Edge Hill University

The Department of Computer Science

Research and Development Project CIS3140

BSc Computer Science (Hons)

CW2 Final Year Project

"Using Machine Learning to Detect Phishing Email Attacks to Overcome the Flaws of Traditional Spam Filters"

Vinh-Thong Ta

Harry Ellis / 24990876

26th April 2024

'This Report is submitted in partial fulfilment of the requirements for the BSc Honours Computer Science Degree at Edge Hill University'

Abstract

Phishing email attacks represent a significant amount of the threat in the cyber security landscape, vulnerable individuals and organisations are at risk due to human error in recognition of social engineering techniques and flaws with current phishing detectors such as traditional spam filters which rely on predominantly trigger words for detection. A thorough literature review creates a basis to expand upon existing knowledge analysing the existing methods of evaluation and the current state-of-the-art. This project explored the use of machine learning techniques and the natural language processing tool BERT for creating representations of semantic meaning, while implementing a pragmatic approach to model training. The model is evaluated with industry-standard metrics and compared to other studies concluding that there is evidence of some improvement amongst existing methods, with this project yielding an accuracy and recall of approximately 98%. The model has also been integrated into a piece of software to demonstrate how the model could be incorporated into a live application. The report also provides insight into future work on how this model could be improved to increase its performance metrics.

Table of Contents

Table of Figures

Chapter 1 – Introduction

Phishing emails are a type of spam email that attempts to impersonate a legitimate entity for the purposes of financial gain or to acquire sensitive information from the victim (Zieni, Massari and Calzarossa, 2023). Phishing attacks are one of the most common types of cybercrime, according to the FBI's Internet Crime Report phishing emails were responsible for 44 million USD worth of damages (FBI, 2022). The prevalence of phishing attacks has increased since the start of the pandemic as people start to work from home. The home environment makes phishing attacks more successful as organisations cannot pre-install vital software that can defend against phishing attacks such as Intrusion Detection Systems (IDS) (Abroshan et al., 2021). This highlights the importance of having good defences and mitigations to avoid losses to individuals and organisations.

Spam filters are the most common defence against phishing emails and often come pre-installed on many email clients. Spam filters often employ tactics such as black and white lists of allowed and disallowed emails to send and receive from, as well as using trigger words commonly found in phishing emails (Kaddoura et al., 2022). These methods are very simplistic and do not consider various other factors that indicate an email is an attempted phishing attack, they may also commonly flag emails that are legitimate. User education is often mandated for employees of businesses as humans are commonly the weakest point in any system's security. This practice teaches people how to recognise the signs of a phishing email and report them (Castaño et al., 2023).

Aim:

The project aims to develop a machine learning model and implement it into a piece of software that reads incoming emails and flag them as legitimate or phishing emails and warn the owner of the inbox of this at an accuracy higher than a traditional spam filter.

Objectives:

1. Establish the current common techniques used in machine learning and associated data-gathering techniques by researchers working on detecting phishing attacks with this method. This objective involves reviewing relevant research papers in journals such as IEEE, using search terms such as “machine learning”, “phishing emails, “natural language processing” etc. Only articles published within the last 5 years were considered for review due to the speed at which knowledge and research processes in this field. This helps achieve the aim by identifying the challenges associated with this field so they can be addressed before the beginning of the project. This objective can be considered complete once the literature review is completed. A Timeline for completion is also included in Appendix D. Establishing the current state of research helps in identifying the best practices to use to develop the ML model and program.

2. Acquire a suitably large and representative dataset which contains the features relevant to this project. (I.e. text body, subject header). This objective also includes the cleaning of this data and an exploratory data analysis. This data is needed in order to train the machine learning model to an accurate enough degree. This objective will be completed when the data is able to be read by the machine learning algorithm without error and relevant graphs and statistics have been created. The timeline for this objective is shown in Appendix D. Acquiring a cleaned dataset is necessary to create the ML model required by the aim.

3. Use the algorithms, Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR), to create machine learning models using the created set of data from objective 2 and then tune the parameters using the validation data set to enhance the accuracy. Then, develop code that incorporates this model to intercept emails and classify them as phishing or legitimate. Once this objective is completed, the code should be able to display emails as plain text and show a classifier for the emails as they arrive in the inbox. The timeline for this objective is shown in Appendix D. The creation of this software will satisfy the development segment of the aim.

4. Evaluate the effectiveness of the created model, made in objective 3, against the accuracy of a traditional spam filter by calculating evaluation metrics such as accuracy, recall and F-Score. This will indicate how the proposed solution performs in comparison to a traditional spam filter. The objective will be completed after the testing data has been inputted into the model and the metrics have been calculated, this is then compared to the metrics from the spam filter in order to see if any improvement to accuracy has been made as outlined in the aim. The timeline for the evaluation stage is included in Appendix D.

Machine learning is a fast-growing area of computing and a branch of artificial intelligence focussed on the creation and use of algorithms that imitates the behaviour of humans. Applying the techniques used in this field could overcome the previously mentioned flaws in current phishing email protections (Mahadevkar et al., 2022). This project will focus on the analysis of the email body, the content of the email, as this often gives many clues on the legitimacy of an email and is not a greatly explored area of analysis in this field. Analysis of natural language in computing is called natural language processing (NLP) (Oswald Campesato, 2021). For text to be used in a machine learning algorithm the text needs to be converted into numerical values through a technique called text vectorisation. Many algorithms exist for this including word2vec, glove, BERT, etc. (István Üveges and Ring, 2023). These algorithms will be evaluated to determine the best candidate for this project.

Report Structure

This report is split into chapters to aid in the comprehension of its contents and make clear distinctions between these sections. Chapter 2 will cover the background on the topic areas that will be relevant to this research project, explaining relevant terminology used when present. The relevant pre-existing works from journals and conferences will also be critically analysed. Chapter 3 will discuss the methods used in data gathering, developing the model, and the evaluation of the model's success in detecting phishing emails. Chapter 4 will expand upon and implement the evaluation methods discussed in Chapter 3. Chapter 5 will expand upon the findings from Chapter 4 by talking about the project as a whole and making and discussing if the project can be considered successful and the limitations of the proposed model, this section will also compare the results of this project to other attempts of using machine learning to detect phishing attacks. Finally, Chapter 6 will demonstrate how the model could be used as part of a commercial system for an email client.

Chapter 2 – Background & Literature Review

2.1 Background

2.1.1 Background Introduction

This section will develop further the terminology used throughout this project as well as to aid readers in the understanding of the topics pertaining to phishing and machine learning and how they are relevant to this project. The motivation and reasonings for the project will also be highlighted throughout.

2.1.2 Social Engineering

Humans, be they individuals or staff of organisations, often form the weakest part of a system's security due to humans' susceptibility to being deceived. Social engineering is a term used in cyber security to describe the techniques of deceiving individuals in order to manipulate them into revealing sensitive information, to gain access to valuables, or to install malware which may be used in additional cybercrimes (Zieni, Massari and Calzarossa, 2023). Social engineering attacks often use emotions as a way of fooling their victims, exploiting fear, excitement, curiosity, anger, and guilt is commonplace due to these emotions' effectiveness (Al-Thani, 2022).

2.1.3 Phishing

The MITRE ATT&CK (Adversarial, Tactics, Techniques and Common Knowledge) framework is a matrix that describes the stages of a cyber-attack lifecycle. Phishing falls under the initial access section of this framework leading to the next phase, execution (MITRE, 2024).

Phishing is a type of social engineering attack in which the attacker poses as a legitimate source such as the victim's bank. There are numerous vectors used to carry out a phishing attack. Some common types are:

- Spear phishing. Attackers often gather information about a group of people to make the success rate of their attack higher.
- Vishing. Voice phishing are attacks that are carried out over the phone.

- Website Phishing AKA HTTPS Phishing. A malicious website which has likeness from other sites the user subscribes to such as online banking or social media sites.
- Email Phishing. The victim receives a deceptive email. These are often sent en masse so they often direct users to other types of phishing vectors such as websites.
- SMS Phishing AKA Smishing. In this attack, the victim receives an SMS text (Thakur and Pathan, 2020).

There are many more types of phishing attacks and new tactics are constantly sought after by attackers as new methods of phishing are more likely to be successful. New methods of attack or new vulnerabilities in a system are called ‘zero-day’ attacks or ‘zero-day’ vulnerabilities (Sarhan et al., 2023).

2.1.4 Phishing Emails

This project focuses on defending against phishing email attacks as previously stated they are the most common for typical users. Phishing emails also encompass a few other types of phishing attack which all use emails as a vector for sending messages; Spear phishing, which has been previously mentioned, as well as ‘Whaling’ which targets high-value email inboxes such as chief officers of a company who have more money or more powerful account details to steal. This project will develop a solution that helps both common users and specific demographics (Leonov et al., 2021).

The existence of such attacks highlights that phishing emails pose a threat to every individual irrespective of their status or position and that nobody is immune to the techniques involved in this social engineering attack. These kinds of attacks are also so prevalent as they are so simple that they do not require the attacker to have any technical knowledge beyond sending an email to be performed (IBM, 2023).

2.1.5 Spam Filters

Most users will have email clients that include a spam filter, but third parties do offer spam filter services (Microsoft, 2024). Spam filters rely on a few different methods to detect that an email is illegitimate, however, they often have flaws, especially when applied in certain scenarios. Spam filters can be configured to use an allowed list of emails that they can receive from while blocking all others, this is known as a whitelist. An organisation may want to use a whitelist if they only want to receive emails from others inside the organisation (Das, Ahuja and Pandey, 2021). However, if one of those emails were to be compromised the filter would be bypassed allowing phishing emails to be sent by a trusted source.

Alternative to whitelists, spam filters can also contain a blacklist which disallows certain domains or IP address ranges from being received by a potential victim. This can be configured manually or by using a pre-determined and regularly updated list held by the creator of the spam filter (Fan and Yuan, 2022). However, this may cause problems if legitimate communication is happening where one of the senders is from a country in which sending spam emails is common such as India or Russia. This can occur if the ISP reassigns an IP address from a malicious entity to a legitimate one (Statista, 2022).

Spam filters also scan the email for trigger words, which if contained in the email immediately flag it as spam. Depending on the word list used, which in some email clients cannot be changed, this may be problematic if a commonly used word pertaining to legitimate traffic is part of that list (Kaddoura et al., 2022). Attackers can easily avoid triggering trigger word scanners by using similar-looking characters such as the Cyrillic letter ‘En’ (H н) which looks identical to the Latin letter (H h) (Asselborn et al., 2021).

While spam filters offer a reasonable amount of protection it only requires one successful attack to cause devastating effects. Spam filters also do not analyse the context of the email and the overall language used, hence the need for a better method that can use this aspect.

2.1.6 User Education

Social engineering based attacks are often successful due to the average user not being technologically minded and miss the signs of a phishing email such as peculiar hyperlinks, subtly different spellings, and recognition of social engineering tactics etc. As email is such an important part of working life, organisations will often give phishing training to their staff to avoid them falling for phishing attempts (Sutter et al., 2022). Despite training, some phishing attacks can be very sophisticated and fool some educated users, so other defences should be in place in the event of this. However, regardless of the number and effectiveness of defences user education should always be used as a last line of defence as no solution is 100% effective and helps foster a sense of security within the organisation leading to better following of security procedures.

2.1.7 Machine Learning

Machine learning as previously stated imitates the behaviour of humans using algorithms and statistical models. The goal of a machine learning model in this context would be to tag each email as part of two groups, phishing or legitimate. These groups are already defined, and the machine learning model will not be looking for patterns not already designated, this kind of problem of sorting entities into predefined groups is called a classification problem (Mueller and Massaron, 2021). The alternative to a classification problem is a regression problem which aims to find a matching numerical value, this will not be used as it is not relevant to this project (Beheshti et al., 2022). Some classification algorithms also produce a confidence level of high sure it is that the entity matches the predicted classifier.

Machine learning algorithms produce a ‘model’ which is used to make the predictions. This model needs to be trained using a dataset. Training, or learning, can either be supervised or unsupervised. Classification algorithms use supervised learning, in which all the elements of the dataset are already labelled, in a phishing email dataset each email will be labelled as phishing or legitimate (Lee, 2019). Unsupervised learning uses an unlabelled dataset with the machine learning algorithm identifying its own patterns, and clusters the data into its own groups. Using this strategy may be advantageous to identify novel phishing email attack techniques that may not be present in a labelled dataset. This project will focus on using classification algorithms with supervised learning as the groups needed can be easily defined (Sinaga and Yang, 2020). The specific algorithms to use will be decided after the exploratory data analysis as this will give insight into which ones may work best.

2.1.8 Benefits of Machine Learning

Machine learning has the potential to have many benefits that could give them an edge in accuracy over traditional spam filters and user education. Machine learning algorithms cannot be influenced by social engineering techniques, unlike humans, using such techniques would only increase the likelihood of an algorithm flagging the email as a potential phishing attack as it senses the use of the social engineering techniques and associates them with phishing attacks, unlike spam filters which would not detect any usage of these techniques (Lopez and Camargo, 2022). The techniques used may even be subtle and a human may not realise they are in use, this may also be influenced by the time of day, alertness, focus etc. These problems are not relevant to a machine learning algorithm as they are deterministic, meaning they provide the same result given the same data regardless of these factors. This makes them more reliable at detecting phishing attacks (Zhou, 2021).

As previously mentioned, machine learning algorithms can establish a baseline of normal behaviour given relevant data. This means that commonly used words in a particular industry that may be trigger words as part of a spam filter will not immediately flag the email as untrustworthy. Following a similar logic the issues mentioned with white and black lists become less of a problem using machine learning as normal expected traffic can arrive to the inbox while phishing attacks can be blocked even if they originate on the same IP address or from within the organisation. Establishing a normal baseline behaviour also helps to mitigate the likelihood of a successful zero-day attack by flagging an email that does not follow the norm (Hashim, Medani and Attia, 2021).

2.1.9 Text Vectorisation & NLP

The goal of Natural Language Processing (NLP) is to allow machines to interpret human-readable media such as documents. This can be done through the use of natural language datasets and using rule-based logic or probabilistic methods to apply tags, or ‘embeddings’, to words or sentences. Using NLP for detecting phishing attacks will allow for a machine learning algorithm to consider the context and nuances of an email when deciding its legitimacy (Rawat et al., 2022).

The inputs to a machine learning model are called the features which are always numerical values. For non-numerical values such as images or text to be used as a feature of a machine learning model it must be converted into ‘embeddings’ using an appropriate technique (Nwanganga and Chapple, 2020). For this project, the text of an email must be converted to text using text vectorisation.

Many text vectorisation techniques exist but many of them would not be appropriate for this project. For example, Bag of Words (BoW) uses tokenisation, which means breaking the document down into smaller parts called tokens, in this case into singular words and computing the frequency of them into a matrix (Yan et al., 2020). A more useful technique for creating embeddings is 'word2vec' which uses neural networks, part of another subsection of machine learning called deep learning. Word2Vec represents each word as a vector of numbers of some size 'n', these values map the word to its semantic meaning this allows synonymous words and likely next and previous words to be found. However, using word2vec would not take into account the semantic meaning of whole sentences, only individual words, potentially missing out on context cues that would be obvious to a human (Hendrawan, Utami and Hartanto, 2022). A more likely candidate for this project is BERT (Bidirectional Representation from Transformers) which looks at the words around the current target to establish some context of the word even if it is not known in its dictionary. this has particular benefit to phishing emails as they commonly contain misspelt words or strange characters to obscure text readers found in spam filters (Anggrainingsih, Hassan and Datta, 2023).

2.2 Related Works

2.2.1 Related Works Introduction

This section discusses the difficulties and research challenges that researchers face when applying machine learning to counter and defend against phishing attacks and how to overcome these difficulties such as data scarcity. It also analyses several research strategies used by researchers in these fields starting with data gathering and cleaning methods, followed by feature engineering techniques and model selection. Finally, this section will discuss how researchers can evaluate their final solution and the performance metrics that can be calculated to compare between models.

2.2.2 Research Challenges

In comparison to legitimate emails, phishing emails are rare. This causes a problem when attempting to train a machine-learning model to detect them. As previously stated, machine learning models require a large dataset to learn the pattern in the data, and without enough data, the model is likely to have a low accuracy (Bagui et al., 2019). Furthermore, if an organisation would like the model to be trained on their own data to better avoid trigger words incorrectly flagging emails, internal emails will need to be used in the dataset. This can be problematic if the emails contain confidential information which in many cases cannot legally be used to train models under the General Data Protection Regulation (GDPR) (GDPR, 2013) and the UKs Data Protection Act (Data Protection Act, 2018). As a result, this may leave a blind spot in the model for detecting sensitive information which may be relevant for spear phishing and whaling attacks or blackmailing the release of some information.

Feature selection can have many difficult decisions in this area. If a researcher decides to include the sender IP address as a feature to train the machine learning model this may cause complications. IP addresses can be spoofed to look as if they came from a legitimate individual therefore not identifying the email as a phishing attempt (Fonseca et al., 2021). This shows it is important to consider the effects of including some features in their contexts and weigh the pros and cons of their inclusion to improve accuracy.

As the strategies of attackers develop and change over time to bypass existing security methods, machine learning models trained on old emails become less effective at detecting new emails. Circumvention of spam email filters has already been done using strategies mentioned in this document, and as machine learning becomes more accessible, workarounds will be found for machine learning based phishing email detection (Karim et al., 2023). For this reason, researchers are looking into deep learning to continually train models as new data is fed into them to learn new circumvention techniques used by malicious actors.

If the machine learning model is to be deployed across multiple industries the vocabulary used in the dataset must reflect the diversity for optimal accuracy. Additionally, if the model is deployed internationally, it may be prudent to consider multiple models per language. Training a single model on multiple languages may reduce the overall accuracy compared to a single-language model.

2.2.3 Current Use of Machine Learning in Phishing Email Detection

To some extent, machine learning has already been applied in phishing email detection. Many studies focused on the application of neural networks, used in deep learning, which simulate the neurons on the brain. However, this type of machine learning requires larger amounts of data to function, unlike traditional machine learning languages which can function on a more minimal dataset yielding a higher accuracy compared to a neural network on the same data (Karim et al., 2019). Neural networks are also more susceptible to inaccuracies when encountering data far outside the expected range called outliers. They are also susceptible to an effect called overfitting in which the model becomes so accustomed to the input data that it cannot accurately predict new instances of a phishing email as it only expects very closely related emails like in its training data, in a sense the model is not broad enough (Do et al., 2022).

Many studies focus on specific aspects of the email that are not the body's content, such as attached documents which may contain malicious code, and URLs which are the links to other content which may lead to a phishing website. The exclusion of aspects of the emails such as the body may lead to obvious phishing attacks being missed by not analysing the context of the email (Sun et al., 2021).

2.2.4 Common Research Strategies & Evaluation Methods

Machine learning research typically follows a pattern to develop its results reliably. Researchers often emphasise and define their goal in the project, so no unnecessary resource is consumed on unrelated or tangential ventures. As previously mentioned, machine learning requires a large dataset that a model can be created from. This data can be gathered from various sources such as publicly accessible datasets from websites such as Kaggle, or self-sourced using web scraping, sensors etc. (Krawczuk et al., 2021)

To create the model without causing an error or crash the data must be cleaned which involves the following removing missing data values, removing outliers of the data which may have been included by mistake, and removing duplicate data so as to not introduce bias into the model. Many researchers will also choose to perform an exploratory data analysis and create diagrams and a summary of data, this process may expose further problems relevant to the research question that must be rectified before proceeding (Dasari and Varma, 2022). From the data researchers then choose and create the features they wish to use in the machine learning model, this may also include the creation of features from the combination of two or more other features. Reducing the number of features in this way can reduce the time required to complete training. The selection and creation of features is called feature engineering (Kumar and Makkar, 2020).

The final set of data is then divided into training and testing data into roughly 70% and 30% respectively, some researchers may choose slightly varying amounts, but training is overwhelmingly more common to be the largest portion. Training data is used to create the machine learning model, and the testing split is used to evaluate the model's effectiveness on unseen data. This split is done as testing the model on the same data that it is trained on would yield an accuracy of 100% and would not be representative of real-world performance (Alhogail and Alsabih, 2021). Deep learning models would also have a split for validation data, which is used for fine-tuning the model. This set needs to be included for deep learning as deep learning models continually train, the models used in this project do not require validation data to be used as they can be tuned without training on additional data (Sahingoz, Buber and Kugu, 2024).

Machine learning models are evaluated using a series of metrics which can be calculated using a confusion matrix as shown in Table X.

		Actual Values	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 1 Confusion Matrix

Confusion matrices show the number of correctly predicted entities as well as the incorrect ones for false positives and false negatives. They are used to calculate the following:

- Accuracy. The overall percentage of the testing data that was correctly identified.
  - $Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)}$
- Precision. The percentage of the data that is precited positive that the model got correct.
  - $Precision = \frac{TP}{(TP+FP)}$
- Recall / Sensitivity. The percentage of a specified classifier that the model correctly predicted.
  - $Recall = \frac{TP}{(TP+FN)}$

- Specificity. In a way specificity is the opposite of recall, in that it is the measure of the model's incorrect predictions.
    - $Specificity = \frac{TN}{(TN+FP)}$
- F1 Score. This is a combined value of the model's precision and recall.
    - $F1 = \frac{2 \times Precision \times Recall}{Precision+Recall}$

(Sun et al., 2021)

These metrics are important to consider for the selection of a model for a final deployment in a piece of software as an organisation may not mind a high number of false positives if it had a high recall.

2.2.5 Project Direction

The purpose of this project is to overcome the limitations of traditional spam filters and user education by making use of machine learning algorithms. This is done in an attempt to yield a higher accuracy of detection on phishing emails over spam filters. Machine learning could have a higher accuracy due to its pattern recognition and semantic analysis abilities. This will be done by leveraging the use of the body of the text as features using text vectorisation instead of associated media such as attachments, URLs, and the email's metadata (the data that describes an email that is partially hidden from a user, including elements such as IP address) (Hickey, 2020).

Chapter 3 – Methods

3.1 Methods Introduction

This section of the report will detail the methods used for different aspects of the project such as data gathering, data analysis, and machine learning training. This section will also briefly discuss the outcomes of these methods and any issues encountered in their execution.

3.2 Pragmatism

This project addresses a real-world problem, so a pragmatic approach is beneficial to use for this project. Using pragmatism allows techniques and methods to be added or removed from the development process as more information is acquired. This approach can be beneficial as switching or adding more appropriate methods can lead to decreased complexity or cost, reduced development or run time, yield better results etc. (O'leary, 2021). This project could benefit from a pragmatic approach by switching between different machine learning algorithms to find a better result, this would greatly benefit the progress of objective 3.

3.3 Qualitative & Quantitative

Qualitative and Quantitative research are both types of research methodologies. Qualitative research aims to identify and define meanings, behaviours, and patterns. The data involved in this research is often textually based which may be acquired through surveys and interviews. Quantitative research focuses on the development of facts and results often derived from the processing and analysis of numerical data. As this project involves the gathering of data and obtaining metrics such as accuracy, part of objective 4, this project qualifies as qualitative research (Ajimotokan, 2022).

3.4 Software Development Life Cycle

Software development life cycles (SDLC) govern how a project is to be completed to reach the final product. The SDLC is generally comprised of 6 steps that form a circular flow i.e. it is repeated continually until the project is discontinued.

1. Planning – This stage requires a list of requirements to be made that must be completed for a prototype to be developed and considered acceptable for release. Requirements are sorted into functional, which describe what the system will do, and non-functional requirements which describe how the system must perform. The requirements can be gathered through interviews with clients and prospective users of the software. Creation, documentation, analysis verification etc. of requirements is called requirement engineering (Bass, 2023).

2. The second stage of the SDLC outlines the design of the system and how it should be implemented. The results of this stage vary depending on the project. One example of a deliverable for this stage is unified modelling language (UML) diagrams which describe the files of a software project.

3. The design is then implemented and developed into working software following the guidelines set out in the second stage (Wysocki, 2019).

4. The next stage is for the code to be tested to ensure no problems will arise in the final software. This stage is separate from any evaluation stage and just ensures proper execution of the program without crashing.

5. In the deployment stage, any issues found in the previous stage are corrected and retested so that the software is ready for release.

6. Finally, the software receives ongoing support by receiving bug fixes and additional new features. New functionality may be added after repeating the first step of gathering new requirements, the rest of the cycle then repeats for this new feature (Hughes, 2016).

As this project is not set for public release, the SLDC will be implemented by the continual development and improvement of a prototype and a model that will be evaluated each time. However, the methodology described above represents a stricter methodology to software design called 'waterfall'. While this methodology proves useful when creating software for systems that cannot be repeatedly tested, such as the software used by NASA on spacecraft (Yuchnovicz, 2018), it may become a hindrance if newer, better methods are found feeding into a pragmatic approach. Other methodologies exist such as 'agile' and 'spiral' that allow developers to more freely move around the stages covered to improve upon the design and implementation without being locked into the first design (Phillips, 2019). This project could benefit from these more pragmatic approaches by the change of the model's parameters. This project will use the spiral methodology as agile is more suitable for a team-based project with multiple developers.

3.5 Data Gathering

Data gathering is the process of collecting data and preparing it for a purpose. For this project, the data must be prepared for input into a machine-learning model.

3.5.1 Dataset Collection

The dataset used for this project has been sourced from a popular dataset repository called Kaggle (Kaggle, 2023).

	Email Text	Email Type
0	re : 6 . 1100 , disc : uniformitarianism , re ...	Safe Email
1	the other side of * galicismos * * galicismo *...	Safe Email
2	re : equistar deal tickets are you still avail...	Safe Email
3	\nHello I am your hot lil horny toy.\n I am...	Phishing Email
4	software at incredibly low prices (86 % lower...	Phishing Email
...	...	...
18646	date a lonely housewife always wanted to date ...	Phishing Email
18647	request submitted : access request for anita	Safe Email
18648	re : important - prc mtg hi dorn & john , as y...	Safe Email
18649	press clippings - letter on californian utilit...	Safe Email
18650	empty	Phishing Email

Figure 1 Initial Data

As shown in Figure 1, the dataset contains the body of the email with the label of safe or phishing email. This dataset was chosen for its simplicity and realism. All the datasets used in this project fall under the GNU Lesser General Public License. As this project has modified the dataset extensively and transformed it into other forms, it is classified as a derivative work and must be open-sourced with the same license applied (Free Software Foundation, 2016).

3.5.2 Data Cleaning

Some of the emails contain data that may not be able to be processed. The data has been cleaned using Python, and a library for data handling called Pandas to read the comma-separated value (CSV) file that contains the emails. Several steps were involved to clean the data, without the cleaning of data, errors or bias can occur in the training or the resulting model (Li et al., 2021). Many of the rows also contained only URLs, while this may help with the indication of phishing emails, it is not within the scope of this project to develop additional features around these, so they have been removed.

For text to be interpretable by computers it must be expressed in numerical forms called an encoding system. The American Standard Code for Information Interchange (ASCII) is one of these encoding systems. Ascii only includes alphanumeric characters as well as some symbols, this excludes characters from other languages as well as characters like emojis which are often found in an encoding system called Unicode (Groote et al., 2021). Text vectorisation techniques may be incompatible with these characters not inside of the ASCII standard so emails containing them have been removed.

Below includes the type of data that was removed and the quantity that met the removal criterion.

- Empty rows/values – 533
- URLs – 744
- Non-ascii emails – 1257
- Duplicate rows – 453

3.5.3 Exploratory Data Analysis

Upon exploration of the data, a scatter graph (See Figure 2) revealed an email of enormous length that made further analysis challenging. This was removed as an outlier that is likely to have been included in the dataset as an error. To avoid further problems, emails above 10000 words in length were removed, it is important to note that this will not decrease the effectiveness of the model on longer-length emails.

A histogram, in Figure 3, has also been created to show the lengths of the emails. From this, we can deduce the vast majority of the emails are below 500 words in size with the mean being 282 words.

Calculating the same statistics for each group individually, like in Figure 4, shows that safe emails, which have a mean length of 302 words, tend to be slightly longer than a phishing email, which has a mean length of 250 words. This indicates that including the length of an email would be an important feature to include.

Word frequency analysis reveals that the most common words in the emails are what are known as stop words, the most frequent can be found in Figure 5. These words such as “the”, “in”, and “and” are unimportant to the meaning of the sentence and just complete the structure. In NLP tasks it may prove beneficial to remove the stop words to not provide unnecessary input to machine learning algorithms and models (Desai, Saini and Bafna, 2022). However, as the text vectorisation technique used in this project uses the context of the email in its technique, it is currently unknown if the stop words will be a detriment or benefit to the model’s accuracy.

Another technique that could be applied at this step is lemmatisation. This technique reduces the inflected forms of words down to the root word, or ‘lemma’. For example, the words: “Changed”, “Changes”, and “Changer”, will all be reduced to just “Change”. This also strips words of affixes, indications of gender, tense etc. Lemmatisation is important as compound words with many language features, can be mapped to the same meaning. Without using this technique some text vectorisation algorithms may not be able to assign values to more complex words (Kowsher et al., 2020).

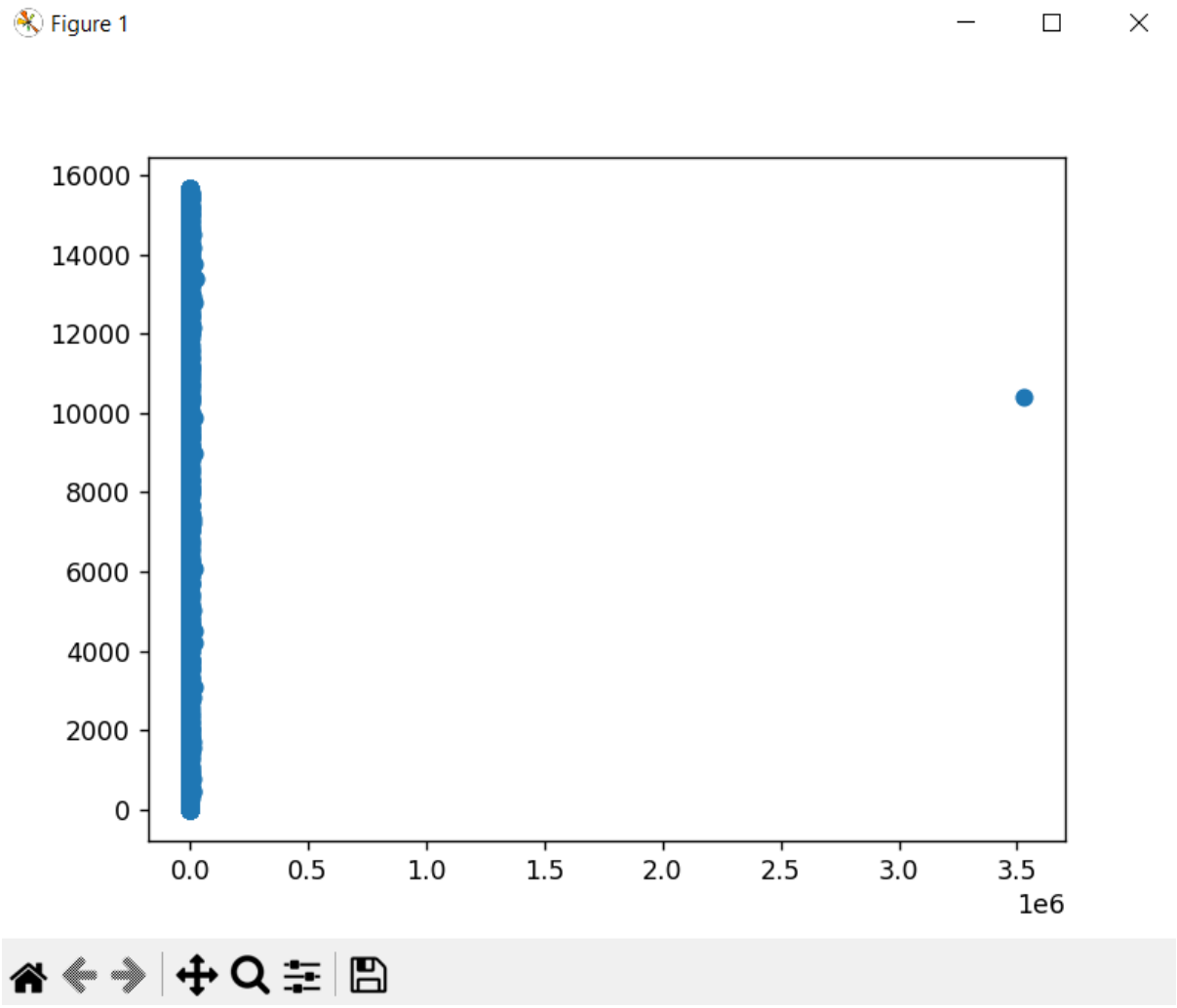


Figure 2 Scatter Graph Showing Outlier

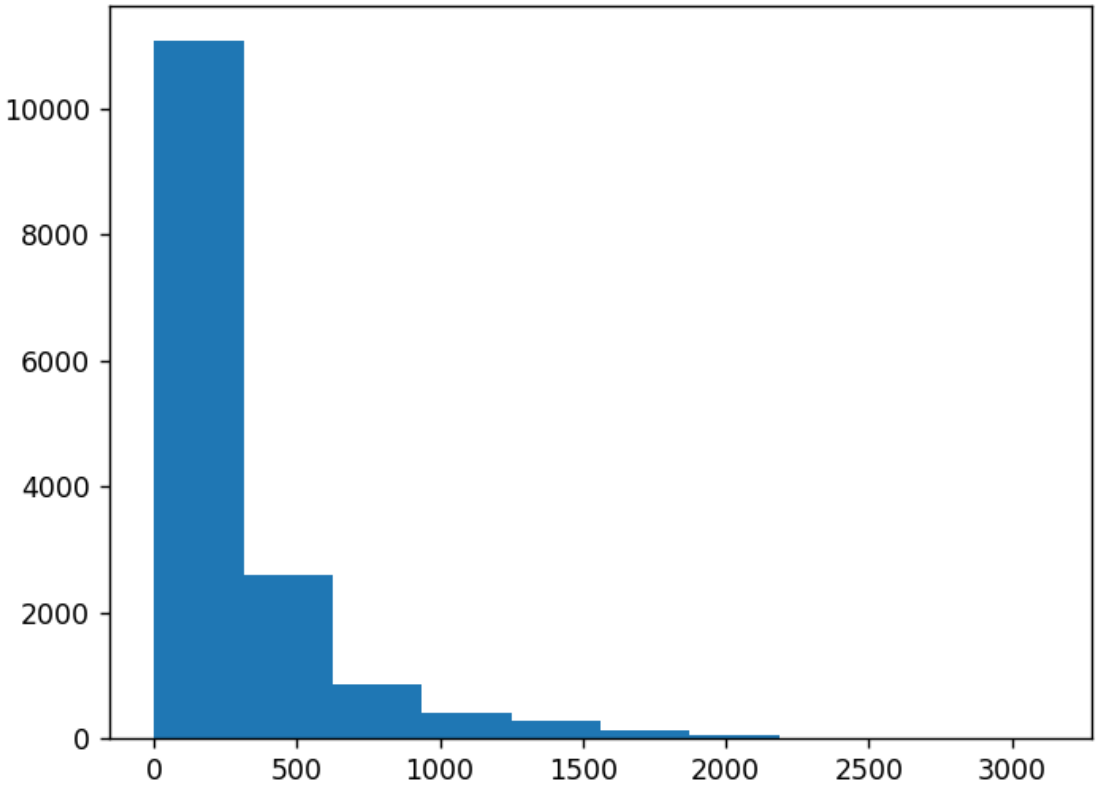


Figure 3 Histogram Showing Email Lengths

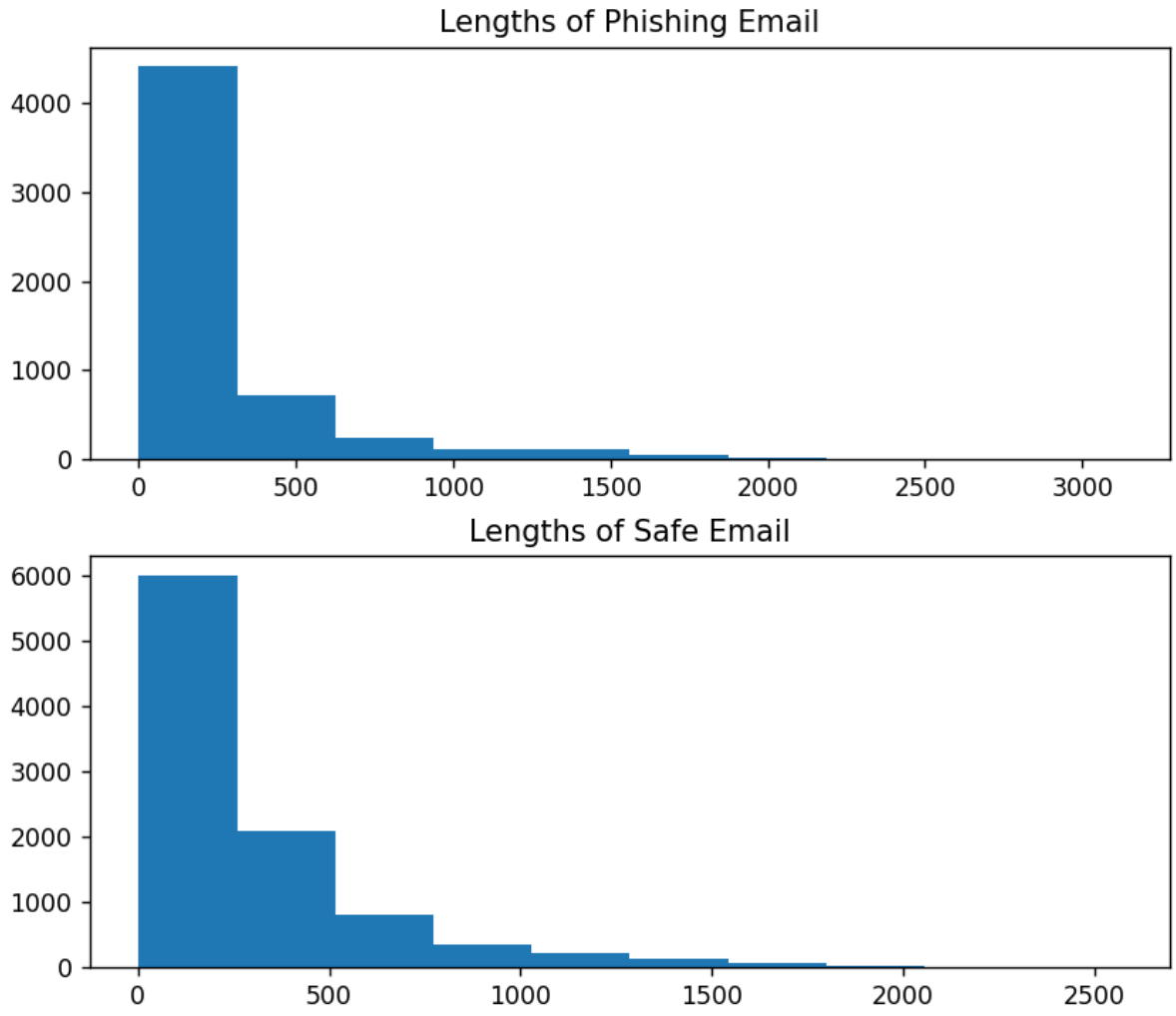


Figure 4 Histograms Showing Length of Emails by Type

Figure 5 Most Common Words in Emails

The remaining data leaves the following email quantities:

- 9735 Safe emails
- 5645 Phishing emails

These have been visualised in Figure 6. As previously noted, for the machine learning model to have a higher accuracy the dataset must be imbalanced to represent the ratio imbalance between phishing and safe emails.

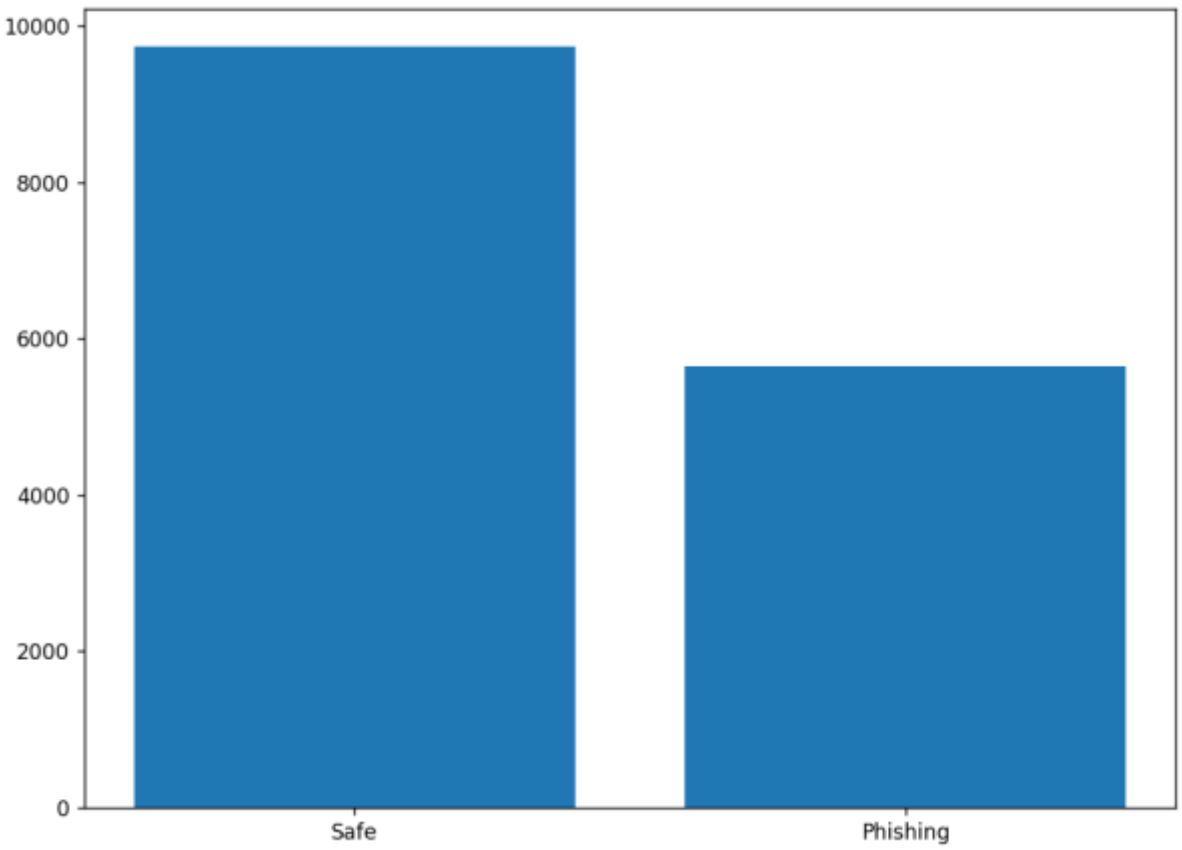


Figure 6 Bar Chart Showing Split of Email Types

3.5.4 BERT-Based Text Vectorisation & One-Hot Encoding

As previously mentioned, before the data can be used in a machine learning model it must be converted, or 'vectorised', into text. This project uses BERT to vectorise text using a pre-trained model developed by Google. BERT uses a dictionary to tokenise words, for example, the word 'overestimated' would likely only contain 'estimate' with 'over' and 'd' being tokenised with hash symbols to denote they are part of the same word ('over#', 'estimate', '#d'). Each token would then be converted to a list of 768 numbers that link to meanings (Gómez-Pérez, Denaux and Garcia-Silva, 2020).

One-hot encoding is a method of converting a finite set of categorical data categories into denary values so that they can be used as part of a machine-learning model. For example, red, green, and blue can be encoded as follows:

- Red → 0
- Blue → 1
- Green → 2

(Yu et al., 2020)

This project will use one-hot encoding to convert the labels of the emails (i.e. phishing or safe emails) to numerical values as follows:

- Safe Email → 0
- Phishing Email → 1

3.6 Machine Learning

3.6.1 Algorithms & Justification

Many machine learning algorithms exist, each with their own strengths and weaknesses. For this project the following algorithms have been chosen:

1. Random Forest (RF)
2. Support Vector Machine (SVM)
3. Logistic Regression (LR)
4. Ensemble (RF + SVM + LR)

Random Forest:

The RF algorithm is a collection of decision trees, as shown in Figure 7. A decision tree is a structure that represents possible decisions to be made from the start called the root node, they also contain subsequent decisions based on the previous decision called branches. The branches all lead to a final decision at the ending 'leaf'. The leaf nodes decide on the outcome of the whole model by way of majority voting (Wei, 2023).

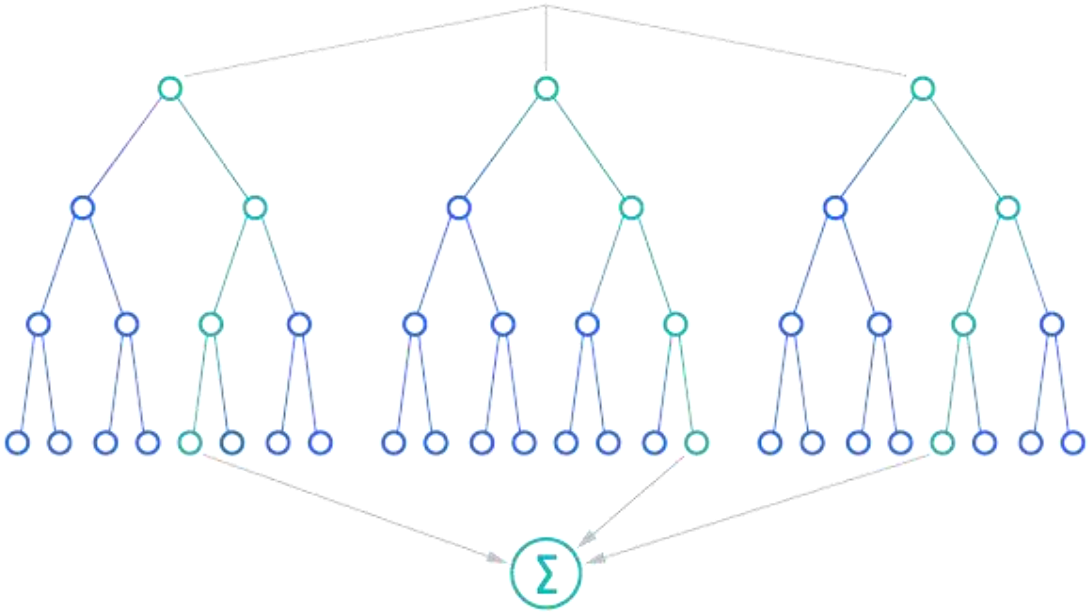


Figure 7 Random Forest Example

RFs have the advantage of handling much more complicated tasks than a single decision tree with a higher degree of accuracy. This machine learning algorithm is also more robust than other models being less susceptible to overfitting. RF can be very computationally expensive, especially as the number of trees increases consuming more memory, this also increases the training time for the model (Ubels et al., 2020).

Support Vector Machine:

SVMs classify an item by drawing the best line between the two sets of data like in Figure 8. The testing data is classified by seeing where new information would be plotted on either side of the line. SVMs are also capable of classifying numbers of classes above two by using multiple equations of different types as in Figure 8 (Chen et al., 2022).

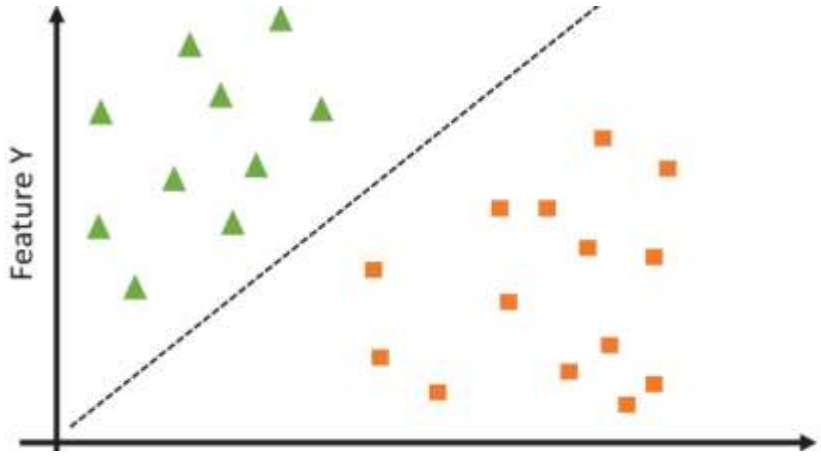


Figure 8 Linear SVM

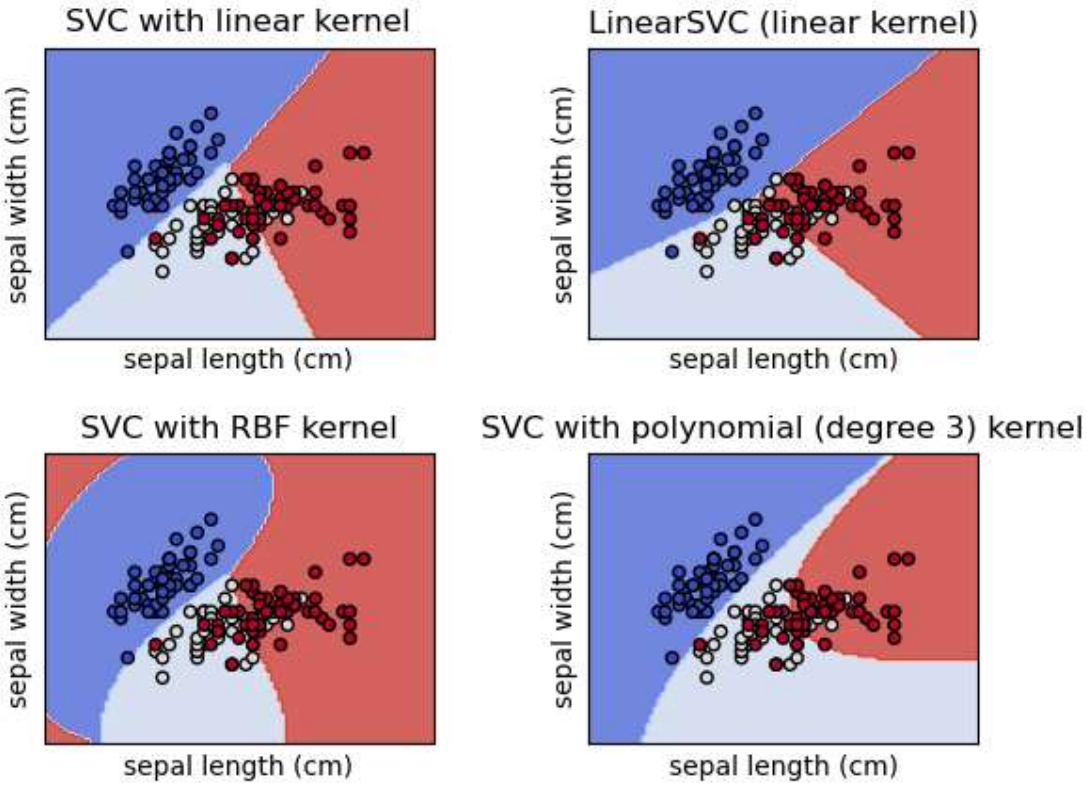


Figure 9 Three Zone SVM with Different Kernels

SVMs can be very effective when dealing with a high number of dimensions as they are equated to a lower number. This is especially useful for this project as there is an effective total of 768 features, the condensing of the dimensions also makes an SVM more memory efficient than alternative algorithms. However, as the data is classified by data points sitting on either side of a definitive boundary, classification tasks where the classes overlap would perform very poorly using an SVM (Kurani et al., 2021).

Logistic Regression:

Logistic regression is a classification algorithm that is closely related to linear regression. The algorithm works by calculating the probability of an instance being a part of one of the classes. This probability is then plotted on a sigmoid curve, like in Figure 10. A threshold is set for what probability is required for the algorithm to evaluate the instance as the select classifier. An example is included in Figure 11 (Pampel, 2021).

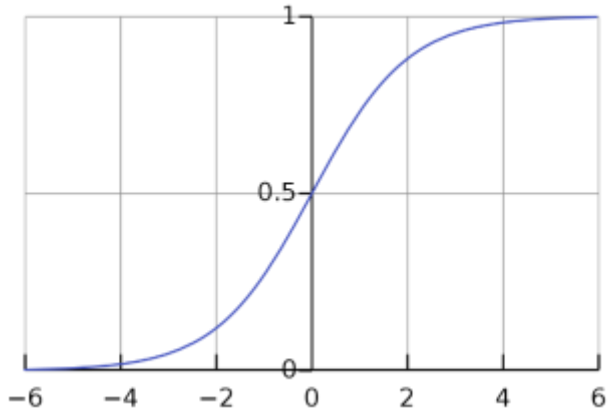


Figure 10 Sigmoid Curve

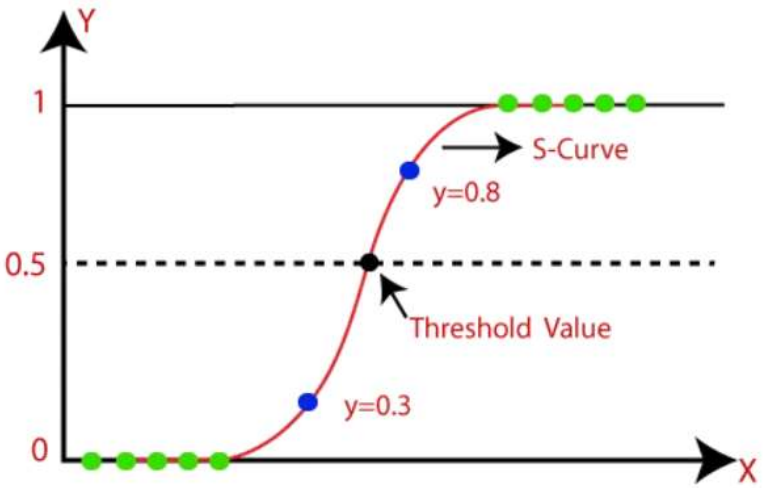


Figure 11 Logistic Regression Example

Logistic regression requires more assumptions of the data than other models so cannot be used in many cases such as the assumption that all data points are independent from each other (one does not affect the others) etc (Nwanganga and Chapple, 2020).

Ensembles:

An ensemble model can be created by combining the outputs from several other models. The final decision of the ensemble model is created through hard or soft voting. Hard voting uses the majority vote from the models, i.e. if two or more models vote phishing the result will be phishing and vice versa for safe classification. Soft voting uses the mean of the output from regression models as well as a threshold to determine the classification of an instance (Puneet et al., 2021). This project will use a hard voting ensemble as the algorithms used as part of it are classification and not regression algorithms, thus soft voting is not possible.

Ensembles do not guarantee an increase in the model's performance. While they can benefit from the strengths of multiple models, they also combine the deficits of each. This means that an ensemble model can have a lower performance than one or more of the individual models used as a part of the ensemble (Ramteke and Maidamwar, 2023).

3.6.2 Model Tuning

The machine learning algorithms used in this project all have modifications, called hyperparameters, that can be changed to increase or decrease the performance of the algorithms. The process of continually changing and retraining algorithms with different hyperparameters is called hyperparameter tuning (Hoque and Aljamaan, 2021). Appendix A shows a list of executions made with different algorithms, their associated hyperparameters, and evaluation metrics. The algorithm with index 12 was the best performer found so will be used as part of the prototype in Chapter 6.

Chapter 4 – Testing & Evaluation

4.1 Testing & Evaluation Introduction

This section of the report will outline how the models will be evaluated using standardised metrics to express the model's effectiveness in determining if an instance of an email is safe or a phishing attempt. This section will also visualise these results and identify any weaknesses of the proposed solution giving examples of commonly misclassified emails and possible reasonings for the model's failure in these cases.

4.2 Metrics & Explanation

As previously mentioned, accuracy, precision, recall, and F1 score are common metrics used to evaluate the performance of a machine learning model. Looking at the table in Appendix A the highest accuracy is given by model 12 with an accuracy of 98.5%. However, accuracy cannot always be used in evaluating the performance of a model. Using multiple metrics is especially important when using unbalanced datasets as misidentifying the minority class is more detrimental. For example, if a dataset contained 98 safe emails and 2 phishing emails, the model would be likely to identify all instances as safe, giving a 98% accuracy, this in theory is a very good result, but would fail in actual deployment. This shows that it is necessary to use all metrics available in evaluation to avoid this critical error (Agrawal, 2021).

For evaluating the effectiveness of the model produced in this project, it is prudent to take note of the recall, as this is a measure of how many phishing emails were correctly predicted (Medeiros et al., 2020). The reasoning behind this metric's importance is that it only requires a single successful phishing email to compromise an organisation's security, so it is favourable to reduce the number of false negatives (phishing emails being classified as safe) (University of Oxford, 2024) (BBC, 2020) (NCSC, 2018). Another method of visualising recall is a receiver operating characteristic (ROC) graph. The ROC graph for the best model found in this project is shown in Figure 12. The blue line represents the performance of a random classifier (predicting at random), a visual aid is displayed in Figure 13 showing the performance of the model is very good (Aslam and Ali Bou Nassif, 2023).

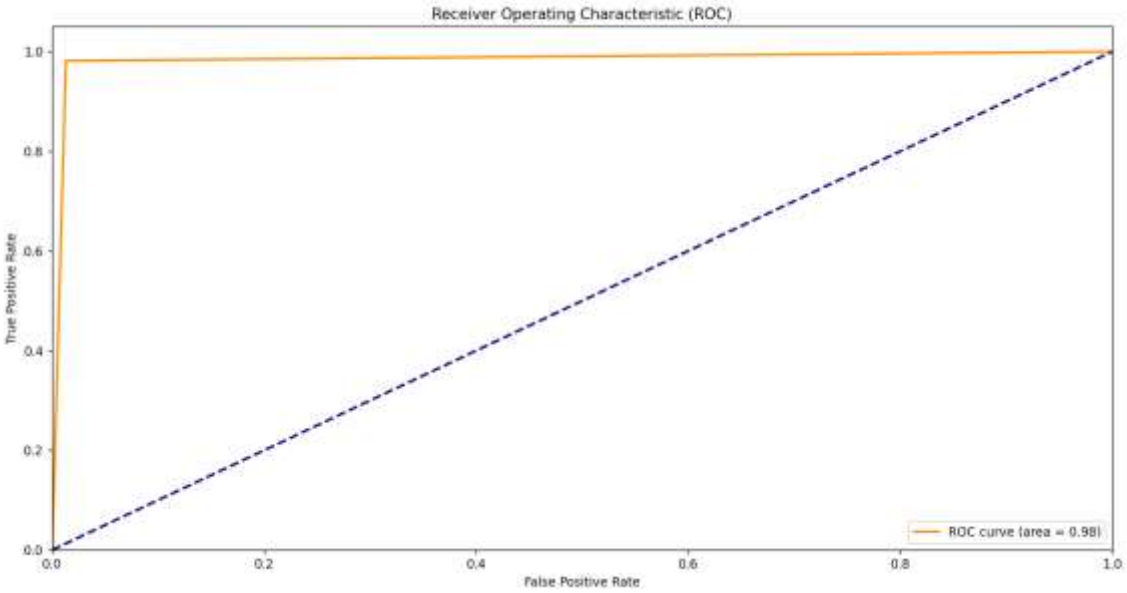
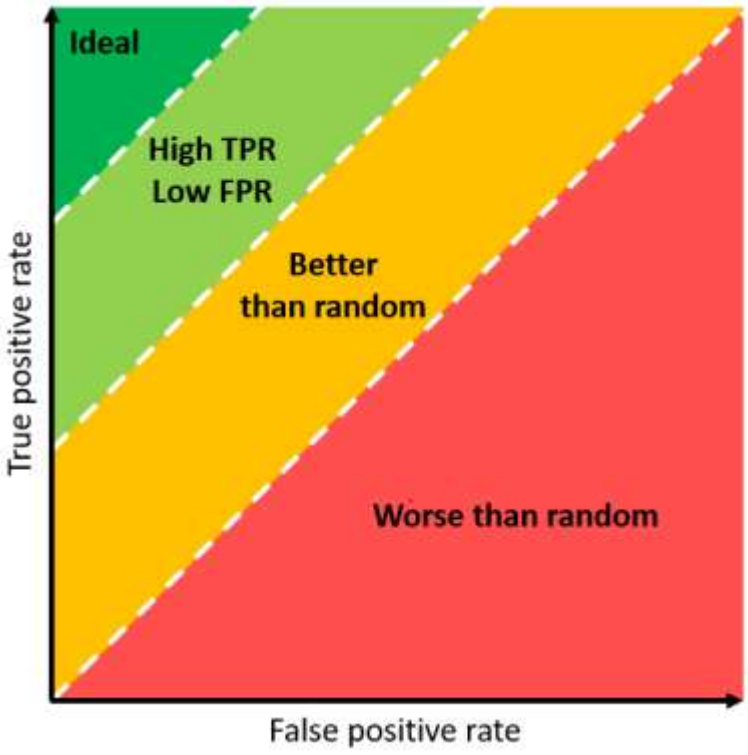


Figure 12 ROC Graph



(Demeusy, 2023)

Figure 13 ROC Analysis Graph

In addition to manual input of hyperparameters, a parameter grid can be used to run many combinations in succession, these 'grid searches' are also included in Appendix A. The use of grid searches requires a lot of time to complete due to every permutation needing to be tested, this is called an exhaustive search (Li et al., 2020). This gives grid searches a time complexity of $O(n \times p^x)$ where p is the number of dimensions and x is the dimensions length. Time complexity is a mathematical representation of how long an algorithm will take to run (Ndiaye et al., 2019).

The highlighted model in Appendix A is the best performer in all metrics except precision being outperformed by only one other model. Due to the reasons above the highlighted model will still be referred to as the best model as it has the best recall score. This model will be the furthest analysed in the remainder of this chapter as well as the model used as part of the prototype in Chapter 6.

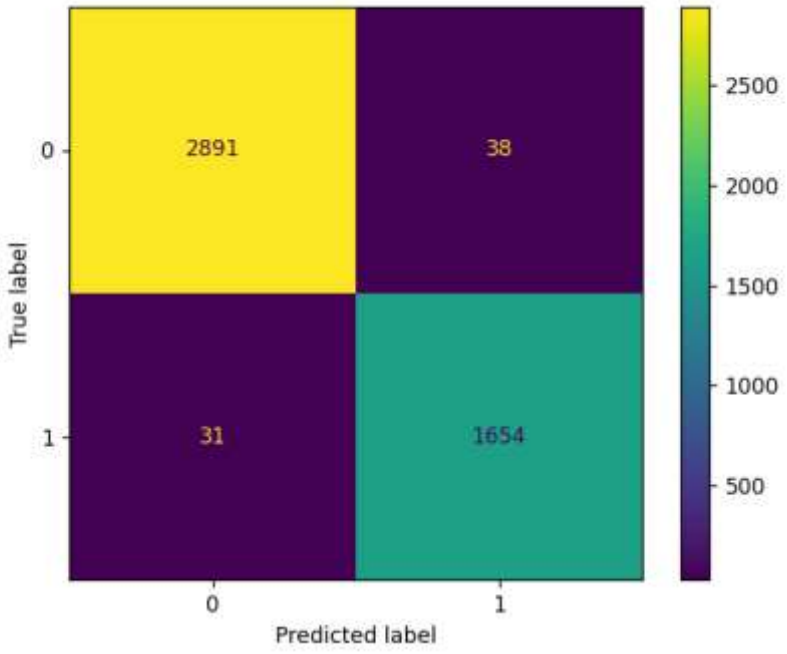


Figure 14 Model Confusion Matrix

Metrics for the model:

- Accuracy: 0.985045513654096
- Precision: 0.977541371158392
- Recall: 0.98160237388724
- F1 Score: 0.979567663606751

Lemmatisation and stop word removal were also applied in an attempt to improve the model's performance. However, this only led to a slight detriment. The results of these attempts can be found in Appendix A and Appendix E.

4.3 Results of Traditional Spam Filter

The vast majority of commercial spam filters are closed source, meaning the exact methodology used to determine an email's legitimacy is unknown (Kochhar et al., 2019). Additionally, testing these emails would require many emails to be sent to an inbox via a commercial server, this may invoke the server to flag the sending testing email account or IP address as spam and the account deactivated or the IP being blacklisted, resulting in the test being incomplete (Ferreira et al., 2021). For these reasons, a simple spam filter has been created that checks for the presence of 769 trigger words. If an email contains any of the words/phrases in the list, the filter predicts it to be a phishing email.

Using this local method of testing for a traditional spam filter has the drawback of not including metadata such as IP address, sender country, time etc. However, the NLP-based system proposed by this project also does not have access to these so one does not have an advantage over the other in textual-based analysis.

Calculating the same metrics used for the machine learning approach, the traditional spam filter yields the following results:

- Accuracy: 0.6332899869960988
- Precision: 0.5609756097560976
- Recall: 0.004074402125775022
- F1 Score: 0.00809004572634541

With a confusion matrix shown in Figure 15.

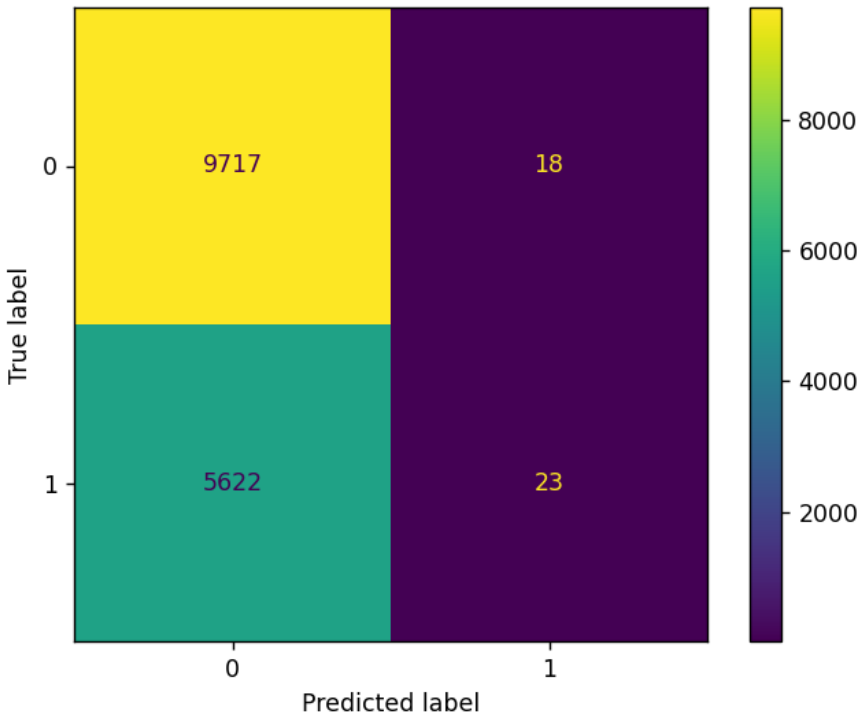


Figure 15 Spam Filter Confusion Matrix

These results show that the accuracy of the filter is above random guessing, which would be 50%, with a result of 63%. However, as previously mentioned recall is more important for this application as only one phishing email attempt needs to be successful for an entire system to be compromised. The spam filter misidentified 5622 phishing emails as legitimate, much higher than 31 from the machine learning model. This shows that the machine learning model is much more effective and relies much less on user education as a defence against phishing email attacks.

In addition to testing on the primary dataset, the model has been evaluated on a second unseen dataset of phishing emails. By attaining predictions on this second dataset, it is possible to estimate the efficiency on new emails that may contain different writing styles to the training data of the primary dataset. The results of the model on this data are as follows:

- Accuracy: 0.957166673791305
- Precision: 0.9373780277809666
- Recall: 0.9470540477847367
- F1 Score: 0.942191195984538

The confusion matrix and ROC graph are included in Appendix F.

This information shows that while the model is not as performant on a second dataset, it does still yield reasonable results for detection. The reason for the loss in performance could be any of the following:

- Domain Shift – Trends in phishing emails have changed in the time, location, or industry (Akrout et al., 2023).
- Overfitting – The model has become too accustomed to the data of the first dataset (Horenko, 2020).
- Nuance – The second dataset may contain some nuance that is not present in the primary dataset, so the model fails in these instances (Mills, de Silva and Alahakoon, 2020).

Chapter 5 – Demonstration Prototype Development

5.1 Prototype Introduction

A prototype has been developed to illustrate how such a model could be implemented into a commercial piece of software as a standalone application.

5.2 Email Interception

In order for the model to be used there must be some method to acquire the emails in the inbox. The ultimate prototype created is simply a Python application that can run in the background of a machine locally, there were numerous reasons for this:

- Integration into an existing popular email client would necessitate the learning of another language namely JavaScript or Visual Studio, which is outside the scope of the project.
- BERT is not able to be integrated using Visual Studio (Microsoft, 2024b).
- As BERT and machine learning are computationally expensive, running the application in a browser could form complications if the browser in question limits resource use.
- Loading the model each time the email client is opened online may cause frustration in loading times for the model (Fargose et al., 2022).
- Browsers and local email clients are plentiful and often updated, ensuring compatibility across different software and versions adds significant complications to implementation (Witte, 2022).

For the reasons outlined above the prototype was chosen to run as a local script, this ensures compatibility across email domains as long as the email is based on IMAP (Internet Messaging Access Protocol). Some examples of popular emails that implement IMAP are:

- Microsoft Outlook
- Windows Mail
- Google Gmail
- Apple Mail
- Mozilla Thunderbird (Google, 2024)

The program runs on a loop and can be minimised to the taskbar. The program reads the current list of unread emails and extracts the body text. This is then vectorised by BERT and fed into the model. An activity diagram showing the program execution can be found in Appendix C.

5.3 Warning the User

At the start of the execution, the program launches an instance of an IMAP server, logged in with the credentials provided by the user. If a spam email is detected the program sends an email to its own inbox warning the user with information on the subject and date/time to aid the user in identifying the correct email, See Figure 17. As the warning email is not a phishing email, any email that is from the user's own account is discounted and not fed into the model or vectorised.

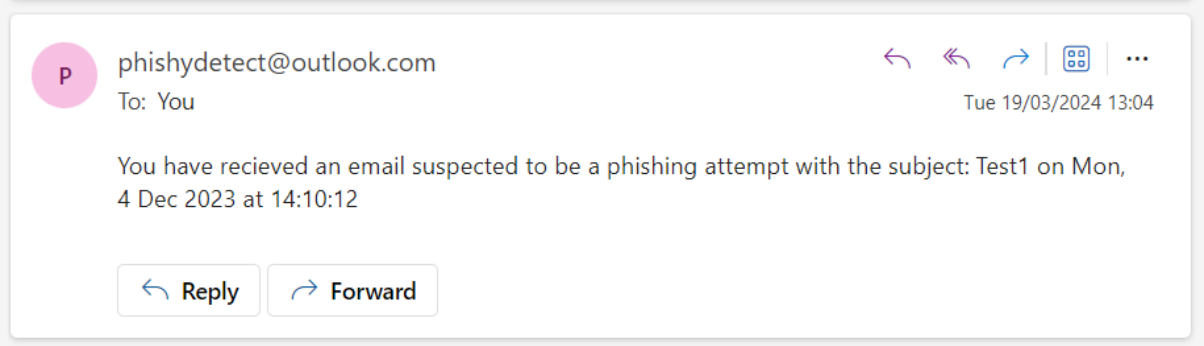


Figure 16 Example Warning

Chapter 6 – Outcome & Discussion

6.1 Outcome & Discussion Introduction

This section will evaluate the success of the project as a whole and the aims and objectives set out from the start. This section will also discuss the limitations of the model with the relevant legal, social, and ethical implications of the results and the model's usage in detecting phishing emails. The results from this project will be compared to existing results from researchers and make mention of any future work that can be done by others to improve upon the results of this project.

6.2 Project Success

The outcomes of this project have shown that the use of BERT and machine learning can be an effective way of detecting spam emails. The use of these methods has also shown to be more effective than a traditional spam filter that uses trigger words as a means of detection, with justification for the importance of specific metrics over others to promote a safer overall system.

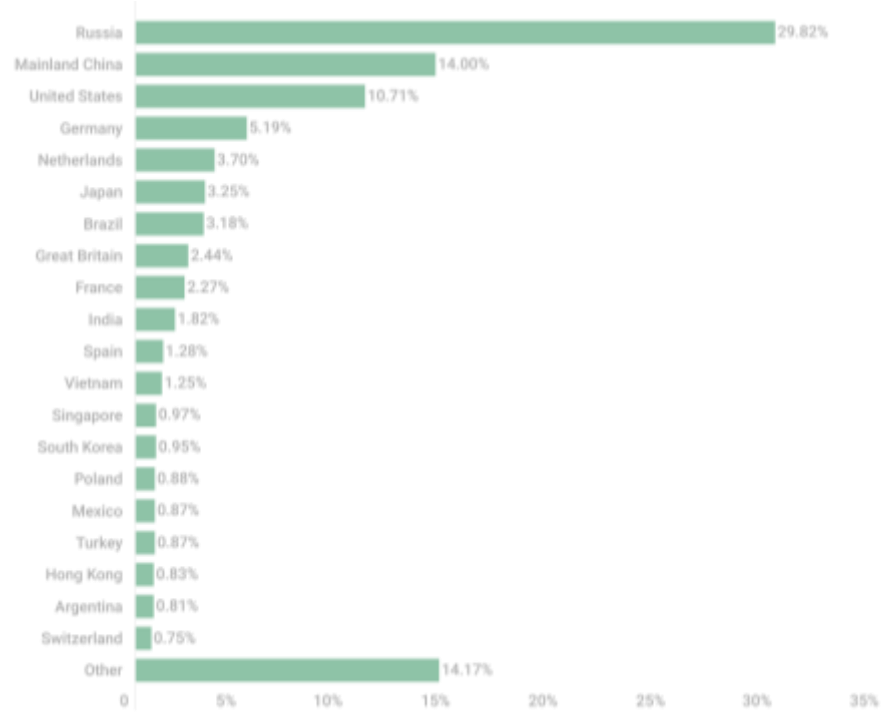
6.3 Limitations

As this project only focuses on the body of the text, email attributes such as URLs, subject headers etc. will not be checked and could lead to obvious phishing attacks being misclassified. Furthermore, this project has a significant reliance on the BERT text vectorisation library. While this library is open source, it could become closed off if decided by Google (Devlin and Chang, 2018), making future models unusable may decrease the effectiveness of this project's solution over time due to changes in language, trends, or colloquialisms (Rodina et al., 2021). The use of BERT also limits the project's feasibility to only English-speaking applications. While other BERT models do exist for multilingual operation, they are likely to have decreased accuracy and larger file size, leading to longer vectorisation time (Thin et al., 2022).

6.4 Social, Legal & Ethical Problems

The use of a model of this kind also has some legal, social, and ethical implications. If the model is to be trained on any sensitive data from a company, such as names, addresses, etc. laws surrounding data protections such as GDPR will apply, and the organisation must take steps to avoid data breaches and make the owners of the data aware of this practice in a privacy policy (GDPR, 2013).

One social concern that is important to note is that it is likely that emails written where the sender's English skills are less advanced than that of training data, will be flagged as a phishing attempt. This is because many phishing attempts originate from countries where the first language is not English lowering the English literacy in these regions. Figure 16 shows the origin of phishing emails showing that Russia is the highest producer (Kaspersky, 2022).



(Kaspersky, 2022)

Figure 17 Origins of Phishing Emails by Country

To ensure the use of the model created in the project is ethical, if the model is used in a commercial solution users must be warned that the predictions made by the model cannot be trusted implicitly as the model does not have a 100% accuracy. User education must be used alongside any developed software.

6.5 Comparisons to Other Studies

Other studies have attempted to classify phishing attempts based on vectorisations of their text. While this section only analyses three of the studies in this field, they can be considered representative of the current state-of-the-art as the main difference between the studies is the use of different vectorisation techniques and different machine learning models with little to no deviation in methodology. Mambina, Ndibwile and Michael's study focussed on Swahili smishing attacks and achieved an accuracy of 99.8% (recall was not a used metric in this study), they achieved this result by using random forest and a text vectorisation technique called term frequency-inverse document frequency (TF-IDF). This vectorisation technique gives a ranking of importance to each word of the text (Mambina, Ndibwile and Michael, 2022). Possible reasons for the study's higher accuracy in comparison to this project are:

- Text messages may be easier to identify than emails due to their tendency to be shorter in length and more direct.
- Text messages may be more suited to TF-IDF than BERT is to emails.
- Swahili may be an easier language to detect phishing attacks for.
- The dataset may make a clearer distinction between phishing and safe messages. This would allow the machine learning model to have higher confidence as the two classifiers are dimensionally further apart.

Specific to email text classification, a study by Ahammad et al. achieved an accuracy of 96% accuracy. This study also did not use a recall metric, so the accuracy will be used as a comparison. While the study does not explicitly mention the text vectorisation used, it is likely to be Bag of Words (BoW) based on the following quote: "Words with the most frequency have the highest intensity as compared to the less frequent words. By interpreting the different frequency words, manually made a corpus of 100 phishing related words (these words are manually taken based on true knowledge and experiences in fraud mails)" (Ahammad et al., 2022). Using BoW likely led to the lower accuracy of 96% in this study in comparison to this project's 98%. This is because BoW is a very primitive NLP technique and does not utilise any sentiment assignment (Denecke, 2023).

Another study aims to identify the use of persuasion cues in phishing emails as a means of classification. Valecha, Mandaokar and Rao, 2021, used persuasion cues as another feature in addition to Word2Vec to achieve a recall of 94.2% (Valecha, Mandaokar and Rao, 2021). While this score is lower than this project, it is likely due to the vectorisation choice of Word2Vec. BERT has superior results to word2vec as it is bidirectional and context-specific whereas word2vec only looks at a singular word (Ji et al., 2019). The use of persuasion cues likely aided the model's performance so could be used in combination with BERT to achieve even better performance.

6.6 Future Work

Machine learning methods are very computationally intensive, leading to training times being days to complete a grid search of many parameters. If the project had more time, then better hyperparameters could be found to increase the performance of the model. As the number of features increases so too does the computation time (Pandit et al., 2022). A method of feature removal exists that limits the loss of the additional information from these features called principal component analysis. This method works by linearly combining multiple features to a specified number that is representative of the original values. However, it is important to note that a PCA does lose some information so would thus lower the metrics values (Bhatia, 2020). PCA has not been used in this project as it is unlikely that better parameters gained from more training time would produce better metrics that exceed that of the model with the original feature set. This is due to the low variance found between the metrics of the models, parameter to parameter.

As previously mentioned in Chapter 3.5 Stop word removal and lemmatisation could aid in the model's detection of phishing emails by removing redundant words and word affixes. However, the removal of these words and language features could reduce the model's performance by removing important context from the email body. This project made a brief attempt at stop word removal and lemmatisation, and although the model was not improved it is possible it could improve the results if used in combination with PCA, or through more extensive grid searches that are more suited to this new data (Hafeez and Nikhila Kathirisetty, 2022).

There are many more machine learning algorithms beyond Support Vector Machines, Random Forest, and Logistic Regression. Due to time and resource constraints, it was possible to test these additional algorithms which may be more suited to this task. As mentioned in Chapter 2.2, researchers are also looking into the application of deep learning which could improve upon this model provided a suitably large dataset is found (Harikrishnakumar et al., 2019).

This project only uses the body of the text for semantic analysis, studies such as the one from Sameen, Han, and Hwang focus on the detection of phishing URLs using machine learning. Adding the URL as a feature to this project's model could improve the performance, as well as the addition of email metadata such as sending email, subject header etc. Additionally, the models could be used in tandem with a voting classifier (Sameen, Han and Hwang, 2020). It is important to note however, that the increased number of features or voting could increase the time required to classify incoming emails, this latency could then allow some phishing attempts to be successful, so future work should find a balance between increased accuracy and the time taken to classify. Future work could also be done in combining many datasets to create a more generalised model more capable of detecting nuanced data.

Chapter 7 – Conclusions

This project has addressed the severity and damage that can be caused by phishing email attacks in the modern world to individuals and organisations. Through a background and literature review, the current techniques of both traditional spam filters and machine learning approaches to tackling this problem have been established and problems with each identified, completing the first objective set out in the introduction.

Leveraging the use of data gathering techniques and data enrichment such as data cleaning, one hot encoding etc. a suitably large dataset of phishing emails was acquired and prepared for the use of training this project's model, satisfying objective 2. Through the use of this data and hyperparameter tuning, a sufficiently performant model was found demonstrating the potential of machine learning to overcome the flaws of traditional spam filters, with the machine learning approach achieving higher accuracy than the spam filter. While the machine learning model does have great potential it is important to note its limitations which have been discussed, completing objective 4.

This project has also shown how the model could be used in a commercial piece of software to enhance the security posture for IMAP-based email users. The software takes the form of a Python script that runs in the background and warns users of incoming phishing attacks via an email warning, completing objective 3.

The development of the machine learning model in this project represents a significant step towards improving the email security landscape and safeguarding users from associated cyber threats. By continuing research and implementing methods talked about in the future works section the model could be improved to further increase effectiveness.

References

ABROSHAN, H., DEVOS, J., POELS, G., and LAERMANS, E., 2021. COVID-19 and Phishing: Effects of Human emotions, behaviour, and Demographics on the Success of Phishing Attempts during the Pandemic. IEEE Access [online]. 9 (1), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/9525387 [Accessed 22 Feb 2024].

AGRAWAL, S.K., 2021. Evaluation Metrics for Classification Model | Classification Model Metrics. Analytics Vidhya [online]. Available from: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/ [Accessed 8 Apr 2024].

AHAMMAD, S.M.M., RAVITEJA, T., KOUSHIK, J., DINESH, P.V., and ASHOK, A., 2022. Machine Learning Approach Based Phishing Email Text Analysis (ML-PE-TA). 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9917765 [Accessed 15 Apr 2024].

AJIMOTOKAN, H.A., 2022. Research Techniques. Springer Nature.

AKROUT, M., FERIANI, A., BELLILI, F., MEZGHANI, A., and HOSSAIN, E., 2023. Domain Generalization in Machine Learning Models for Wireless Communications: Concepts, State-of-the-Art, and Open Issues. IEEE Communications Surveys and tutorials/IEEE Communications Surveys and Tutorials [online]. 25 (4), pp. 3014–3037. Available from: https://ieeexplore.ieee.org/document/10288574 [Accessed 18 Apr 2024].

AL-THANI, N.A., 2022. Adolescents' and Social engineering: the Role of Psychometrics Factors in Determining Vulnerability and Designing Interventions. IEEE Xplore [online]. Available from: https://ieeexplore.ieee.org/document/9995705 [Accessed 21 Feb 2024].

ALHOGAIL, A. and ALSABIH, A., 2021. Applying Machine Learning and Natural Language Processing to Detect Phishing Email. Computers & Security [online]. 110 (1), p. 102414. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0167404821002388 [Accessed 23 Feb 2024].

ANGGRAININGSIH, R., HASSAN, G.M., and DATTA, A., 2023. CE-BERT: Concise and Efficient BERT-based Model for Detecting Rumours on Twitter. IEEE Access [online]. 11 (1), pp. 80207–80217. Available from: https://ieeexplore.ieee.org/document/10196451 [Accessed 22 Feb 2024].

ASLAM, S. and ALI BOU NASSIF, 2023. Phish-identifier: Machine Learning Based Classification of Phishing Attacks. Advances in Science and Engineering Technology International Conferences (ASET) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10180869 [Accessed 17 Apr 2024].

ASSELBORN, T., JOHAL, W., TLEUBAYEV, B., ZHEXENOVA, Z., DILLENBOURG, P., MCBRIDE, C., and SANDYGULOVA, A., 2021. The Transferability of Handwriting skills: from the Cyrillic to the Latin Alphabet. npj Science of Learning [online]. 6 (1). Available from: https://www.nature.com/articles/s41539-021-00084-w#citeas [Accessed 18 Dec 2023].

BAGUI, S., NANDI, D., BAGUI, S., and WHITE, R.J., 2019. Classifying Phishing Email Using Machine Learning and Deep Learning. IEEE Xplore [online]. 1 (1), pp. 1–2. Available from: https://ieeexplore.ieee.org/document/8885143 [Accessed 23 Feb 2024].

BASS, J.M., 2023. Agile Software Engineering Skills. Springer Nature.

BBC, 2020. Twitter hack: Staff Tricked by Phone spear-phishing Scam. BBC News [online]. 31 July. Available from: https://www.bbc.co.uk/news/technology-53607374 [Accessed 8 Apr 2024].

BEHESHTI, I., GANAIE, M.A., PALIWAL, V., RASTOGI, A., RAZZAK, I., and TANVEER, M., 2022. Predicting Brain Age Using Machine Learning Algorithms: A Comprehensive Evaluation. IEEE Journal of Biomedical and Health Informatics [online]. 26 (4), pp. 1432–1440. Available from: https://ieeexplore.ieee.org/document/9439893 [Accessed 22 Feb 2024].

BHATIA, S., 2020. A Comparative Study of Opinion Summarization Techniques. IEEE Transactions on Computational Social Systems [online]. 8 (1), pp. 1–8. Available from: https://ieeexplore.ieee.org/document/9262854 [Accessed 16 Apr 2024].

CASTAÑO, F., FERNAÑDEZ, E.F., ALAIZ-RODRÍGUEZ, R., and ALEGRE, E., 2023. PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification. IEEE Access [online]. 11 (1), pp. 40779–40789. Available from: https://ieeexplore.ieee.org/document/10103863 [Accessed 22 Feb 2024].

CHEN, Y., MAO, Q., WANG, B., DUAN, P., ZHANG, B., and HONG, Z., 2022. Privacy-Preserving Multi-Class Support Vector Machine Model on Medical Diagnosis. IEEE Journal of Biomedical and Health Informatics [online]. 26 (7), pp. 3342–3353. Available from: https://pubmed.ncbi.nlm.nih.gov/35259122/ [Accessed 8 Apr 2024].

DAS, L., AHUJA, L., and PANDEY, A., 2021. Existing Spam Filtering Methods considering Different technique: a Review. IEEE Xplore [online]. Available from: https://ieeexplore.ieee.org/document/9673294 [Accessed 22 Feb 2024].

DASARI, D. and VARMA, P.Suresh., 2022. Employing Various Data Cleaning Techniques to Achieve Better Data Quality Using Python. 6th International Conference on Electronics, Communication and Aerospace Technology [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10009079 [Accessed 23 Feb 2024].

DATA PROTECTION ACT, 2018. Data Protection Act 2018. Legislation.gov.uk [online]. Available from: https://www.legislation.gov.uk/ukpga/2018/12/section/2/enacted [Accessed 19 Apr 2024].

DEMEUSY, A., 2023. ROC Curve and AUC: an Intuitive Approach and Implementation Guide. Medium [online]. Available from: https://medium.com/@anthony.demeusy/roc-curve-and-auc-an-intuitive-approach-and-implementation-guide-b245b060fced [Accessed 17 Apr 2024].

DENECKE, K., 2023. Sentiment Analysis in the Medical Domain. Springer Nature.

DESAI, P., SAINI, J.R., and BAFNA, P.B., 2022. POS-based Classification and Derivation of Kannada Stop-words Using English Parallel Corpus. 2022 3rd International Conference for Emerging Technology (INCET) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9825429 [Accessed 6 Mar 2024].

DEVLIN, J. and CHANG, M.-W., 2018. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. blog.research.google [online]. Available from: https://blog.research.google/2018/11/open-sourcing-bert-state-of-art-pre.html [Accessed 8 Apr 2024].

DO, N.Q., SELAMAT, A., KREJCAR, O., HERRERA-VIEDMA, E., and FUJITA, H., 2022. Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. IEEE Access [online]. 10 (1), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/9716113 [Accessed 23 Feb 2024].

FAN, J. and YUAN, F., 2022. Recognition of Junk Short Messages Based on Local Sensitive Hash KNN Algorithm. In: International Conference on Artificial Intelligence, Information Processing and Cloud Computing (AIIPCC) [online]. Available from: https://ieeexplore.ieee.org/document/10070193 [Accessed 22 Feb 2024].

FARGOSE, R., GAONKAR, S., JADHAV, P., JADIYA, H., and LOPES, M., 2022. Browser Extension for a Safe Browsing Experience. 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) [online]. 1 (1), pp. 1–6. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9885551 [Accessed 10 Apr 2024].

FBI, 2022. Internet Crime Report. ic3 [online]. Available from: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf [Accessed 22 Feb 2024].

FERREIRA, I.C., ARAGÃO, M., OLIVEIRA, E.M., KUEHNE, B.T., MOREIRA, E.M., and CARPINTEIRO, O.A.S., 2021. The Development of the Open Machine-Learning-Based Anti-Spam (Open-MaLBAS). IEEE Access [online]. 9 (1), pp. 138618–138632. Available from: https://ieeexplore.ieee.org/document/9565223/authors#authors [Accessed 9 Apr 2024].

FONSECA, O., CUNHA, I., FAZZION, E., MEIRA, W., ALVES DA SILVA, B., FERREIRA, R.A., and KATZ-BASSETT, E., 2021. Identifying Networks Vulnerable to IP Spoofing. IEEE Transactions on Network and Service Management [online]. 18 (3), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/9360876 [Accessed 23 Feb 2024].

FREE SOFTWARE FOUNDATION, 2016. GNU Lesser General Public License v3.0 - GNU Project - Free Software Foundation. Gnu.org [online]. Available from: https://www.gnu.org/licenses/lgpl-3.0.html [Accessed 22 Apr 2024].

GDPR, 2013. Art. 32 GDPR – Security of Processing | General Data Protection Regulation (GDPR). General Data Protection Regulation (GDPR) [online]. Available from: https://gdpr-info.eu/art-32-gdpr/ [Accessed 22 Feb 2024].

GÓMEZ-PÉREZJ.M., DENAUX, R., and GARCIA-SILVA, A., 2020. A Practical Guide to Hybrid Natural Language Processing : Combining Neural Models and Knowledge Graphs for NLP. Cham: Springer International Publishing.

GOOGLE, 2024. About IMAP and POP Clients - Google Workspace Admin Help. support.google.com [online]. Available from: https://support.google.com/a/answer/12103?hl=en [Accessed 10 Apr 2024].

GROOTE, J.F., MOREL, R., SCHMALTZ, J., and WATKINS, A., 2021. Logical gates, circuits, processors, Compilers and Computers. Cham: Springer.

HAFEEZ, S. and NIKHILA KATHIRISETTY, 2022. Effects and Comparison of Different Data pre-processing Techniques and ML and Deep Learning Models for Sentiment analysis: SVM, KNN, PCA with SVM and CNN. First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9844192 [Accessed 16 Apr 2024].

HARIKRISHNAKUMAR, R., DAND, A., NANNAPANENI, S., and KRISHNAN, K., 2019. Supervised Machine Learning Approach for Effective Supplier Classification. IEEE Xplore [online]. Available from: https://ieeexplore.ieee.org/document/8999282 [Accessed 17 Apr 2024].

HASHIM, A., MEDANI, R., and ATTIA, T.A., 2021. Defences against Web Application Attacks and Detecting Phishing Links Using Machine Learning. IEEE Xplore [online]. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9429609 [Accessed 22 Feb 2024].

HENDRAWAN, I.R., UTAMI, E., and HARTANTO, A.D., 2022. Comparison of Word2vec and Doc2vec Methods for Text Classification of Product Reviews. 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10057702 [Accessed 22 Feb 2024].

HICKEY, M., 2020. Hands on Hacking. Wiley & Sons Canada, Limited, John.

HOQUE, K.E. and ALJAMAAN, H., 2021. Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting. IEEE Access [online]. 1 (1), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/9643011 [Accessed 8 Apr 2024].

HORENKO, I., 2020. On a Scalable Entropic Breaching of the Overfitting Barrier for Small Data Problems in Machine Learning. Neural Computation [online]. 32 (8), pp. 1563–1579. Available from: https://ieeexplore.ieee.org/document/9142620 [Accessed 18 Apr 2024].

HUGHES, T.M., 2016. SAS Data Analytic Development. John Wiley & Sons.

IBM, 2023. What Is Phishing? | IBM. www.ibm.com [online]. Available from: https://www.ibm.com/topics/phishing [Accessed 22 Feb 2024].

ISTVÁN ÜVEGES and RING, O., 2023. HunEmBERT: a Fine-Tuned BERT-Model for Classifying Sentiment and Emotion in Political Communication. IEEE Access. 11 (1), pp. 60267–60278.

JI, S., SATISH, N., LI, S., and DUBEY, P.K., 2019. Parallelizing Word2Vec in Shared and Distributed Memory. IEEE Transactions on Parallel and Distributed Systems [online]. 30 (9), pp. 2090–2100. Available from: https://ieeexplore.ieee.org/document/8663393 [Accessed 15 Apr 2024].

KADDOURA, S., CHANDRASEKARAN, G., ELENA POPESCU, D., and DURAISAMY, J.H., 2022. A Systematic Literature Review on Spam Content Detection and Classification. PeerJ Computer Science. 8 (1), p. e830.

KAGGLE, 2023. Phishing Email Detection. www.kaggle.com [online]. Available from: https://www.kaggle.com/datasets/subhajournal/phishingemails/data [Accessed 17 Apr 2024].

KAGGLE, 2024. Phishing Emails 2. www.kaggle.com [online]. Available from: https://www.kaggle.com/datasets/shallykandoi/phishing-emails-2 [Accessed 17 Apr 2024].

KARIM, A., AZAM, S., SHANMUGAM, B., KANNOORPATTI, K., and ALAZAB, M., 2019. A Comprehensive Survey for Intelligent Spam Email Detection. IEEE Access [online]. 7 (1), pp. 168261–168295. Available from: https://ieeexplore.ieee.org/document/8907831 [Accessed 23 Feb 2024].

KARIM, A., SHAHROZ, M., MUSTOFA, K., BELHAOUARI, S.B., and JOGA, S.R.K., 2023. Phishing Detection System through Hybrid Machine Learning Based on URL. IEEE Access [online]. 11 (1), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/10058201/authors#authors [Accessed 23 Feb 2024].

KASPERSKY, 2022. Spam and Phishing in 2022 – Securelist. securelist.com [online]. Available from: https://securelist.com/spam-phishing-scam-report-2022/108692/ [Accessed 18 Dec 2023].

KOCHHAR, P.S., KALLIAMVAKOU, E., NAGAPPAN, N., ZIMMERMANN, T., and BIRD, C., 2019. Moving from Closed to Open Source: Observations from Six Transitioned Projects to GitHub. IEEE Transactions on Software Engineering [online]. 47 (9), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/8812899 [Accessed 9 Apr 2024].

KOWSHER, Md., TAHABILDER, A., HOSSAIN SARKER, M.M., ISLAM SANJID, Md.Z., and PROTTASHA, N.J., 2020. Lemmatization Algorithm Development for Bangla Natural Language Processing. 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9306652 [Accessed 6 Mar 2024].

KRAWCZUK, P., PAPADIMITRIOU, G., TANAKA, R., ANH, T.M., SUBRAMANYA, S., NAGARKAR, S., JAIN, A., LAM, K., MANDAL, A., POTTIER, L., and DEELMAN, E., 2021. A Performance Characterization of Scientific Machine Learning Workflows. IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9652609 [Accessed 23 Feb 2024].

KUMAR, N. and MAKKAR, A., 2020. Machine Learning in Cognitive IoT. CRC Press.

KURANI, A., DOSHI, P., VAKHARIA, A., and SHAH, M., 2021. A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting. Annals of Data Science [online]. 10 (1), pp. 183–208. Available from: https://link.springer.com/article/10.1007/s40745-021-00344-x [Accessed 8 Apr 2024].

LEE, W.-M., 2019. Python Machine Learning. Indianapolis, In: Wiley.

LEONOV, P.Y., VOROBYEV, A.V., EZHOVA, A.A., KOTELYANETS, O.S., ZAVALISHINA, A.K., and MOROZOV, N.V., 2021. The Main Social Engineering Techniques Aimed at Hacking Information Systems. 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9455031 [Accessed 22 Feb 2024].

LI, P., RAO, X., BLASE, J., ZHANG, Y., CHU, X., and ZHANG, C., 2021. CleanML: a Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks. 2021 IEEE 37th International Conference on Data Engineering (ICDE) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9458702 [Accessed 6 Mar 2024].

LI, Y., LIU, G., GAO, L., JIAO, L., MARTURI, N., and SHANG, R., 2020. Hyper-Parameter Optimization Using MARS Surrogate for Machine-Learning Algorithms. IEEE Transactions on Emerging Topics in Computational Intelligence [online]. 4 (3), pp. 287–297. Available from: https://ieeexplore.ieee.org/document/8735959 [Accessed 16 Apr 2024].

LOPEZ, J.C. and CAMARGO, J.E., 2022. Social Engineering Detection Using Natural Language Processing and Machine Learning. 2022 5th International Conference on Information and Computer Technologies (ICICT) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9844996 [Accessed 22 Feb 2024].

MAHADEVKAR, S.V., KHEMANI, B., PATIL, S., KOTECHA, K., VORA, D.R., ABRAHAM, A., and GABRALLA, L.A., 2022. A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions. IEEE Access [online]. 10 (1), pp. 107293–107329. Available from: https://ieeexplore.ieee.org/document/9903420 [Accessed 22 Feb 2024].

MAMBINA, I.S., NDIBWILE, J.D., and MICHAEL, K.F., 2022. Classifying Swahili Smishing Attacks for Mobile Money Users: a Machine-Learning Approach. IEEE Access [online]. 10 (1), pp. 83061–83074. Available from: https://ieeexplore.ieee.org/document/9849641 [Accessed 15 Apr 2024].

MEDEIROS, N., IVAKI, N., COSTA, P., and VIEIRA, M., 2020. Vulnerable Code Detection Using Software Metrics and Machine Learning. IEEE Access [online]. 8 (1), pp. 219174–219198. Available from: https://ieeexplore.ieee.org/document/9272730 [Accessed 8 Apr 2024].

MICROSOFT, 2024a. Manage Mail Flow Using a third-party Cloud Service with Exchange Online. learn.microsoft.com [online]. Available from: https://learn.microsoft.com/en-us/exchange/mail-flow-best-practices/manage-mail-flow-using-third-party-cloud [Accessed 22 Feb 2024].

MICROSOFT, 2024b. Build Your First Outlook add-in - Office Add-ins. learn.microsoft.com [online]. Available from: https://learn.microsoft.com/en-us/office/dev/add-ins/quickstarts/outlook-quickstart?tabs=yeomangenerator [Accessed 10 Apr 2024].

MILLS, N., DE SILVA, D., and ALAHAKOON, D., 2020. Generating Situational Awareness of Pedestrian and Vehicular Movement in Urban Areas Using IoT Data Streams. IEEE Internet of Things Journal [online]. 7 (5), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/8960359 [Accessed 18 Apr 2024].

MITRE, 2024. MITRE ATT&CKTM. Mitre.org [online]. Available from: https://attack.mitre.org/ [Accessed 19 Apr 2024].

MUELLER, J. and MASSARON, L., 2021. Machine Learning for Dummies. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

NCSC, 2018. Phishing attacks: Defending Your Organisation. NCSC.gov.uk [online]. Available from: https://www.ncsc.gov.uk/guidance/phishing [Accessed 8 Apr 2024].

NDIAYE, E., LE, T., FERCOQ, O., SALMON, J., and TAKEUCHI, I., 2019. Safe Grid Search with Optimal Complexity. 36th International Conference on Machine Learning [online]. 97 (1). Available from: https://proceedings.mlr.press/v97/ndiaye19a/ndiaye19a.pdf [Accessed 16 Apr 2024].

NWANGANGA, F.C. and CHAPPLE, M., 2020. Practical Machine Learning in R. Indianapolis, Indiana: John Wiley & Sons, Incorporated.

O'LEARY, Z., 2021. Essential Guide to Doing Your Research Project. 4th ed. S.L.: Sage Publications.

OSWALD CAMPESATO, 2021. Natural Language Processing Fundamentals for Developers. Mercury Learning and Information.

PAMPEL, F.C., 2021. Logistic Regression : a Primer. Thousand Oaks, California: Sage Publications, Inc.

PANDIT, A., GUPTA, A., BHATIA, M., and SUBHASH CHAND GUPTA, 2022. Filter Based Feature Selection Anticipation of Automobile Price Prediction in Azure Machine Learning. 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9850615 [Accessed 16 Apr 2024].

PHILLIPS, M., 2019. The Practitioner's Handbook of Project Performance. Routledge.

PUNEET, DEEPIKA, SINGH, P., BANSAL, R., and SHARMA, S., 2021. Coronary Heart Disease Prediction Using Voting Classifier Ensemble Learning. 2021 3rd International Conference on Advances in

Computing, Communication Control and Networking (ICAC3N) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9725705 [Accessed 8 Apr 2024].

RAMTEKE, N. and MAIDAMWAR, P., 2023. Cardiac Patient Data Classification Using Ensemble Machine Learning Technique. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10307702 [Accessed 8 Apr 2024].

RAWAT, A., MAHESHWARI, H., KHANDUJA, K., KUMAR, R., MEMORIA, M., and KUMAR, S., 2022. Sentiment Analysis of Covid19 Vaccines Tweets Using NLP and Machine Learning Classifiers. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/9850629 [Accessed 3 Jun 2023].

RODINA, J., TROFIMOVA, Y., KUTUZOV, A., and ARTEMOVA, E., 2021. ELMo and BERT in Semantic Change Detection for Russian. Analysis of Images, Social Networks and Texts [online]. 1 (1), pp. 175–186. Available from: https://link.springer.com/chapter/10.1007/978-3-030-72610-2_13 [Accessed 8 Apr 2024].

SAHINGOZ, O.K., BUBER, E., and KUGU, E., 2024. DEPHIDES: Deep Learning Based Phishing Detection System. IEEE Access [online]. 12 (1), pp. 8052–8070. Available from: https://ieeexplore.ieee.org/document/10388305 [Accessed 23 Feb 2024].

SAMEEN, M., HAN, K., and HWANG, S.O., 2020. PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System. IEEE Access [online]. 8 (1), pp. 83425–83443. Available from: https://ieeexplore.ieee.org/document/9082616 [Accessed 16 Apr 2024].

SARHAN, M., LAYEGHY, S., GALLAGHER, M., and PORTMANN, M., 2023. From zero-shot Machine Learning to zero-day Attack Detection. International Journal of Information Security. 22 (1).

SCIKIT LEARN, 2019. sklearn.model_selection.GridSearchCV — scikit-learn 0.22 Documentation. Scikit-learn.org [online]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed 16 Apr 2024].

SINAGA, K.P. and YANG, M.-S., 2020. Unsupervised K-Means Clustering Algorithm. IEEE Access [online]. 8 (1), pp. 80716–80727. Available from: https://ieeexplore.ieee.org/document/9072123 [Accessed 22 Feb 2024].

STATISTA, 2022. Spam e-mail: Countries of Origin 2020. Statista [online]. Available from: https://www.statista.com/statistics/263086/countries-of-origin-of-spam/ [Accessed 22 Feb 2024].

SUN, B., BAN, T., HAN, C., TAKAHASHI, T., YOSHIOKA, K., TAKEUCHI, J., SARRAFZADEH, A., QIU, M., and INOUE, D., 2021. Leveraging Machine Learning Techniques to Identify Deceptive Decoy Documents Associated with Targeted Email Attacks. IEEE Access [online]. 9 (1), pp. 87962–87971. Available from: https://ieeexplore.ieee.org/document/9435284 [Accessed 23 Feb 2024].

THAKUR, K. and PATHAN, A.-S.K., 2020. Cybersecurity Fundamentals. CRC Press.

THIN, D.V., LE, L.S., HOANG, V.X., and NGUYEN, N.L.-T., 2022. Investigating Monolingual and Multilingual BERT Models for Vietnamese Aspect Category Detection. 2022 RIVF International Conference on Computing and Communication Technologies (RIVF) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10013792 [Accessed 8 Apr 2024].

UBELS, J., SCHAEFERS, T., PUNT, C., GUCHELAAR, H.-J., and DE RIDDER, J., 2020. RAINFOREST: a Random Forest Approach to Predict Treatment Benefit in Data from (failed) Clinical Drug Trials. Bioinformatics [online]. 36 (2), pp. i601–i609. Available from: https://academic.oup.com/bioinformatics/article/36/Supplement_2/i601/6055917 [Accessed 8 Apr 2024].

UNIVERSITY OF OXFORD, 2024. Avoid Email Scams. www.infosec.ox.ac.uk [online]. Available from: https://www.infosec.ox.ac.uk/phishing#tab-450966 [Accessed 8 Apr 2024].

VALECHA, R., MANDAOKAR, P., and RAO, H.R., 2021. Phishing Email Detection Using Persuasion Cues. IEEE Transactions on Dependable and Secure Computing [online]. 19 (2), pp. 1–1. Available from: https://ieeexplore.ieee.org/document/9565347 [Accessed 15 Apr 2024].

WEI, L., 2023. Genetic Algorithm Optimization of Concrete Frame Structure Based on Improved Random Forest. 2023 International Conference on Electronics and Devices, Computational Science (ICEDCS) [online]. 1 (1). Available from: https://ieeexplore.ieee.org/document/10361776 [Accessed 8 Apr 2024].

WITTE, F., 2022. Strategy, Planning and Organization of Test Processes. Springer Nature.

WYSOCKI, R.K., 2019. Effective Project Management : Traditional, Agile, Extreme, Hybrid. 8th ed. Indianapolis, Indiana: Wiley.

YAN, D., LI, K., GU, S., and YANG, L., 2020. Network-Based Bag-of-Words Model for Text Classification. IEEE Access [online]. 8 (1), pp. 82641–82652. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9079815 [Accessed 22 Feb 2024].

YU, L., ZHOU, R., CHEN, R., and LAI, K.K., 2020. Missing Data Preprocessing in Credit Classification: One-Hot Encoding or Imputation? Emerging Markets Finance and Trade. 1 (1), pp. 1–11.

YUCHNOVICZ, D., 2018. Spotlight on Lessons Learned: Aligning System Development Models with Insight Approaches | APPEL Knowledge Services. Nasa.gov [online]. Available from: https://appel.nasa.gov/2018/11/27/spotlight-on-lessons-learned-aligning-system-development-models-with-insight-approaches/ [Accessed 8 Apr 2024].

ZHOU, Z.-H., 2021. Machine Learning. Gateway East, Singapore: Springer.

ZIENI, R., MASSARI, L., and CALZAROSSA, M.C., 2023a. Phishing or Not Phishing? a Survey on the Detection of Phishing Websites. IEEE Access [online]. 11 (1), pp. 18499–18519. Available from: https://ieeexplore.ieee.org/document/10049452 [Accessed 22 Feb 2024].

ZIENI, R., MASSARI, L., and CALZAROSSA, M.C., 2023b. Phishing or Not Phishing? a Survey on the Detection of Phishing Websites. IEEE Access [online]. 11 (1), pp. 18499–18519. Available from: https://ieeexplore.ieee.org/document/10049452 [Accessed 21 Feb 2024].

Appendices

Appendix A

Index	Algorithm	Accuracy	Precision	Recall	F1-Score	Date	Parameters
1	Random Forest	0.96532293	0.95850872	0.94599407	0.95221027	14/03/2024	Default Settings (n_estimators=100)
2	Random Forest	0.96684005	0.96256039	0.94599407	0.95420533	15/03/2024	max_features='sqrt'
3	Random Forest	0.9692245	0.96280744	0.95252226	0.95763723	15/03/2024	max_features='sqrt', n_estimators=200
4	Random Forest	0.97030776	0.96347305	0.95489614	0.95916542	15/03/2024	max_features='sqrt', n_estimators=300
5	Support Vector Machine	0.97702644	0.96523276	0.97210682	0.9686576	15/03/2024	Default Settings (kernel='rbf', degree=3, gamma='scale')
6	Support Vector Machine	0.97312527	0.95038788	0.96715905	0.96315697	15/03/2024	kernel='linear'
7	Support Vector Machine	0.98157781	0.97281324	0.9768546	0.97482973	15/03/2024	kernel='poly'
8	Support Vector Machine	0.95036844	0.9401451	0.92284866	0.93141659	15/03/2024	kernel='sigmoid'
9	Support Vector Machine	0.98309493	0.97516263	0.97863501	0.97689573	15/03/2024	kernel='poly', degree=4
10	Support Vector Machine	0.98396186	0.97634536	0.97982196	0.97808057	15/03/2024	kernel='poly', degree=5
11	Support Vector Machine	0.98461205	0.97638725	0.98160237	0.97898787	15/03/2024	kernel='poly', degree=6
12	Support Vector Machine	0.98504551	0.97754137	0.98160237	0.97956766	15/03/2024	kernel='poly', degree=7
13	Support Vector Machine	0.98461205	0.97808057	0.97982196	0.97895049	15/03/2024	kernel='poly', degree=8
14	Support Vector Machine	0.63480711	crash	crash	crash	15/03/2024	kernel='poly', degree=7, gamma='auto'
15	Logistic Regression	0.97550932	0.96508876	0.96795252	0.96651852	15/03/2024	Default Settings (solver='lbfgs', max_iter=100)
16	Logistic Regression	0.97615951	0.96515062	0.96973294	0.96743635	15/03/2024	solver='sag'
17	Logistic Regression	0.97637625	0.96517119	0.97032641	0.96774194	15/03/2024	solver='saga'
18	Logistic Regression	0.97637625	0.96572104	0.96973294	0.96772283	15/03/2024	solver='saga', max_iter=200
19	Ensemble	0.98268147	0.97626113	0.97626113	0.97626113	16/03/2024	Best of above (RF, SVM, LR)
							Grid Searches
1	SVM	0.98461205	0.97751479	0.98041543	0.97896296	12/04/2024	{'C': [0.1, 1, 10], 'kernel': ['linear', 'poly', 'rbf'], 'degree': [3, 4, 5, 6, 7, 8]}
2	SVM (Expanded)	0.98461205	0.97751479	0.98041543	0.97896296	13/04/2024	{'C': [0.1, 1, 10], 'kernel': ['linear', 'poly', 'rbf', 'sigmoid'], 'degree': [1, 2, 3, 4, 5, 6, 7, 8], 'gamma': ['auto', 'scale']}
3	Random Forest	0.97009103	0.96345117	0.95430267	0.9588551	15/04/2024	{'n_estimators': [50, 100, 200, 300], 'criterion': ['gini', 'entropy', 'log_loss'], 'max_features': ['sqrt', 'log2', None],}
4	Logistic Regression	0.97269181	0.95934001	0.96617211	0.96274394	16/04/2024	{'penalty': ['l1','l2', 'elasticnet', None], 'solver': ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga']}
							Enrichments
1	Lemmatisation	0.98439532	0.97806758	0.97922849	0.97864769	17/04/2024	
2	Stop Word Removal	0.97897703	0.97037915	0.97210682	0.97124222	17/04/2024	
3	Combination	0.97724317	0.96800948	0.96973294	0.96887044	17/04/2024	
							Second Dataset
1	Second Dataset	0.95716667	0.93737803	0.94705405	0.9421912	18/04/2024	

Appendix B

Required Python Libraries:

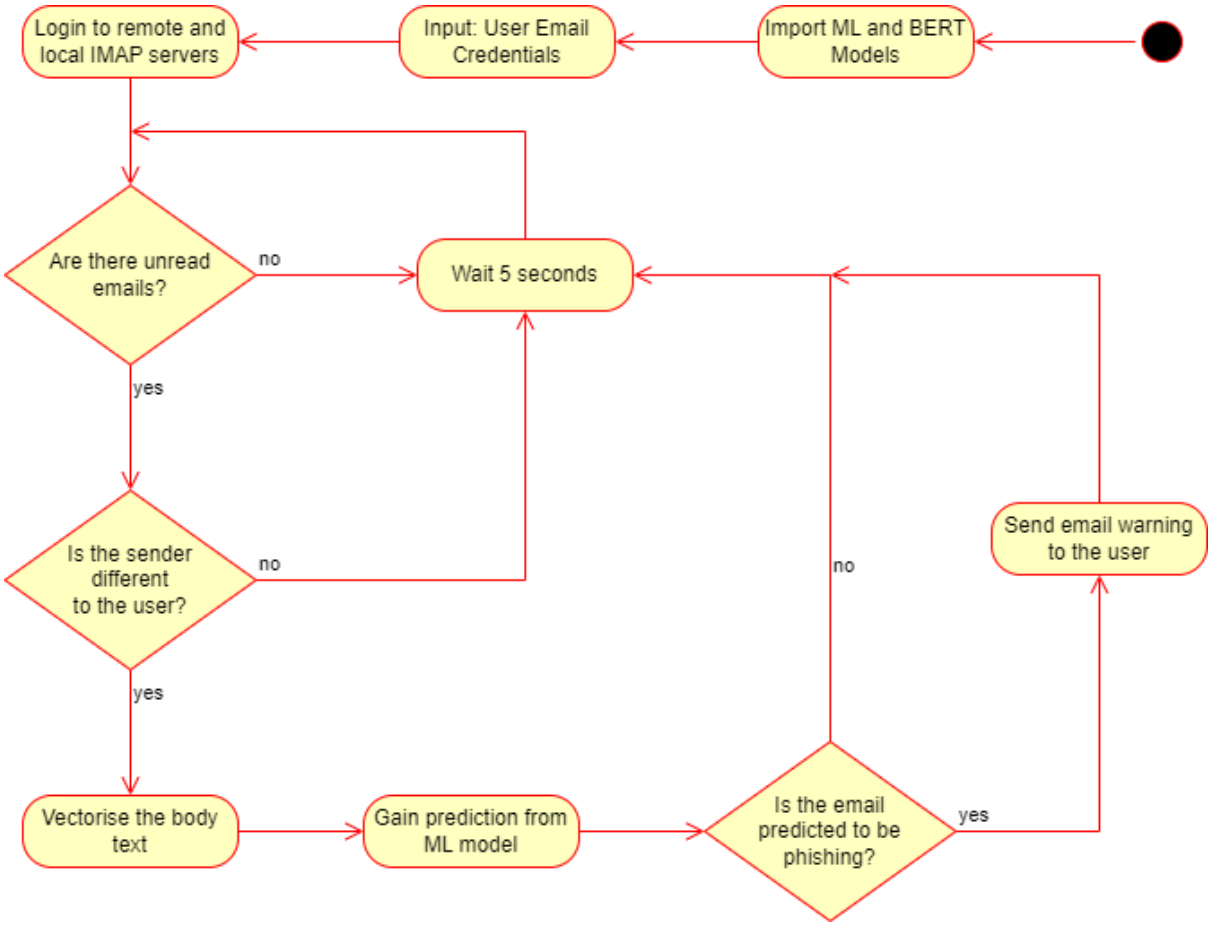
Format: library_name==version

- joblib==1.2.0
- matplotlib==3.6.3
- nltk==3.8.1
- numpy==1.24.1
- pandas==1.5.3
- scikit_learn==1.3.2
- scipy==1.13.0
- torch==2.2.1+cu121
- transformers==4.38.2

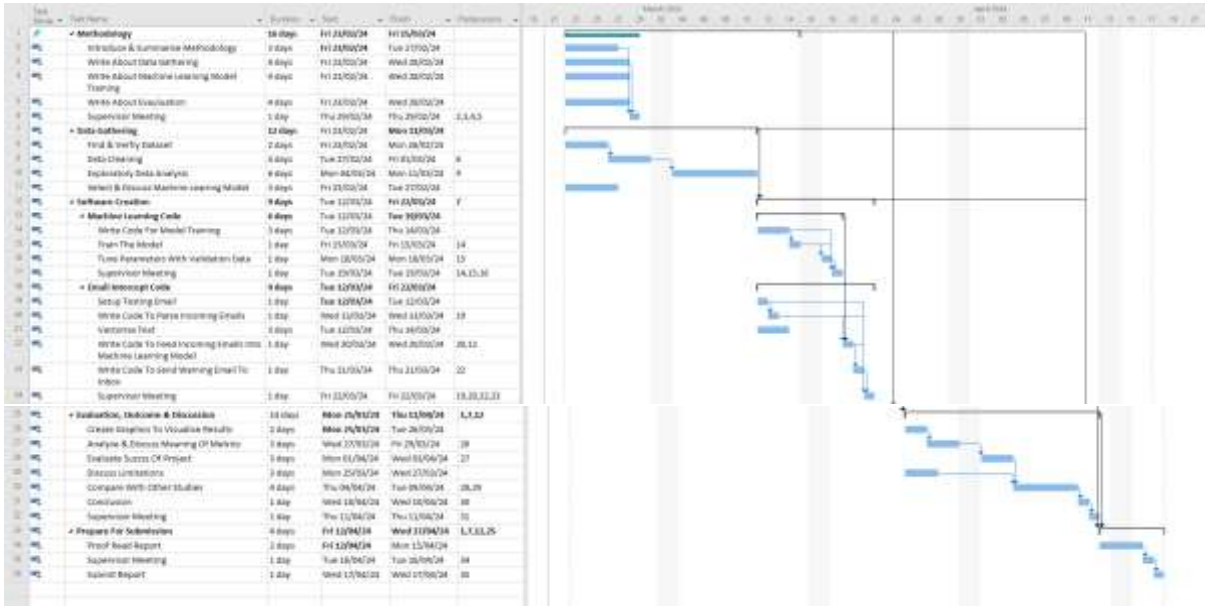
Please note the following:

- Many files within the artefact require file paths to be changed to function as desired.
- The machine learning scripts have certain functional sections commented out such as grid searches. If the grid search functionality is desired, please comment out the code from the best model (SVM) and uncomment the grid search while changing the estimator's variable name.

Appendix C

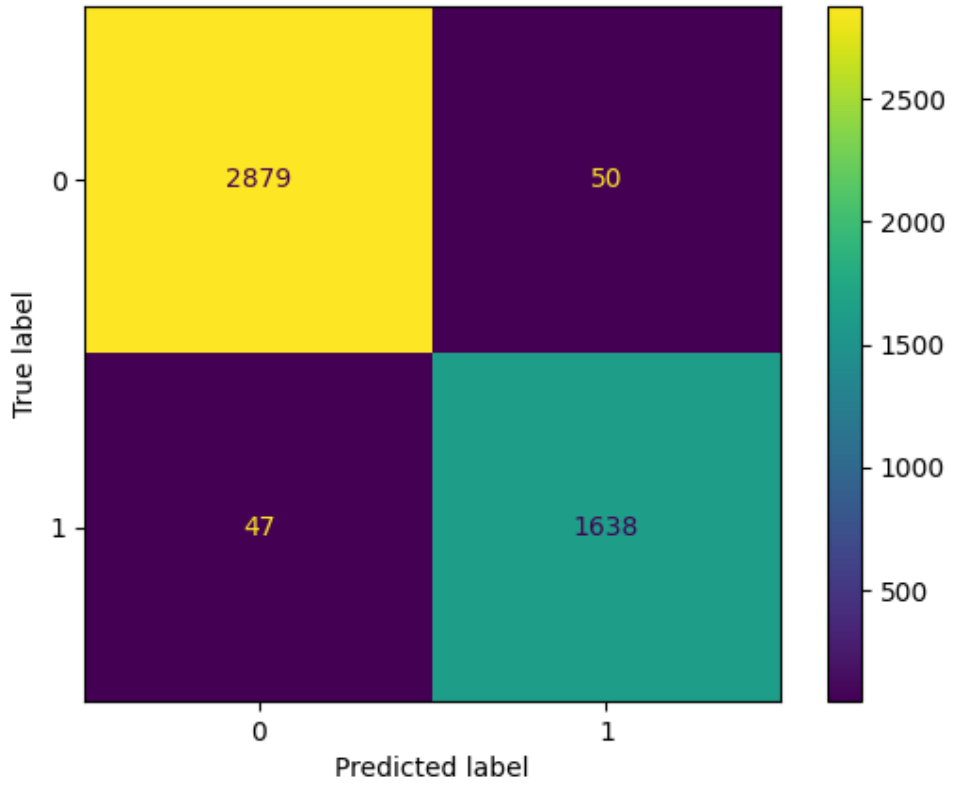

Login to remote and local IMAP servers
Input: User Email Credentials
Import ML and BERT Models
Are there unread emails?
no
Wait 5 seconds
yes
Is the sender different to the user?
no
yes
Send email warning to the user
no
Vectorise the body text
Gain prediction from ML model
Is the email predicted to be phishing?
yes

Appendix D

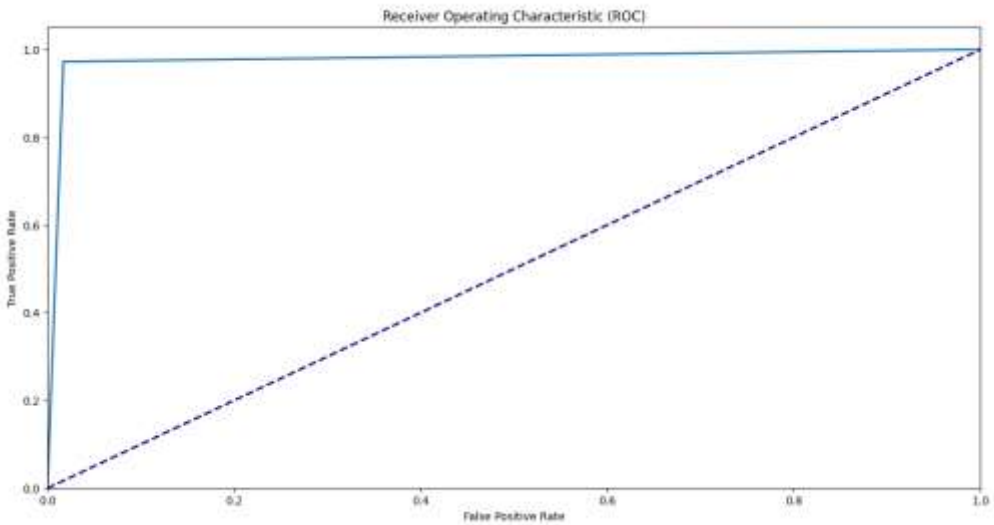


Appendix E

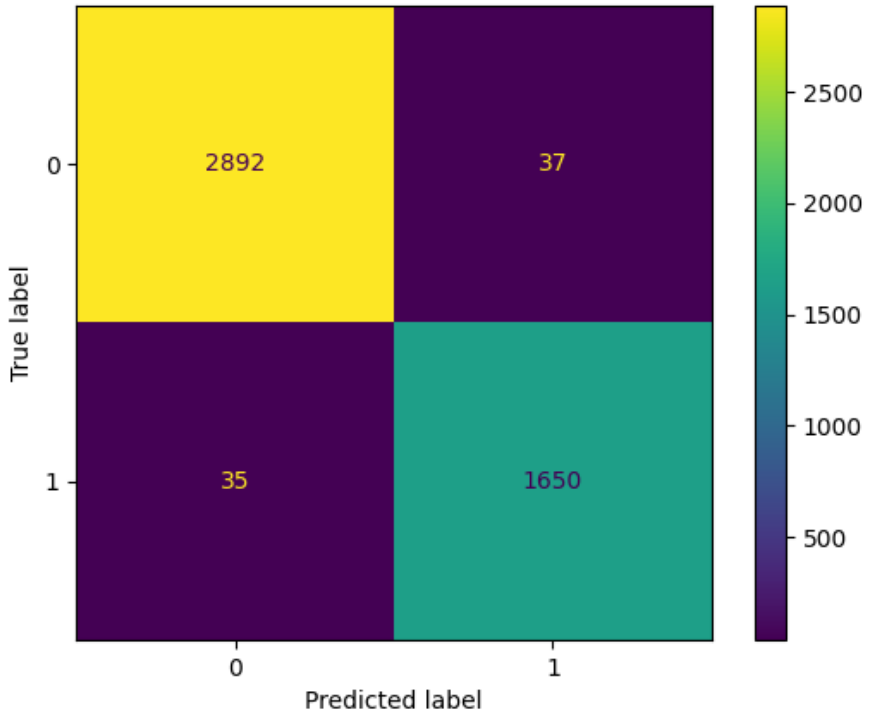
Stop word removal confusion matrix:



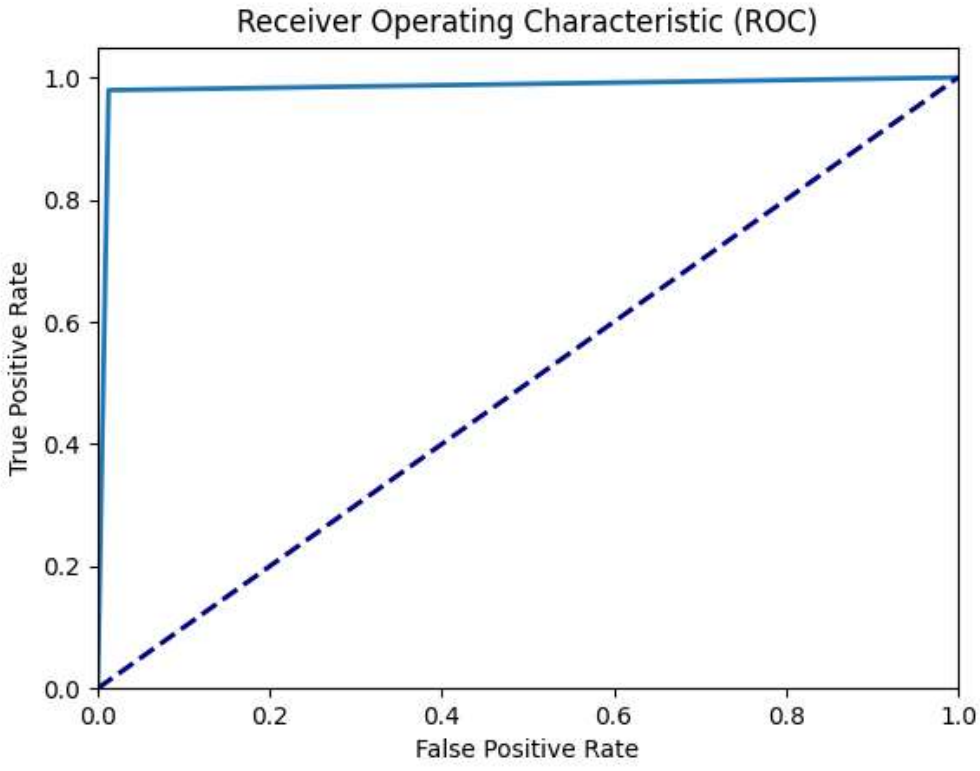
Stop word removal ROC graph:



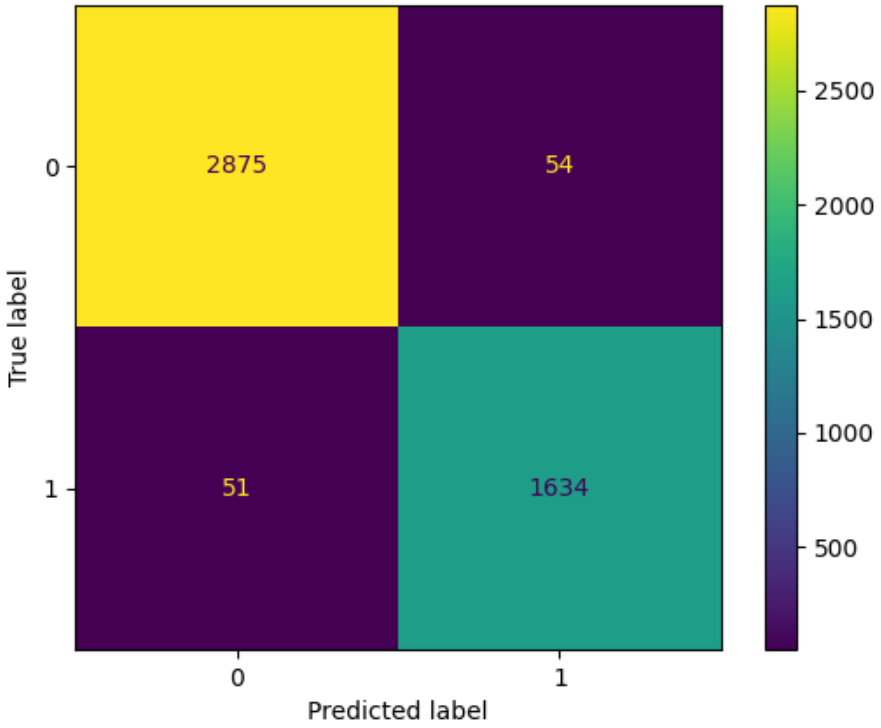
Lemmatisation confusion matrix:



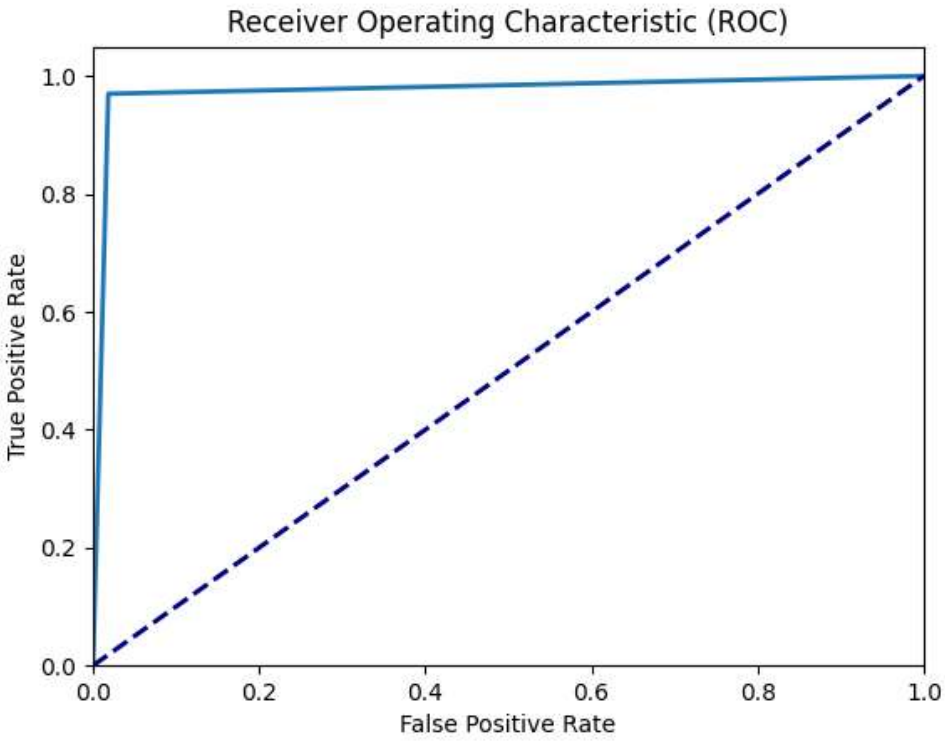
Lemmatisation ROC graph:



Stop word removal and lemmatisation combination confusion matrix:

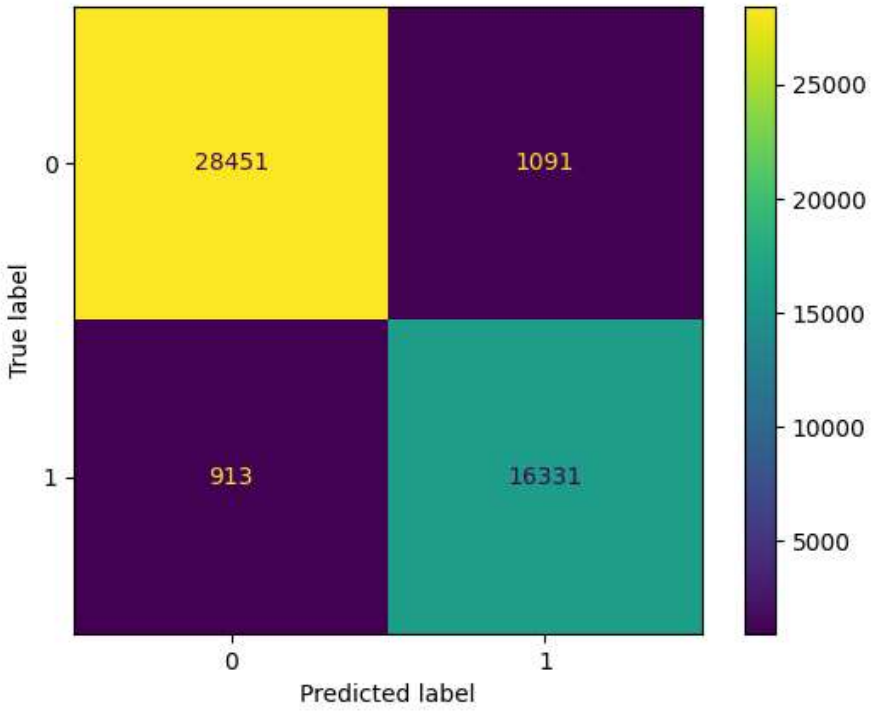


Stop word removal and lemmatisation combination ROC graph:

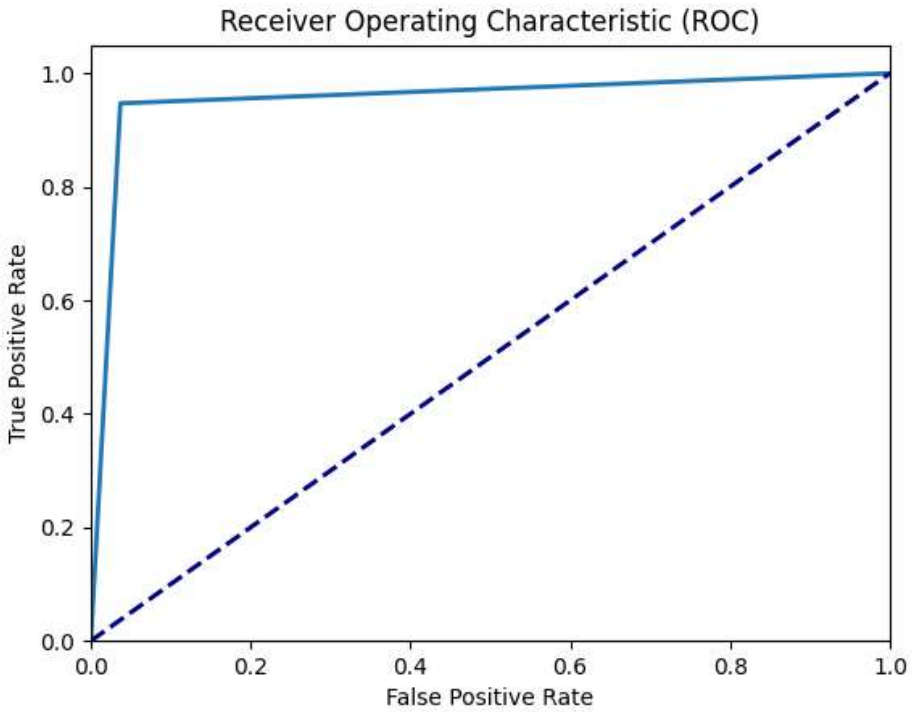


Appendix F

Second dataset confusion matrix:



Second dataset ROC graph:


Receiver Operating Characteristic (ROC)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0