Table of Contents

List of Figures

Abstract

Phishing is a significant cybersecurity threat that targets organisations as well as individuals. The aim of this project is to provide a comprehensive machine learning model that can accurately detect LLM generated phishing with high accuracy from a dataset of four different classes of emails: LLM phishing, LLM non-phishing, Human phishing and Human non-phishing.

This balanced and diverse dataset of 4000 emails acts as a real-world representation of the different types of emails that are sent daily that include different distinct features, allowing for an accurate feature differentiation from the classes of the dataset. The five machine learning algorithms that were used for this research are: Decision Tree, Support Vector Machine (SVM), Random Forest, Gradient Boost and K-Nearest Neighbours (KNN). These algorithms were tuned to evaluate the performance of the models after hyperparameter tuning. The highest accuracy achieved from the model before tuning was the SVM with an accuracy of 97.3%. The subsequent highly accurate models are Random Forest of 96.9%, KNN of 96.8% and Gradient Boosting of 96.7%. The model that achieved the lowest accuracy was Decision Tree, achieving an accuracy of 90.7%. Hyperparameter tuning was applied to models and the performance was re-evaluated to investigate if hyperparameter tuning enhanced the performance of the models. Other metrics such as precision, recall and F1-score were also measured.

The developed and trained models were then integrated with a web page developed using streamlit for a user-friendly interface for the classifications of the emails. Overall, this research aims to provide a framework for detecting LLM phishing emails. The results of this research signify that with the correct methodologies, we can enhance the detection of LLM generated phishing, contributing to robust defences against emerging cyber threats.

1. Introduction

1.1 Background Context

The rise and sophistication of technology has caused an advancement in cybercrime tactics that are constantly becoming more sophisticated and evolving to bypass current detection mechanisms. The utmost prevalent form of cybercrime, known as phishing, is a type of cybercrime that is categorised as a type of social engineering that aims to deceive recipients to reveal personal information leading to financial and data losses. Phishing can emerge in different forms such as SMS-Phishing, Voice-Phishing and Email Phishing. Despite their differences, they all share the same goal: to deceive the recipient for the attacker's personal gain.

In previous years, phishing tactics were more widespread, easy to recognize, and became fairly simple to identify them. According to Cardona (2024), phishing emails have evident features such as poor grammar and spelling mistakes that expose their fraudulent nature. Nonetheless, the constant sophistication of these phishing tactics have led emails to become more vigilant, with almost perfect spelling and grammar. This sophistication indicates that while poor grammar and spelling can affect the perception, it cannot be the sole reason behind the judgement.

1.2 Problem Statement

In recent years, Large Language Models (LLMs) have emerged and have been fairly used by many individuals for different purposes. Cybercriminals did not take long to utilise these LLMs for criminal activities, such as crafting phishing emails. These advanced models are capable of creating rather realistic, human-like emails that can be personalised to an individual where it seems legitimate.

According to Eze and Shamir (2024), LLM generated phishing emails presented exceptional grammatical accuracy and significant level of linguistic consistency. Contrairly, Cardona (2024) stated that traditional phishing emails showcased certain features such as grammatical and spelling errors. The evolution of phishing techniques, where LLM generated phishing demonstrates distinct features that differentiate them from human generated phishing, denotes that there is a need for an improved detection mechanism that can adapt to these new tactics.

This study will focus on developing a machine learning model that is able to effectively detect LLM generated phishing emails. A web page will be integrated with the developed model where an email text can be checked whether it is LLM Phishing, Human Phishing, LLM Legitimate or Human Legitimate.

1.3 Research Objectives

The primary objective of this research is to provide a comprehensive detection model that is able to detect LLM generated phishing emails with high accuracy. This model will then be used to develop a web page for the purpose of classifying the email messages to check if they are LLM Phishing, Human Phishing, LLM Legitimate or Human Legitimate. The specific goals to be achieved are:

- To develop and evaluate the performance of the machine learning model with the chosen algorithms: Decision Tree, Support Vector Machine (SVM), Random Forest, Gradient Boosting and K-Nearest Neighbours (KNN) on their ability to detect LLM phishing emails, Human phishing, LLM legitimate and Human legitimate emails.
- To evaluate the use of hyperparameter tuning to improve the accuracy of the developed models.

- To develop a web page that is integrated with the developed machine learning model for the purpose of email classification.

The evaluation of the developed model will be on the basis of these objectives. The existence of a machine learning model that detects traditional phishing email is acknowledged. However, due to the rise of new phishing tactics, such as LLM generated phishing, there arises a need for a model that is able to classify phishing emails based on the authors of these emails being either human or LLM. This will contribute to understanding the certain features that are exhibited in LLM phishing, therefore aiding defensive mechanisms that can be developed to detect and combat this issue. This paper aims to contribute a feasible and effective solution for dealing with complex forms of phishing in contemporary threats surrounding cybersecurity.

1.4 Purpose Of The Research

In a digital era where emails are the main form of communication between users and legitimate entities such as banks, service providers and employees, there is a need to ensure that cybersecurity measures are also constantly evolving and adapting to develop defensive methods against these attacks. LLM phishing is considered a novel concept and a current topic, therefore influencing this research to contribute to the development of appropriate and effective measures against this type of phishing. Phishing emails, if mistaken for legitimate emails, can contribute to major issues such as financial losses, identity theft and sensitive information disclosure. Therefore, developing a robust and accurate machine learning model that can detect these types of emails is crucial.

1.5 Research Methodology

This research will follow a multi-class, supervised approach. The dataset, that was sourced from Kaggle, that will be used in the development of this model is a dataset that includes LLM phishing, Human phishing, LLM legitimate and Human legitimate. Appropriate preprocessing techniques will be applied to remove any meaningless information that does not provide any contribution to the trained model, to enhance the performance of the model. Subsequently, the data will be trained using different algorithms such as: Decision Tree, SVM, Random Forest, Gradient Boosting and KNN. Metrics such as accuracy, precision, recall and F1-score will be measured to assess the performance of the models. Hyperparameter tuning will then be utilised to further evaluate the model's performance. Lastly, a web page will be developed that is integrated with the machine learning model for the purpose of providing a user-friendly interface for email classification.

1.6 Organization of Research

This research has been thoroughly organised to provide a structured approach for the purpose of guiding the reader through the chapters of this study. The outline of this study is explained below:

1. **Chapter 1: Introduction:** This chapter introduces the background of the topic, problem statement, research objectives, purpose of this research, research methodology and the organisation of this research.
2. **Chapter 2: Literature Review:** This chapter discusses relevant literature and previous work that has been conducted in this research area. This aims to identify the gaps that this research will address.
3. **Chapter 3: Methodology:** This chapter discusses the detailed approach that was undertaken in this research, including the data used and the evaluation metrics.

4. **Chapter 4: Results:** The findings of the research are presented in this chapter. A comprehensive analytical discussion is included in this chapter where results are discussed.
5. **Chapter 5: Discussion:** A thorough decision of results with the support of relevant literature is presented in this chapter. The findings are compared and contrasted with existing literature. Limitations are also addressed in this chapter.
6. **Chapter 6: Conclusion:** This chapter provides an overview of the whole research and suggestions for future work

This structure aims to provide a structured approach to understanding the fundamental concepts of LLM phishing detection using machine learning models. These concepts are then built with the progression of chapters to the overall methods used to achieve the results.

2. Literature Review

2.1 Introduction to LLM Phishing Emails

The accuracy of machine learning models in detecting phishing emails has been significantly studied. Phishing emails and machine learning models that are used for the purpose of detection is not considered a new concept. However, with the emergence of Artificial Intelligence, cybercriminals tend to use these technologies in crafting phishing emails that make them harder to detect as phishing due to the sophisticated nature of these emails, according to Bethany et al. (2024). A study by Francia et al. (2024) shows that there is a clear insignificant differentiation between human generated spear phishing and AI generated spear phishing, where AI generated spear phishing emails performed better than the human-generated spear phishing by 19.9% above 50%. A study by Francia et al. (2024) investigated the effectiveness of spear phishing messages that were crafted by GPT-4 compared to those crafted by humans. No significant difference was observed with the click rates between AI generated spear phishing and human generated, even though the perception of AI messages appeared more convincing. Moreover, AI spear phishing created more personalised and highly convincing messages compared to human spear phishing by 80%. A supporting study by Greco et al., (2024) highlighted that AI can create human-like messages that assist cybercriminals in generating phishing emails that are highly effective at deceiving recipients by providing convincing messages. AI-generated phishing emails can produce many convincing phishing emails in a fraction of the time that it would take a cybercriminal to generate them manually, allowing these emails to become more widespread to reach and deceive more people. An article by Petrosyan (2023) revealed that AI-generated phishing emails deceived 65% of the users into disclosing personal information, where human generated emails were successful in deceiving 60% of the users. Figure 1 shows that there is a 5% difference from AI

generated emails to human generated emails when deceiving users, however AI is constantly advancing and becoming more sophisticated with time to successfully deceive a higher percentage into believing that an email is genuine.

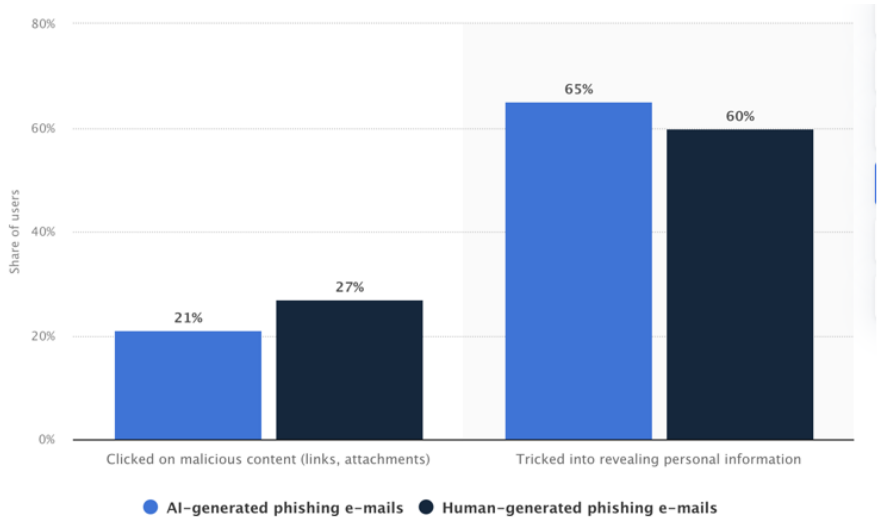

Figure 1: AI-generated phishing emails from Human-generated phishing email (Petrosyan, 2023)

2.2 Advancements in Phishing Detection Models and Approaches

Jamal et al. (2023) proposed a novel workaround for the phishing, spam, and ham email detection through the newly developed improved phishing spam detection model, namely IPSDM that uses an approach where translators like DistilBERT and Roberta, upon fine-tuning can be utilised optimally. The study notes the longstanding issue of phishing and spam filtration, two threats with major financial and resource consequences for the global community. It can be said that the idea of the proposed work is based on the use of the abilities of LLMs, and the use of the transformations based on the models on the transformer basis to increase detection accuracy. The authors gathered a set of phishing, spam, and ham mails, where they deal with the class imbalance issue using the adaptive synthetic sampling. Their methodology involved pre-training followed by fine-tuning of the DistilBERT and Roberta models to work on this dataset. The current findings revealed that the

proposed IPSDM significantly boosts the performance of the basic models in balanced and imbalanced cases. The findings of this work attest to the seemingly endless possibilities of LLMs and transformer-based models in the improvement of email security and lays a solid framework for further studies on the said field.

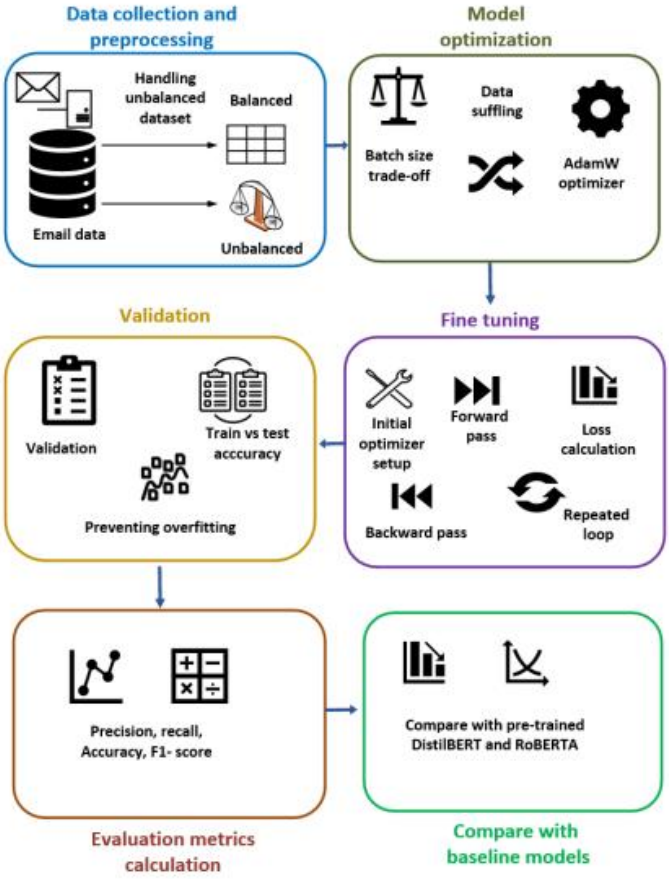

Figure 2: Overall methodology. Illustrates data collection and preprocessing, model optimization, fine tuning, evaluation metrics (Jamal et al., 2023)

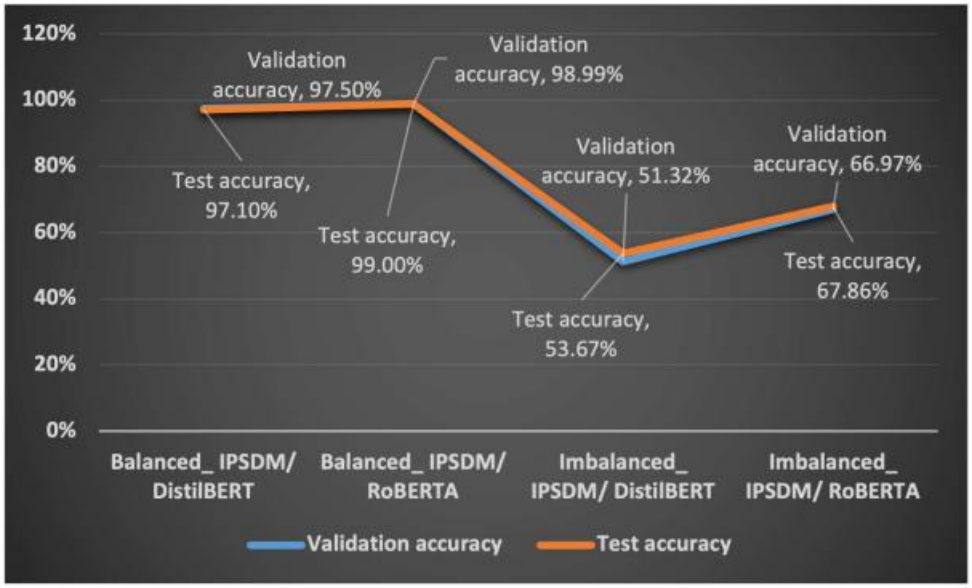

Figure 3: Validation vs test accuracy graph (Jamal et al., 2023)

Phishing attacks remain widespread and have developed into rather complex threats that are aimed at exploiting people's weak spots to take unauthorised control over personal data. Modern development utilises machine learning and natural language processing (NLP) in the improved detection of phishing that has made them more challenging to pull off. The research (Chinnasamy, 2024) presented the concept of designing an AI-based solution for phishing detection. This solution involves the use of Decision Trees and Naive Bayes algorithm for better classification of phishing URLs. Their research uses labelled datasets for the training of models, known as a supervised learning approach, and it aids in a powerful distinction between the legitimate and the fake websites. Moreover, this system also integrates feedback and reinforcement learning as mechanisms of learning to adapt to new strategies used in phishing and consequently, the model adapts to new forms of phishing improving the detection accuracy. Similarly, the literature presents other techniques, namely the deep learning models of CNN and LSTMs, which have proved to have high precision in identifying phishing sites and emails. The comparative analysis within the research shows that the use of the Random Forest algorithm both in the case of the accuracy and

the precision of the analysis distinguish the effectiveness of this method for solving complex problems of phishing. In summary, the approaches incorporated with AI in the existing paper have come up with a strong model for enhancing cybersecurity against phishing, hence a strong declaration of the role of AI in shaping the future of cybersecurity.

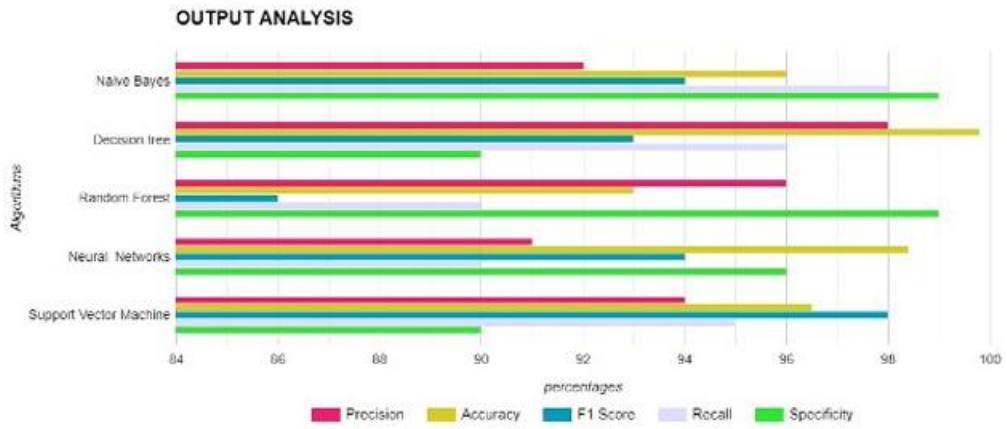

Figure 4: The Overall Analysis of Various Parameters (Chinnasamy, 2024)

The research (Thakur et al., 2023) provides an overview of using deep learning for the detection of phishing emails and several issues that have been identified in recent research. Deep learning techniques are seen to have the ability to predict the phishing threats, however, restrictions like small datasets and the absence of explicit dataset characteristics prevent the assessment of model performance. The problem has remained to be the classification of phishing emails as false positives and false negatives. Prior research has mainly concentrated on the formal features of emails while neglecting other significant components such as the sender image (multimedia) and behavioural history. The limited applicability to the English language and moderate addressing of complex phishing methods specifically reduces the models' efficacy. Furthermore, the limitations arising from the usage of the old or not very diverse data sets and the improper tools make the models unable to counterbalance new trends in phishing strategies. The gaps in the current literature should be filled in future studies by employing more private approaches, using larger

collections of data, and fine-tuning the selection of the features. There is also a need to expand the analysis of the email content to go beyond words, such as multimedia and new emerging spam strategies and reconsider models regarding present day phishing and concept drift. Comparing them with the modern approaches and making a proper fine-tuning of the hyperparameters could increase the detection accuracy dramatically. Therefore, the enhancement of techniques for the development of real-time processing systems and the combination of the various kinds of algorithms used in machine learning can enhance the effectiveness of the approach to fight phishing, providing better cybersecurity solutions for various complex and constantly changing environments.

2.2.1 Comparative Analysis

When comparing the given articles, the literature review shows considerable progress in the sphere of phishing email identification, main methodologies used and the efficiency. DistilBERT and Roberta are transformer-based models, and Jamal et al. (2023) study shows that they can be used, combined with oversampling, to address the imbalance problem and significantly increase detection rates. Yet, the strategy employed in the study comes with some drawbacks since the exploitation of adaptive synthetic sampling and model fine-tuning may not efficiently manage the dynamic nature of phish threats. The research (Chinnasamy, 2024) presents the implementation of AI approaches such as Decision Trees and Naive Bayes with feedback mechanisms and reinforcement learning in view of new phishing strategies. However, the focus of the study on supervised learning and conventional methodologies may confine it to slow reacting to new forms which phishing exacts frequently. Reviewing the work, (Thakur et al., 2023) has identified the flaws of the current deep learning models, such as problems with small sample datasets and the selection of features, which cause false-positive and false-negative results. These and other such

observations have forced the review and underlined the need for more elaborate datasets, enhanced features, as well as real-time processing systems to arrive at a higher degree of accurate detection. Overall, each of the works under consideration provides a significant contribution to the subject and reveals that the literature points to the need for the further improvement of the approaches and the implementation of novel technologies for preventing highly sophisticated phishing attacks.

2.3 Comparative studies on Human vs AI Phishing Emails

Recent studies have shown that AI can be used for the writing of phishing emails, where they seem believable, but mostly they are not able to surpass the effectiveness of human generated phishing emails. Research by Eze and Shamir (2024) also investigated if automatic text analysis methods are able to detect AI generated emails better than human phishing. It was found that automatic text analysis is able to identify AI generated phishing emails with high accuracy compared to traditional, human written phishing emails. The messages generated by both AI and humans are compared using a rigorous method, where a combination of different methods was used, and a qualitative and quantitative analysis was performed. This kind of approach focuses on showing how the message length, style, and personalization can impact the credibility and effectiveness of the message. The use of these methods produced promising results, with an accuracy of 99.3% when identifying AI phishing emails. The study highlights the development of AI capability in creating more persuasive messages and the implied significance of literacy and constant wariness of cyber threats.

	AI-generated	Enron	Ling-Spam	Nigerian phishing
AI-generated	100	0	0	0
Enron	0	97	0	3
Ling-Spam	0	4	96	0
Nigerian phishing	0	0	0	100

Figure 5: Confusion matrix of the classification results of the AI-generated emails, the Enron emails, the manually crafted Nigerian phishing scam emails, and the Ling-Spam emails. The numbers show the percentage of total number of test samples from each class (Eze and Shamir, 2024)

New approaches have been introduced in identifying the differences resulting from original human-written content and AI-written content. The research (Greco et al., 2024) analysed and compared several machine learning architectures regarding authorship of phishing emails where the focus is put on simpler models than the deep neural networks. The authors employed a set of 1000 genuine from "Nazario" and "Nigerian Fraud" dataset and 1000 emails produced with WormGPT, to evaluate the models' effectiveness and tested Random Forest, SVM, XGBoost, Logistic Regression, KNN, Naïve Bayes, Neural Network with and without transfer learning. On their findings, they showed that the best model the experiment delivered was the Neural Network with transfer learning model, with a correct classification rate of 99%. The accuracy of Random Forest comes to 98.16%, the accuracy of SVMs and Logistic Regression is 99.20% and 99.03%, respectively. Although the focus was made on the outstanding performance of the Neural Network model, the study also underlined the practical benefits of applying the Logistic Regression since this model is easy to interpret and computationally less demanding. Accordingly, Greco et al. (2024) the obligation of ensuring how the users understand the approach to distinguishing phishing emails, pointing out that the chosen Logistic Regression is easy to explain to a computerised system at an informational level. The research suggests that further studies are recommended to expand the sets of features, to consider multi-class classification, and also to understand the most

common features of phishing emails that are not technical so that they can be easily explained to individuals without technical knowledge.

Table 1
Average accuracy throughout the repeated 10-fold cross-validation

	Random Forest	SVM	XGBoost	Logistic Regression	KNN	Naïve Bayes	NN (Transfer Learning)	Neural Network
Average	98.16%	99.20%	97.50%	99.03%	97.67%	94.10%	99.06%	99.78%
Standard Deviation	0.0103	0.0057	0.0108	0.0055	0.0105	0.0165	0.0062	0.0034

Figure 6: Avg Accuracy for 10-Fold Cross Validation (Greco et al. 2024)

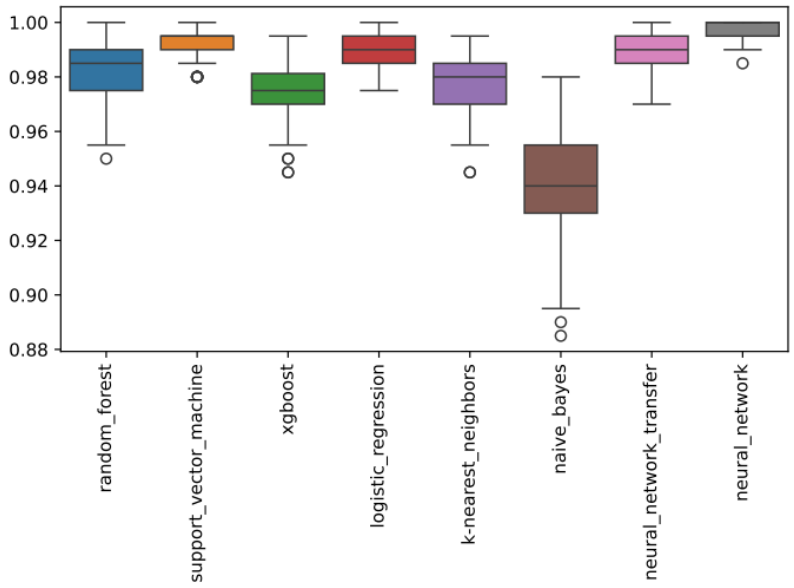

Figure 7: Box plot of the accuracy of each model for each cross-validation fold (Greco et al. 2024)

The sophistication of phishing attacks is rapidly increasing especially in the light of LLMs. The research by Bethany et al. (2024) had a strong study comparing lateral phishing emails written by humans with those composed by the LLM in organisational context. The research, which took approximately 11 months and involved about 9,000 employees of a large public university,

indicates that first, human generated messages are more captivating to the recipient. It establishes a significant variation in the outcome is dependent on the relevancy of the content of the phishing emails. Nonetheless, emails generated by LLM are notably weak in contexts that relate to less sensitive areas compared to the human-written emails. Moreover, the study also assesses the effectiveness of LLM in the generation of phishing in case of internal and external organisational information where it shows that phishing emails with internal information gain higher engagement than those with external contents. Regarding detection and defence, the work compares different machine-generated text detection models, noting that fine-tuning these models has a relative impact on enhancing the current models' performance in detecting LLM phishing emails. Text-to-Text Transfer Transformer -LLM Email Defence improved the F1 score of 98.96. This study thus fleshes the current knowledge of the phishing strategies along with escalating calls for raising awareness and developing the countermeasures to effectively curb the dangerous LLMs.

Model	Data Source	P	R	F1
BERT-Defense	WebText/GPT2-Large	33.3	15.0	20.7
RoBERTa-Defense	RealNews/GROVER	66.0	68.2	67.1
GLTR-BERT	WebText/GPT2-XL	100	22.4	36.6
GLTR-GPT2	WebText/GPT2-XL	94.4	3.4	6.6
BERT-DefenseEmail	LLM Generated Emails	87.6	92.0	89.8
RoBERTa-DefenseEmail		97.1	100	98.5
GLTR-BERTEmail		86.5	90.0	88.2
GLTR-GPT2Email		80.0	72.0	75.8
T5-LLMEmail Defense (ours)	LLM Generated Emails	99.03	98.89	**98.96**

Figure 8: Performance metrics of existing base detectors, fine-tuned versions of the existing base detectors, and proposed detector with respect to the machine class (Bethany et al. 2024)

2.3.1 Critical Analysis

The literature review synthesises the effectiveness of human and AI-generated phishing emails, and the factors encountered in the differentiation process. The findings indicate a nuanced view: thereby, even though the use of AI in phishing messages is inexpensive and even convincing, it is

highly unpredictable, its efficiency will greatly depend on the specific context of employing, as well as the algorithms of AI. Eze and Shamir (2024) point to the general problem in the distinction between the contents created by AI and human messages, calling for better media literacy as well as people's resistance to cyber threats. Greco et al. (2024) also adds value by showing that, while neural networks, trained through transfer learning show very good performances in training a classifier for phishing emails, simpler models such as Logistic Regression are however also beneficial as they are easy to explain and impose low costs in terms of computational resources. Bethany et al. (2024) examines how LLMs enhance the employment of phishing attacks and while emails originating from the LLM may not be so effective in other lower risk environments, it remains an effective threat where internal organisational information is at stake.

This review collectively emphasises the enhancement of the complex detection system and awareness level for preventing new trends in phishing that is being encouraged by AI technologies.

2.4 Machine Learning Techniques for Phishing Detection

Phishing attacks which are sharply deceptive continue to remain a big threat in modern technology due to their continuing enhancement in their features. Thus, the study by P.M et al. (2023) shows that phishing is still a continual threat; it also discusses the use of more progressive forms of machine learning to address the problem. It captures the increase of phishing attacks through email, phone calls, fake websites and therefore the need to develop better detection techniques. The authors suggest the following vast model that incorporates four machine classifications, namely Random Forest, XGBoost, and Logistic Regression in identifying phishing sites. Through a series of experiments on a dataset collected from PhishTank and consisting of phishing URLs, the effectiveness of these algorithms in correctly classifying between the safety or otherwise of a site

is determined. It is observed that the highest accuracy of 94.2% is being achieved by XGBoost, which validates the findings of the study which holds a better position compared to other models in this regard. This work advances the knowledge in the area by looking into the various strategies of phishing attacks and proposing an effective approach to the problem using the machine learning classification technique. The findings support the published research pointing out that machine learning can improve the general effectiveness of phishing identification. Altogether, this research supports the increasing significance of machine learning in the prevention and fight against phishing and emphasises the necessity for the further developments of detections' approaches.

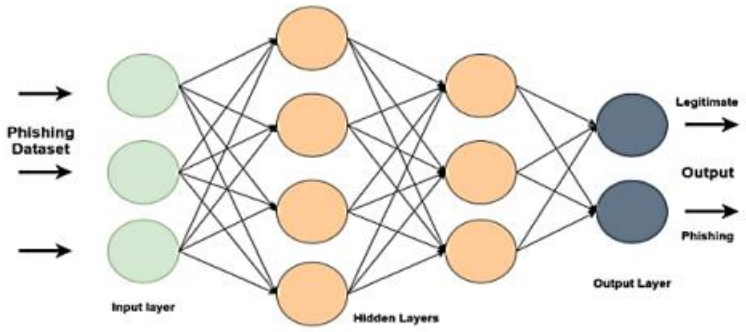

Figure 9: Machine Learning model for Phishing attack detection P.M et al. (2023)

The Role of Large Language Models

The paper published by (Hazell, 2023) focuses on the relationship between the LLM and spear phishing which has numerous consequences in the sphere of cybersecurity. The paper shows how LLMs can help improve the effectiveness of cybercrimes, specifically spear phishing by automating and multiplying them. In Hazell's investigation, GPT-4 pose high risks since they can help generate highly personalised phishing emails hence lowering the cognitive workload, costs, and skill set needed by the attacker. It would also subvert and facilitate the generation of plausible emails easily and at a significantly lower cost. Moreover, investigations on how LLMs can be utilised in enhancing the formation of rudimentary malware, which degrades cybersecurity even

more. This gives rise to questions as to whether LLMs are not a double-edged sword, with their helpful uses like in customer service and content generation on the one hand and able to be weaponized for cyber crimes on the other. The study demonstrates that the problem of controlling misuse arises from the very nature of the governing system, because AI solutions are created to perform a vast number of tasks, which makes it extremely difficult to prevent their unlawful use without limiting beneficial application. In addition, it outlines other possible solutions such as structured access schemes and use of LLM-based defensive systems while stating that the solution to governance of AI should be balanced taking into consideration the advancing capabilities of AI. Thus, investigation strengthens the need for constant safeguard and the application of new types of approaches in counteraction of risks, connected with the progress of AI technologies.

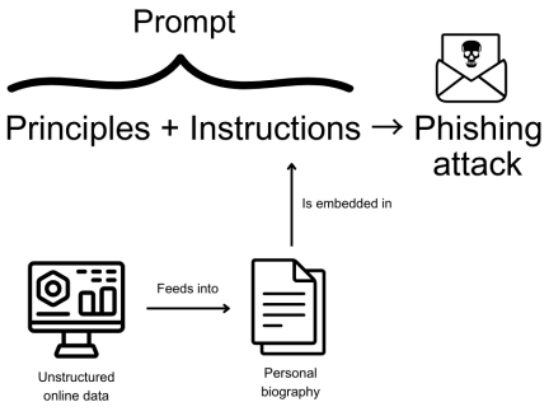

Figure 10: A demonstration of the collect and contact phases of an LLM-based spear phishing attack (Hazell, 2023)

2.4.1 Critical Analysis

The two sides of the technologically enhanced features are discussed, especially regarding phishing attacks and the use of machine learning. The study by P.M et al. (2023) supports different machine learning models and claims that Random Forest, XGBoost, and Logistic Regression models are effective in identifying phishing sites with the highest result found in the XGBoost

model. This suggests a new way of enhancing phishing by using advanced classification methods as indicated. However, while these models demonstrate improved detection rates, the review also reveals a critical concern addressed by Hazell (2023), the concept of twinning and the possibility of using LLMs both for ‘benign’ uses such as language translation, and for malicious uses, such as the creation of deep fakes. In this case, therefore, by spear phishing, Hazell demonstrates how LLMs like GPT-4 can be used to develop toxic solutions built on the very ambition of creating safer worlds, which only deepens the contradiction.

The proposal of this dual perspective is further balanced AI governance and an ongoing process of the cybersecurity approach adjustment to the opportunities and threats connected with the advancement of technologies. Hence, the review captures the virtuous relationship between innovation and security while noting that promising new solutions based on machine learning could also cause new security problems if not well monitored.

3. Methodology

This chapter will be describing the dataset that was used in this project as well as the methodology followed. This dataset consists of a combination of 4000 LLM phishing, Human phishing, LLM legitimate and Human legitimate emails. The machine learning model was developed using a supervised learning approach. The algorithms that were used for the machine learning model were

Decision Tree, SVM, Random Forest, KNN and Gradient Boosting. Hyper-parameter tuning was performed on all the algorithms used to further optimise the models performance. The goal was to achieve a model that has an accuracy of over 90% in detecting LLM generated phishing emails. The dataset is a multiclass balanced dataset. The emails are then preprocessed and transformed into feature space through TF-IDF vectorization, while Truncated Singular Value Decomposition (SVD) allows to reduce text data's dimension to improve model's performance.

3.1 Dataset Description

The dataset that was used for this project was a multiclass dataset of 4000 LLM generated and Human generated emails that were in the context of both legitimate and phishing nature. Human generated emails were from the "Nazario" and "Nigerian Fraud" dataset that included 1000 legitimate emails and 1000 phishing emails. The LLM generated emails were generated by ChatGPT and WormGPT that also included 1000 legitimate emails and 1000 phishing emails. This dataset is publicly available on Kaggle "Human-LLM generated phishing-legitimate emails". The dataset is a balanced dataset, where there is an equal representation from all four classes. It is beneficial to have equal proportionality of Human Legitimate, Human Phishing, LLM Legitimate, and LLM Phishing. It is very important to have a balanced dataset in the context of training a machine learning model to detect LLM generated phishing emails because a balanced dataset ensures that the model is not biassed towards any "dominant" class. No additional work was performed to balance the dataset as it was already balanced.

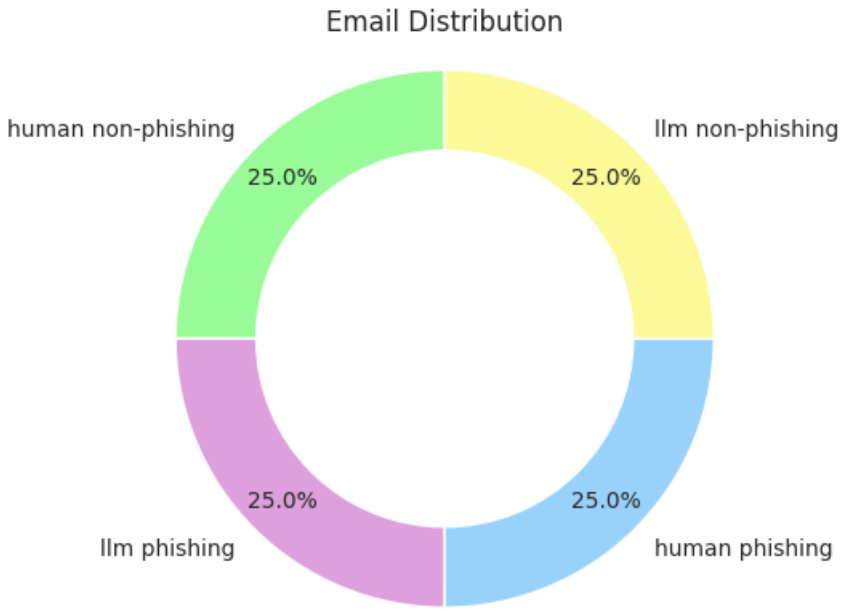

Figure 11: Dataset Distribution

The dataset was labelled as: 0 for human non-phishing, 1 for LLM phishing, 2 for human phishing and 3 for LLM non-phishing. This is known as a supervised learning approach. This allows for the model to understand the features of each class, where the model is able to make clear distinctions between the classes making it easily interpreted and processed by computers and any other algorithms based on machine learning.

This dataset was employed in the paper (Greco et al., 2024) . This paper was recited as a speech in the ITASEC 2024 conference on cybersecurity, with the title, “Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails.” Due to its diverse scope, the given dataset can become useful in training and testing ML algorithms focused on phishing emails’ identification and the ability to distinguish between human-generated and AI-based emails.

3.2 Text Preprocessing

The text preprocessing stage is a crucial stage in machine learning as it determines how effectively the model will learn and how accurate the predictions will be. It is the process of transforming raw data into a suitable format for model training. The transformation process of the email data can be divided into several essential steps to get the texts ready for further analysis and modelling. The initial step of the text preprocessing is to check if the text is a string. All the texts are converted to lowercase to make the set of texts in the emails more unified. The cleaned text was used to tokenize it into individual words, and this made it easy to analyse the text thoroughly.

Preprocessing is an important step in the whole process of ML, especially for the NLP tasks, including the classification of emails. This phase aims to pre-process the raw textual data into a format that can be fed into the machine learning algorithms so that the raw data is most suitable for model discovery of useful patterns, since the classification of LLM phishing emails requires a rigorous process of extracting patterns and features for differentiation.

Text Cleaning: This stage consists of the preprocessing techniques where the data from the dataset is preprocessed using different forms to transform the raw data into a clean, structured format that is suitable for model training.

> **Converting the text to lowercase:** This step in text preprocessing ensures that “Urgent” and “urgent” are treated the same. They both have the same meaning but one is capitalised and one is lowercase, so transforming all the text to lowercase ensures standardisation.
>
> **Remove extra whitespaces:** This step strips the extra white spaces since they can cause inconsistencies in the data. Data would be treated differently due to the extra whitespace. Removing extra whitespaces ensures consistency and reduces noise.

Expanding contractions: The function expand_contractions expands contractions such as “don’t” and “do not”. This step ensures the text is uniform making it easier for the model to interpret.

Reduce repeated characters: This step ensures that words that include repeated characters such as “cooool” are reduced to two consecutive letters “cool”. This eliminates noise which is good for a better performing model.

Removing punctuation: Punctuation is considered as noise and does not provide meaningful contribution to the model so removing punctuation reduces the noise in the data.

Replace newlines and tabs with spaces: This step replaces any multi-line or tabbed text into a single line, making the text more uniform and easier to interpret.

Tokenization: The text is tokenized into individual words. This step is performed because there is a need to remove stop words from the text in the email data. The sentence is converted into tokens where it is checked for any stop words.

Removing stop words: Stop words are removed in this step in order to allow for a reduction of dimensionality as well as focusing on more important words that are meaningful.

Lemmatize: During this step, the words are converted into their base form to achieve consistency and reduce the vocabulary size.

This preprocessing pipeline ensures that only the key concepts of the email body are used for analysis while at the same time retaining all the information that is relevant in the detection of LLM phishing email whilst reducing the noise.

3.3 Word cloud

Word clouds are visual representations of the frequency of a particular word in a given dataset. This enables for a better understanding of the common words in the dataset therefore a better distinction of the common words that are used for each of the classes. The bigger the words in the word cloud the more common they are.

As shown in Figure 12, the most common words in the context of the dataset used in this research that consists of: Human non-phishing are submission id, added. Similarly, the common words LLM non-phishing are account, find, please. In case of LLM phishing, the words please, click, link, account are the most common, which shows that it includes a sense of urgency. Lastly, Human phishing includes words such as message, account, click and payment.

In the context of LLM phishing and non-phishing, similar words are present in the word clouds. This shows that there could be certain difficulties in distinguishing between LLM phishing and LLM legitimate, and a possibility of misclassifications between the two if the specific distinctive features are not identified.

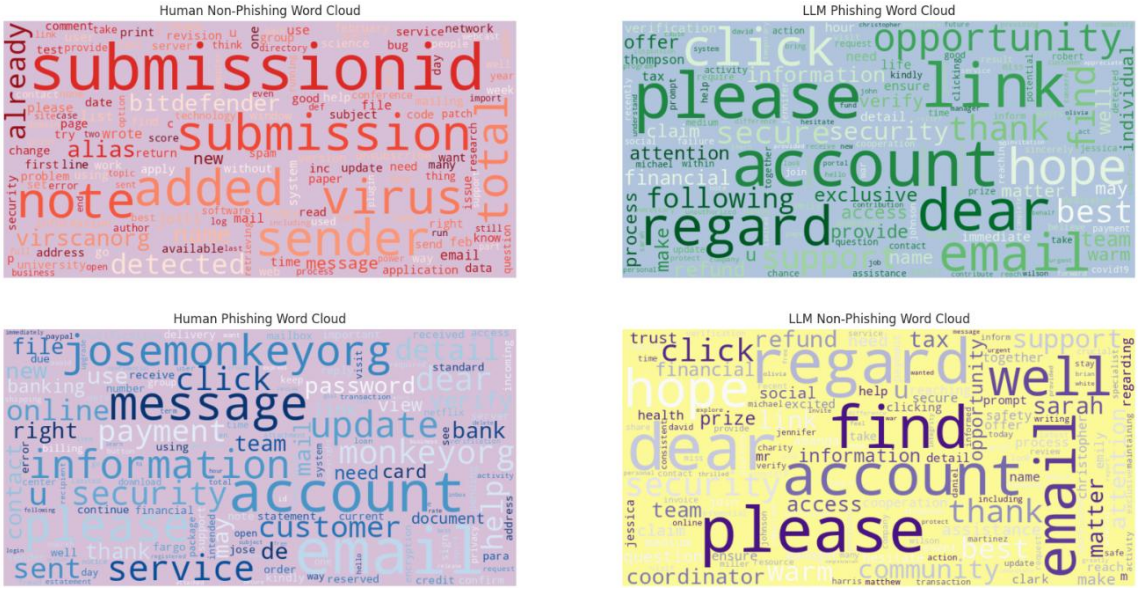

Figure 12: Word Cloud of Human and LLM Generated Data

3.4 Data Splitting

Further enhancing the convenience of the models' training and testing; the given dataset was divided into feature matrix X (comprising the processed textual data) and target variable from Y (the corresponding labels). To split the dataset the train_test_split function from sklearn package is used. As decided in model_selection, the dataset was split to a training set (70%) and testing set (30%). This allows the model to have enough data to be trained on, and the remaining data to be tested on. To make the results reproducible, a 'random' state was fixed to bring consistency in the assessment across different runs.

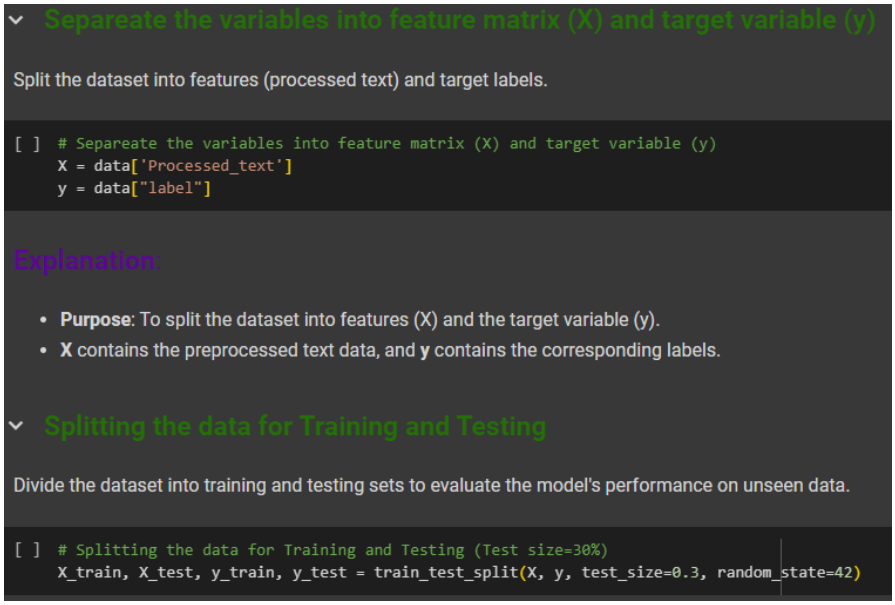

Figure 13: Data Splitting for Training and Testing

3.5 Text Vectorization and Dimensionality Reduction

Vectorization is an important process in text classification where the raw text data is transformed to numerical form that is understandable by the ML algorithms. For this research, two primary techniques were employed to transform the email text into structured numerical data: Most notably for message vectorization through TF-IDF and for dimensionality reduction through Truncated Singular Value Decomposition (SVD). This guaranteed that the email data was not only presented in a manner that could be understood by the ML models, but the data was also preprocessed in a manner that also maximised its efficiency for computation.

To capture the importance of a word in a document to be able to perform the classifications, Term Frequency - Inverse Document Frequency (TF-IDF) was applied. This technique involves affording a set of numerical values to the respective words in a document by comparing the occurrence of the word frequently used in the document with that which occurs per thousand words in the entire corpus; makes the capturing of such important words more accurate as compared to

mere count of occurrences. To reduce computational complexities and improve the model's high predictiveness, the max_features parameter limits the number of the features to 1000. This limits the vocabulary to the top 1000 with the highest TF-IDF score. The number of features is important to take into consideration because this determines the model's accuracy and ensures that it is not too low to be under-fitted or too high that it is overfitted and complex.

TF-IDF Vectorization: TF-IDF vectorization was chosen because it is best suited for this context. TF-IDF adds weight to words that appear in LLM phishing emails but not in human generated phishing emails. The weight of common words in emails that are present are less important, which allows it to give more attention to distinct words that only appear in the context of the four classes. This process allows for an effective distinction. TF-IDF focuses on the features that differ between the categories.

Dimensionality Reduction: Truncated Singular Value Decomposition (SVD) was utilised, which is normally used to minimise the number of feature vectors in the text data. The preprocessing stage involved converting the actionable text data into dense format vectors after which feature dimensionality was reduced. In particular, 50 components were selected based on preliminary experiments and analysis giving account of significant semantic features while avoiding potential overfitting. The SVD model was then built based on the training data and this model was used to transform both the training and testing data into a smaller representation of text content. This dimensionality reduction step played an important role in improving the reliability of the implemented machine learning model and a faster training time.

3.6 Machine Learning Algorithms

Five machine learning models were considered in this study. These are:

3.6.1 Decision Tree

A decision tree is a widely used model for classification tasks that systematically breaks down data by making a series of decisions based on feature values. This model is beneficial for phishing detection as it can handle complex, non-linear relationships in the data and provides easily interpretable decision rules. Decision trees have been shown to perform well in scenarios where decision rules can be clearly defined, such as identifying phishing patterns (Quinlan, 1986).

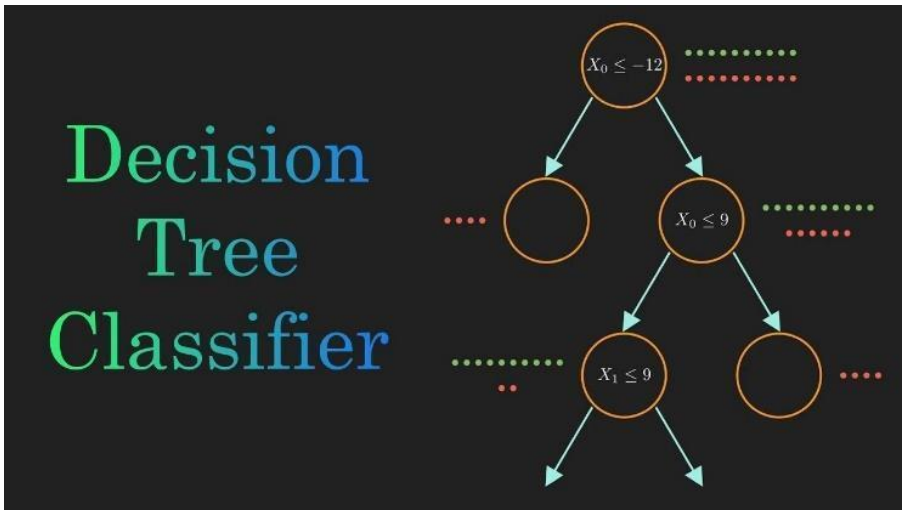

Figure 14: Decision Tree (Nerd, 2021)

3.6.2 Support Vector Machine (SVM)

SVM works by finding the optimal hyperplane that separates different classes in a high-dimensional space (Cortes and Vapnik, 1995). SVM is effective for phishing detection because it can handle high-dimensional feature spaces and is robust against overfitting, especially in cases with limited or imbalanced data. Its ability to maximise the margin between classes helps in distinguishing subtle differences between phishing and legitimate messages (Cristianini and Taylor, 2024).

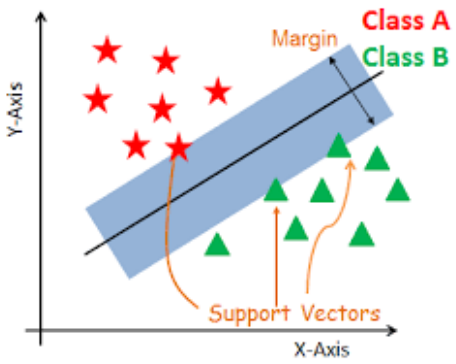

Figure 15: SVM (Navlani, 2019)

3.6.3 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to enhance predictive accuracy and reduce overfitting (Breiman, 2001). This model is well-suited for phishing detection as it leverages the strengths of multiple trees to improve generalisation and robustness. By averaging the predictions of various trees, Random Forest can capture diverse phishing tactics while minimising the risk of being misled by noise in the data (Ho, 1995).

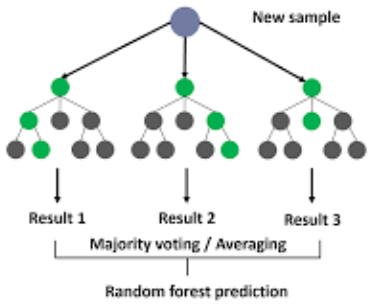

Figure 16: Random Forest (Yehoshua, 2023)

3.6.4 K-Nearest Neighbours

K-Nearest Neighbours (KNN) was selected because it is straightforward and performs well on classification problems where the decision limit between two classes is not well-defined. KNN

operation uses the closest points in the feature space to classify a given data point of the majority class of these neighbours. KNN computes the distance of the classification point to all other points in the training data. This is followed by finding k nearest neighbours based on which it assigns the class label which is most frequent among these neighbours.

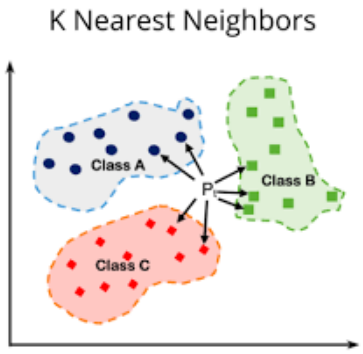

Figure 17: K-Nearest Neighbours (Sachinsoni, 2023)

3.6.5 Gradient Boosting

Gradient Boosting was chosen as a more developed type of ensemble learning, which works with trees of decision. Each tree learns from the erroneous results of the other and therefore Gradient Boosting is effective in case other models cannot identify complex relationships. It is expected to deliver relatively high accuracy especially once the hyperparameters have been optimised. Gradient Boosting combines many weak learners in which each tree tries to minimise the errors made by the previous trees. This is achieved by minimising a loss function, which can normally be done using gradient descent.

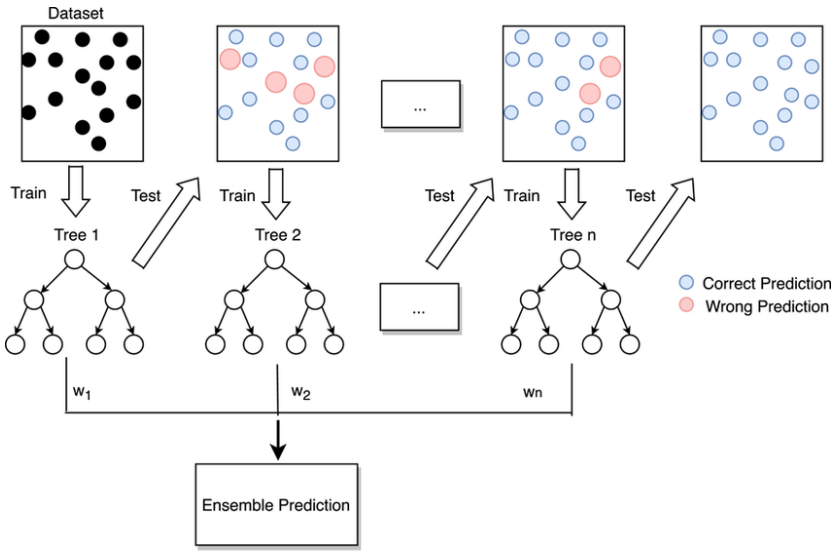

Figure 18: Gradient Boosting (Tao Zhang1, 2021)

3.7 Hyperparameter Tuning

Grid Search was used for hyperparameter tuning in order to get the right set of parameters for each of the models. Using GridSearchCV, hyperparameter tuning was utilised to improve the performance of the model. Thus, the process of evaluating models entailed proceeding through different parameter configurations to determine the right one for the model. The tuned models were then repurposed and tested with an aim of making a comparison with the basic models. This process entailed assessing various options of hyperparameters. For each model, a range of hyperparameters was specified, which are summarised in a table, shown in Figure 19.

MODEL	PARAMETERS
DT	'criterion': ['gini', 'entropy'], 'Max_depth': [None, 10, 20, 30, 40, 50], 'Min_samples_split': [2, 5, 10, 15, 20], 'Min_samples_leaf': [1, 2, 4]
SVM	'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['linear', 'rbf']
RF	'n_estimators': [100, 200], 'Max_depth': [10, 20], 'Min_samples_split': [2, 5]
GB	'n_estimators': [100, 300], 'Max_depth': [3, 5], 'Min_samples_split': [2, 5]
KNN	'n_neighbors': [3, 5], 'metric': ['euclidean', 'manhattan', 'minkowski']

Figure 19: Hyperparameters that were tested

GridSearchCV was used in conjunction with cross-validation on five folds to assess the performance of the created model depending on different hyperparameters. Cross validation allows for a better model generalisation so that the model is not fixed only on the trained or tested data. The parameters selected offering the best performance have been used for every model.

3.8 Evaluation Metrics

Performance metrics such as accuracy, precision, recall and F1-score were used in the assessment of the classification to evaluate and compare the performance of different models.

Accuracy: It assesses the accuracy of the forecasts by comparing ratio of correct forecasts to the total number of forecasted instances. It is used to evaluate the performance universally; its disadvantage is that it is inclined to give misleading results when the dataset is imbalanced.

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Number of Instances (TP + TN + FP + FN)}}$$

Figure 20: Accuracy Equation

Precision: It demonstrates the proportion of all of the true positive predictions amongst all the positive predictions that are made by the model. In the context of LLM generated phishing emails, it is the fraction of emails classified as LLM generated phishing that were actually LLM generated phishing.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Figure 21: Precision Equation

Recall: Recall measures the proportion of positive observations included under the positive class among all the observations included in this actual class. Measures its capacity for correctly recognizing phishing emails.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Figure 22: Recall equation

F1-Score: The F1-score is the harmonic mean of the precision and recall, giving a single measure that does both. This becomes important when there is an imbalance in the class setup.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 23: F1-Score equation

Confusion Matrix: The confusion matrix gives a clear understanding of the actual prediction and mis-prediction, it is easy to determine which class is most often misclassified. This matrix is particularly useful in the context of detecting phishing emails since the cost of false negatives may cause major consequences.

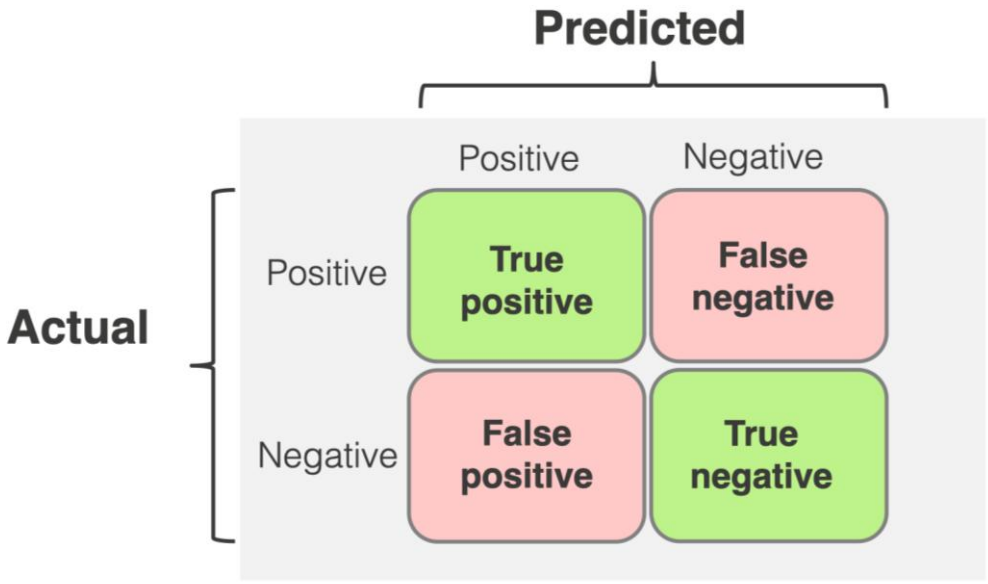

Figure 24: Confusion Matrix Explanation (Evidently AI Team, 2024)

All these metrics were used for the evaluation of the developed model. These metrics allow for a clear assessment of the performance of the model and how well it is at predicting LLM generated phishing emails.

3.9 Webpage for Email Classification

A web page was created that was linked with the model to be able to classify emails into one of the four categories for classifications: LLM phishing, LLM non-phishing, Human phishing, Human non-phishing. This webpage was developed using Streamlit for the sole purpose of

classifying emails based on pre-trained models. The pre-trained machine learning models as well as TF-IDF vectorization and Truncated SVD were all integrated into the streamlit application for the development of the webpage. The user can use this interface to input an email text and click the “classify” button which triggers a process of text pre-processing, vectorization and dimensionality reduction before being classified by one of the pre-trained models. The user can use any desired classification algorithm from the dropdown menu.

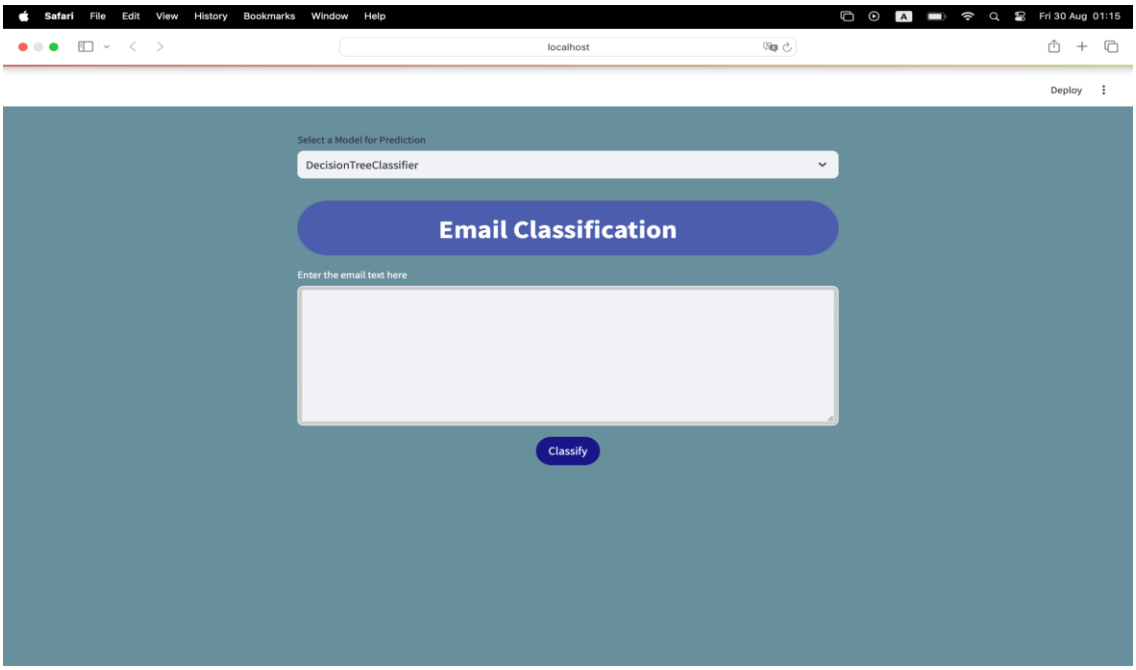

Figure 25: Web- page Interface

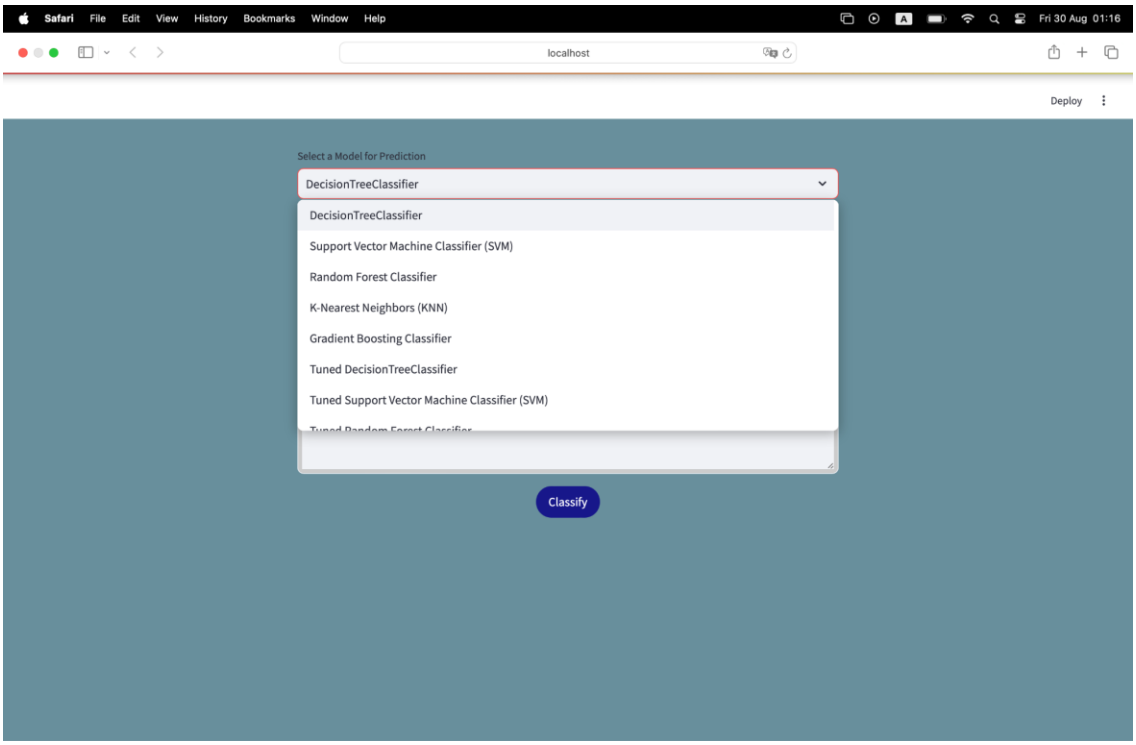

Figure 26: Web-Page Interface Classifier Selection

3.10 Experimental Setup

The performance of the developed machine learning model is highly influenced by the environment of its implementation. Therefore, providing an insight on the resources that were used for the development of the model is important for understanding its performance. The model training was performed on a MacBook Air 2022, Apple M2 Chip, running on a MacOS Sonoma 14.6.1 with an 8 GB of memory.

3.10.1 Programming Language, Libraries and Software

The model was developed on Google Colab for its ease of use and free access to its powerful computational resources. The main programming language that was used was Python since it is known to support libraries such as Pandas for data manipulation and analysis, NumPy for numerical computations, Scikit-learn which is an important library for the machine learning

algorithm implementation, NLTK for natural language processing requirements and lastly Matplotlib and seaborn for data visualisation.

3.11 Ethical Considerations

Ethical considerations cannot be overlooked in the context of detecting LLM phishing emails. This research was conducted with the highest ethical standards. The dataset that was used for this research is a publicly available dataset from Kaggle. However, it is important to state that this dataset was only used for the training of the proposed machine learning models. To ensure privacy, it was important to remove any private information and only work with relevant contextual features that impact the detection of phishing emails without any PII (Personal Identifiable Information). The dataset was not used for any malicious activities, but solely for the training of the machine learning models to produce effective methods for protecting the cybersecurity of our digital world. This research also follows the BCS (British Computer Society) code of conduct that requires the maintenance of high standards of integrity, accountability and confidentiality, where this work is only used for serving the public good to build models that can defend individuals and organisations against phishing attacks.

4. Model Evaluation and Results

This chapter will discuss the results of the developed model with the results of the performance metrics. The metrics are measured before and after the hyperparameter tuning to check if the model's performance is enhanced with hyperparameter tuning. These results represent fairly how well the models are at differentiating between the different classes to determine the best model for this task.

4.1 Decision Tree

According to the results using the Decision Tree, it scored an accuracy, precision and recall of 90.7% and an F1-score of 90.6% and a training time of 0.253 seconds. However, concerning category specific analysis it was found that some categories did relatively better than others.

Human non-phishing performed significantly better than the rest of the categories in terms of the highest percentage of precision, recall, and F1-score of 97%. This implies that the model is very accurate in predicting this category.

On the other hand, LLM Non-Phishing had an 82% precision, showing that the model is less accurate in making the predictions for this category, 88% recall indicating that the model can still correctly identify emails in this category but not as reliable as other categories, 85% F1-score.

Moreover, LLM phishing has a precision of 87% which indicates that it can accurately predict most of the emails in this category correctly, the recall is only 80% which shows that some emails in this category are missed and classified as emails of other categories. This lower percentage of recall affects the F1-score of 84%.

Lastly, the Human phishing category performed well with a precision of 95%, recall of 97% and F1-score of 96%. This shows that the model can correctly classify these emails and rarely misclassify others as Human phishing. The confusion matrix shows that the model provides high accuracy in classifying human generated emails.

However, it also shows that the model encounters difficulty in differentiating between LLM phishing and non-phishing emails. This demonstrates that these particular categories may include features that make them harder to differentiate causing certain misclassifications. Relating to the word cloud that is available in the methodology section, it was visually clear that LLM phishing and non-phishing included very similar recurring words which could be the reason behind the difficulty of differentiation. As seen in figure 30, 32 LLM Non-phishing emails were classified as LLM phishing, and 52 LLM phishing emails were misclassified as LLM Non-phishing.

```
================================================================================
                    DecisionTreeClassifier Model Evaluation
================================================================================
Accuracy:      0.907
Precision:     0.907
Recall:        0.907
F1-score:      0.906
Training Time: 0.253 seconds
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.97      0.97      0.97       315
           1       0.87      0.80      0.84       307
           2       0.95      0.97      0.96       294
           3       0.82      0.88      0.85       284

    accuracy                           0.91      1200
   macro avg       0.91      0.91      0.91      1200
weighted avg       0.91      0.91      0.91      1200
```

Figure 27: Decision Tree Model Evaluation

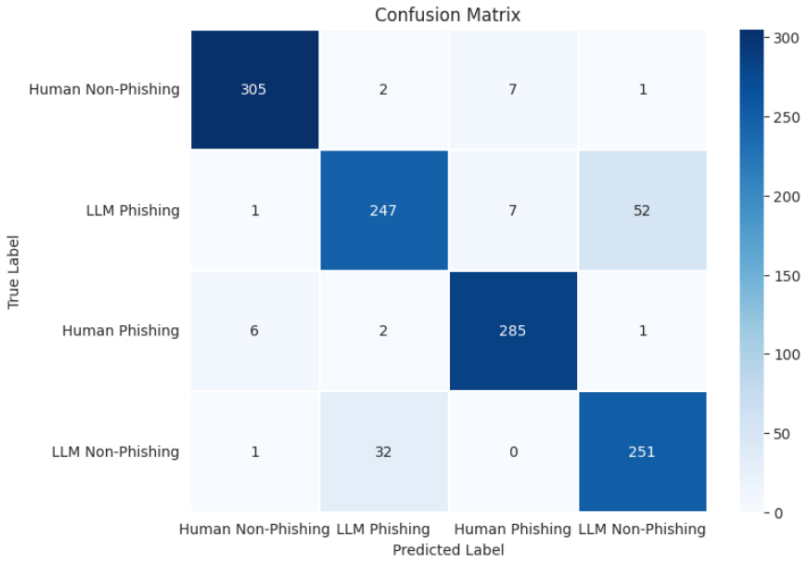

Figure 28: Decision Tree Confusion Matrix

After hyperparameter tuning, the model showed trivial improvement. The model's accuracy and recall increased slightly to 91.3%, and 91.2% for precision and F1-score. Conversely, the training time increased to 104.63 seconds, indicating that the tuning process added complexity but yielded more reliable results.

The best parameters identified were a maximum depth of 30, a minimum samples leaf of 1, and minimum samples split of 2, using entropy as the criterion.

The classification report demonstrates that precision, recall, and F1-score were improved for the "LLM Phishing" class, where precision increased to 85%, and recall to 87%. This indicates that the tuned model became more adept at correctly identifying phishing attempts. The confusion matrix reveals fewer misclassifications, suggesting that the tuning process enhanced the model's ability to accurately differentiate between these categories.

```
================================================================================
                Tuned DecisionTreeClassifier Model Evaluation
================================================================================
Accuracy:      0.912
Precision:     0.912
Recall:        0.912
F1-score:      0.912
Training Time: 104.634 seconds
================================================================================
                    Best Parameters Found by GridSearchCV
================================================================================
criterion: entropy
max_depth: 30
min_samples_leaf: 1
min_samples_split: 2
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.98      0.98       315
           1       0.84      0.87      0.85       307
           2       0.97      0.98      0.98       294
           3       0.85      0.82      0.83       284

    accuracy                           0.91      1200
   macro avg       0.91      0.91      0.91      1200
weighted avg       0.91      0.91      0.91      1200
```

Figure 29: Decision Tree Model Evaluation After Hyperparameter Tuning

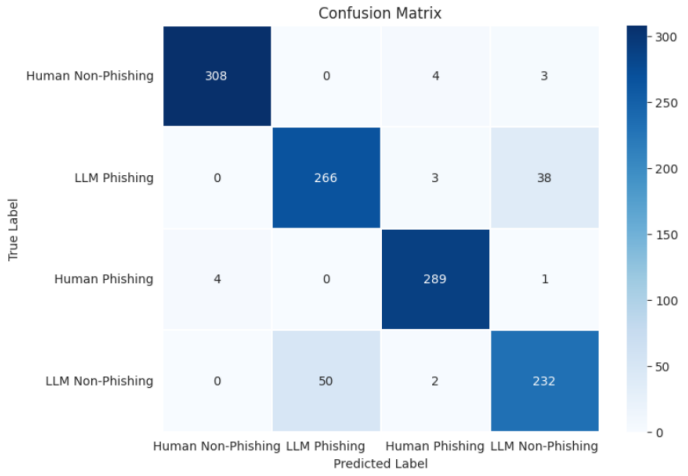

Figure 30: Decision Tree Confusion Matrix after Hyperparameter Tuning

Overall, the Decision Tree model resulted in an accuracy, precision and recall of 90.7%, and an F1-score of 90.6%. After hyperparameter tuning using GridSearchCV, the model showed increased accuracy and recall slightly to 91.3%, with a precision, and F1-score of 91.2%.

4.2 Support Vector Machine

The performance of SVM was high with an accuracy, precision and recall of 97.3%, and an F1-score of 97.2%. The training time was reasonable, taking only 0.20 seconds.

The classification report shows the SVM offered high performance especially when classifying the various classes with precision, recall and F1-score of 98% in Human non-phishing and Human phishing. This implies that the model is exceptionally accurate at accurately predicting these categories and correctly identifies almost all of the instances.

LLM Phishing and LLM Non-Phishing achieved 96% and 97% in precision respectively. For LLM Phishing, it resulted in a 97% in recall and F1-score whereas LLM Non-phishing achieved a 96% in both recall and F1-score.

The confusion matrix shows a few misclassifications where 12 LLM Non-phishing was misclassified as LLM Phishing, and 8 LLM Phishing was misclassified as LLM Non-phishing.

In summary, the high accuracy and balanced metrics of this model show that this model is exceptionally reliable for the classification of the different email categories. This model's performance significantly surpassed the performance of Decision Tree.

```
================================================================================
                Support Vector Machine (SVM) Model Evaluation
================================================================================
Accuracy:      0.973
Precision:     0.973
Recall:        0.973
F1-score:      0.972
Training Time: 0.203 seconds
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.98      0.98       315
           1       0.96      0.97      0.97       307
           2       0.98      0.98      0.98       294
           3       0.97      0.96      0.96       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 31: SVM Model Evaluation

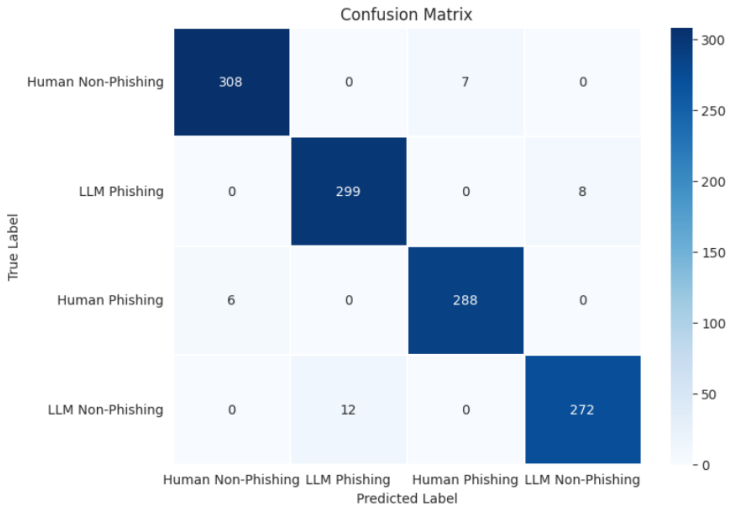

Figure 32: SVM Confusion Matrix

After applying hyperparameters tuning, the SVM is seen to have better performance with slight improvements. The accuracy slightly increased at 97.6% precision, the recall and F1-score increased to 97.6%. The training time period was carried out to 32.114 seconds, which is longer than the time required for the non-tuned model since tuning requires more computations to optimise.

The optimum values detected for the parameters were C = 10, gamma = 1 and kernel type is RBF. This optimization contributed to a slight overall model improvement.

An increase was observed for the Human non-phishing class where it achieved a 99% in precision. However, in the case of LLM phishing and LLM non-phishing, the only increase was that observed in the LLM phishing where the recall increased from 97% to 98% which shows that with the optimization the identification of this instance increased.

The classification report highlights that the tuned SVM model achieved near-perfect scores. The confusion matrix supports this, showing less misclassifications compared to the untuned model. These results suggest that the hyperparameter tuning process effectively enhanced the model's ability to accurately classify different email categories.

Overall, the SVM model demonstrated that it is a high performing model in the context of this research, where both the untuned and tuned models demonstrated excellent performances, where the latter offered a slight overall improvement in its performance.

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
================================================================================
             Tuned Support Vector Machine (SVM) Model Evaluation
================================================================================
Accuracy:      0.976
Precision:     0.976
Recall:        0.976
F1-score:      0.976
Training Time: 32.114 seconds
================================================================================
                    Best Parameters Found by GridSearchCV
================================================================================
C: 10
gamma: 1
kernel: rbf
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.97      0.98       315
           1       0.96      0.98      0.97       307
           2       0.97      0.99      0.98       294
           3       0.97      0.96      0.97       284

    accuracy                           0.98      1200
   macro avg       0.98      0.98      0.98      1200
weighted avg       0.98      0.98      0.98      1200
```

Figure 33: SVM Model Evaluation after Hyperparameter Tuning

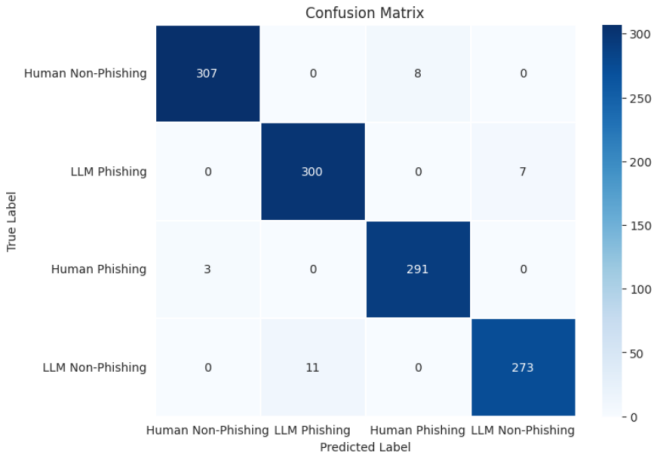

Figure 34: SVM Confusion Matrix after Hyperparameter Tuning

4.3 Random Forest

The Random Forest results demonstrate a good performance across multiple metrics. It achieved an overall accuracy, precision, recall and F1-score of 96.9%. This illustrates the model's ability to correctly classify instances of both phishing and non-phishing emails of human and LLM nature.

The category classification-based report reveals that the model demonstrated a strong performance in identifying "Human Non-Phishing" emails, with a precision and recall of 99%, and an F1-score of 98%. Nonetheless, the "LLM Non-Phishing" and “LLM Phishing” category showed a slight drop in recall at 96% and 95% respectively, which suggests that a small number of phishing and non-phishing emails generated by language models were misclassified.

The confusion matrix further supports these findings, showing that most of the errors occurred in distinguishing between "LLM Phishing" and "LLM Non-Phishing" emails, with 11 instances of LLM non-phishing emails being misclassified as LLM phishing, and 12 instances of LLM Phishing being misclassified as LLM non-phishing.

```
================================================================================
                  Random Forest Classifier Model Evaluation
================================================================================
Accuracy:      0.969
Precision:     0.969
Recall:        0.969
F1-score:      0.969
Training Time: 2.071 seconds
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.98      0.98       315
           1       0.95      0.95      0.95       307
           2       0.98      0.98      0.98       294
           3       0.96      0.96      0.96       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 35: Random Forest Model Evaluation

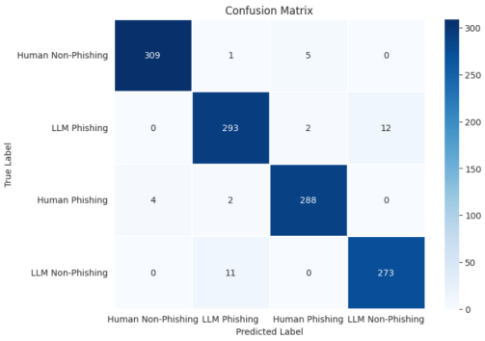

Figure 36: Random Forest Confusion Matrix

After applying hyperparameter tuning, the performance of Random Forest improved slightly. The tuned model achieved the accuracy of 97.3%, precision, recall, and F1-score improved to 97.3%.

This improvement is the result of fine-tuning parameters like max_depth=20, min_samples_split=2 and n_estimators=200 which improves the model's capability to differentiate between the two classes. The confusion matrix of the tuned model indicates that there are fewer misclassified instances, especially the LLM Non-Phishing where there is a fall from 11 to 10 instances of wrong classification as phishing. The "Human phishing" category also improved with lesser misclassifications of phishing emails from 4 to 2 misclassifications.

Overall, Random Forest achieved an overall accuracy of 96.9%, with precision, recall, and F1-score all standing at 96.9%. After applying hyperparameter tuning, the performance achieved the accuracy of 97.3% where the precision, recall, and F1-score improved to 97.3%.

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
================================================================================
               Tuned Random Forest Classifier Model Evaluation
================================================================================
Accuracy:      0.973
Precision:     0.973
Recall:        0.973
F1-score:      0.973
Training Time: 73.782 seconds
================================================================================
                    Best Parameters Found by GridSearchCV
================================================================================
max_depth: 20
min_samples_split: 2
n_estimators: 200
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.98      0.99       315
           1       0.96      0.95      0.96       307
           2       0.98      0.99      0.99       294
           3       0.95      0.96      0.96       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 37: Random Forest Model Evaluation after Hyperparameter Tuning

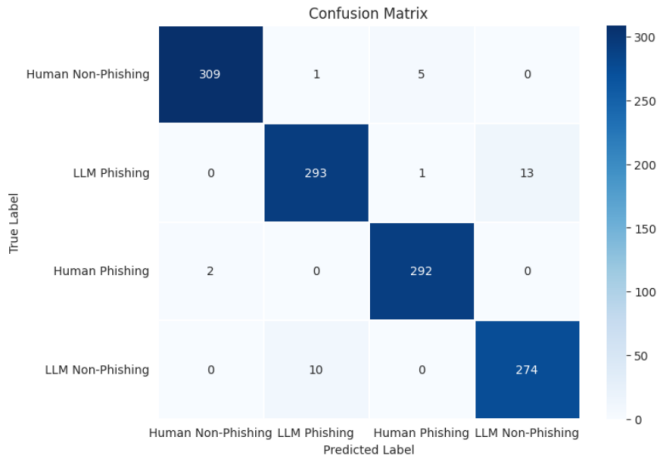

Figure 38: Random Forest Confusion Matrix after Hyperparameter Tuning

4.4 Gradient Boosting

Gradient Boosting overall results demonstrate that the model performed well in identifying the nature of the phishing and non-phishing emails, whether they were LLM or human generated. The models overall performance metrics resulted in an accuracy, precision, recall and F1-score of 96.7% and a training time of 22.64 seconds. The training time for this algorithm was high compared to other algorithms that were trained.

The category classification demonstrates that both “Human Phishing” and “Human Non-phishing” scored 98% in precision, recall and F1-score. On the other hand, “LLM Phishing” has a precision of 94% which is slightly lower compared to other algorithms, indicating that the model had slight difficulties in predicting this class. A recall of 96% indicates that the model was effective at identifying the “LLM Phishing” emails.

Lastly, “LLM Non-Phishing” has a precision of 96%, which is almost similar to the other algorithms, indicating that this model was effective at predicting this class. A recall of 94% indicates that some emails were missed and not identified as “LLM Non-Phishing”. The confusion matrix reflects these findings, highlighting minor misclassifications, particularly within the "LLM Non-Phishing" category, where 16 instances were incorrectly classified as LLM phishing and 10 LLM phishing were misclassified as LLM Non-phishing, demonstrating an area where the model could improve.

```
================================================================================
                Gradient Boosting Classifier Model Evaluation
================================================================================
Accuracy:      0.967
Precision:     0.967
Recall:        0.967
F1-score:      0.967
Training Time: 22.640 seconds
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.98      0.98       315
           1       0.94      0.96      0.95       307
           2       0.98      0.98      0.98       294
           3       0.96      0.94      0.95       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 39: Gradient Boosting Model Evaluation

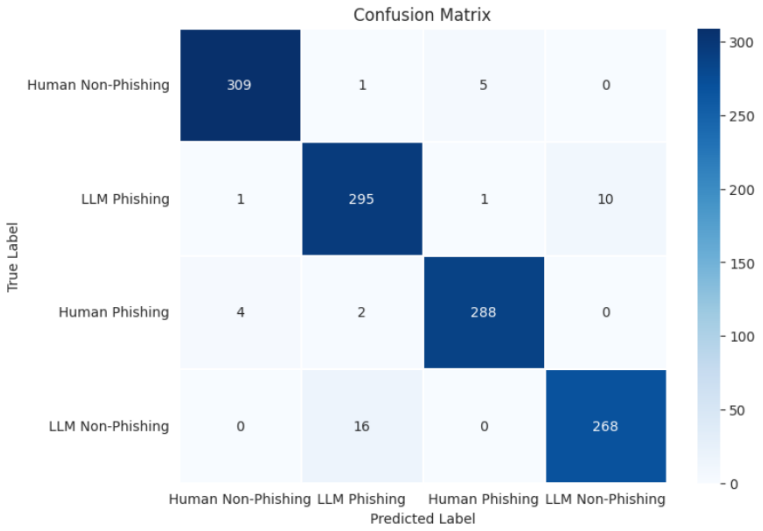

Figure 40: Gradient Boosting Confusion Matrix

After implementing hyperparameter tuning, the model exhibited a slight improvement in performance, achieving a 96.8% in accuracy, precision and recall. The F1-score achieved 96.7%. The training time had a significant noticeable increase to 1472.071 seconds, which is the highest amongst all the models.

The tuning process, which adjusted parameters such as max_depth=5, min_samples_split=5, and n_estimators=300, shows that the model's performance improved slightly by 0.1%.

The confusion matrix indicates a reduction in misclassifications within the "LLM non-phishing", where the number of misclassifications decreased from 16 to 12 of LLM non-phishing misclassified as LLM phishing. Additionally, the model's performance in distinguishing between "LLM phishing" and "LLM non-phishing" emails also improved slightly, suggesting that hyperparameter tuning effectively enhanced the model's ability to accurately classify emails generated by language models.

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
================================================================================
             Tuned Gradient Boosting Classifier Model Evaluation
================================================================================
Accuracy:      0.968
Precision:     0.968
Recall:        0.968
F1-score:      0.967
Training Time: 1472.071 seconds
================================================================================
                    Best Parameters Found by GridSearchCV
================================================================================
max_depth: 5
min_samples_split: 5
n_estimators: 300
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.98      0.98      0.98       315
           1       0.95      0.95      0.95       307
           2       0.97      0.98      0.98       294
           3       0.96      0.95      0.96       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 41: Gradient Boosting Model Evaluation after Hyperparameter Tuning

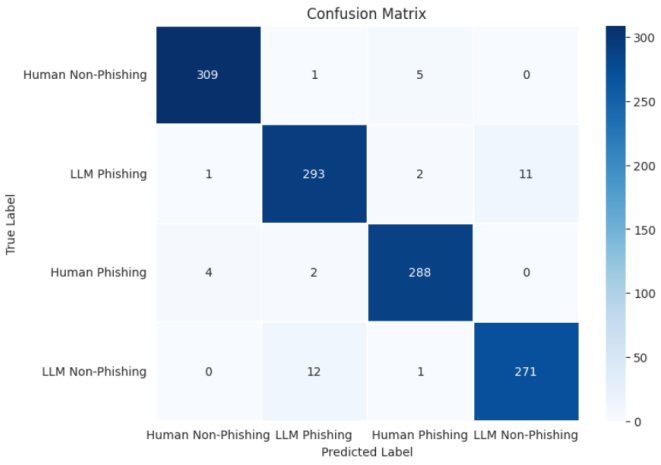

Figure 42: Gradient Boosting Confusion Matrix after Hyperparameter Tuning

Overall, Gradient Boosting is quite impressive with an overall accuracy of 96.7%. The precision, recall, and F1-score for the model were consistent at 96.7%. Hyperparameter tuning demonstrated a slight improvement in performance, achieving an accuracy of 96.8%. The precision and recall both increased to 96.8%, with the F1-score remaining steady at 96.7%.

4.5 KNN

The results from the KNN demonstrate a relatively good performance, achieving an overall accuracy, recall and F1-score of 96.8% and a precision of 96.9%.

The classification report reveals that “Human phishing” achieved a precision of 98%, demonstrating the model’s ability to almost accurately predict the majority of Human phishing emails. Subsequently, LLM phishing scored a precision of 97%, demonstrating a good performance, where this algorithm was able to accurately predict most of this category. Both LLM non-phishing and human non-phishing achieved a 96% in precision, showing that their precisions were not as high as LLM and Human phishing. However, a 96% precision is high enough to indicate that the predictions were accurate for most of the emails. Human non-phishing achieved a recall of 99%, indicating that the model could correctly identify 99% of the emails as Human non-phishing. LLM non-phishing achieved a 98% recall indicating that the model could also identify the majority of the LLM non-phishing emails. Lastly, both LLM phishing and Human phishing scored a recall of 95%, meaning 95% of actual phishing emails were accurately identified. The two classifications were relatively lower than the previous two, but it still performed great overall.

```
================================================================================
                K-Nearest Neighbors (KNN) Model Evaluation
================================================================================
Accuracy:      0.968
Precision:     0.969
Recall:        0.968
F1-score:      0.968
Training Time: 0.048 seconds
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.96      0.99      0.98       315
           1       0.97      0.95      0.96       307
           2       0.98      0.95      0.97       294
           3       0.96      0.98      0.97       284

    accuracy                           0.97      1200
   macro avg       0.97      0.97      0.97      1200
weighted avg       0.97      0.97      0.97      1200
```

Figure 43: KNN Model Evaluation

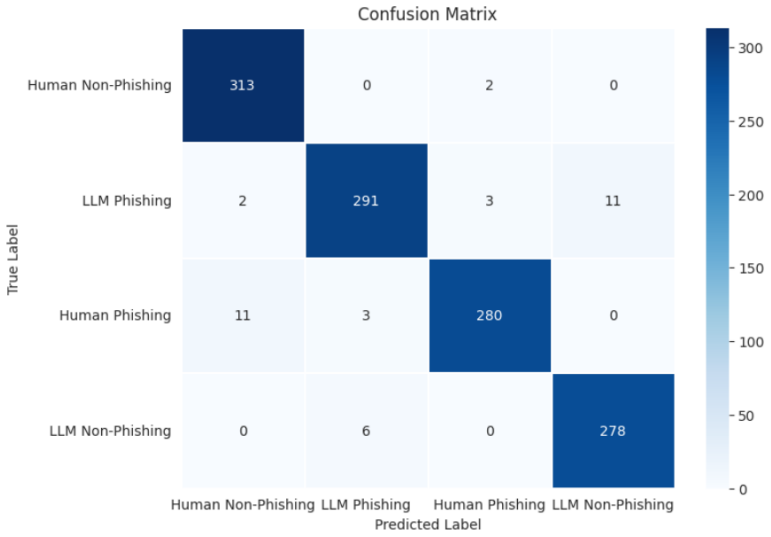

Figure 44: KNN Confusion Matrix

As a result of the hyperparameter tuning, the performance reduced slightly with accuracy at 96.4%. Precision, recall, and F1-score also were slightly lower achieving a 96.4%.

In the tuning process the parameters were: metric=Euclidean and number of neighbours n_neighbors=3, we observed that the performance declined slightly. Different hyper parameters were tested, and these two produced the best performance. The confusion matrix shows an increase in misclassifications, where 13 Human phishing emails were misclassified as Human-non phishing, 8 LLM non-phishing were misclassified as LLM phishing and 12 LLM phishing were

misclassified as LLM non-phishing. This implies that in this particular case, the hyperparameter tuning was not able to improve the generality of the model and could have slightly worsened the ability of the model to classify between the categories.

The KNN before hyperparameter tuning demonstrates strong performance, with an overall accuracy of 96.8%. The precision is 96.9% while recall and F1-score is 96.8%, indicating a reliable ability to classify both phishing and non-phishing emails accurately.

Overall, in the case of KNN, the use of hyperparameter tuning slightly hindered the performance by achieving a lower overall performance.

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
================================================================================
               Tuned K-Nearest Neighbors (KNN) Model Evaluation
================================================================================
Accuracy:      0.964
Precision:     0.964
Recall:        0.964
F1-score:      0.964
Training Time: 1.171 seconds
================================================================================
                    Best Parameters Found by GridSearchCV
================================================================================
metric: euclidean
n_neighbors: 3
================================================================================

Classification Report:

              precision    recall  f1-score   support

           0       0.95      0.98      0.97       315
           1       0.97      0.94      0.96       307
           2       0.97      0.96      0.96       294
           3       0.96      0.97      0.97       284

    accuracy                           0.96      1200
   macro avg       0.96      0.96      0.96      1200
weighted avg       0.96      0.96      0.96      1200
```

Figure 45: KNN Model Evaluation after Hyperparameter Tuning

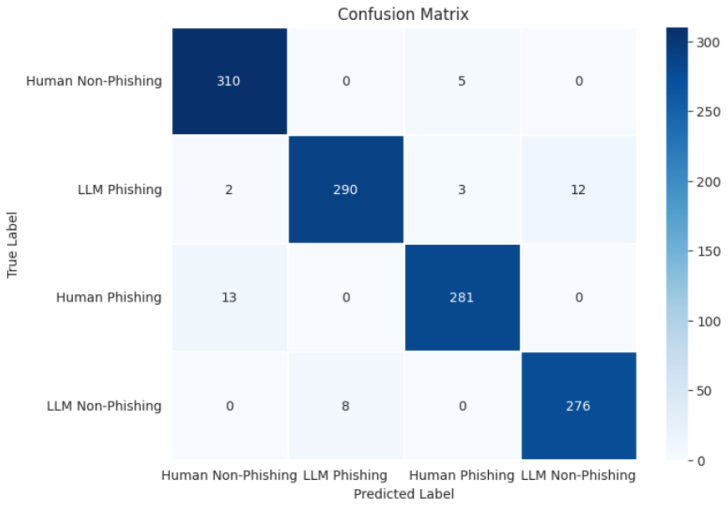

Figure 46: KNN Confusion Matrix after Hyperparameter Tuning

4.6 Comparative Evaluation of All Machine Learning Classifiers

Comparison of the performance of five different classifiers is conducted in this section. In all the models, accuracy, precision, recall, F1-score and training time were assessed. The classifiers were used in their default settings as well as with the best hyperparameters characterised previously to define the model that would be most effective in the given task.

Among all the models that were employed, SVM yielded the best outcomes that proved it more effective with a measure of a 97.6% accuracy when hyperparameters were tuned. Subsequently, the next best performing model was Random Forest, where it demonstrated a high accuracy score of 96.9% before tuning and 97.3% after tuning. KNN and Gradient Boosting also showed good results of an accuracy of 96.8% and 96.7% respectively before hyperparameter tuning.

However, as shown in the results section, KNN's performance declined slightly after tuning the hyperparameters, from 96.8% to 96.4%. The reasoning behind the decrease in performance could be because the default parameters for the model were the best possible parameters in case of this algorithm. This proves that while hyperparameter tuning was observed to improve the performance

of the remaining four models, this was not the case for KNN, implying that hyperparameter tuning can be a great option for optimization but it can also cause a decrease in performance if the default parameters for the model are optimal.

In the case of Gradient Boosting, it was observed that whilst the accuracy increased slightly to 96.8%, the training time was considerably long, taking 1472.07 seconds. This shows that this model required a lot of computational resources due to the complexity of the model.

Lastly, Decision Tree, both in the basic and fine-tuned version, was providing rather stable results and-perhaps due to its nature- was slightly less effective compared to other models.

In conclusion, it is crucial in the context of detecting LLM generated phishing to develop a high performing model that is able to accurately and effectively detect phishing attempts since they can cause significant monetary and data losses. However, it should not be computationally intensive, where training time is an important metric since these models need to be continuously improved and deployed to adapt to new phishing tactics.

Model Name	Accuracy Score	Precision	Recall	F1-score	Training Time
Decision Tree Classifier	0.906667	0.907373	0.906667	0.906389	0.253388
Support Vector Machine Classifier (SVM)	0.972500	0.972537	0.972500	0.972495	0.202995
Random Forest Classifier	0.969167	0.969200	0.969167	0.969179	2.070634
K-Nearest Neighbors (KNN)	0.968333	0.968551	0.968333	0.968259	0.047691
Gradient Boosting Classifier	0.966667	0.966827	0.966667	0.966689	22.640116
Tuned Decision Tree Classifier	0.912500	0.912478	0.912500	0.912363	104.634436
Tuned Support Vector Machine Classifier (SVM)	0.975833	0.975939	0.975833	0.975831	32.114043
Tuned Random Forest Classifier	0.973333	0.973402	0.973333	0.973334	73.781536
Tuned K-Nearest Neighbors (KNN)	0.964167	0.964373	0.964167	0.964125	1.171091
Tuned Gradient Boosting Classifier	0.967500	0.967507	0.967500	0.967497	1472.070783

Figure 47: Overall Performance of All 5 Models

4.7 Results of Web-Page

The webpage was linked with the developed models; therefore, it is linked with the accuracies of the models for the categorization of the emails. The webpage accurately classifies the emails into their respective categories as seen in the figure below.

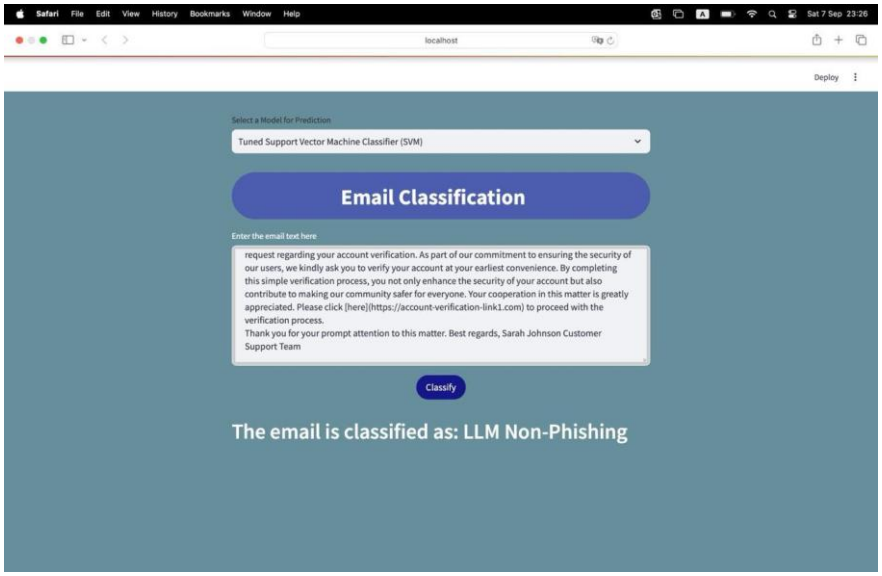

Figure 48: LLM Non-Phishing Classified Email

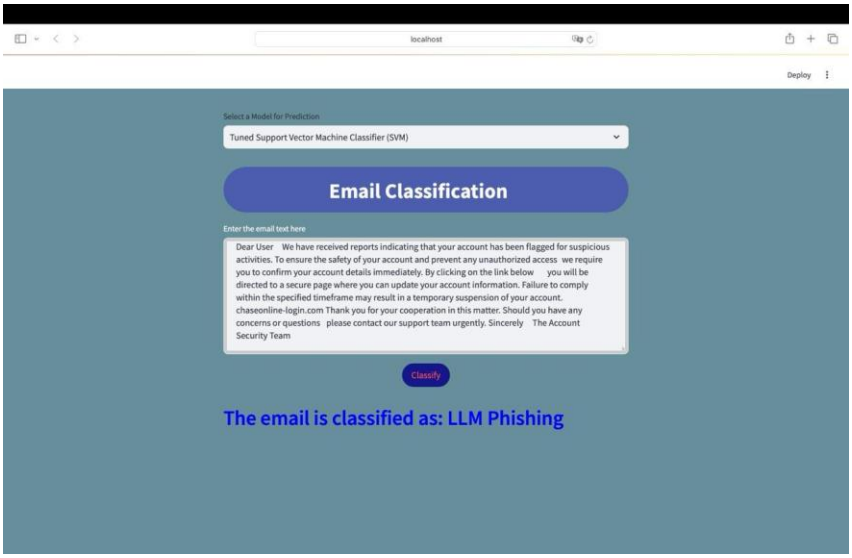

Figure 49: LLM Phishing Classified Email

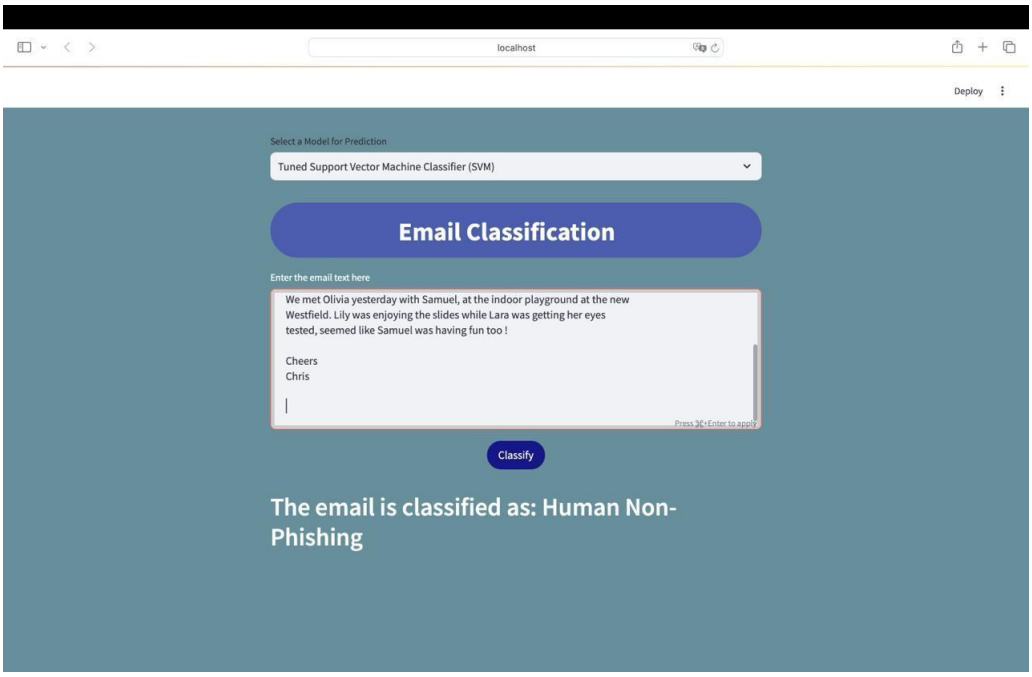

Figure 50: Human Non-Phishing Classified Email

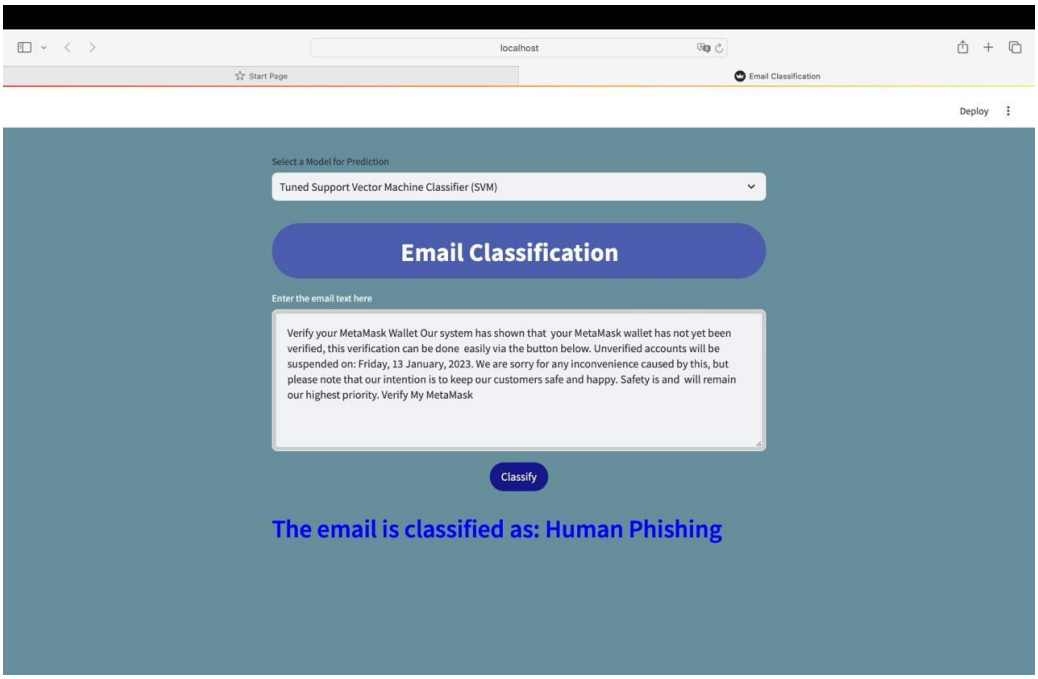

Figure 51: Human Phishing Classified Email

5. Discussion

A thorough discussion of the presented results will be presented in this chapter. The results will be compared with existing relevant literature and the limitations of this study will be addressed.

5.1 Research Aims and Key Findings

The research was conducted with the aim of identifying the phishing emails generated by LLM using supervised ML algorithms such as Decision Tree, SVM, Random Forest, Gradient Boosting and KNN. The study showed that applying appropriate preprocessing techniques such as the TF-IDF vectorization and Truncated SVD as the dimensionality reduction contributed significantly to boosting the performance of the models. Among all the models tested, the accuracy of SVM was the highest, as it reached 97.3% and further increased to 97.6% after hyperparameter tuning. Subsequently, Random Forest with an accuracy of 96.9% before hyperparameter tuning and 97.3% after tuning also shows great results for the context of LLM phishing detection. Nevertheless, Decision Tree had the least accuracy of 90.7% before hyperparameter tuning and 91.3% after tuning, which shows that this algorithm may not be the best for detecting LLM phishing emails, as SVM and Random Forest showcased a noticeable outstanding performance.

As seen in the results, KNN's performance was great before tuning the hyperparameters were an accuracy of 96.8% was achieved. However, it was observed that hyperparameter tuning in the case of KNN did not contribute to an overall model enhancement. This indicates that in the case of KNN, the default parameters might have been the best suited.

In all the algorithms, the confusion matrix illustrates clear misclassifications between LLM phishing and LLM non-phishing. This misclassification is significant compared to Human phishing and Human non-phishing misclassification. This could signify that LLM phishing and

LLM non-phishing exhibit extremely similar features. Future work should focus on the use of a more discriminative feature engineering method that can accurately capture precise and relevant features of these two email categories.

Overall, SVM has shown great results across all metrics, especially accuracy and recall, where in the context of phishing emails the rate of false negatives is expensive. The high recall and accuracy score of 97.6% show that this model is the best suited for this context.

5.2 Comparison with Relevant Literature

Referring to a similar study obtained by Greco et al. (2024), in their paper that is titled "*David versus Goliath: Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails*" which is the main influence of this research, focused mainly on binary classification of phishing emails, whether they were LLM generated or human generated. As seen in Greco et al. (2024) research, they performed a binary classification to detect whether an email was LLM or human generated. Binary classification typically leads to higher accuracy because of its simplicity compared with the multi-classification model, where more complexity is added due to the classification of multiple classes rather than two classes, therefore increasing the complexity of the decision boundary during the learning process of the model. With respect to the differences in the classification approaches, the study conducted by Greco et al. (2024) also achieved a high SVM score of 99.20%, supporting the findings by confirming that SVM is a suitable model for the purpose of detecting LLM generated phishing emails. Hence, the binary classification approach can be considered to provide a more nuanced read as the task at hand was confined to differentiating between AI and human written emails only.

In the binary case, they found that essentially simpler models like Logistic Regression in fact have high accuracy and are easier to interpret and explain. However, the multiple-class solution

of this study, as more complex, speaks of a better representation of the real-world problem of phishing emails classification since multiple sources and multiple types of content may exist. The results indicate that implementing the multi-class classification indeed is more complicated but requires the precise identification of the phishing messages in the more real-life settings where different emails are present.

In discussion the classifications of phishing emails, a study by Eze and Shamir (2024), which utilised more complex ensemble models as well as automatic text analysis for the classification of AI phishing emails. This research utilised complex ensemble models such as MALLET with three algorithms that are: Winnow, maximum entropy, and Naive Bayes to classify between AI-Generated, Enron, Ling-Spam and Nigerian Phishing, where a high accuracy of 99.3% was achieved for Naive Bayes, 99.2% for maximum entropy and 97% for winnow.

The findings in their study shows that several algorithms that work together are shown to be better than a singular approach. The compromise of one defence does not affect the overall security. It shows that a multi-defence approach requires an attacker to adjust their phishing attack to bypass the multiple defences that are put in place, rather than a single machine learning approach.

However, this approach is computationally expensive and requires a lot of training to achieve. Dealing with the constant advancements of technology and the constant change of phishing tactics requires a simple, computationally inexpensive and fast training model to keep up with upcoming phishing tactics. Whilst the approach of Eze and Shamir's study shows near-perfect exceptional results with minor misclassifications, our study adopts a simpler approach with the highest accuracy achieved by SVM of 97.6%.

5.3 Evaluation

It is crucial to evaluate the objectives that were set at the start of this research to ensure that they have been met. The main objective of this study was to develop a machine learning model that can effectively identify and differentiate LLM generated phishing with high accuracy using a multi classification model. This objective has been met with the use of the algorithms in this study such as Decision Tree, SVM, Random Forest, Gradient Boosting and KNN. All these algorithms demonstrated a high performance in the detection of LLM generated phishing emails with minor misclassifications.

The second objective that was set was to evaluate the use of hyperparameter tuning for the optimization of the model, where it was observed that the performance of the models were enhanced slightly after tuning the hyperparameters, indicating that it contributed to an overall enhanced and optimised performance. Although that was not the case for KNN, where the performance decreased by 0.4% after the use of hyperparameter tuning. This signifies that hyperparameters mostly improves the performance of the model, but that is not the case always.

Lastly, the last objective was to design a web-page that can be used for the classification of these emails into their respective categories. It was developed with the help of the machine learning model where an email text was taken as an input for the purpose of classifying it into one of the four categories: LLM phishing, LLM non-phishing, human phishing or human non-phishing. The web-page is able to accurately classify these emails based on the trained models, producing promising results and a user friendly interface.

5.4 Discussion of limitations

A few limitations should be discussed in this study to provide an overall reliable and comprehensive discussion, they should be taken into consideration as they demonstrate transparency of the scope and constraints of this research.

The use of synthetic derived emails generated by LLMs can be considered limited as they do not represent all the real-world phishing attempts. The dataset that was used in this project includes four different classes, providing a level of diversity. However, phishing tactics are constantly evolving to escape present detection methods, therefore the emails used in the training of the machine learning model do not provide an overall representation of emerging LLM phishing tactics. The performance of the model in real world deployments could differ due to emergence of new forms of phishing that are not represented in the training data.

Moreover, the concept of LLM generated phishing emails is considered relatively new compared to traditional phishing emails, making it difficult to find diverse datasets. Therefore, linking to the previous point, that the training of these models was limited to the number of available LLM generated phishing emails in the dataset. Although the developed web-page demonstrated promising results when tested with LLM generated phishing emails that were not from the trained dataset, a consideration should be taken regarding this limitation.

Lastly, the LLMs utilised in this study were only limited to certain models such as WormGPT and ChatGPT. However, as other LLMs come into existence, which may have improved phishing capability, the models examined in this research may require further enhancement and training on the new LLMs phishing capabilities.

5.5 Practical Implications

The findings of this study contribute significantly to the practical implications for enhancing cybersecurity measures in the prevention and detection of phishing. As LLMs become more complex overtime, the threats of phishing emails also rise, posing serious cyber threats to individuals as well as organisations.

This study demonstrates positive results in the ability of machine learning models to detect LLM generated phishing emails. This is useful since most traditional spam filtering systems may not be trained to distinguish LLM generated phishing from normal phishing, where they can be misinterpreted and mistaken for legitimate emails. These machine learning models can be incorporated into current approaches used by organisations to combat these new email threats. According to the results of the study, it is possible to apply machine learning for the protection against phishing, even if the attackers use AI to create their phishing emails.

6. Conclusion

This research raises concerns on current and emerging threats of LLMs such as GPT used in generating phishing emails and points to the need for improved methods for detection of such threats in today's cyber world. This study provides a more robust solution and increased ability to detect both LLM and human generated phishing emails by using a rich dataset that was trained for this basis. A diverse range of algorithms were employed such as Decision Tree, SVM, Random Forest, Gradient Boosting and KNN, where each of the algorithms provided a unique perspective. The models in this study show impressive performances, where SVM achieved an accuracy of 97.6% when tuned. This reassures that LLM phishing can be detected using machine learning

algorithms. Moreover, a website was built on the trained models where the models are used to classify inputs of emails into their respective categories with the assistance of the trained models.

Future Research

It is important to acknowledge that this study has provided valuable insights, yet there is room for future research to build on the current findings, especially by leveraging the limitations for future research.

Building on current findings for future research could investigate ensemble models using multi classification that do not require significant computational resources whilst still yielding better results. According to Sagi and Rokach (2018), ensemble models are the use of a combination of models to increase the overall accuracy by leveraging the strengths and mitigating the weaknesses of the models. The principle behind ensemble models is the use of a combination of base classifiers to create a robust model where the accuracy is boosted.

Another possible area for future research is further expanding the dataset to include a larger and more diverse dataset to train the model for the purpose of improving the overall generalisation capabilities. According to Choudhari and Choudhari (2024), having a diverse dataset contributes to an overall improved accuracy by allowing the algorithms to set their weights more efficiently. A broader range of data allows for improved generalisation on unseen data which is beneficial in real-world implementations to enhance reliability.

The constant and expeditious evolution of the nature of phishing tactics means that a more robust, real-time solution needs to be developed to detect phishing attempts in real-time to avoid any

potential harm. Future work must keep these recommendations in mind and build upon the foundations in this research.

References

Cardona, J. (2024). Grammatical Deviations in Philippine Phishing Emails. *International Journal of English Language Studies*, [online] 6(2), pp.124–129. doi:https://doi.org/10.32996/ijels.2024.6.2.18.

Eze, C.S. and Shamir, L. (2024). Analysis and Prevention of AI-Based Phishing Email Attacks. *Electronics*, [online] 13(10), p.1839. doi:https://doi.org/10.3390/electronics13101839.

Francia, J., Hansen, D., Schooley, B., et al. (2024) *Assessing AI vs Human-Authored Spear Phishing SMS Attacks: An Empirical Study Using the TRAPD Method*. Available at: https://arxiv.org/pdf/2406.13049.

Greco , F., Desolda, G., Esposito, A., et al. (2024) *David versus Goliath: Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails David versus Goliath: Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails*.

Petrosyan, A. (2023) *Interaction with AI and human-generated phishing e-mails Europe 2023 | Statista*. Available at: https://www.statista.com/statistics/1420881/ai-and-human-generated-phishing-e-mails-interaction-europe/

Jamal, S., Wimmer, H. and Sarker, I.H. (2024) An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. *Security and privacy*. doi:https://doi.org/10.1002/spy2.402.

Ponnusamy, Chinnasamy & Krishnamoorthy, P. & Alankruthi, K. & Mohanraj, T. & Kumar, B. & Chandran, Likha. (2024). AI Enhanced Phishing Detection System. 1-5. 10.1109/INCOS59338.2024.10527485.

Thakur, K., Ali, M.L., Obaidat, M.A., et al. (2023a) A Systematic Review on Deep-Learning-Based Phishing Email Detection. *Electronics*, 12 (21): 4545. doi:https://doi.org/10.3390/electronics12214545.

Bethany, M., Galiopoulos, A., Bethany, E., et al. (2024b) *Large Language Model Lateral Spear Phishing: A Comparative Study in Large-Scale Organizational Settings*. doi:https://doi.org/10.48550/arXiv.2401.09727.

P.M, D., M, M., B, N., et al. (2023) Identification of Phishing Attacks using Machine Learning Algorithm. *E3S Web of Conferences*, 399: 04010. doi:https://doi.org/10.1051/e3sconf/202339904010.

Hazell, J. (2023) *Large Language Models Can Be Used To Effectively Scale Spear Phishing Campaigns*. doi:https://doi.org/10.48550/arXiv.2305.06972.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning*, 1 (1): 81–106. doi:https://doi.org/10.1007/bf00116251.

Nerd, N. (2021) *Decision Tree Classification Clearly Explained!* Available at: https://www.youtube.com/watch?v=ZVR2Way4nwQ.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), pp.273–297. doi:https://doi.org/10.1007/BF00994018.

Cristianini, N. and Taylor, J.S. (2024) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Available at: https://books.google.co.uk/books/about/An_Introduction_to_Support_Vector_Machin.html?id=_PXJn_cxv0AC&redir_esc=y.

Navlani, A. (2019) *Scikit-learn SVM Tutorial with Python (Support Vector Machines)*. Available at: https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python.

Tin Kam Ho (1995) *Random decision forests*. doi:https://doi.org/10.1109/ICDAR.1995.598994.

Yehoshua, D.R. (2023) *Random Forests*. Available at: https://medium.com/@roiyeho/random-forests-98892261dc49.

Sachinsoni (2023) *K Nearest Neighbours — Introduction to Machine Learning Algorithms*. Available at: https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2.

Zhang, Tao & Lin, Wuyin & Vogelmann, Andrew & Zhang, Minghua & Xie, Shaocheng & Qin, Yi & Golaz, Jean-Christophe. (2021). Improving Convection Trigger Functions in Deep Convective Parameterization Schemes Using Machine Learning. Journal of Advances in Modeling Earth Systems. 13. 10.1029/2020MS002365.

Evidently AI Team (2024) *How to interpret a confusion matrix for a machine learning model*. Available at: https://www.evidentlyai.com/classification-metrics/confusion-matrix.

Choudhari, A. and Choudhari, N. (2024) Addressing and Resolving Biases in Artificial Intelligence. *Addressing and Resolving Biases in Artificial Intelligence*. doi:https://doi.org/10.59720/23-249.

Appendix

A. GitLab Link: LLM Detection ML Model

https://git.cs.bham.ac.uk/projects-2023-24/axa2257

Contents of Folder:

ML Model Final.ipynb is the ML model that can be viewed and run on Google Colab.

app.py file is the implementation of the website using Streamlit library.

dataset folder with the datasets.

folder containing all the models for app.py

svd_model.pkl for the dimensionality reduction for app.py

tfidf_vectorizer.pkl for the vectorization for app.py

Software and Libraries used:

- Python
- Google Colab
- NLTK
- Seaborn
- Pandas
- Numpy
- Sklearn

- Streamlit

Q. How to run the webpage?

1. The whole folder should be opened on Visual Studios or any similar application.
2. The datasets should be downloaded and the relevant path should be copied into the loaded dataset section in the .ipynb file
3. After copying all four relative paths for the datasets, open the terminal in the terminal and run “Streamlit run app.py”
4. The webpage should be opened.

To run the ML model on google colab, just open the ML Model Final.ipynb file on google colab and load the datasets in the file. The ML model will then run after the datasets have been loaded.

B. Dataset Link

https://www.kaggle.com/datasets/francescogreco97/human-llm-generated-phishing-legitimate-emails

C. Project Management

This research was conducted within a span of approximately two months. During the initial stage, background work has been conducted to gain insight on previous solutions to LLM generated phishing. Building on that comes the major milestone of developing a machine learning model, which consumed the most time as it included a lot of rigorous activities and trial and error. After finalising the machine learning model, the webpage was designed using streamlit to provide a user-friendly interface for our solution. The technical work of this

research took approximately one month and a half to complete. Finally, the final write up for this research consumed the remaining two weeks.