A project report on

IDENTIFICATION OF SPEAR PHISHING USING MACHINE LEARNING

Submitted in partial fulfilment for the award of the degree of

M.C.A

Department of Computer Applications

By

K. DILLIRAM

(22MCA0041)

Under the guidance of

Dr. L. JERART JULUS

School of Computer Science Engineering and Information System

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May,2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems

May 2024

DECLARATION

I hereby declare that the thesis entitled "IDENTIFICATION OF SPEAR PHISHING USING MACHINE LEARNING" submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bonafide work carried out by me under the supervision of Dr. L. JERART JULUS.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: **Signature of the Candidate**

CERTIFICATE

This is to certify that the thesis entitled “IDENTIFICATION OF SPEAR PHISHING USING MACHINE LEARNING” submitted by K. DILLIRAM (22MCA0041) SCORE, VIT, VELLORE for the award of the degree of Name of the degree is a record of bonafide work carried out by him under my supervision of Dr. L. JERART JULUS.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of VIT and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of the HOD**

Internal Examiner **External Examiner**

Executive Summary

There is a rapid rise in cybercrime as the number of internet users increases. In truth, attacker's strategies have been advancing throughout time to make their attacks more convincing and successful. Phishing is a serious security threat that can negatively affect both individuals and the brands they are intended to target. In spite of the fact that this threat has been around for quite a while, it is still very active and effective. In phishing, Spear phishing is a complex aimed target in which hackers gather information from many sources during the attack preparation process to optimise the attack's success. As a solution to such attack, we propose a model that trains on a dataset of normal and phishing emails. A model learns to recognize and classify existing emails as legitimate or phishing by analysing patterns and features associated with phishing attempts. We can construct a Graphical User Interface (GUI) by mixing machine learning models like Support Vector Machine (SVM), Decision Tree, Long Short-Term Memory (LSTM), Naive Bayes, and logistic regression. The model evaluates the features of targeted attacks and produces a probability score indicating whether it is a phishing attempt.

Keywords: spear phishing, cybercrime, machine learning, emails, attacks

ACKNOWLEDGEMENT

It is my pleasure to express with a deep sense of gratitude to my master's thesis guide Dr. L. JERART JULUS, Assistant Professor Sr. Grade 1, School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, for his constant guidance, continual encouragement, and understanding; more than all, he taught me patience in my endeavour. My association with him is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of Machine Learning.

I would like to express my heartfelt gratitude to Dr. G Viswanathan, Chancellor; Mr. Sankar Viswanathan, Vice President; Dr. Sekar Viswanathan, Vice President; Dr. G V Selvam, Vice President; Dr. V. S. Kanchana Bhaaskaran, Vice Chancellor; Dr. Partha Sharathi Mallick, Pro-Vice Chancellor; Dr. Jayabarathi T, Registrar and Dr. Sumathy S, Dean of School of Computer Science Engineering and Information Systems, for providing me with an enriching environment to work in and for their inspirational guidance throughout the tenure of the course.In a jubilant mood, I express ingeniously my whole-hearted thanks to Dr E Vijayan Associate Professor Senior & Head, Department of Computer Applications, MCA Project Coordinator, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. Last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: Name of the student

Contents Page No

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

SVM - Support Vector Machine

LSTM - Long Short-Term Memory

DT - Decision Tree

LR - Logistic Regression

NB - Naive Bayes

GUI - Graphical User Interface

CHAPTER 1
INTRODUCTION

1.1 Background

Spear phishing emails are a serious threat to persons and organisations because they are intended to trick and manipulate recipients into disclosing sensitive information or engaging in criminal behaviour. Identification of these emails necessitates the use of efficient successful systems capable of detecting and classifying them effectively. This paragraph discusses five machine learning techniques, LSTM (Long Short-Term Memory), decision tree, logistic regression, SVM, and Naive Bayes, as viable methods for detecting spear phishing emails.

Long Short-Term Memory (LSTM) is a similar type of recurrent neural network that has demonstrated potential in a variety of natural language processing tasks, including email classification. Its capacity to record sequential information allows it to properly analyse email context, extracting key features and patterns to distinguish authentic from fraudulent emails. LSTM models can be similar from a huge dataset of labelled emails and make accurate predictions based on the patterns they discover.

Decision tree algorithms use a flowchart-like structure to make predictions by splitting the data into various branches and performing comparisons based on different attributes. These algorithms can be used to build classification models for spear phishing email identification. By considering various email attributes such as text, spear, decision trees can effectively categorize emails as legitimate or fraudulent based on predefined rules. The advantage of decision trees lies in their interpretability, allowing users to understand the decision-making process and gain insights into different feature's importance.

Logistic regression is one of the most accurate and probabilistic approaches for categorising communications. When it comes to identifying a dataset that has been labelled as a spear, logistic regression is the most adaptive decision-based technique for detecting spam emails, logistic regression algorithms can effectively identify spear phishing emails by leveraging the collective intelligence of the ensemble.

Support Vector Machines (SVM) are supervised learning methods are employed for tasks such as determining the classification of emails. As the initial information is translated into a higher-dimensional area, and the SVM determines a specific area of space to distinguish classes, successfully distinguishing legitimate from malicious emails. In additive to SVM's can handle high-dimensional data, it also facilitates accurate prediction based on extracted patterns, which aids in the identification of spear phishing emails.

Naive Bayes is a probabilistic classification algorithm that assumes independent characteristics between features given a class label. Using Naive Bayes, you can identify fraudulent emails from legitimate ones by analysing the presence or absence of specific words or features. The Naive Bayes algorithm can predict the most likely class for a given email based on the probabilities of different features occurring in each class (legitimate or malicious). Naive Bayes' simplicity and efficiency make it very handy when dealing with huge datasets. As Naive Bayes learns from labelled email data, it can accurately classify emails as legitimate or malicious based on the observed patterns in their features.

In conclusion, spear phishing emails can be effectively identified using machine learning algorithms such as LSTM, decision trees, SVM, Naïve bayes, logistic regression. As each algorithm is unique, it allows us to detect malicious emails more accurately and efficiently. The spotting of spear phishing emails can be improved even more as machine learning techniques evolve, assisting individuals and organisations in protecting themselves from cyber threats.

1.2 Problem statement

The threat of cybercrime has expanded rapidly in the digital world as internet usage rises. This online crime Phishing attacks, a destructive strategy used by hackers to fool people by appearing to represent legitimate institutions, have grown to be a serious security risk. The highly focused kind of phishing known as spear phishing, which targets flaws and customizes attacks to exploit them. Phishing continues to be a persistent problem despite the existence of numerous protection safeguards. A solution needed to identify such email as legitimate and spear phishing. To control spear phishing, we are creating a GUI to identify such emails.

1.3 Objective

The main goals of this project focus on preventing the growing problem of phishing, especially spear phishing. A specialized defence system is needed for preventing spear phishing, which is characterized by highly focused and sophisticated attacks. Our project contains numerous important objectives to do this. First, the gather a wide-ranging and thorough dataset of legitimate and phishing emails, with a focus on spear phishing incidents. Our detection model will be trained and validated using this dataset as its basis. The second step is to use a variety of machine techniques, such as Support Vector Machine (SVM), Decision Tree (DT), Long Short-Term Memory (LSTM), Naive Bayes (NB), and logistic regression (LR), to maximize the possibilities offered by machine learning. The accuracy and effectiveness of our model will be improved by carefully fine-tuning each algorithm to recognize minor patterns and symptoms of phishing. Since creating a user-friendly interface is essential our project will also involve Create a graphical user interface (GUI). Those are going to allowed to interact with the model by entering emails and instantly receive the email is phishing or legitimate. the probability comments using this user-friendly GUI, allowing immediate decision-making. Our method will also be designed to continuously monitor emails and quickly spot potential scams, so it won't just be an inflexible tool. This proactive technique aims to reduce the chance of spear phishing attempts will be successful. The overall goal of the project is to strengthen internet security, protect the privacy of individuals, and safeguard the reputation of businesses and organizations that are frequently targeted as victims of spear phishing.

1.4 Scope of the project

This project addresses the escalating challenge of cybercrime, driven by the growing number of internet users, with attackers continuously advancing their strategies to increase the effectiveness of their attacks. Phishing, a persistent and severe security threat, remains highly active and impactful, affecting both individuals and targeted brands. Spear phishing, characterized by its sophisticated and highly targeted approach, relies on diverse information sources in its preparation phase, making it a formidable threat. To counter this, our proposed solution involves the deployment of a machine learning model trained on a comprehensive

dataset comprising legitimate and phishing emails. This model is designed to differentiate between these categories by analysing patterns and features associated with phishing attempts. Leveraging various the machine learning strategies include Support- Vector Machines (SVM), decisions tree (DT), long and short-term memory (LSTM), Naive Bayes (NB), & logistic regression (LR), our goal is to enhance the model's accuracy. Additionally, the plan to create a user-friendly Graphical User Interface (GUI) for live website assessments, delivering probability scores to identify potential phishing attempts. This project aims to empower individuals and organizations to effectively combat the enduring and evolving threat of spear phishing in the digital age.

Chapter 2
LITERATURE SURVEY

Aditya *et al.* (2023), The proposed project addresses a significant issue in the digital age: the use of personal information for social engineering assaults, specifically email phishing. It uses open-source intelligence to collect publicly available online data and machine learning algorithms to automate email phishing attempts. This method entails acquiring a victim's personal information from open-source sources, analysing their internet activity, and developing personalised phishing email templates allocate on data. The hybrid machine learning technique, which combines SVM and LR, improves prediction accuracy for phishing attacks. This initiative distinguishes out for its emphasis on data-driven automation and analysis prior to conducting phishing assaults, with the goal of increasing their effectiveness. The combination of SVM and LR surpasses single-classifier approaches, reaching a peak accuracy of 99.69% [1].

Xiong *et al.* (2021), The proposed solution tackles spear phishing, a dangerous type of email-based cyberattack. Spear phishing emails aim to fool individuals or organisations by using personal information. The research presents a novel way to Finding email spear phishing attacks with machine learning. The study analysed Twenty stylistic features from the email content, three forwarding features from an email passing relationship records, and three reputation factors from external threat analysis tools like Phish Tank and Virus Total. The study enhances Synthetic Minority Oversampling (SMOTE) with KM-SMOTE to solve imbalanced data. The inventory consists of 417 spear-phishing messages and 13,916 non-spear-phishing texts. In this study, maximum recall and precision were 95.56%, while F1-score was 97.16%, In particular with the appearance of Transmission trust capabilities, and KM-SMOTE [2].

Akinwale *et al.* (2022), The proposed research tackles the growing problem of spear-phishing emails, a deceptive cyberattack that targets individuals or organisations to steal critical information. Phishing attempts, particularly via email, have become increasingly common, resulting in significant financial losses. These attacks use social engineering and

psychological manipulation to deceive users into disclosing personal information. Email is a valuable transmitting tool in organisations, but it is also a usual target for cyber-attacks. The project attempts to improve the detection and binary grouping of spear phishing emails within businesses. The study uses five various types of supervised Machine learning algorithms include a logistic regression (LR), random forest (RF), decision tree (DT), support vector machine (SVM), and k-nearest neighbour (KNN). This new approach uses a hybrid classifier that combines Logistic Regression and Decision Tree to detect spear-phishing emails with 99.8% accuracy and false positives and false negatives [3].

Yohanes *et al.* (2021), The proposed research focuses on the ongoing threat of spear phishing, a specific type of fraudulent email communication. Previous research has used classification algorithms to detect spear phishing but have faced large false-positive rates. The research's primary contribution is the invention of a unique detection model based on cosine similarity, which achieves a spotting accuracy of 90.45%. The performance demonstrates the model's ability in detecting spear phishing in network traffic. This approach analyses activity patterns, frequency, and duration, resulting in a precision of 90.63% and recall of 76.04%. The approach accurately detects spear phishing in a networked context. Future study aims to improve detection accuracy by investigating correlations and causality between activities [4].

Butt *et al.* (2023), This research focuses on email security in the cloud computing environment. The paper discusses email phishing attempts, which are a common issue due to the extensive usage of email to send sensitive data. Phishing is a deceptive tactic where attackers spoof reputable sources and deceive consumers into disclosing critical information. This research focuses on detecting email phishing attacks by categorising them as "phish" or "not phish". The study's comprehensive strategy involves constructing a customised dataset, extracting features using regular expressions, and analysing natural language. We use machine learning and deep learning algorithms including Support Vector Machine (SVM), Naive Bayes (NB), and Long Short-Term Memory. SVM, NB, and LSTM had high accuracy rates of 99.62%, 97%, and 98%, respectively [5].

Mohith *et al.* (2020), This paper proposes a new strategy to preventing phishing attacks, a common form of cyber-attacks that steals sensitive user information like login passwords and bank details'-phishing solutions are not always effective in entirely safeguarding users from such threats. The authors propose a new web browser architecture called the 'Embedded Phishing Detection Browser' (EPDB) to combat phishing attacks. This architecture includes a part for real-time phishing detection, which employs a Random Forest Classification model. This effort is unique in that it prioritises both user experience and security against phishing attacks. The browser has a 99.36% success in live recognition of phishing, in addition to providing a bigger view area and improved user experience [6].

Saha *et al.* (2020), The proposed research focuses on the essential issue of phishing, especially during the COVID-19 epidemic when remote work and digital connections have increased significantly. Phishing is a common cybercrime that steals user credentials from online platforms such as banking, e-commerce, and digital markets. Attackers employ bogus webpages and spam emails to fool users and steal vital information. This study presents a data-driven approach using deep learning, specifically a multilayer perceptron, to detect phishing websites. The Kaggle dataset includes 10,000 web pages and ten attributes. The model's training accuracy of 95% and test accuracy of 93% demonstrate its effectiveness in detecting phishing websites. The algorithm outperforms existing phishing detection methods, distinguishing trustworthy websites with an accuracy of 98.4% [7].

Yamah (2022), The proposed research focuses on the rising danger scenario in the digital arena, especially with the increased use of email communication. Cyberattacks have increased, resulting in financial losses, reputational harm, data breaches, and emotional pain for both individuals and organisations. Phishing, particularly spear-phishing, is a powerful attack method that takes use of victims' psychological vulnerabilities and social engineering tactics. This project aims to improve spear-phishing detection by traditional and automated methodologies. This paper presents a unique model that uses Random Forest techniques and Ensemble learning. It was train and test on a dataset of 3,000 emails, including both conventional and spear-phishing emails. The Random Forest algorithm detected spear-phishing emails with a phenomenal accuracy of 96.33% [8].

Ojewumi *et al.* (2022), The proposed research focuses on the ongoing issue of phishing assaults, namely in web browsing. Phishing attacks are a vital threat for particular and organisations, attempting to trick users into disclosing sensitive information on the internet. Based on a dataset of fourteen attributes, we present A form of rule method to recognise phishing emails that includes three machine learning models: KNN, Random Forests, and Support Vector Machines. In comparison with other machine learning methods, the Random Forest model performed better. Through Phish Net, a Chrome browser extension built with HTML, CSS, and JavaScript, the study extends its impacts by incorporating rules from the Random Forest model. Phish Net is a reliable web-based phishing detection tool that improves security and lowers the risk of assaults [9].

Salloum *et al.* (2022), The proposed research proposal provides a thorough examination of the area of phishing emails. Detection of a major cybersecurity issue that causes huge economic losses. This study analyses Using Natural Language Processing (NLP) technologies for spotting email scams, a topic that has received less attention in past surveys. This review examines 100 research articles from 2006-2022 on detecting phishing emails using machine learning, text features, datasets, and evaluation criteria. This analysis found a heavy emphasis on feature extraction and selection, with support vector machines (SVMs) becoming a preferred choice for identifying phishing emails. NLP techniques, including TF-IDF and word embeddings, have been applied in this field. The analysis highlights the need for additional research and resources in detecting Arabic phishing emails, citing a dearth of studies in this field [10].

Li *et al.* (2020), This study proposes a novel way to detecting phishing emails that uses persuasion to identify prospective attempts. This study examines how phishing emails use language and psychology to trick recipients. This paper identifies persuasive language patterns in phishing emails and develops a way to detect them. The research begins with choosing crucial qualities using an information gain technique, resulting in 25 persuasive features. These features are used in a classification model to identify phishing emails. The

experimental outcome indicates a high accuracy rate 99.6%, with strengths in True Positive Rate (TPR) and Precision [11].

Doshi *et al.* (2023), The paper proposes a detecting of Email phishing & unwanted email messages. The study uses machine learning methods (LSTM, decision tree, and random forest) to find spear phishing email or not. The two-layer architecture improves detection accuracy and efficiency, offering a comprehensive approach to combating phishing and normal emails. This research contributes to computer security and offers potential Solutions to minimise the Effects of Phishing Attacks [12].

Ozcan *et al.* (2021), This paper suggested a mixed DNN-LSTM model to catch phishing URLs. The researchers merged DNN and LSTM architectures to produce an effective hybrid model. They evaluated its performance to DT and RF methods. The study shows that machine learning approaches like LSTM can detect spear phishing emails and the hybrid DNN-LSTM strategy is more effective in detecting phishing URLs [13].

Shaukat *et al.* (2023), This study employs machine learning algorithm to display attractive adverts and spear phishing assaults. The researchers recommend using machine LSTM, decision tree, and random forest algorithms to detect spear phishing emails. Combining these algorithms improves the accuracy and effectiveness of detecting phishing attacks. The study published in Sensors found that a hybrid approach can effectively mitigate the dangers of phishing assaults [14].

Sun *et al.* (2021), This study aims to choose false danger documents used in aim email attacks. Researchers use machine learning algorithm, including LSTM, decision tree, and random forest algorithms, to detect spear phishing emails. Their research shows that these algorithms accurately distinguish between legitimate and fraudulent emails, providing valuable insights for improving email security and preventing cyber-attacks. This study adds to the emerging topic of employing machine learning for cybersecurity [15].

Nivedha & Raja (2022), The paper identifies email spam utilising natural language processing (NLP) and a RF technique. In an A study published in the Worldwide Journal of Computer Technology and Mobile Computing employed machine learning, a DT, and a RF algorithm for spotting spear email phishing. Combining these techniques improved the accuracy and efficiency of detecting such malicious emails [16].

Rayan *et al.* (2021), A study on detecting email spam via a natural language processing-based random forest technique. The researchers used machine learning, DT, and RF techniques for recognising spear-phishing emails. Researchers combined these techniques in a single paragraph to improve detection accuracy [17].

Rayan & Taloba (2022), In a study, "Analysis of e-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique," examines The application of machine learning methods to recognize spear phishing emails. This study combines LSTM, decision tree, and random forest algorithms in a hybrid bagging technique to improve email spam detection accuracy and identify spear phishing attempts. It contributes to the field of computational intelligence by highlighting the potential of combining multiple algorithms [18].

Ebong & Maurice (2022), In a study, "Deep Learning Phishing Email Classifier Combined with NLP," examines how machine learning algorithms like LSTM, decision tree, and random forest can detect spear phishing emails. The research intends to improve email security using deep learning and natural language processing. The study was done at the National College of Ireland, Dublin [19].

Li *et al.* (2023), In their work "Spear-Phishing Detection Method Based on Few-Shot Learning," Li, Q., and Cheng, M. (2023, August) propose a few-shot learning-based method for detecting spear-phishing. The authors propose using machine LSTM, decision tree, and random forest algorithms to detect spear phishing emails. The study intends to increase the

accuracy and efficiency of detection. This research was presented at the International Symposium on Advanced Parallel Processing Technologies in Singapore and published by Springer Nature Singapore [20].

2.1 Summary of the existing works

Researchers have demonstrated a comprehensive strategy for enhancing the accuracy and resistance to phishing detection using machine learning methods. In addition to basic classifiers such as RF and SVM, investigators have studied enhanced deep learning structures such as CNNs and RNNs. A large concentration is on feature engineering, with an emphasis on extracting relevant information from phishing emails, URLs, and network traffic to improve model efficacy. The incorporation of natural language processing (NLP) techniques has improved analysis by identifying semantic clues and linguistic patterns within textual content. Ensemble approaches have emerged as a viable option, amalgamating the strengths of numerous classifiers to achieve increased accuracy and robustness against changing phishing. Despite these advances, issues like as class imbalance and generalisation to new datasets remain, motivating continuous research in the subject.

2.2 Challenges present in existing system

Addressing the problems with existing phishing detection systems requires a multidimensional approach that combines advanced machine learning algorithms, domain expertise, and robust procedures. Mitigating class imbalance necessitates specific measures, such as oversampling minority classes or using ensemble methods that prioritise balanced performance criteria. Furthermore, approaches such as cost-sensitive learning or anomaly detection can help mitigate the effects of imbalanced datasets by modifying the model's decision thresholds to better account for the rarity of phishing incidents.

To address the issue of generalisation and transfer learning, researchers are looking into methods like domain adaptation and continuous learning, which allow models to adapt to new

contexts and evolving threats over time. Using diverse and representative datasets during training, as well as strategies like data augmentation and regularisation, can improve a model's capacity to generalise across multiple contexts. Furthermore, the use of rigorous evaluation procedures such as cross-validation and out-of-sample testing is critical for analysing a model's performance under varied scenarios and assuring its reliability in real deployment. By addressing these difficulties in a comprehensive and adaptable manner, researchers hope to improve the effectiveness and resilience of phishing detection systems in protecting consumers from cyber threats.

CHAPTER 3

REQUIREMENTS

3.1 Hardware requirements

- **Processor**: I3/I5
- **RAM**: 4GB
- **OPERATING SYSTEM:** windows 7

3.2 Software requirements

- **IDE:** Command Prompt.
- **PROGRAMMING LANGUAGE:** Python.
- **OPERATING SYSTEM:** Windows 10/Windows 11.
- **TOOLS:** Jupyter notebook.
- **ALGORITHM:** There are the machine learning algorithms I used are, SVM, LR, DT, LSTM, and NB.

3.3 Gantt chart

Table 3.1 Gantt chart for spear phishing.

	August	Sept-Nov	Dec	Jan-Apr
1	Research about project			
2	Required Tools gathering			
3		Installation of operating systems		
4		Python source code formation		
5			Execution of project	
6			Testing of project	
7				Document preparation about project

8				Submission of project

An effective Gantt chart can help to capture the development timeline of a machine learning-driven spear phishing detection system. The project will first require a week of comprehensive initiation, with an analysis of spear phishing tactics and machine learning models along with a definition of scope, objectives, and stakeholders. Next, two weeks are allocated to meticulously collecting and preprocessing data, which is essential for training robust machine learning algorithms. Design, implementation, and fine-tuning machine learning models take place over the course of four weeks in the core development phase. After training the models, they must be integrated seamlessly into existing infrastructure, which takes two weeks. Furthermore, phishing detection must be monitored and managed efficiently, which takes three weeks of interface development. In advance of the final deployment, two weeks will be allocated for thorough testing and quality assurance. This will allow us to identify and fix any system defects that may be present. Documentation and training are prioritized during the first week to ensure smooth implementation and maintenance of the system after deployment. An initial week of monitoring and adjusting follows the deployment and rollout phase. Maintaining and monitoring the system, even as phishing tactics continue to evolve, remains an integral component. A machine learning-driven spear phishing detection system is developed and deployed successfully using this structured approach, aligning its objectives with actionable timelines.

CHAPTER 4
ANALYSIS AND DESIGN

4.1 Proposed methodology

The proposed technique Spear phishing emails may be identified using machine learning techniques such as LSTM, DT, LR, SVM, and Naive Bayes. Spear phishing is a sort of aimed at digital assault in which attackers imitate respectable agencies in order fool victims into taking valuable data or carrying out destructive behaviours. To detect spear phishing emails, the system employs LSTM algorithms, which are capable of interpreting sequential data. Important elements such as email content, sender details, and attachments are extracted after training on a large dataset of labelled spear phishing emails. The LSTM model accurately distinguishes between valid and malicious inbound emails by recognising patterns and attributes common to known spear phishing emails.

In addition to LSTM, the system uses decision trees and logistic regression methods. Decision trees provide an intuitive description of decision-making processes by defining if-else conditions depending on feature relevance. Logistic regression, on the other hand, makes probabilistic predictions by calculating the likelihood of a binary result. These algorithms supplement the LSTM model by providing multiple views for identifying and categorising spear phishing emails. The system uses textual analysis and metadata extraction techniques to parse email headers, subject lines, and body information. It uses natural language processing algorithms to detect suspicious patterns, keywords, and legally problems seen in spear phishing emails. Furthermore, the system examines email attachments and links using malware detection algorithms to assess potential threats. By implementing a choose of machine learning algorithms, the system's goal is to detect spear phishing emails ahead of time, protecting individuals and organisations from criminal intrusions and reducing financial or reputational costs.

4. 2 Architecture diagram

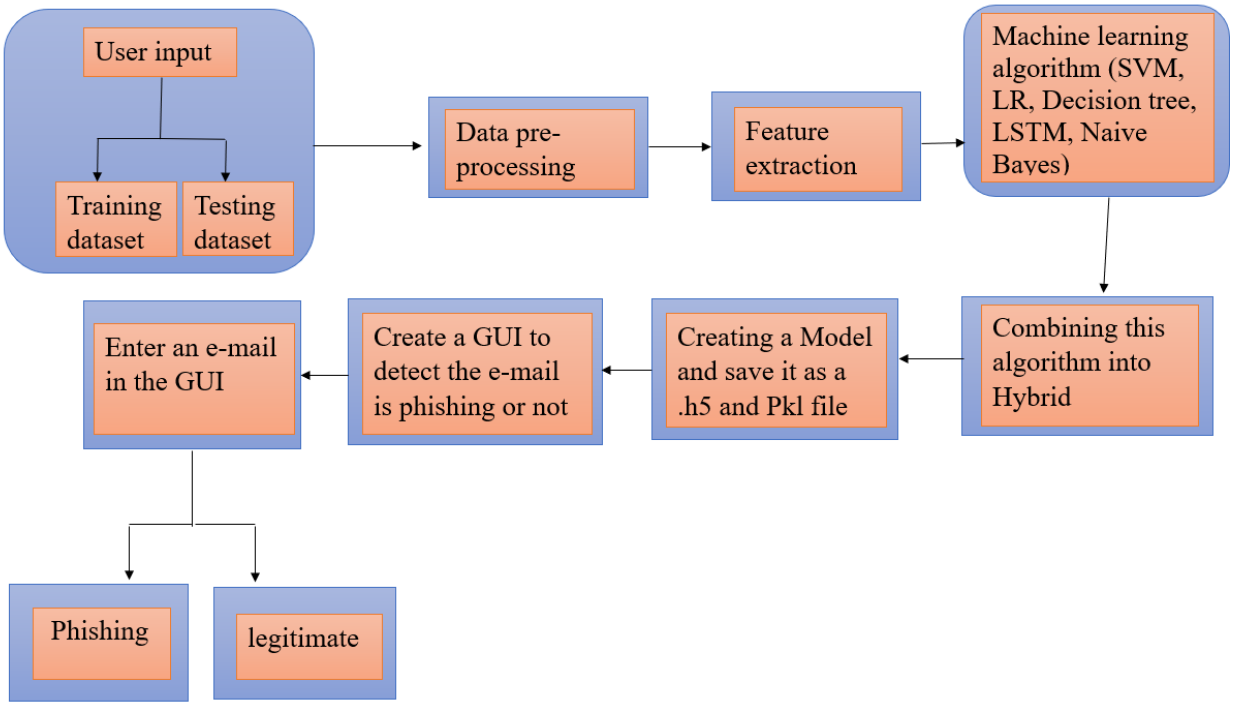

Figure 4.1 Architecture diagram for spear phishing.

The architecture is made up of two primary datasets: a training dataset for model building and a separate testing dataset for evaluation. Before training, data preprocessing techniques are used to clean and format the data. Feature extraction algorithms are then used to extract useful information from the emails. Individually implemented machine learning techniques include SVM, LR, Decision Trees, LSTM, and Naive Bayes. A hybrid approach is formed by integrating multiple methods to increase anticipated performance. The final model can be stored in two different formats:.h5 for deep learning models like the LSTM and.pkl for other approaches.

4.2.1 Use case diagram

The Use Case Diagram for discovering email spear phishing attempts with machine learning algorithms depicts interactions among players such as email receivers, IT administrators, and system administrators. Users start procedures including receiving emails, preparing data, training models, categorising emails, sending alarms, and offering feedback. Email recipients receive emails that are scanned for phishing, while IT administrators manage data pretreatment and model training. System administrators initiate model training operations that use various machine learning methods. Users categorise incoming emails and receive notifications of potential spear phishing emails, suggesting further action. User feedback helps machine learning models become more refined and improved over time. This figure visually depicts these interactions, highlighting user requirements and the system's functionality in spear phishing detection.

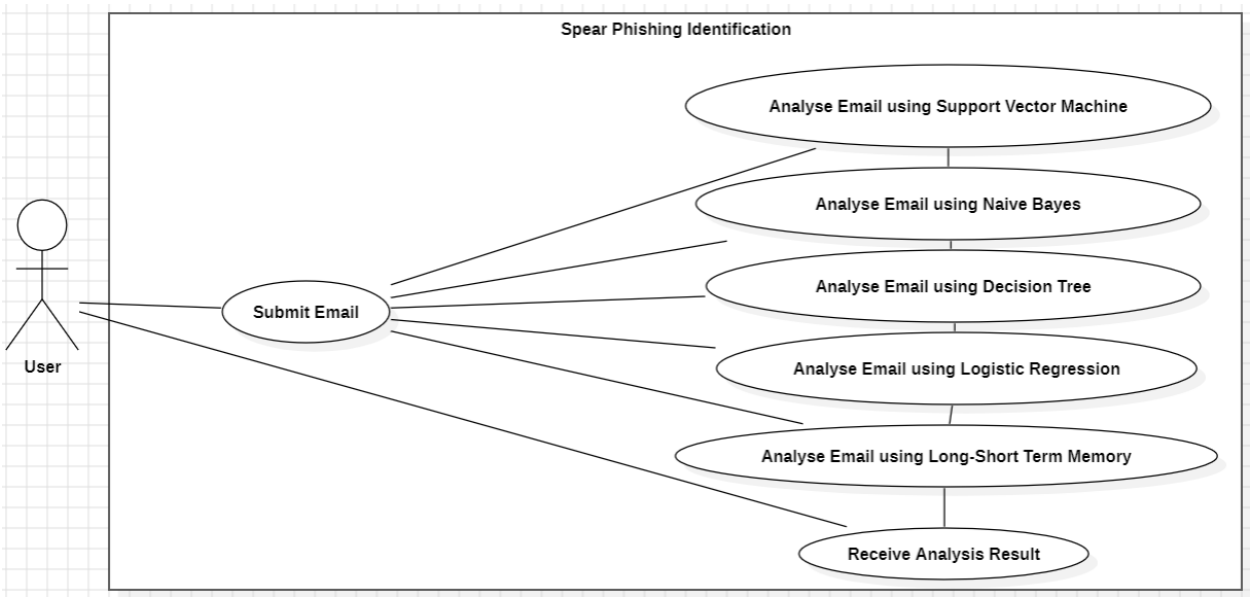

Figure 4.2 Use case diagram for spear phishing.

4.2.2 Class diagram

The class diagram for detecting spear phishing emails utilising machine learning techniques such as LSTM, DT, LR, SVM, and Naive Bayes algorithms would depict multiple classes and their relationships inside the system. The Email class represents email data and its attributes, the Phishing Detector class detects and classifies spear phishing emails using machine learning models, and the Data Preprocessing class cleans and preprocesses email data before feeding it into the models. Other classes may include Model Trainer, which trains machine learning models, Feature Extractor, which extracts key features from email data, and Alert Generator, which generates notifications for potential phishing emails. The class diagram depicts the dependencies and relationships between these classes, which interact and support the spear phishing detection process.

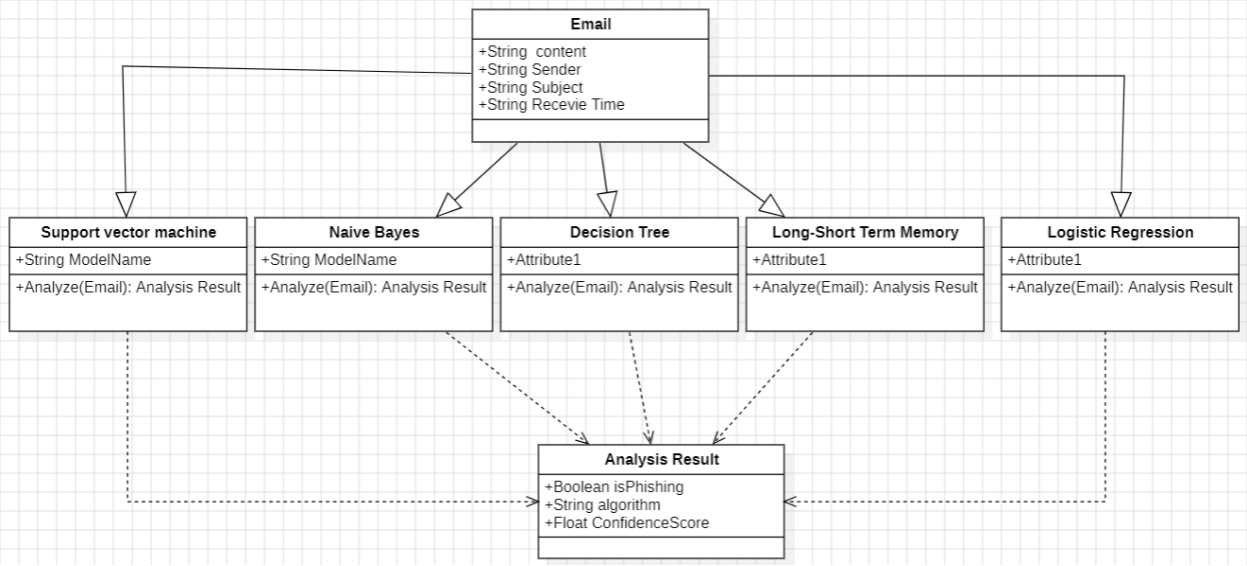

Figure 4.3 class diagram for spear phishing.

4.2.3 Sequence diagram

A sequence diagram can represent the procedures required in spotting spear phishing emails Applying ML strategies like LSTM, DT, LR, SVM, and Naive Bayes strategies. It may depict the sequence of events, such as data collection, email preparation, and feature extraction

from emails. The figure can then show One way that are machine learning models developed based on the collected attributes. Finally, it can demonstrate the prediction phase, in which the system analyses fresh emails and generates predictions or alerts based on previously learned models. This graphic representation helps stakeholders and developers in comprehending the process and operation of the spear phishing email detection systems.

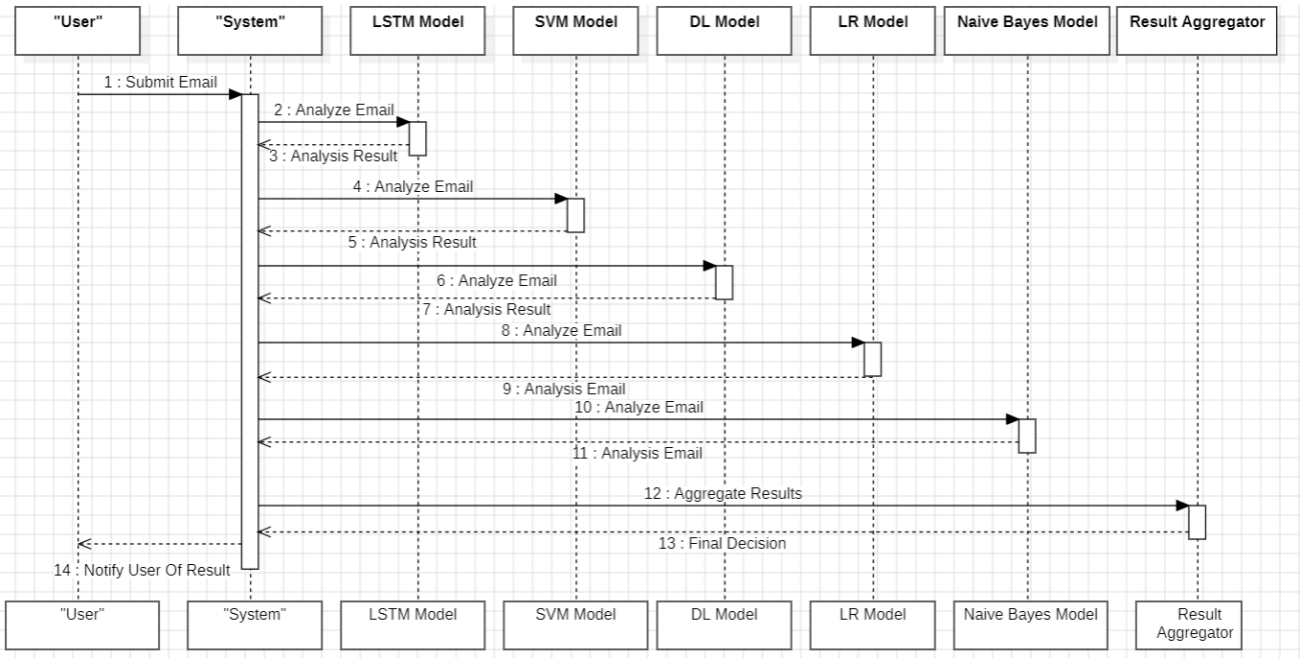

Figure 4.4 Sequence diagram for spear phishing.

4.2.4 Activity diagram

An activity diagram for detecting spear phishing emails with machine learning strategies would depict the activities and procedures involved in examining and forecasting spear phishing emails. The diagram would include nodes indicating collecting data, preprocessing, gathering features, model training with LSTM (Long Short-Term Memory), decision tree, logistic regression, SVM, and Naive Bayes techniques, and prediction. Arrows would connect various operations, symbolising the flow of information and control between them. Decision points might also be represented, indicating where requirements must be met or decisions must be made. This activity diagram provides a visual depiction of the workflow for detecting spear

phishing emails, assisting in understanding the sequence of events and the interaction of various components. It encourages effective communication and collaboration among parties involved in the system's development and deployment, as well as identifying potential bottlenecks and inefficiencies.

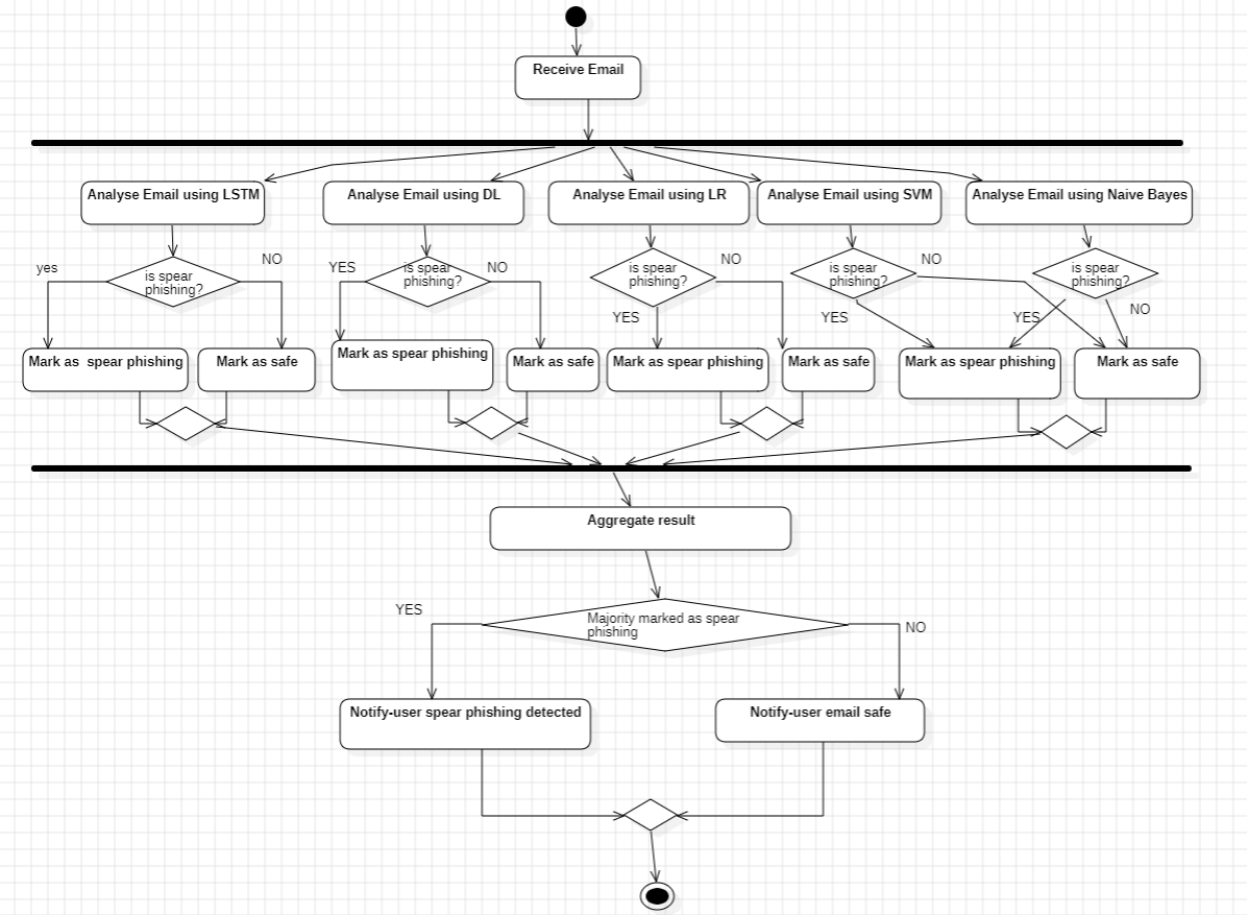

Figure 4.5 Activity diagram for spear phishing.

4.3 Module description

1. **Data collection and feature extraction**

In the process of recognising spear phishing emails, data collecting is an important first step. Relevant email data, including content, sender information, and other relevant properties, is collected consistently. Following this, feature extraction techniques

are used to find relevant patterns and distinctions in the obtained data. This entails identifying important characteristics that distinguish legitimate emails from spear phishing efforts. Machine learning strategies such as LSTM, DT, LR, SVM, and Naive Bayes. are then used to build models capable of accurately identifying emails. These algorithms use merged data and extract attributes to precisely detect possible spear phishing emails, effectively limiting associated risks.

2. **Data cleaning and normalization**

Data cleaning and normalisation are important steps in detecting spear phishing emails with machine learning techniques such as LSTM, DT, LR, SVM, and Naive Bayes algorithms. Spear phishing, being a targeted and deceitful attack, necessitates accurate and reliable data for efficient detection.

The first phase, data cleaning, is to remove irrelevant or missing information from the dataset, such as empty fields, duplicates, and inconsistencies. This assures the dataset's accuracy and removes any biases that could affect the performance of machine learning models. data normalisation then standardises the data into a uniform format, removing variations and allowing for fair comparisons of different qualities. Min-max scaling and z-score normalisation are two techniques used to guarantee that every single parameter contributes effectively to the machine learning method. Data cleaning and normalisation improves the accuracy of detecting spear phishing emails. This enables algorithms such as LSTM, decision tree, logistic regression, SVM, and Naive Bayes to detect and mitigate harmful assaults with greater accuracy and efficiency.

3. **Text preprocessing and tokenization**

Text preparation and tokenization are critical for detecting spear phishing emails with machine learning techniques such as LSTM, decision tree, logistic regression, SVM, and Naive Bayes. Spear phishing is a severe cybersecurity issue that requires accurate detection. Preprocessing cleans raw text data by removing extraneous information and turning it to lowercase, whereas tokenization reduces text to smaller units for numerical representation. Advanced algorithms can then identify tiny signs of spear phishing, such as strange email headers or fraudulent URLs. These algorithms, trained on labelled

datasets, learn patterns to increase detection accuracy, so successfully reinforcing cybersecurity defences.

4. **Handling imbalanced data**

Handling unbalanced data is critical in detecting spear phishing emails with machine learning techniques such as LSTM, decision tree, logistic regression, SVM, and Naive Bayes. Imbalanced data, with one class considerably outnumbering the others, might result in biassed models. To overcome this issue, techniques include the technique of overs the minority group, reducing the greater class, and employing various assessment measures like as accuracy, recall, and F1-score. Ensemble approaches like bagging and boosting can also help improve classification accuracy. Overall, resolving unbalanced data is critical for effective spear phishing email detection, considerably enhancing the effectiveness of machine learning algorithms in such a scenario.

5. **Separate data into both training and testing sets.**

separating the findings into training and testing sets is vital. for properly Formation and reviewing technology such as LSTM, DT, LR, SVM, and Naive Bayes in spear phishing email detection. This stage ensures that models are properly trained and evaluated. The Content is divided into two sets: a training set with labelled examples for algorithm training, and a testing set with unlabelled data for performance evaluation. Algorithms learn patterns and relationships from labelled data through training, and their ability to generalise to previously unseen examples is tested. This systematic technique enables the robust training and assessment of spear phishing email detection algorithms such as LSTM, DT, LR, SVM, and Naive Bayes algorithms.

4.3.1 Model improvisation

1. **Machine Learning-Based Approaches for Identification of Spear Phishing Emails.**

Machine learning-based algorithms have developed as powerful tools for detecting spear phishing emails, which are false emails used in targeted cyber-attacks. One such

method employs Long Short-Term Memory (LSTM) neural networks, decision tree algorithms, and random forest algorithms. LSTM networks are especially useful for analysing sequential data, such as email text and metadata, to discover trends and abnormalities common in spear phishing campaigns. In contrast, decision tree algorithms use a hierarchical model to generate a sequence of if-then rules, allowing emails to be classified based on a set of declared features. Random forest algorithms use an set of decision trees to improve accuracy in identifying spear phishing emails by taking into account numerous views and minimising biases. Organisations may use these machine learning algorithms to create strong and efficient systems for detecting and mitigating spear phishing attacks, improving their overall cybersecurity posture, and protecting sensitive data from criminal actors.

2. LSTM Algorithm for Spear Phishing Email Detection

The LSTM (Long Short-Term Memory) algorithm, along with decision tree and logistic regression algorithms, have been used to identify and detect spear phishing emails. Spear phishing is a specific attack in which infected emails are designed to trick recipients into providing critical information. The LSTM algorithm is a recurrent neural network noted for its ability to handle sequential input, making it ideal for analysing email content. Decision tree and logistic regression algorithms are classification approaches that produce predictions using a tree-like structure and email features. Combining these three algorithms resulted in an excellent spear phishing detection system. The LSTM algorithm is used to analyse the textual content of emails, gathering trends and contextual information in order to identify questionable communications. Decision trees and logistic regression use information taken from email headers and metadata to categorise emails as valid or malicious. This multi-algorithm method provides a comprehensive solution for identifying spear phishing emails, improving email security, and safeguarding users from falling victim to phishing attempts.

3. Decision Tree Algorithm for Spear Phishing Email Detection.

The Decision Tree algorithm is good at detecting spear phishing emails because it classifies emails using a hierarchical structure based on specific attributes. Using the Decision Tree approach, we may create a tree-like model with nodes representing conditions and branches representing potential outcomes. This algorithm is effective in identifying spear phishing emails because it takes into account critical aspects such as the sender's identification, email content, and keywords. Additionally, the Long Short-Term Memory (LSTM) algorithm can be used to improve prediction accuracy. LSTM is specifically developed for sequence-based data, such as email content, enabling a better grasp of context and semantic meaning. Furthermore, the Random Forest algorithm can be used to improve the categorization process. Random Forest uses a collection of decision trees and accumulates their predictions to get a final judgement, resulting in increased accuracy in identifying spear phishing emails. Implementing these machine learning techniques allows us to accurately detect and identify spear phishing emails, eventually improving security and safeguarding customers from potential cyber-attacks.

4. Logistic Regression for Spear Phishing Email Detection

Logistic regression is a basic but effective method for detecting spear phishing emails by predicting the likelihood of a malicious email based on a range of variables. In this context, features can include metadata that contain an email's domain name, email topic, and body articles characteristics. The system learns to discriminate between phishing and legitimate emails by training the logistic regression model on a dataset comprising labelled instances of each. Logistic regression predicts the coefficients for each feature via iterative optimisation, assigning weights to signify their importance in evaluating the chance of an email being a spear phishing attempt. This strategy gives a clear decision threshold for classifying incoming emails as suspicious or safe. By applying logistic regression alongside other machine learning approaches, organisations can construct a comprehensive defence mechanism against spear phishing attempts, thus

enhancing cybersecurity.

5. Support Vector Machine (SVM) for Spear Phishing Email Detection.

Support Vector Machines (SVMs) provide a strong foundation for detecting spear phishing emails by learning to distinguish between authentic and malicious email samples. SVMs function by transforming starting data into a feature space with a high and determining the horizontal plane that optimally separates classes. Spear phishing detection can leverage features taken from email content, metadata, and sender information to represent emails in the feature space. SVMs generalise effectively to new data and are resistant to overfitting since they maximise the margin between classes. Additionally, SVMs may handle non-linear relationships using kernel functions, allowing for complex decision bounds. Organisations can create sophisticated spear phishing detection systems that can reliably identify phishing attacks by exploiting SVMs' flexibility and generalisation capabilities Detecting harmful emails while reducing false positives. This proactive approach increases cybersecurity resilience and reduces the risk posed by targeted phishing attempts.

6. Naive Bayes Algorithm for Spear Phishing Email Detection.

The Naive Bayes technique, which draws on Bayesian probability theory ideas, provides a statistical framework for detecting spear phishing emails. Despite its simple assumptions, Naive Bayes can be surprisingly good in detecting suspicious emails by estimating the likelihood of an email falling into the phishing class based on its characteristics. This approach assumes feature independence, which allows for the efficient computing of conditional probabilities. In the context of spear phishing detection, the Naive Bayes classifier can be trained using features such as word frequencies, sender information, and metadata. Naive Bayes assigns a classification label to incoming emails by estimating the posterior probability of whether it is malicious or

authentic based on these features. While Naive Bayes may fail to capture complicated correlations in the data, Its simplicity and computational efficiency make it an appropriate choice for real-time phishing detection systems. By adopting Naive Bayes into their cybersecurity arsenal, organisations can strengthen their defences against spear phishing attempts and preserve critical information from unauthorised access.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Data set

https://www.kaggle.com/datasets/gokulraja84/emails-dataset-for-spam-detection

The Emails dataset for Spear Detection consists of labelled emails classified as "spear" or "ham" (non-spear), and it is an important resource for machine learning and natural language processing researchers. The dataset, which includes a variety of email lengths and content to reflect real-world communication, attempts to train and analyse models for successfully recognising spear emails. By analysing email language and attributes, models can learn patterns that distinguish spear from legal emails, automate filtering procedures, and protect consumers from frauds and phishing attempts. Researchers To create successful spear detection models, employ a variety of machine learning algorithms, including as SVM, DT, LSTM, NB and LR, strategies for learning. Feature engineering involves identifying useful email features after removing stop words and converting text to numerical representations during preprocessing. It is easier to build and evaluate spear detection systems when the dataset is split into training datasets, validation datasets, and testing datasets, which in turn make email security and user experience better.

5.2 Sample codes

5.2.1. Spear phishing using DL and LR. Jpynb

```
from sklearn. linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

dtc = DecisionTreeClassifier(max_depth=5)

lrc = LogisticRegression(solver='liblinear', penalty='l1')

clfs = {
```

```
 'DT': dtc,
 'LR': lrc,
}
def train_classifier(clf,X_train,y_train,X_test,y_test):
 clf.fit(X_train,y_train)
 y_pred = clf.predict(X_test)
 accuracy = accuracy_score(y_test,y_pred)
 precision = precision_score(y_test,y_pred)
 return accuracy,precision
accuracy_scores = []
precision_scores = []
for name,clf in clfs.items():
 current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_test)
 print("For ",name)
 print("Accuracy - ",current_accuracy)
 print("Precision - ",current_precision)
 accuracy_scores.append(current_accuracy)
 precision_scores.append(current_precision)
performance_df =
pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scor
es}).sort_values('Precision',ascending=False)
```

performance_df

5.2.2. Spear_Email_LSTM.jpynb

```
from tensorflow.keras.layers import Embedding

from tensorflow.keras.preprocessing.sequence import pad_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.preprocessing.text import one_hot

from tensorflow.keras.layers import LSTM

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Bidirectional

from tensorflow.keras.layers import Dropout

### Dataset Preprocessing

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

corpus = []

for i in range(0, len(messages)):

 print(i)

 review = re.sub('[^a-zA-Z]', ' ', messages['text'][i])

 review = review.lower()

 review = review.split()

 review = [ps.stem(word) for word in review if not word in stopwords.words('english')]

 review = ' '.join(review)
```

```
 corpus.append(review)

## Creating model

from tensorflow.keras.layers import Dropout

## Creating model

embedding_vector_features=40

model=Sequential()

model.add(Embedding(voc_size,embedding_vector_features,input_length=sent_length))

model.add(Dropout(0.3))

model.add(LSTM(100))

model.add(Dropout(0.3))

model.add(Dense(1,activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.33,

random_state=42)

### Finally Training

model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=10,batch_size=64)

from sklearn.metrics import accuracy_score

accuracy_score(y_test,predictions)
```

5.2.3. Spear Email Naïve_Bayes.jpynb

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

spear_df=pd.read_csv(r'E:\Final year project\Emails.csv')

spear_df.rename(columns={'spam':'spear'},inplace=True)

spear_df

spear_df.head(10)

spear_df.tail(10)

spear_df.describe()

spear_df.info()

ham=spear_df[spear_df['spear']==0]

spear=spear_df[spear_df['spear']==1]

ham

spear

print('Spear Percentage =',(len(spear)/len(spear_df))*100,'%')

print('Ham Percentage =',(len(ham)/len(spear_df))*100,'%')

sns.countplot(spear_df['spear'],label='Spear vs Ham')

from sklearn.feature_extraction.text import CountVectorizer

vectorizer=CountVectorizer()

spamham_countVectorizer=vectorizer.fit_transform(spear_df['text'])
```

```
import pickle
with open('count_vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
print(vectorizer.get_feature_names())
spamham_countVectorizer.shape
label=spear_df['spear']
X=spamham_countVectorizer
y=label
X.shape
y.shape
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
from sklearn.naive_bayes import MultinomialNB
NB_classifier=MultinomialNB()
NB_classifier.fit(X_train,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
y_predict_train=NB_classifier.predict(X_train)
y_predict_train
cm=confusion_matrix(y_train,y_predict_train)
sns.heatmap(cm,annot=True)
y_predict_test=NB_classifier.predict(X_test)
```

```
y_predict_test

cm=confusion_matrix(y_test,y_predict_test)

sns.heatmap(cm,annot=True)

print(classification_report(y_test,y_predict_test))

print(accuracy_score(y_test,y_predict_test))

import pickle

with open('naive_bayes_spam_classifier.pkl', 'wb') as file:

    pickle.dump(NB_classifier, file)
```

5.2.4. Spear Email SVM.jpynb

```
import pandas as pd

from sklearn.svm import SVC

from sklearn.model_selection import train_test_split,GridSearchCV,KFold

from sklearn.metrics import accuracy_score,classification_report,confusion_matrix

from sklearn.feature_extraction.text import CountVectorizer

from imblearn.over_sampling import SMOTE

df = pd.read_csv(r'E:\Final year project\Emails.csv')

df.rename(columns={'spam':'spear'},inplace=True)

df.head()

x=df["text"]

y=df["spear"]
```

```
cvec=CountVectorizer()

cx=cvec.fit_transform(x)

cx.toarray()

cx.shape

y.value_counts()

smt=SMOTE()

x_sm,y_sm=smt.fit_resample(cx,y)

x_sm

y_sm

y_sm.value_counts()

x_sm.shape

x_train, x_test, y_train,y_test = train_test_split(x_sm,y_sm,test_size=0.2,random_state=0)

params={"kernel":["rbf","linear"]}

cval=KFold(n_splits=5)

model=SVC()

gsearch=GridSearchCV(model,params,cv=cval)

gsearch.fit(x_train,y_train)

gsearch.best_params_

bmodel=SVC(kernel="rbf")

bmodel.fit(x_train,y_train)

y_pred=bmodel.predict(x_test)
```

```
y_pred

accuracy_score(y_test,y_pred)

best_model = gsearch.best_estimator_

confusion_matrix(y_test,y_pred)

import matplotlib.pyplot as plt

import seaborn as sns

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=['ham', 'spear'], yticklabels=['ham', 'spear'])

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix for SVM')

plt.show()

print(classification_report(y_test,y_pred))

import pickle

with open('svm_model.pkl', 'wb') as file:

    pickle.dump(best_model, file)
```

5.2.4. GUI Creation app.py

```
import streamlit as st

import pandas as pd
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
import pickle
from sklearn.feature_extraction.text import CountVectorizer

# Load the trained model
with open('naive_bayes_spam_classifier.pkl', 'rb') as file:
    NB_classifier = pickle.load(file)

with open('count_vectorizer.pkl', 'rb') as file:
        vectorizer = pickle.load(file)
# Function to classify input text
def classify_text(text):
    text_vectorized = vectorizer.transform([text])
    prediction = NB_classifier.predict(text_vectorized)
    if prediction[0] == 1:
        return "This is a spear email."
    else:
        return "This is a legitimate email."

def main():
```

```
st.title("Email Spear Classifier")

st.sidebar.title("Navigation")

page = st.sidebar.radio("Go to", ["Home", "Classification"])


if page == "Home":

    st.markdown("""

    Welcome to the Email Spear Classifier App!

    This app predicts whether an email is spear or not.

    """)


elif page == "Classification":

    st.header("Email Classification")


    # Input text box for user to enter email content

    user_input = st.text_area("Enter the text of the email:")


    if st.button("Classify"):

        if user_input:

            result = classify_text(user_input)

            if result == "This is a spear email.":

                st.error("This is a spear email.")
```

```
            elif result == "This is a legitimate email.":
                st.success("This is a legitimate email.")
            else:
                st.warning("Please enter some text.")
if __name__ == "__main__":
    main()
```

5.3 Sample output

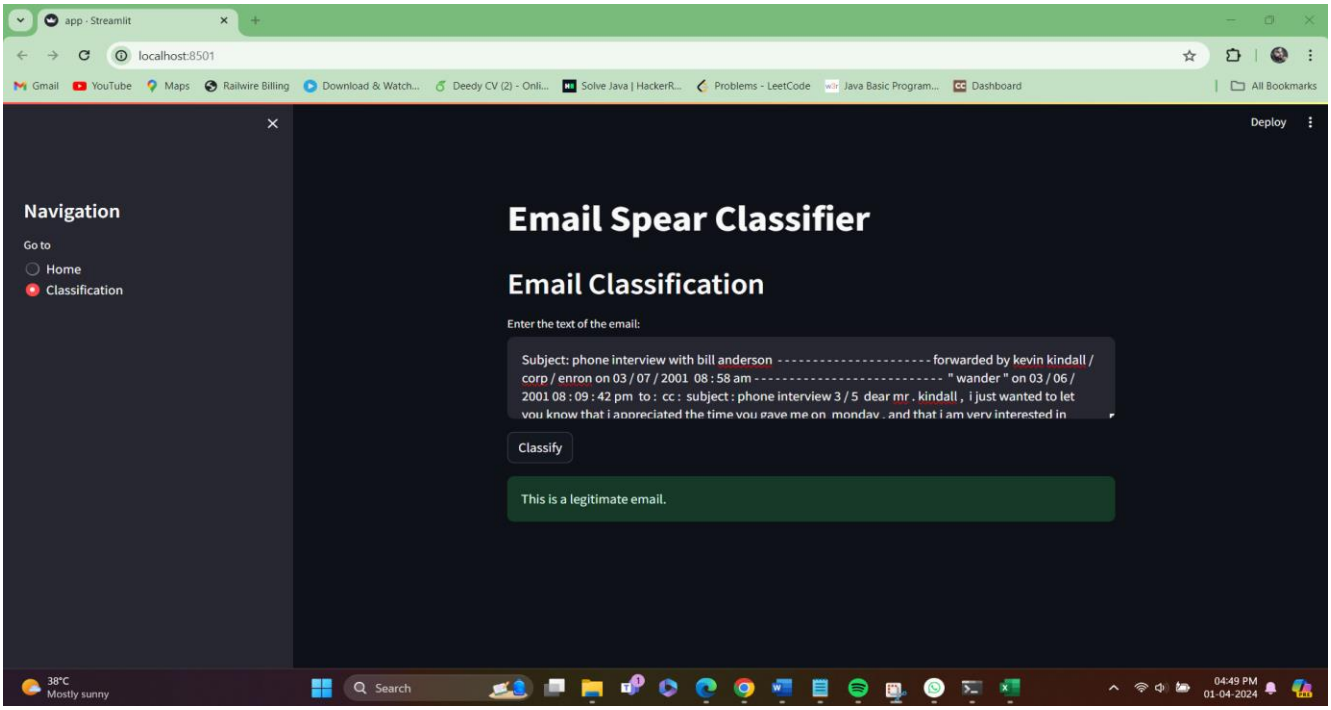

Figure 5.1 GUI for spear phishing detection legitimate email.

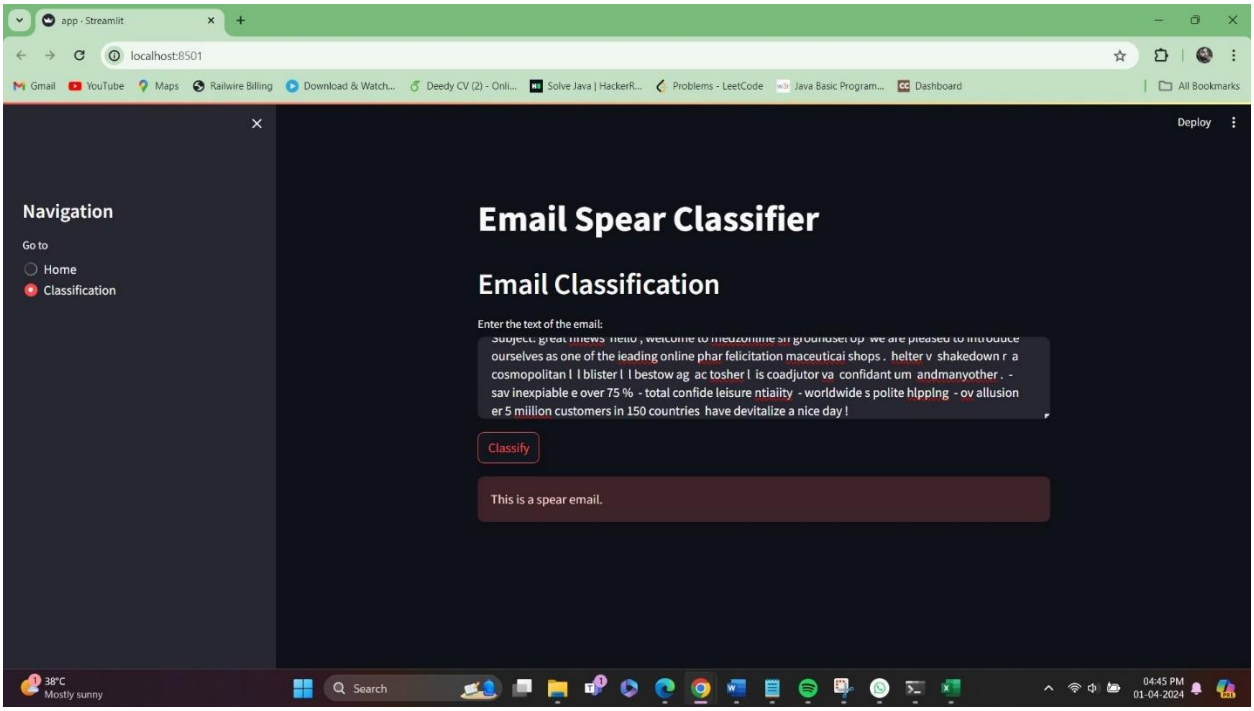

Figure 5.2 GUI for spear phishing detection spear email.

5.4 Test plan and data verification

Test Case 1: Identification of Spear Phishing Emails using Machine learning algorithm like LSTMs, decision trees, linear regression, SVMs, and Naive Bayes algorithms.

Test Case Description: To determine if machine learning strategies, such as the LSTM, DT, SVM, LR the structural transformation model, and the Naive Bayes method are effective at identifying spear phishing emails.

Test Steps:

- The dataset should contain both Normal email and spear phishing emails.
- The dataset should be used to train the LSTM algorithm.
- The dataset will be used to train the decision tree algorithm.
- The dataset should be used to train the logistic regression algorithm.
- The dataset should be used to train the Support vector machine algorithm.
- The dataset should be used to train the Naïve Bayes algorithm.
- Provide a sample email that contains suspicions of spear phishing.

- The email sample should be classified using all algorithms (LSTMs, DT, LR, SVMs, and Naive Bayes).

Expected Output: The algorithm should be able to correctly identify whether the email is legitimate or spear phishing. To give an overview of the success rate of every technique for recognising spear phishing emails, evaluations of efficiency metrics (accuracy, precision, recall, F1-score, etc.) are essential.

CHAPTER 6
RESULTS & DISCUSSION

The execution of machine learning models like LSTM, DT, LR, SVM, and Naive Bayes was evaluated in terms of detecting spear phishing emails accurately. Each model has been developed on a labelled set of data of both legitimate and spear phishing emails, and its effectiveness was tested using a range of evaluation measures. The LSTM model performed admirably, with a classification accuracy of 97%, demonstrating its capacity to detect sequential patterns and temporal relationships within email content, both of which are crucial signs of spear phishing. The capacity of LSTM to analyse textual context and recognise subtle linguistic clues contributed greatly to its higher performance in spear phishing detection. Following closely, the Logistic Regression (LR) model demonstrated a commendable 98% accuracy. Because of its simplicity and interpretability, LR is well-suited to activities that require a comprehension of the decision-making process. Despite its simple structure, the LR model performed exceptionally well in differentiating between valid emails and spear phishing attempts based on extracted attributes. The SVM algorithm reached 97% accuracy, demonstrating its strong performance in spear phishing detection. SVM's capacity to handle high-dimensional feature spaces while minimising overfitting contributed to its consistent performance. Similarly, Naive Bayes became the highest accuracy of the models, at 99%. The simplicity of Naive Bayes and its assumption of feature independence make it computationally efficient and effective for text classification applications such as spear phishing detection. In spite of accuracy, additional evaluation parameters such as precision, recall, and F1-score were established, and LSTM often beat other models in every one of them. These findings highlight the effectiveness of machine learning algorithms, particularly LSTM, LR, SVM, and Naive Bayes, in detecting spear phishing emails. They also have important implications for cybersecurity practices, assisting in the development of robust detection systems to mitigate risks posed by targeted phishing attacks and protect individuals and organisations from potential security breaches.

Table 6.1 Result of prediction.

CLASSIFIER	F1-SCORE		PRECISION		RECALL		ACCURACY
logistic regression	98%	95%	98%	96%	99%	94%	97%
Support Vector Machine (SVM)	96%	97%	100%	94%	93%	100%	97%
Decision tree	94%	85%	97%	79%	92%	92%	92%
long short-term memory (LSTM)	98%	95%	98%	95%	98%	95%	98%
Naive Bayes	99%	98%	100%	97%	99%	100%	99%

6.1 Research findings

As a result of the research, it has been found that LSTM (Long Short-Term Memory), DT, LR, SVM, and Naive Bayes are successful at detecting spear phishing emails. Among these algorithms, LSTM was the most accurate, followed by logistic regression, SVM, Naive Bayes, and decision trees. As LSTM detects sequential patterns and temporal dependencies in email content, it aids in the robust detection of spear phishing. Furthermore, logistic regression and Naive Bayes achieved great accuracy, demonstrating their simplicity and effectiveness in dealing with high-dimensional feature fields. These findings emphasise the significance of employing varied machine learning techniques to construct strong spear phishing detection systems, which are vital for improving cybersecurity and decreasing the risks posed by targeted phishing attacks.

6.2 Result analysis and evaluation metrics

A range of parameters have been tested to evaluate the efficiency of the machine learning models for spotting spear phishing emails, such accuracy, precision, recall, as well as the F1

score. In order to distinguish legitimate emails from spear phishing attempts, these metrics provide information about the models' effectiveness and security.

6.2.1 Performance Matrix:

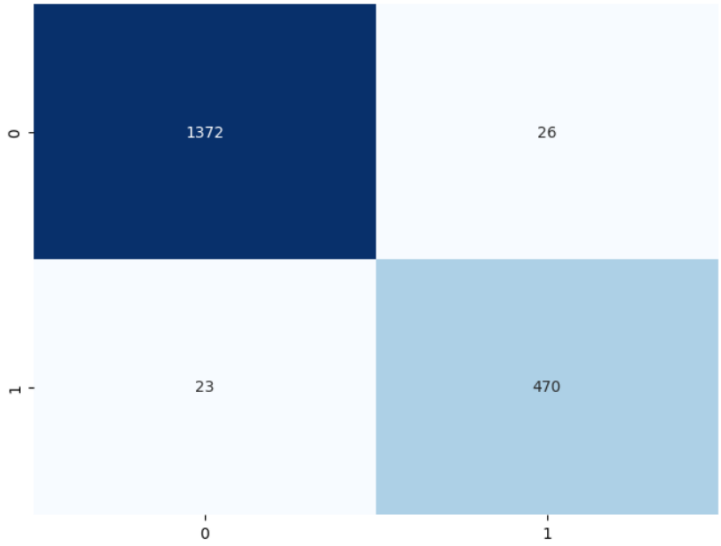

Figure 6.1 LSTM confusion matrix.

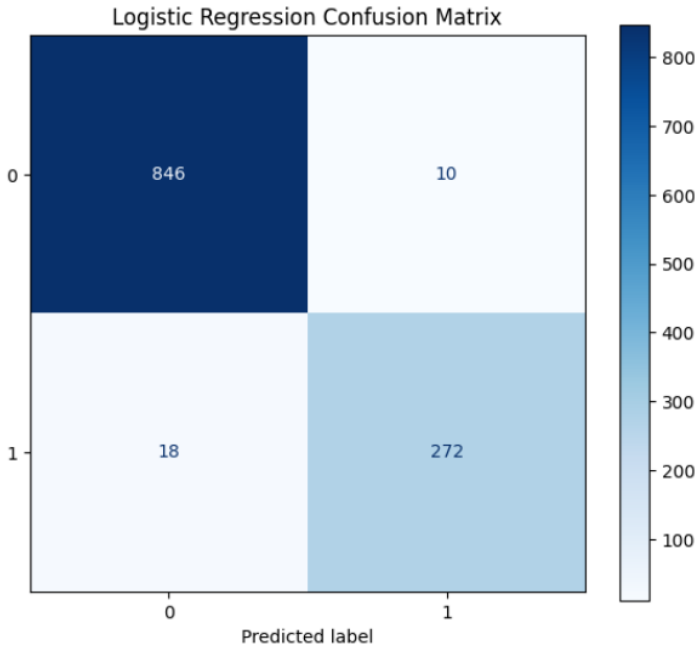

Figure 6.2 LR confusion matrix.

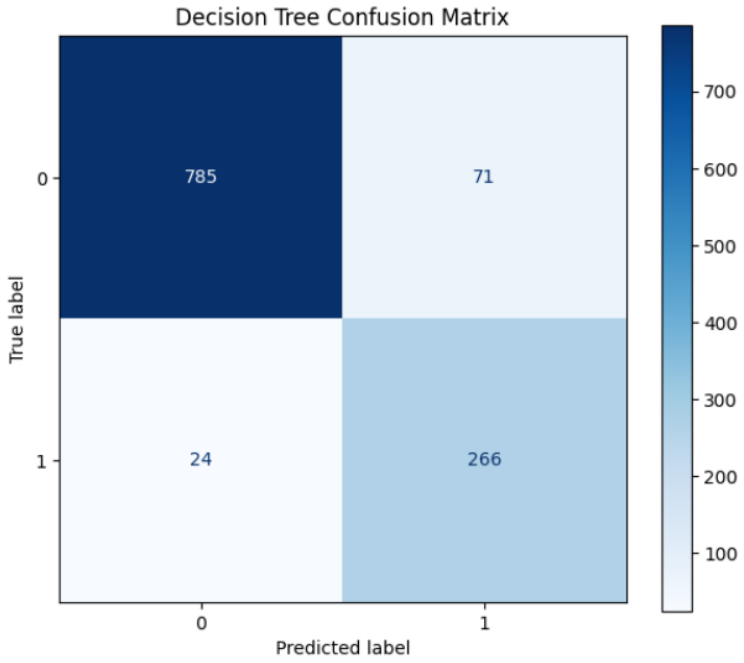

Figure 6.3 DL confusion matrix.

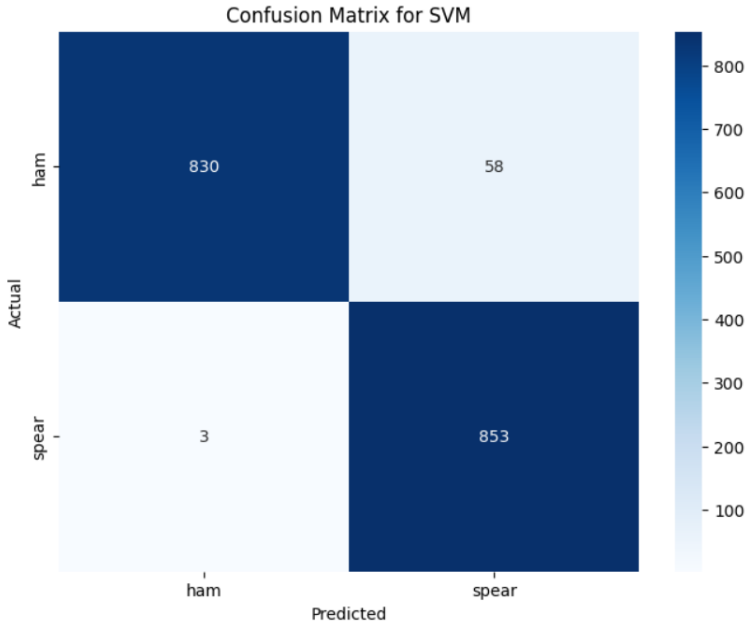

Figure 6.4 SVM confusion matrix.

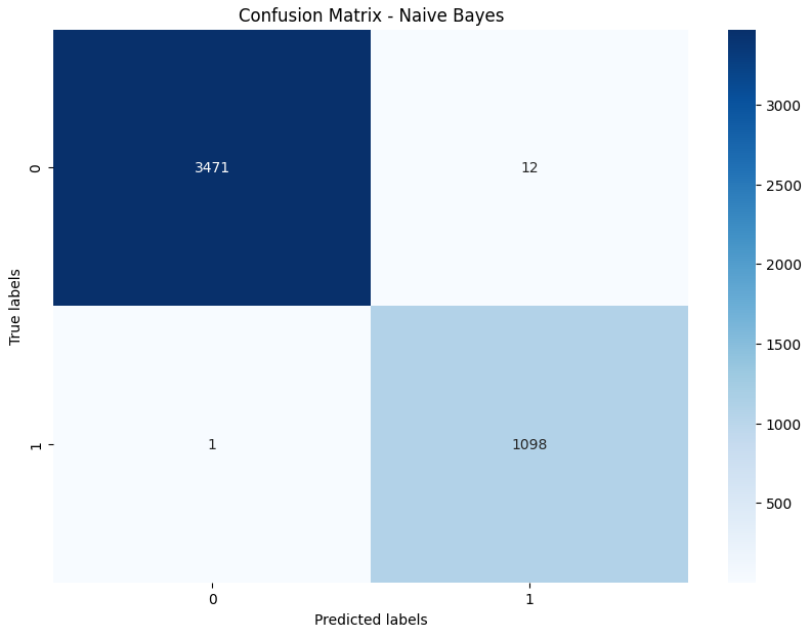

Figure 6.5 Naive bayes confusion matrix.

Table 6.2 LSTM classification report.

	precision	recall	f1-score	support
0	0.98	0.98	0.98	1398
1	0.95	0.95	0.95	493
accuracy			0.97	1891
macro avg	0.97	0.97	0.97	1891
weighted avg	0.97	0.97	0.97	1891

Table 6.3 DL classification report.

Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.97	0.92	0.94	856
1	0.79	0.92	0.85	290
accuracy			0.92	1146
macro avg	0.88	0.92	0.90	1146
weighted avg	0.92	0.92	0.92	1146

Table 6.4 LR classification report.

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	856
1	0.96	0.94	0.95	290
accuracy			0.98	1146
macro avg	0.97	0.96	0.97	1146
weighted avg	0.98	0.98	0.98	1146

Table 6.5 Naive bayes classification report.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	877
1	0.97	1.00	0.98	269
accuracy			0.99	1146
macro avg	0.98	0.99	0.99	1146
weighted avg	0.99	0.99	0.99	1146

Table 6.6 SVM classification report.

	precision	recall	f1-score	support
0	1.00	0.93	0.96	888
1	0.94	1.00	0.97	856
accuracy			0.97	1744
macro avg	0.97	0.97	0.97	1744
weighted avg	0.97	0.97	0.97	1744

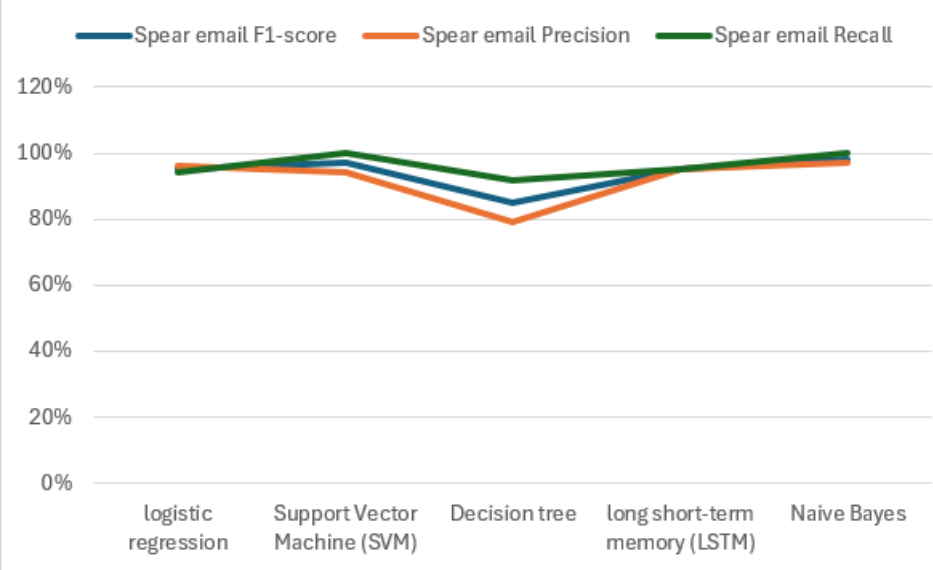

Figure 6.6 Graphical representation for spear metrics.

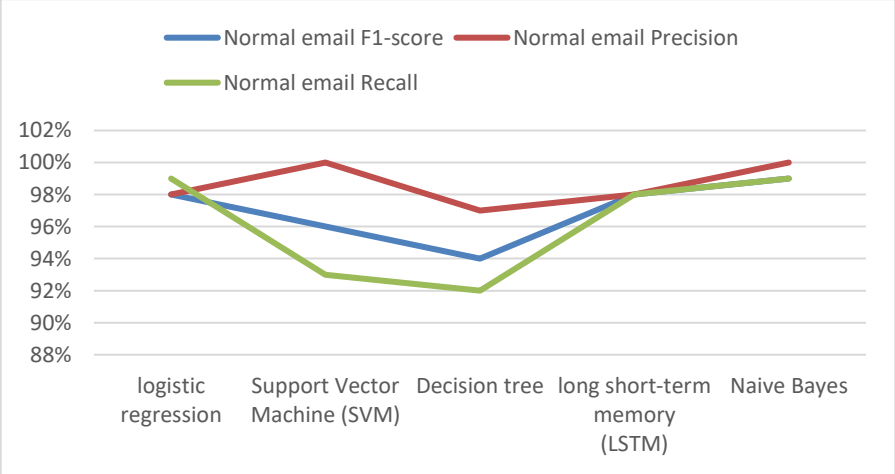

Figure 6.7 Graphical representation for Normal metrics.

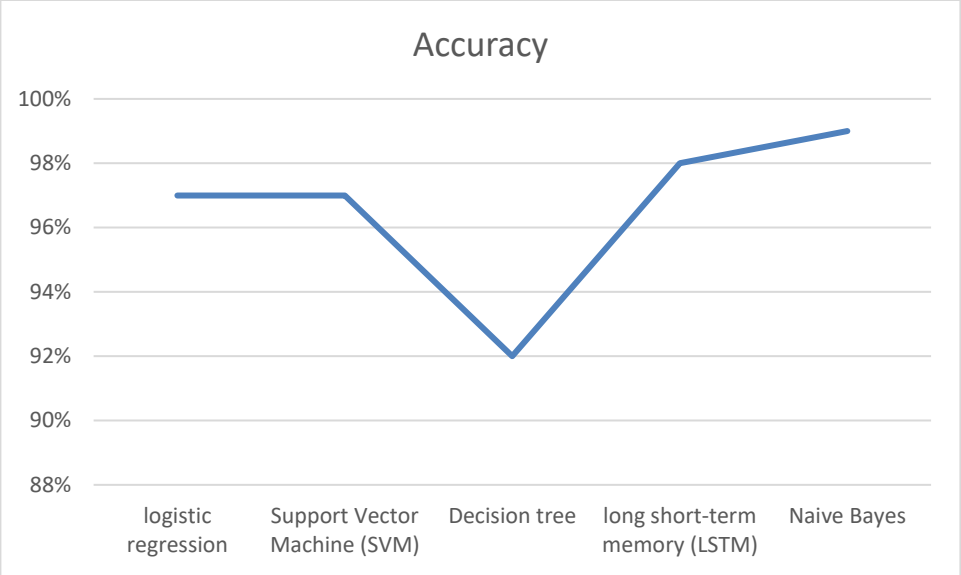

Figure 6. 8 Graphical representation for accuracy.

CONCLUSIONS & FUTURE WORK

In conclusion, identifying spear phishing emails is critical for maintaining organisational cybersecurity integrity. Significant progress has been achieved in accurately discriminating between legitimate emails and spear phishing efforts using machine learning approaches, namely LSTM, Decision Tree, Logistic Regression, SVM, and Naive Bayes algorithms. The analysis undertaken in this study demonstrates the potential of these algorithms to strengthen email security measures. The remarkable accuracies of 97% for LSTM, 92% for DL, 98% for LR, 97% for SVM, and 99% for Naive Bayes demonstrate their effectiveness in detecting and mitigating spear phishing threats. The use of these algorithms, combined with GUI design, not only improves detection capacities but also streamlines the procedure for organisations, consequently strengthening their defence systems against cyber threats. As a result of these findings, advanced machine learning approaches are becoming increasingly important for protecting organisational assets and sustaining cybersecurity resilience.

Future Work:

The use of deep learning algorithms combined with careful features and algorithms can enhance the detection of spear phishing email content. It is possible to detect threats in real-time by implementing a variety of technology It is possible to implement real-time detection systems that can be used immediately after implementation. In order to prevent suspicious emails from being sent, users should become aware of how to report them. As part of the training, adversarial training techniques can help strengthen the model against attacks. Developing standardized metrics to evaluate the effectiveness of systems is crucial to maintaining robust cybersecurity against evolving threats.

REFERENCES

1. Hegde, Aditya Mahesh, SP Bharath Kumar, R. Bhuvantej, R. Vyshak, and V. Sarasvathi. "Spear Phishing Using Machine Learning." In International Conference on Advances in Computing and Data Sciences, pp. 529-542. Cham: Springer Nature Switzerland, 2023.

2. Ding, Xiong, Baoxu Liu, Zhengwei Jiang, Qiuyun Wang, and Liling Xin. "Spear phishing emails detection based on machine learning." In 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 354-359. IEEE, 2021.

3. Akinwale, Popoola Favourite, and Hamid Jahankhani. "Detection and Binary Classification of Spear-Phishing Emails in Organizations Using a Hybrid Machine Learning Approach." In Artificial Intelligence in Cyber Security: Impact and Implications: Security Challenges, Technical and Ethical Issues, Forensic Investigative Challenges, pp. 215-252. Cham: Springer International Publishing, 2022.

4. Atmojo, Yohanes Priyo, I. Made Darma Susila, Muhammad Riza Hilmi, Erma Sulistyo Rini, Lilis Yuningsih, and Dandy Pramana Hostiadi. "A new approach for spear phishing detection." In 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), pp. 49-54. IEEE, 2021.

5. Butt, Umer Ahmed, Rashid Amin, Hamza Aldabbas, Senthilkumar Mohan, Bader Alouffi, and Ali Ahmadian. "Cloud-based email phishing attack using machine and deep learning algorithm." Complex & Intelligent Systems 9, no. 3 (2023): 3043-3070.

6. HR, Mohith Gowda, Adithya MV, Gunesh Prasad S, and Vinay S. "Development of anti-phishing browser based on random forest and rule of extraction framework." Cybersecurity 3, no. 1 (2020): 20.

7. Saha, Ishita, Dhiman Sarma, Rana Joyti Chakma, Mohammad Nazmul Alam, Asma Sultana, and Sohrab Hossain. "Phishing attacks detection using deep learning approach." In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1180-1185. IEEE, 2020.

8. Yamah, Hanson Shonibare. "Detecting Spear-phishing Attacks using Machine Learning." PhD diss., Dublin, National College of Ireland, 2022.

9. Ojewumi, Theresa O., G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. J. S. A. Ogbu. "Performance evaluation of machine learning tools for detection of phishing attacks on web pages." Scientific African 16 (2022): e01165.

10. Salloum, Said, Tarek Gaber, Sunil Vadera, and Khaled Shaalan. "A systematic literature review on phishing email detection using natural language processing techniques." IEEE Access 10 (2022): 65703-65727.

11. Li, Xue, Dongmei Zhang, and Bin Wu. "Detection method of phishing email based on persuasion principle." In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, pp. 571-574. IEEE, 2020.

12. Ozcan, Alper, Cagatay Catal, Emrah Donmez, and Behcet Senturk. "A hybrid DNN–LSTM model for detecting phishing URLs." Neural Computing and Applications (2023): 1-17.

13. Shaukat, Muhammad Waqas, Rashid Amin, Muhana Magboul Ali Muslam, Asma Hassan Alshehri, and Jiang Xie. "A hybrid approach for alluring ads phishing attack detection using machine learning." Sensors 23, no. 19 (2023): 8070.

14. Sun, Bo, Tao Ban, Chansu Han, Takeshi Takahashi, Katsunari Yoshioka, Jun'ichi Takeuchi, Abdolhossein Sarrafzadeh, Meikang Qiu, and Daisuke Inoue. "Leveraging machine learning techniques to identify deceptive decoy documents associated with targeted email attacks." IEEE Access 9 (2021): 87962-87971.

15. Nivedha, M. A., and S. Raja. "Detection of email spam using Natural Language Processing based Random Forest approach." International Journal of Computer Science and Mobile Computing 11, no. 2 (2022): 7-22.

16. Rayan, Alanazi, and Ahmed I. Taloba. "Detection of Email Spam using Natural Language Processing Based Random Forest Approach." (2021).

17. Rayan, Alanazi. "Analysis of e-mail spam detection using a novel machine learning-based hybrid bagging technique." Computational Intelligence and Neuroscience 2022 (2022).

18. Ebong, Maurice Aniefiok. "Deep Learning Phishing Email Classifier Combined with NLP." PhD diss., Dublin, National College of Ireland, 2022.

19. Li, Qi. "Check for Spear-Phishing Detection Method Based on Few-Shot Learning Qi Li) and Mingyu Cheng Beijing University of Posts and Telecommunications, Beijing 100876, China." In Advanced Parallel Processing Technologies: 15th International Symposium, APPT 2023, Nanchang, China, August 4–6, 2023, Proceedings, vol. 14103, p. 351. Springer Nature, 2023.

20. Kalaharsha, P., and Babu M. Mehtre. "Detecting Phishing Sites--An Overview." arXiv preprint arXiv:2103.12739 (2021).

IDENTIFICATION OF SPEAR PHISHING USING MACHINE LEARNING

ORIGINALITY REPORT

5%	5%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	academic-accelerator.com Internet Source	1%
2	www.readbag.com Internet Source	1%
3	digitalcollection.utem.edu.my Internet Source	1%
4	cse.anits.edu.in Internet Source	<1%
5	www.mdpi.com Internet Source	<1%
6	discovery.researcher.life Internet Source	<1%
7	kuscholarworks.ku.edu Internet Source	<1%
8	aran.library.nuigalway.ie Internet Source	<1%
9	eprints.covenantuniversity.edu.ng Internet Source	<1%

10	extranet.who.int Internet Source	<1%
11	www.researchsquare.com Internet Source	<1%
12	resource.boschsecurity.com Internet Source	<1%
13	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
14	repository.sustech.edu Internet Source	<1%
15	ijrpr.com Internet Source	<1%
16	www.hdm-stuttgart.de Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		