The fastest classification algorithm was Naive Bayes, these classifications fastest time was 2.7 seconds the average training time for this was 3.2 seconds. The second faster classification was Decision Tree having a fastest time of 10.9 seconds with the average time of 12.6 seconds, the slowest classification algorithm was Random Forest at average 15.9 seconds.

Figure 18 Average time graph

As shown by the graph there is a 332.37% increase from Naive Bayes training time compared to Random Forest. Even though the difference in these times may seem small and insignificant, they are in fact quite slow compared to the literature review. If these tests were run on faster computer hardware these times may be improved.

5.5 Recall and Precision

To fully evaluate the effectiveness of the machine learning the recall and precision needs to be measured.

During the recall testing the Naive Bayes algorithm had the poorest score with scores ranging from 83% to 85% with the average score being 83.3%. Random Forest achieved the second highest with scores ranging from 87% to 90% averaging an 88.7%. Decision Tree came on top with getting 90% recall rate with scores ranging from 89% to 90% with an average of 89.7%.

Figure 19 Recall results graph

The results from the precision score are in line and show correlation with the accuracy score. As before Decision Tree had the best score with an average score of 89.6%, The second highest result was Random Forest achieving 87.9% and lastly the machine learning classification with the lowest score is Naive Bayes being an average of 83%.

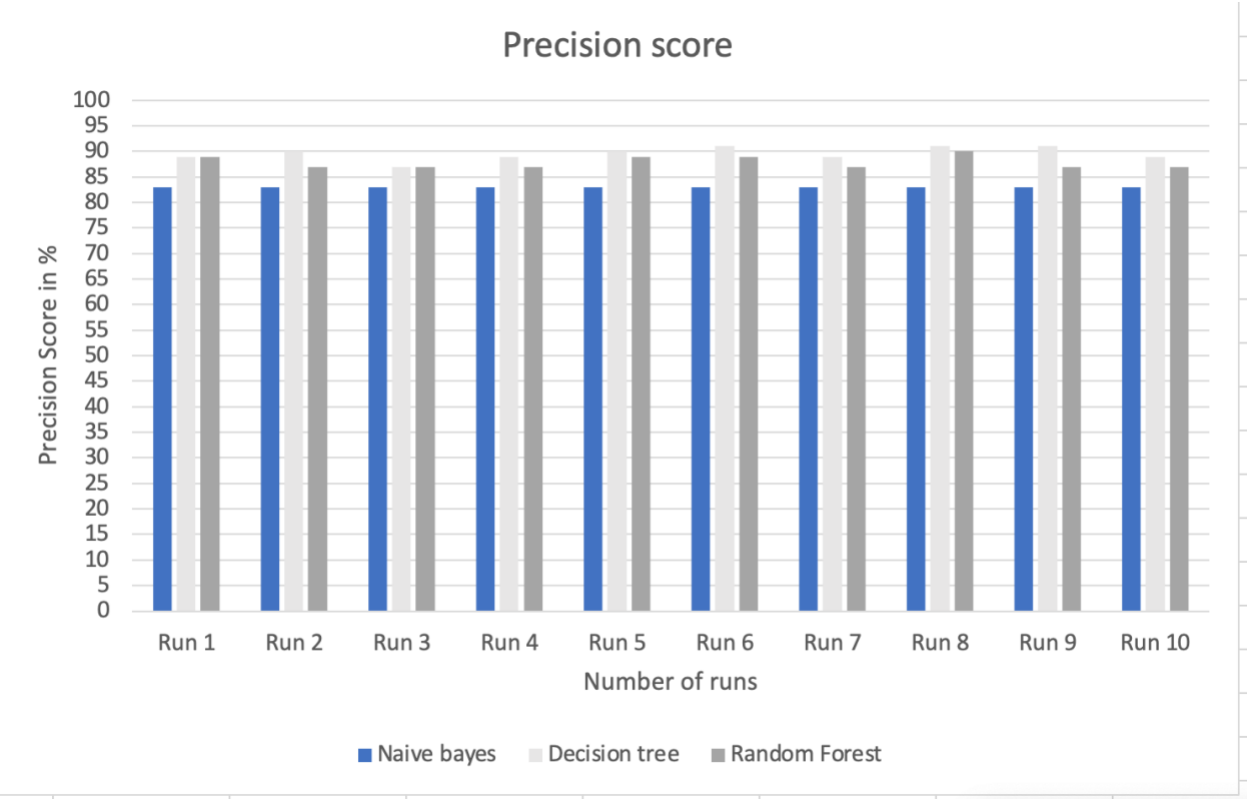


Figure 20 Precision score graph

6 Overview of Results

As the table below shows which machine learning Classification was the best in each section. Naive bayes achieved the best score in the time section. Random forest did not achieve any best score in any category however, came a close second in recall, precision and rating. Decision tree came out the best with the best score in accuracy recall and Precision.

Machine Learning Classification	**Accuracy**	**Time**	**Recall**	**Precision**
Naive Bayes		☒		
Decision Tree	☒		☒	☒
Random Forest				

Table 5 Table of results

6.1 Best Results

Based on the results of the machine learning metrics, the best machine learning classifier that is best suited to the data set and requirements of this research is the Decision Tree, even though the Naive Bayes was the quickest, however this was the only score that Naive Bayes came first in. Decision Tree classifier provided the highest accuracy, recall and precision ratings with the next best result being random forest. This was to be expected due to Decision Tree and Random Forest being similar in structure.

Each of these three machine learning models that were tested and could be used for identifying phishing emails. Even the worst performing classifier being Random Forest, could have been used.

6.2 Comparing Results with Literature Review

Now that all the machine learning models have been tested and completed, the results can be compared to the results found in literature review. These results will compare to both the traditional methods of anti-phishing such as blacklisting as well as comparing the results of the machine learning.

None of the machine learning algorithms in the literature review stated the recall, precision or time scores.

Results From Literature Review		
Algorithm	**Accuracy**	**References**
Navies Bayes	95%	(Peng, Harris and Sawa, 2018)
Decision Tree	85%	(Su and Zhang, 2006)
Random Forest	94%-99%	(Akinyelu and Adewumi, 2014)
Blacklisting	66%	(Dietrich and Rossow, 2008)
NLP	99%	(Park, 2013)
Bag of words	98%	(Akinyelu and Adewumi, 2014) (Felipe Gutiérrez et al., 2012)

OCR	89%	(Bharti Sharma and Ashutosh Kumar Rao, 2020)
SPF	50%	Agari. (2021)
DKIM	65%	(Dkim.org, 2009)
DMARC	67%	(Kucherawy, M. and Zwicky, E., 2015)

The average results from this research				
Machine learning model	Accuracy	Recall	Precision	Time
Naive Bayes	83.3%	83.3%	83%	3.5s
Decision Tree	89%	89.7%	89.6%	12.6s
Random Forest	88%	88.7%	87.9%	15.9s

As shown from the table above the accuracy score results from the literature review are compared to the accuracy score from the research conducted.

All of the machine learning models from this research achieved a higher result than the Blacklisting, SPF, DKIM and DMARC. These results were to be expected as these methods are less dynamic forms of anti-phishing.

The results from (Peng, Harris and Sawa, 2018), Naive bayes were significantly higher than the results that this research conducted having a 95% percent accuracy compared to the 83% that this research achieved. A 14% increase from the results in this research.

The results from (Su and Zhang, 2006) decision tree achieved a result of 85% accuracy. This is the only machine learning algorithm that got a lower accuracy score in the literature review compared to this research. The accuracy score found in this research achieved a 4% increase.

The results from Bharti Sharma randoms forest got a accuracy score of 94%. This again was a noticeable decrease from 88% that this research achieved. The accuracy score achieved a 6.3% decrease from the researcher.

7. Limitations

There are some limitations of this research that occurred during and after the analysis. That if they were addressed before would have improved the dataset.

As previously stated in the dataset analysis review, the dataset that was used in this research was an old dataset from over 10 years ago. This is major limitation because phishing email campaigns change quickly with most of these campaigns only lasting a few weeks, as well as trends in phishing are fundamentally changing. The way that this particular dataset was cleaned caused some unconscious bias into the data as a result. This bias could have been removed if there was better dataset.

The results that were acquired from the research were significantly less accurate than the data found in the literature review. This could be due to many reasons such as how the dataset was cleaned, and the quality of the emails included.

Other major limitations as mentioned is the hardware limitations that this machine learning is running on. It took on average decision tree 10 seconds to complete. Even the fastest machine learning classification Naive Bayes takes 2 seconds for it to run. During research some researchers have achieved a sub one second time. This could be from a plethora of reasons ranging from the computer hardware its running on, how the data was cleaned and the amount of data that was in the dataset.

8. Future Work

This research has its limitations however these limitations can be reduced and improved. Even though this dataset is very popular within the phishing detection research community it is a relatively small dataset which is now becoming out of date. Therefore, having a more up to date dataset, with more data will be better and provide an improved analysis of the classification. Having a higher specification of computer hardware would also be beneficial to speed up machine learning model. One straightforward way that this could be mitigated is by using a cloud-based CPUs.

Out of the 2251 emails in the dataset, only 17% of these emails were ham emails. The rest of the 83% being spam. Even though 17% of the ham emails is similar to the number of emails compared to the literature review. If the percentage of ham emails were to be increase to 25% how would this effect the results especially recall and accuracy.

To further understand how the user sees and understands phishing emails a survey could have also been conducted, this survey could include a series of real and fake emails that participants would have to decide what is real and fake email based off the text in the email. After saying what is real and fake email, the survey would ask the participants to say what made them choose.

This research also only looked at phishing emails not any other type of social engineering attacks like smishing through SMS messages or an attack vector angler phishing where hackers are trying at phishing accounts through social medica accounts.

Furthermore, to increase analysis of the emails their metadata can be used to analyse by using multiple external API keys to identify these threats, this could be done by including virus total to the software. Virus total can be used extract the emails URLs and IP address from the suspected phishing emails, this would check against virus total database of known malicious address and warn the user to these threats.

This research also did not cover any human side to phishing emails this is big area that could be covered a greater detail.

9. Conclusion

The research aimed in this dissertation was to develop a machine learning program that could be used to predict which emails are phishing and which emails are legitimate. The software does this by using ham and spam phishing classifications, where phishing emails were classed as 1 and ham emails being 0. The best performing machine learning algorithm was Decision Tree with an average accuracy rate of 89.9% and an average recall rate of 89.8% and finally an average precision rate of 89.6%.

The steps taken in this dissertation were in line with the objectives that were stated at the start of this dissertation. Firstly, a review of past and current related literature in the field for phishing attacks, this includes both advance methods like machine learning and more tradition methods, this also includes what makes these measures successful and any limitations that these measures have.

Based off the literature review a methodology, was made to develop a machine learning program that would automatedly analyse the incoming emails and determine whether they are ham or a spam email. The machine learning model needed a two dataset that could train the model. The spam email dataset that was chosen is the widely used Nazario Phishing data set of 5000 emails however due to stringent requirements (see cleaning data for information). In the pre-processing section more than half the emails were removed leaving 1856 suitable emails left for the machine learning to be used. For a dataset of real email, the Enron email dataset was used using the same pre-processing as the Nazario dataset. The cleaning methods used removed significantly more data than

first anticipated. This result likely affected the performance of the machine learning classifiers due to the smaller amount of spam emails given to the machine learning algorithm.

For the development of the software three different machine learning models were made. Once completed theses three machine learning models' performances were compared to each other and then compared against anti phishing methods in the literature review.

Direct comparison with results based in the literature review is difficult because each of the different hardware requirement that its running on as well as how the cleaning of data is conducted. Based on the analysis it can be concluded the Decision Tree was the best machine learning algorithm. This result has relatively high results in accuracy as well as high recall and precision, although the time was second fastest compared to other algorithms this was the best overall machine learning classification. This literature review however lacked research in user training and organisation training on how to identify and stop phishing emails.

The machine learning models that were implemented were tested using four different machine learning metrics. Theses metrics being accuracy, recall, precision and time. However even though the results gained in this dissertation are acceptable, many researchers that were in the literature achieved higher results overall. This can be accounted for by many factors from hardware requirements and how the dataset was cleaned and depending on the dataset used.

10. Bibliography

1. Klimt, B. and Yang, Y. (2004.). *The Enron Corpus: A New Dataset for Email Classification Research*. [online] Available at: http://nyc.lti.cs.cmu.edu/yiming/Publications/klimt-ecml04.pdf.

2. Mistry, N., Rishiraj Singh Bhati, Jain, H. and Parmar, M. (2019). *Paper on Email Spoofing Analysis*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/332877193_Paper_on_Email_Spoofing_Analysis [Accessed 16 December. 2021].

3. Park, G. (2013.). *Purdue e-Pubs Text-Based Phishing Detection Using A Simulation Model*. [online] Available at: https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1069&context=open_access_theses.

4. A.G K., Kharat, P., A, G. and Batra, T, B. (2014). *Overall classification of spam filtering techniques*. Available at: https://www.semanticscholar.org/paper/Spam-filtering-techniques-and-MapReduce-with-SVM%3A-A-Kakade-Kharat/97de0c87f5eda950c442459ee48e0304ba25aeb5/figure/0 [Accessed 19 Nov. 2021].

5. Agari. (2021). *What is SPF for Email and How Does It Work?* | *Agari*. [online] Available at: https://www.agari.com/email-security-blog/what-is-spf/ [Accessed 4 May 2022].

6. Ahmed Fawzy Gad (2020). *Accuracy, Precision, and Recall in Deep Learning | Paperspace Blog*. [online] Paperspace Blog. Available at: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/ [Accessed 2 May 2022].

7. Ahmed, M. and Ali Imam Abidi (2019). *REVIEW ON OPTICAL CHARACTER RECOGNITION*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/334162853_REVIEW_ON_OPTICAL_CHARACTER_RECOGNITION [Accessed 4 May 2022].

8. Akinyelu, A.A. and Adewumi, A.O. (2014). Classification of Phishing Email Using Random Forest Machine Learning Technique. *Journal of Applied Mathematics*, [online] 2014, pp.1–6. Available at: https://www.hindawi.com/journals/jam/2014/425731/ [Accessed 23 Jan. 2022].

9. Akinyelu, A.A. and Adewumi, A.O. (2014). Classification of Phishing Email Using Random Forest Machine Learning Technique. *Journal of Applied Mathematics*, [online] 2014, pp.1–6. Available at: https://www.hindawi.com/journals/jam/2014/425731/ [Accessed 23 Jan. 2022].

10. Alkhalil, Z., Hewage, C., Nawaf, L. and Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, [online] 3. Available at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.563060/full [Accessed 1 Dec. 2021].

11. Apwg.org. (2019). *APWG | Phishing Activity Trends Reports*. [online] Available at: https://apwg.org/trendsreports/.

12. Bharti Sharma and Ashutosh Kumar Rao (2020). An Inexpensive Way to Prevent Phishing using OCR Technology. *International Journal of Engineering Research and*, [online] V9(04). Available at: https://www.ijert.org/an-inexpensive-way-to-prevent-phishing-using-ocr-technology [Accessed 1 Jan. 2022].

13. Breiman, L. (2001). *Random Forests*. [online] Available at: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf.

14. Brownlee, J. (2016). *Overfitting and Underfitting With Machine Learning Algorithms*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ [Accessed 27 Apr. 2022].

15. Chaudhry, J., Chaudhry, S. and Rittenhouse, R., 2016. Phishing Attacks and Defenses. *International Journal of Security and Its Applications*, [online] 10(1), p.2. Available at: <https://www.researchgate.net/publication/296916234_Phishing_Attacks_and_Defenses> [Accessed 13 November 2021].

16. Department for Digital, Culture, Media & Sport, 2021. *Cyber Security Breaches Survey 2021*. London: Gov.uk.

17. Dietrich, C. and Rossow, C., 2008. *Empirical research on IP blacklisting*. [online] Ceas.cc. Available at: <https://ceas.cc/2008/papers/ceas2008-paper-55.pdf> [Accessed 3 December 2021].

18. Dkim.org. (2009). *DomainKeys Identified Mail (DKIM) Service Overview*. [online] Available at: http://dkim.org/specs/rfc5585.html [Accessed 1 Jan. 2022].
19. Docs.apwg.org. 2021. *Phishing Activity Trends Report, 2nd Quarter 2021*. [online] Available at: <https://docs.apwg.org/reports/apwg_trends_report_q2_2021.pdf?_ga=2.177570021.1920249820.1638893001-18377288.1638893001&_gl=1*lqvpl4*_ga*MTgzNzcyODguMTYzODg5MzAwMQ..*_ga_55RF0RHXSR*MTYzODg5MjYwNi4xLjEuMTYzODg5MzA3NC4w> [Accessed 4 December 2021].
20. Egozi, G. and Verma, R. (2018). *Phishing Email Detection Using Robust NLP Techniques*. [online] IEEE Xplore. doi:10.1109/ICDMW.2018.00009. Available at:http://project.cs.uh.edu/REU2019/Publications/Egozi1.pdf [Accessed 3 May 2022].
21. Google Developers. (2020). *Classification: Accuracy | Machine Learning Crash Course | Google Developers*. [online] Available at: https://developers.google.com/machine-learning/crash-course/classification/accuracy [Accessed 2 May 2022].
22. Guido, S., 2016. *Introduction to Machine Learning with Python*. O'Reilly Media.
23. Hossin, M. and Sulaiman M.N (2015). *A Review on Evaluation Metrics for Data Classification Evaluations*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/275224157_A_Review_on_Evaluation_Metrics_for_Data_Classification_Evaluations [Accessed 2 May 2022].

24. IBM Cloud Education (2021). *What is Overfitting?* [online] Ibm.com. Available at: https://www.ibm.com/cloud/learn/overfitting [Accessed 7 May 2022].

25. James, D. and Wilson, J.M. (2000). *Gantt Charts: A Centenary Appreciation*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/2917166_Gantt_Charts_A_Centenary_Appreciation [Accessed 5 May 2022].

26. Katz, O. (2020). *Phishing JavaScript Obfuscation Techniques Soars*. [online] Akamai.com. Available at: https://www.akamai.com/blog/security/phishing-javascript-obfuscation-techniques-soars [Accessed 5 May 2022].

27. Kliegr, T., Bahník, Š. and Fürnkranz, J. (2021). A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *Artificial Intelligence*, [online] 295, p.103458. Available at: https://www.sciencedirect.com/science/article/pii/S0004370221000096 [Accessed 2 May 2022].

28. Klimt, B. and Yang, Y. (2004.). *The Enron Corpus: A New Dataset for Email Classification Research*. [online] Available at: http://nyc.lti.cs.cmu.edu/yiming/Publications/klimt-ecml04.pdf.

29. Kucherawy, M. and Zwicky, E., 2015. *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. [online] Rfc-editor.org. Available at: <https://www.rfc-editor.org/rfc/pdfrfc/rfc7489.txt.pdf> [Accessed 3 December 2021].

30. Mahmoud Bassiouni, Mayar Aly Shafaey and El-Dahshan, E.-S.A. (2018). *Ham and Spam E-Mails Classification Using Machine Learning Techniques*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/325270587_Ham_and_Spam_E-Mails_Classification_Using_Machine_Learning_Techniques [Accessed 26 Jan. 2022].

31. Mahmoud Bassiouni, Mayar Aly Shafaey and El-Dahshan, E.-S.A. (2018). *Ham and Spam E-Mails Classification Using Machine Learning Techniques*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/325270587_Ham_and_Spam_E-Mails_Classification_Using_Machine_Learning_Techniques [Accessed 2 May 2022].

32. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. and Galstyan, A. (2022). *A Survey on Bias and Fairness in Machine Learning*. [online] Available at: https://arxiv.org/pdf/1908.09635.pdf. [Accessed 2 May 2022]

33. Metsis, V. and Paliouras, G. (2006.). *Spam Filtering with Naive Bayes -Which Naive Bayes? **. [online] Available at: https://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf. [Accessed 2 May 2022]

34. Moradpoor, N., Clavie, B. and Buchanan, B. (2017). *Employing Machine Learning Techniques for Detection and Classification of Phishing Emails*. [online] Available at: https://www.napier.ac.uk/~/media/worktribe/output-

461545/employing-machine-learning-techniques-for-detection-and-classification-of-phishing-emails.pdf.

35. Moradpoor, N., Clavie, B. and Buchanan, B. (2017). *Employing Machine Learning Techniques for Detection and Classification of Phishing Emails*. [online] Available at: https://www.napier.ac.uk/~/media/worktribe/output-461545/employing-machine-learning-techniques-for-detection-and-classification-of-phishing-emails.pdf.

36. Ncsc.gov.uk. (2019). *Email security and anti-spoofing*. [online] Available at: https://www.ncsc.gov.uk/collection/email-security-and-anti-spoofing [Accessed 23 Jan. 2022].
37. Nightingale, S.J. (2017). Email authentication mechanisms: DMARC, SPF and DKIM. [online] doi:10.6028/nist.tn.1945.
38. Peng, T., Harris, I. and Sawa, Y. (2018). Detecting Phishing Attacks Using Natural Language Processing and Machine Learning. *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. [online] Available at: https://ieeexplore.ieee.org/document/8334479 [Accessed 23 Jan. 2022].
39. Raschka, S., Patterson, J. and Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, [online] 11(4), p.193. doi:10.3390/info11040193.
40. Rish, I. (2001). *An Empirical Study of the Naïve Bayes Classifier*. [online] ResearchGate. Available at:

https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier [Accessed 3 May 2022].

41. Saenko, K. (2020). *It takes a lot of energy for machines to learn – here's why AI is so power-hungry*. [online] The Conversation. Available at: https://theconversation.com/it-takes-a-lot-of-energy-for-machines-to-learn-heres-why-ai-is-so-power-hungry-151825 [Accessed 2 May 2022].

42. Salloum, S., Gaber, T., Vadera, S. and Shaalan, K. (2021). Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey. *Procedia Computer Science*, 189, pp.19–28.

43. Scikit Learn (2016). *Scikit Learn algorithm cheat sheet*. [Online image] *scikit-learn.org*. Available at: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html [Accessed 1 Jan. 2022].

44. Sheng, S., Wardman, B., Warner, G. and Zhang, C. (2009). *An Empirical Analysis of Phishing Blacklists*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/228932769_An_Empirical_Analysis_of_Phishing_Blacklists [Accessed 3 May 2022].

45. Su, J. and Zhang, H. (2006.). *A Fast Decision Tree Learning Algorithm*. [online] Available at: https://www.aaai.org/Papers/AAAI/2006/AAAI06-080.pdf.

46. TechRepublic. (2020). *Number of spoof attempts on domains drops to 'near zero' within months of DMARC enforcement*. [online] Available at: https://www.techrepublic.com/article/number-of-spoof-attempts-on-domains-

drops-to-near-zero-within-months-of-dmarc-enforcement/ [Accessed 5 May 2022].

47. Vakili, M., Ghamsari, M. and Rezaei, M. (2020.). *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification*. [online] Available at: https://arxiv.org/pdf/2001.09636.pdf.

48. Vanderplas, J.T. (2017). *Python data science handbook : essential tools for working with data*. Beijing Etc.: O'reilly, Cop, p.363. [Accessed 7 May 2022]

49. Wang, Y., Liu, Y., Wu, T. and Duncan, I. (2020). A Cost-Effective OCR Implementation to Prevent Phishing on Mobile Platforms. *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. [online] Available at: https://ieeexplore.ieee.org/document/9138873 [Accessed 30 December . 2021].

9. Appendix's

Time		Naviv bayes	Desction Tree	Random Forest
	Run 1	4.51	11.89	12.34
	Run 2	2.76	11.83	15.45
	Run 3	3.67	13.25	23.34
	Run 4	3.82	14.29	10.24
	Run 5	2.79	11.84	15.34
	Run 6	4.66	11.92	16.56
	Run 7	3.67	10.96	14.32
	Run 8	2.78	12.25	11.54
	Run 9	2.91	12.46	16.67
	Run 10	3.42	15.55	23.23
Average		3.499	12.624	15.903
Accuarty		Naviv bayes	Desction Tree	Random Forest
	Run 1	83	90	89
	Run 2	83	90	89
	Run 3	83	90	89
	Run 4	85	90	89
	Run 5	84	89	89
	Run 6	83	90	89
	Run 7	83	90	90
	Run 8	83	89	87
	Run 9	83	90	88
	Run 10	83	90	89
Average		83.3	89.8	88.8
Recall		Naviv bayes	Desction Tree	Random Forest
	Run 1	83	90	89
	Run 2	83	90	89
	Run 3	83	90	89
	Run 4	85	90	89
	Run 5	84	89	89
	Run 6	83	90	89
	Run 7	83	90	90
	Run 8	83	89	87
	Run 9	83	90	88
	Run 10	83	90	89
		83.3333333	89.77777778	88.77777778
Precision		Naviv bayes	Desction Tree	Random Forest
	Run 1	83	89	89
	Run 2	83	90	87
	Run 3	83	87	87
	Run 4	83	89	87
	Run 5	83	90	89
	Run 6	83	91	89
	Run 7	83	89	87
	Run 8	83	91	90
	Run 9	83	91	87
	Run 10	83	89	87
		83	89.6	87.9

Appendix 1 Screenshot of table of results

All of the code and the dataset used can be found by clicking the link below.

https://github.com/AnthonySaich/Dissertation-CT6039-