Spear Phishing Email Detection Using Machine Learning

Report to be submitted in partial fulfillment of the requirements for the degree

of

Bachelor of Computer Applications (BCA)

VIDYASAGAR UNIVERSITY
Academic Year:2024-2025

SUPERVISED BY:
Dr. Chiranjit Changdar
Assistant Professor
Department of BCA &
Computer Science
Belda College

SUBMITTED BY:
Shibnath Rana
(VU221021817 of 2022-2023)
Roll - 4116102 No - 22020

Department of Computer Sc. & Application

Kedar Memorial Building (1st Floor), Belda College
Belda, Paschim Medinipur, WB - 721424
compsc@beldacollege.ac.in, bca@beldacollege.ac.in

Certificate

This is to certify that we have examined the thesis entitled **A Study on Spear Phishing Email Detection Using Machine Learning**, submitted by **Shibnath Rana** (Roll Number: 4116102-22020 and Reg. No – VU221021817 of 2022-23) a bachelor student of Department of BCA & Computer Science in partial **fulfillment** for the award of degree of **Bachelor of Computer Application (BCA)**.

We hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial **fulfillment** for the Graduation Degree for which it has been submitted. The thesis has fulfilled all the requirements as per the regulations of the Institute and has reached the standard needed for submission.

(Head of the Department)	**(Supervisor)**
Dr. Debasis Dhal	Dr. Chiranjit Chanda
Assistant Professor	Assistant Professor
Department of BCA &	Department of BCA &
Computer Science	Computer Science
Belda College	Belda College

ACKNOWLEDGEMENTS

ABSTRACT

Spear phishing has emerged as one of the most critical cybersecurity threats in the digital landscape, targeting individuals and organizations through highly personalized and deceptive emails designed to steal sensitive information or carry out malicious activities. Unlike traditional phishing attacks, spear phishing exploits context-specific knowledge and social engineering tactics, rendering conventional rule-based and signature-based detection systems largely ineffective. This thesis presents a comprehensive machine learning-based detection framework enhanced with natural language processing (NLP) techniques to effectively identify and classify spear phishing emails. The study employs a variety of classical machine learning classifiers—Logistic Regression, Naive Bayes, Support Vector Machines (SVM), Decision Trees, K-Nearest Neighbors (KNN), Random Forest, and Gradient Boosting—applied to a combined dataset of near about 5,000 labeled emails from two publicly available sources. These emails were preprocessed and engineered to extract key features such as lexical patterns, header metadata, keyword frequency, sender-recipient relationships, and semantic cues within the message content. The feature selection process was informed by prior phishing research and known attack signatures. Using Python's Scikit-learn library, each model was trained on 80% of the dataset and validated on the remaining 20%. The evaluation metrics—accuracy, precision, recall, F1-score, confusion matrix, and AUC-ROC curves—provided a detailed analysis of the models' performance. Logistic Regression and Naive Bayes achieved the highest accuracy (95%) and F1-score (0.95), while ensemble methods like Random Forest and Gradient Boosting also demonstrated strong results (F1-score of 0.93). Despite achieving high AUC scores, models like KNN and SVM showed comparatively lower recall, which is critical in minimizing false negatives. Beyond numerical performance, this research underscores the importance of integrating machine learning into real-world email filtering systems to detect spear phishing proactively.

Keywords: Spear phishing, Privacy, Machine learning, Natural language processing, Phishing detection

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the modern digital era, email remains a primary communication tool for individuals and organizations. However, this widespread reliance on email has also made it a common target for cyberattacks. Among these, spear phishing has become one of the most dangerous and effective attack vectors. Unlike traditional phishing, which targets a broad audience with generic messages, spear phishing is highly targeted and personalized, often using specific information about the victim to appear legitimate and trustworthy. This makes it significantly harder to detect using conventional security methods.

The increasing sophistication of spear phishing attacks has revealed the limitations of traditional detection techniques such as rule-based filters, blacklists, and signature-based systems. These approaches often fail to recognize new or customized attack patterns, allowing malicious emails to bypass security defenses. As a result, there is a pressing need for more advanced, adaptive, and intelligent detection mechanisms.

Machine learning (ML) offers a promising solution to this challenge. By analyzing large volumes of data and learning from patterns in past attacks, ML models can detect subtle indicators of spear phishing that may not be visible to traditional systems. Techniques such as natural language processing (NLP), anomaly detection, and deep learning enable the identification of unusual communication behavior, suspicious language patterns, and fraudulent sender profiles.

This paper explores the application of machine learning techniques to detect spear phishing emails. It investigates current methodologies, highlights the challenges involved—such as feature selection, dataset limitations, and false positive rates—and evaluates the effectiveness of various ML models. The goal is to contribute to the de-

velopment of more robust and intelligent systems capable of protecting users against increasingly sophisticated spear phishing threats

Anti-Phishing Working Group (APWG) also highlight that phishing at tacks increases in recent years. Fig. 1 illustrates that the APWG Phishing Ac tivity Trends Report from third Quarter 2023 to second Quarter 2024. APWG recorded 1,025,968 phishing attempts in total during the first quarter of 2024. A new record and the worst phishing quarter ever was set by the APWG in the second quarter of 2024, when 932,923 total phishing assaults were recorded. Since early 2023, there have been four times as many phishing attacks reported to APWG. Increase in smishing and vishing, which together have increased by about 70 percent is a major trend in mobile phone-based fraud.

Figure 1.1: Reported Phishing Attacks (2022-2024)

Third Quarter 2024 to Second Quarter 2023 Phishing Activity Trends Report by Anti-Phishing Working Group(APWG)
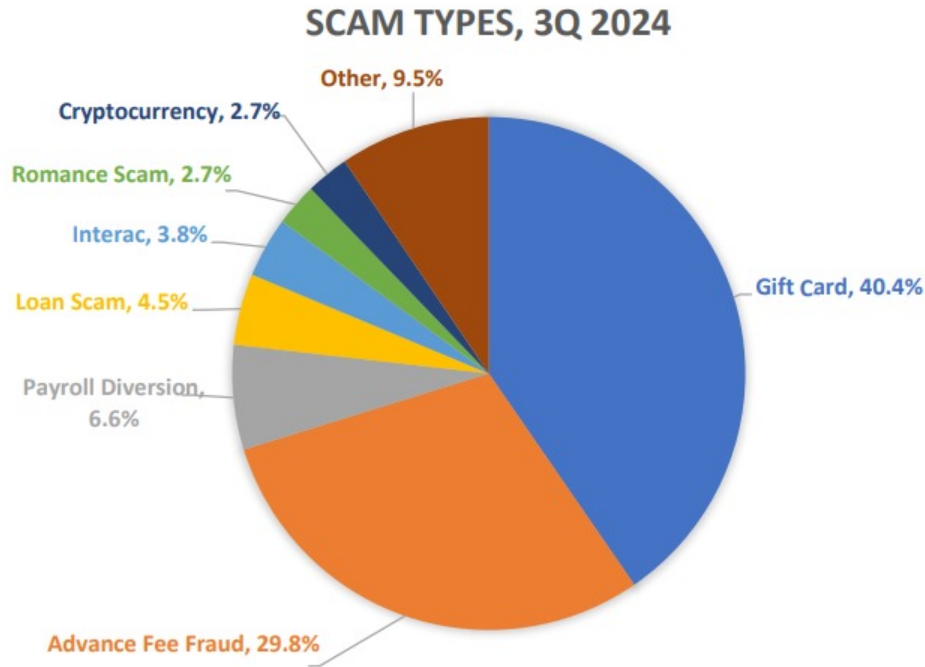
Figure 1.2: SCAM TYPES, 3Q 2024

1.1 Motivation

Schuetz et al. (2016)Phishing is a widespread and escalating form of cybercrime that results in billions of dollars in losses each year. Among its various types, spearphishing is particularly dangerous due to its targeted nature and reliance on social engineering. It involves crafting emails that appear authentic, often tricking more than 70% of recipients, as shown in several studies. Traditional anti-phishing training programs have had limited success in helping users recognize these threats. One reason is that cybersecurity is often viewed as secondary to primary job responsibilities. Another is that users are typically unmotivated to engage in time-consuming training sessions. Therefore, a more effective approach must be short, minimally disruptive, and capable of bringing about meaningful behavioral change. To address this, the study explores the use of fear appeals combined with protection motivation theory (PMT), suggesting that a brief yet impactful message can encourage protective behavior. Additionally, construal-level theory (CLT) is integrated to enhance how users perceive and mentally respond to phishing threats, strengthening the effect of fear-based messages. A real-world field experiment is planned to evaluate

the effectiveness of this new training strategy. The study aims to (1) deliver concise, fear-based training, (2) extend PMT with CLT, and (3) demonstrate a comprehensive application of CLT in cybersecurity education.

1.2   Background

Schuetz et al. (2016)Phishing is a type of cyberattack that uses social engineering to deceive victims and steal sensitive information. It typically involves attackers posing as trustworthy entities to trick individuals into revealing confidential data. These attacks often spread through malicious emails embedded with malware, which, once installed, can compromise the victim's system and infect others. Phishing comes in various forms, such as email phishing, vishing, smishing, angler phishing, whaling, and spear phishing. This study emphasizes spear phishing — a highly targeted attack aimed at specific individuals or institutions, often of influence. Attackers collect detailed information about the target to craft convincing messages tailored specifically for them. Interestingly, many spear-phishing attacks follow identifiable patterns, making it possible to detect threats by analyzing these patterns in emails or social media content. Machine learning (ML) techniques are used in this research to detect such attacks by classifying data into predefined categories. Algorithms such as decision trees, logistic regression, support vector machines, multinomial Naive Bayes, and K-nearest neighbors are used. To enhance accuracy, ensemble methods like Random Forest (parallel) and AdaBoost or Stacking (sequential) are also applied. Results show that ensemble approaches perform better than traditional algorithms in predicting spear-phishing attempts in emails.

1.3   Research Issues

Martin et al. (2018)Spear phishing remains a major cybersecurity concern, with attackers constantly evolving their methods to bypass existing defenses. This study presents a comprehensive analysis of detection and prevention strategies, supported by real-time examples. It explores various spear phishing techniques, such as email spoofing, malware deployment, social engineering, and whaling attacks. Detection methods including signature-based, anomaly-based, and hybrid machine learning approaches are thoroughly examined, highlighting their respective strengths and lim-

Figure 1.3: Type of Phishing Attack

itations. Additionally, key prevention strategies like user awareness training, email authentication, filtering systems, incident response planning, and multifactor authentication are emphasized as critical tools for defense.

The research introduces a systematic taxonomy that categorizes detection and prevention techniques, providing a clear overview of the field. It also integrates legal perspectives, addressing spear phishing prevention through penal jurisprudence and criminological viewpoints, thereby bridging the gap between technical solutions and policy frameworks. Notably, the study identifies significant challenges, such as the ever-changing nature of spear phishing tactics, techniques to evade detection, and the integration of AI into cybersecurity systems.

To address these concerns, the research proposes open questions for future studies and calls for collaboration between academia, industry, and policymakers. These efforts aim to create more robust, adaptive strategies to counter this persistent and evolving threat effectively.

1.4 Research Objectives

Stembert et al. (2015)Cyber criminals use phishing emails in high-volume and spear phishing emails in low volume to achieve their malicious objectives. Hereby they inflict financial, reputational, and emotional damages on individuals and organizations. These (spear) phishing attacks get steadily more sophisticated as cyber criminals

use social engineering tricks that combine psychological and technical deceptions to make malicious emails as trustworthy as possible. Such sophisticated (spear) phishing emails are hard for email protection systems to detect. Security researchers have studied users' ability to perceive, identify and react upon email (spear) phishing attacks.

In this study we have surveyed recent works on understanding how to prevent end-users from falling for email (spear) phishing attacks. Based on the survey we design and propose a novice method that combines interaction methods of reporting, blocking, warning, and embedded education to harness the intelligence of expert and novice users in a corporate environment in detecting email (spear) phishing attacks. We evaluate the design based on a qualitative study, in three experimental steps, by using a mockup prototype, and with 24 participants. We report on the insights gained, indicating that the proposed combination of the interaction methods is promising, and on future research directions.

1.5   Organization of the Report

This report is structured into eight comprehensive chapters, each focusing on a specific aspect of the study to systematically address the detection of spear phishing emails using machine learning techniques.

- **Chapter 1** Introduces the concept of spear phishing, its significance in cybersecurity, and the motivation behind the study. It outlines the problem statement, background, research issues, and objectives.

- **Chapter 2** Provides an overview of existing work in spear phishing detection. It discusses various traditional, rule-based, machine learning, deep learning, and NLP-based approaches, offering theoretical and practical insights.

- **Chapter 3** Describes the classification algorithms used in the study, including Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Trees, Random Forests, Gradient Boosting, and Support Vector Machines, highlighting their strengths and applications.

- **Chapter 4** Details the datasets used in the study, including their sources, composition, preprocessing steps, and statistical characteristics. It explains how the data was prepared for model training and evaluation.

- **Chapter 5** Defines the performance metrics used to evaluate model effectiveness, such as accuracy, precision, recall, F1-score, and confusion matrix. Each metric's relevance to phishing detection is also explained.

- **Chapter 6** Describes the step-by-step process of the model development. It includes data preprocessing, feature extraction and selection, model training, testing, and validation processes using the CRISP-DM framework.

- **Chapter 7** Presents the results of the classification models, including confusion matrix analysis, performance metrics, and AUC-ROC curve comparisons. The strengths and weaknesses of each model are discussed in detail.

- **Chapter 8** Summarizes the key findings, reaffirms the effectiveness of the proposed detection approach, and suggests possible directions for future research, including real-time detection and integration of advanced deep learning models.

Chapter 2

Literature Review

2.1   Introduction

Spear phishing remains one of the most sophisticated and damaging forms of cyber-attacks, targeting specific individuals or organizations with personalized messages. Unlike generic phishing attempts, spear phishing leverages social engineering and contextually rich information, making traditional detection methods less effective. As a result, recent research has increasingly focused on the application of machine learning (ML) techniques to detect and prevent spear phishing attacks with greater accuracy.

2.2   Theoretical Framework

Martin et al. (2018) examined the usefulness of equal-variance signal detection theory (EVSDT) in assessing how people detect phishing and spear-phishing emails. While most cybersecurity breaches stem from mistaken responses to such emails, it remains unclear how to best measure detection performance. Specifically, questions remain about whether the equal-variance assumption in signal detection models holds and how threat level influences detection—through changes in mean separation or evidence variability. Using an online inbox simulation, the study showed that EVSDT can effectively quantify detection accuracy and response bias. Results revealed that threat level influenced mean separation but not signal variance, suggesting that susceptibility is tied to stable individual differences rather than fluctuating variability. The study supports using EVSDT as a practical tool for evaluating phishing detec-

tion and training effectiveness. Notably, participants were especially poor at detecting spear-phishing emails, with accuracy as low as 40%

Tchakounte et al. (2021) Asolution to help victims against phishing is anticipating and leveraging impacts related to phisher actions.In this regard, this work reshapes game theoretical logic between Intrusion Detection System (IDS) agents and insiders to email spear-phishing interactions. The email spear-phishing attack is designed as a non-cooperative and repeated game between opponents. Additionally, thisworkreliesonQuantalResponseEquilibrium(QRE)to build a game theoretical approach to predict the phisher's future intent based on past actions of both players. This approach is coupled with a recommendation strategy appropriate allocation of resources to invest to strengthen user protection. Simulations on spear-phishing scenarios demonstrate the ability of the final system to intuitively guess the most likely phisher decisions. This work provides intelligence to spearphishing detectors and humans such that they can anticipate next phisher actions.

2.3   Spear-phishing email prevention methods

Stembert et al. (2015) sophisticated as cyber criminals use social engineering tricks that combine psychological and technical deceptions to make malicious emails as trustworthy as possible. Such sophisticated (spear) phishing emails are hard for email protection systems to detect. Security researchers have studied users' ability to perceive, identify and react upon email (spear) phishing attacks. In this study we have surveyed recent works on understanding how to prevent end-users from falling for email (spear) phishing attacks. Based on the survey we design and propose a novice method that combines interaction methods of reporting, blocking, warning, and embedded education to harness the intelligence of expert and novice users in a corporate environment in detecting email (spear) phishing attacks. They evaluate the design based on a qualitative study, in three experimental steps, by using a mockup prototype, and with 24 participants. They report on the insights gained, indicating that the proposed combination of the interaction methods is promising, and on future research directions.

Al-Hamar et al. (2021)provided a solution that can detect targeted Spear-phishing attacks based on required similarities in the specific domain which it has been targeted. The strategy is to figure out whether the domain is genuine or a forgery,

which is to be evaluated by multi novel grading algorithms. Therefore, this research addresses targeted attacks on specific organisations by presenting a new enterprise solution. This detection system focuses on domain names, which tend to be registered domain names trusted by the victims. The results from this investigation show that this detection system has proven its ability to reduce email phishing attacks significantly.

2.4   Traditional Approaches

Evans et al. (2022) considerable research has been conducted into the use of Machine Learning (ML) techniques for spear-phishing detection. ML-based solutions are vulnerable to zero-day attacks, as when the algorithms do not have access to the relevant historical data, they cannot be reliably trained. Furthermore, email address spoofing is a low-effort yet widely applied forgery technique in spear phishing which the standard email protocol SMTP fails to detect without the use of extensions. Detecting this type of spear threat requires (i) a close investigation of each sender within the mailbox; and (ii) a thorough exploration of the similarity of its characteristics to the spoofed email. This raises scalability challenges due to the growing number of features relevant for investigation and comparison, which is proportional to the number of the senders within a particular mailbox. This differs from traditional phishing attacks, which typically look at email bodies and are generally limited to a binary classification between 'phishing' and 'benign' emails.

Mondal et al. (2022) These attacks are often termed as "phishing." It can be described as a social engineering attack involving sophisticated attack vectors that can be used to steal sensitive information from a victim, to be more precise as per the definition of phishing. In phishing, the attacker disguises himself as a trustworthy entity, uses several tactics to win the victim's trust, and then convinces the victim to reveal the requisite information to commit the fraud. This kind of attack can be done by sending malicious emails containing malware that can leak information when installed in the victim's system. Such attacks can further be spread through the hacked systems to other victims. Phishing can again be classified into several types: Spear phishing, email phishing, whaling, smishing, vishing, and angler phishing. Among these, this chapter primarily concentrates on spear phishing. Spear phishing is a kind

of email or social media scam that generally targets some specific individual or institution. Most of the time, the target happens to be influential. In spear phishing, the attacker gathers as much information as possible about the target to tailor the attack for that particular victim. However, many times, these attacks are found to exhibit a similar pattern. So, detecting that pattern in any email or social media post can predict any possible threat in the same. This chapter mainly focuses on finding these patterns and labeling posts as possible spear-phishing attacks using machine-learning (ML) algorithms. The classification algorithms categorize structured or unstructured data into a given number of decision classes.

2.5   Rule Based Approaches

Moghimi & Varjani (2016) present a new rule-based method to detect phishing attacks in internet banking. Their rule-based method used two novel feature sets, which have been proposed to determine the webpage identity. Their proposed feature sets include four features to evaluate the page resources identity, and four features to identify the access protocol of page resource elements. They used approximate string matching algorithms to determine the relationship between the content and the URL of a page in their first proposed feature set. Their proposed features are independent from third-party services such as search engines result and/or web browser history. We employed support vector machine (SVM) algorithm to classify webpages. Their experiments indicate that the proposed model can detect phishing pages in internet banking with accuracy of 99.14% true positive and only 0.86% false negative alarm. Output of sensitivity analysis demonstrates the significant impact of their proposed features over traditional features. They extracted the hidden knowledge from the proposed SVM model by adopting a related method. They embedded the extracted rules into a browser extension named PhishDetector to make their proposed method more functional and easy to use. Evaluating of the implemented browser extension indicates that it can detect phishing attacks in internet banking with high accuracy and reliability. PhishDetector can detect zero-day phishing attacks too.

Basnet et al. (2011) propose a rule-based method to detect phishing webpages. They first study a number of phishing websites to examine various tactics employed by phishers and generate a rule set based on observations. They then use Decision

Tree and Logistic Regression learning algorithms to apply the rules and achieve 95-99% accuracy, with a false positive rate of 0.5-1.5% and modest false negatives. Thus, it is demonstrated that our rulebased method for phishing detection achieves performance comparable to learning machine based methods, with the great advantage of understandable rules derived from experience.

2.6   Machine learning Approaches

Samad & Gani (2020) mitigate the impact of spear phishing by utilizing the multi-layer approach. The multi-layer approach is the best method of managing the web interruption, as the intruder needs to experience shift levels. Practically all the scientists are dealing with the content of the email; however, this paper picks a novel method to counter the phishing messages by utilizing both the attachment and content of an email. They applied sentimental analysis on emails, including both content of the email and the attachment, to check whether they are spam or not using SVM classifier and Randomforest Classifier; the former showed 96 percent accuracy while, as later offers 97.66 percent accuracy. SVM showed false-positive 0 percent and false-negative 4 percent, while RandomForest showed 0 percent false-positive and 2.33 percent false-negative ratios. They also performed topic modeling using LDA(Latent Dirichlet Allocation)) from Gensim package to get the dominant topics in our dataset. They visualized the results of our topic model using pyLDvis.

Hegde et al. (2023) discusses the use of machine learning algorithms and tools for automating email phishing attacks. The personal data of the victim is gathered from open-source websites in order to analyze their online activity. This information is then analyzed in an attempt to learn more about the victim's interests and a relevant email template is created based on this information. The machine learning algorithm is then provided with the constructed template, which predicts how successful a phishing attack would be if launched. This paper uses a machine learning algorithm that is composed of support vector machines (SVM) and logistic regression (LR). This hybrid algorithm, which is a combination of SVM and logistic regression, achieves a peak accuracy of 99.69% when compared to using only one type of classification method, such as SVM or LR. The purpose of this paper is to increase the effectiveness of phishing attacks by automating the data extraction process and to analyse the success rate of attack using machine learning before launching it. This paper will

serve the interests of institutions/companies by providing a convenient way to conduct automatic phishing as part of Cyber-Security training to educate employees, giving them a practical experience of social engineering attack.

2.7 Deep Learning Approaches

Birthriya et al. (2025) analyzed various spear phishing techniques which include email spoofing, social engineering, malware, whaling attacks. Detection techniques including signature-based, anomaly-based machine learning and hybrid approaches are examined herein with an emphasis on their strengths and weaknesses. User awareness training, email authentication, filtering, incident response planning and multifactor authentication were identified as important strategies to try out. A structured overview of the field is provided through systematic taxonomy for classification of these methods. The survey also considers legal frameworks and institutional policies that often go unnoticed regarding spear phishing prevention by incorporating perspectives from penal jurisprudence and criminology to bridge technical and legal approaches. Other critical challenges that have been identified include changing nature of spear phishing tactics; detection evasion; integration of artificial intelligence (AI) into cyber security defenses among others. Henceforth it offers insights into these challenges while proposing open research questions aimed at guiding future research initiatives coupled with encouraging collaborations between academia industry policy makers. Such contributions will help develop more effective adaptive strategies against this persistent threat.

Li & Cheng (2023)spear-phishing detection based machine learning and deep learning have become the focus of researchers in recent years. However, because of a smaller range and less attack frequency, the number of spear-phishing emails is very limited. How to detect spear-phishing based on machine learning and deep learning with small samples has become a key issue. Meanwhile, in machine learning and deep learning, few-shot learning aims to study a better classification model trained with only a few samples. Therefore, we propose a spear-phishing detection method based on few-shot learning that combines the basic features and the message body of emails. We propose a simple word-embedding model to analyzes the message body, which can process the message body of different lengths into text feature vectors with the same dimension, thus retaining the semantic information to the greatest extent.

Then the text feature vectors are combined with the basic features of emails and input into commonly used machine learning classifiers for detection. Our proposed simple word-embedding method does not require the complex training of the model to learn a large number of parameters, thereby reducing the dependence of the model on a large number of training data. The experimental results show that the method proposed in this paper achieves better performance than the existing spear-phishing detection method. Especially, Especially, the advantages of our detection method are more obvious with small samples.

2.8 Natural Language Processing Approaches

Shamoo (2025) application of Natural Language Processing (NLP) in detecting and mitigating these malicious activities. By analyzing linguistic patterns and textual characteristics, NLP techniques can distinguish between legitimate and deceptive content, offering a robust, automated defense mechanism against a wide array of cyber threats. The chapter discusses various NLP-based methods, including text classification, feature extraction, sentiment analysis, and machine learning algorithms, that can effectively identify phishing and spam messages. Real-world case studies and practical implementations are also presented, providing insight into the operational efficacy of NLP in cybersecurity. The chapter concludes with future research directions, addressing the evolving tactics of cybercriminals and the potential of NLP to keep pace with emerging threats.

Kotson & Schulz (2015) use Natural Language Processing techniques to investigate a specific real-world phishing campaign and quantify attributes that indicate a targeted spear phishing attack. Our phishing campaign data sample comprises 596 emails - all containing a web bug and a Curriculum Vitae (CV) PDF attachment - sent to our institution by a foreign IP space. The campaign was found to exclusively target specific demographics within our institution. Performing a semantic similarity analysis between the senders' CV attachments and the recipients' LinkedIn profiles, we conclude with high statistical certainty (p ¡; 10-4) that the attachments contain targeted rather than randomly selected material. Latent Semantic Analysis further demonstrates that individuals who were a primary focus of the campaign received CVs that are highly topically clustered. These findings differentiate this campaign from one that leverages random spam.

Shamoo (2025) explores the application of Natural Language Processing (NLP) in detecting and mitigating these malicious activities. By analyzing linguistic patterns and textual characteristics, NLP techniques can distinguish between legitimate and deceptive content, offering a robust, automated defense mechanism against a wide array of cyber threats. The chapter discusses various NLP-based methods, including text classification, feature extraction, sentiment analysis, and machine learning algorithms, that can effectively identify phishing and spam messages. Real-world case studies and practical implementations are also presented, providing insight into the operational efficacy of NLP in cybersecurity. The chapter concludes with future research directions, addressing the evolving tactics of cybercriminals and the potential of NLP to keep pace with emerging threats.

2.9   Artificial Neural Network

Mohamed et al. (2024) examines the integration of Artificial Intelligence (AI) in combating spear phishing, through an in-depth analysis of 23 pivotal studies. It highlights the progression of AI-enabled detection technologies that employ advanced algorithms to identify and mitigate sophisticated email threats. The coverage extends across a variety of AI methodologies, such as machine learning, natural language processing, and behavioral analytics, demonstrating their effectiveness in detecting malicious email activities that conventional security protocols often overlook. Their review identifies significant strides in utilizing AI to combat the subtle and evolving nature of spear phishing attacks but also acknowledges the emergence of new challenges, including adaptability issues, increased false positives, and the necessity for ongoing adaptation to address emerging threats. This paper elucidates the ongoing battle between spear phishing strategies and AI-based defenses, providing insights into future directions for research and innovation in the vital area of cybersecurity.

Mohamed et al. (2025) presents a critical analysis of the role of Artificial Intelligence (AI) in defending against spear phishing attacks, which continue to be a significant cybersecurity threat. By examining 30 seminal studies, we provide an in-depth evaluation of current AI techniques, such as machine learning, natural language processing, and behavioural analytics, which are utilized to detect and mitigate sophisticated email threats. Our review uncovers that AI not only significantly enhances the detection capabilities against these targeted attacks but also faces challenges like

adaptability and false positives. These findings highlight the continuous evolution of AI strategies in spear phishing defense and the need for ongoing innovation to keep pace with advanced threat tactics. This paper aims to guide future research by proposing integrated AI solutions that enhance both detection capabilities and responsiveness to new threats, thereby strengthening cybersecurity defenses in an increasingly digital world.

2.10 Taxonomy of an Email Message

Do et al. (2022) introduces the concept of phishing and deep learning in the context of cybersecurity. Then, taxonomies of phishing detection and deep learning algorithm are provided to classify the existing literature into various categories. Next, taking the proposed taxonomy as a baseline, this study comprehensively reviews the state-of-the-art deep learning techniques and analyzes their advantages as well as disadvantages. Subsequently, the paper discusses various issues that deep learning faces in phishing detection and proposes future research directions to overcome these challenges. Finally, an empirical analysis is conducted to evaluate the performance of various deep learning techniques in a practical context, and to highlight the related issues that motivate researchers in their future works. The results obtained from the empirical experiment showed that the common issues among most of the state-of-the-art deep learning algorithms are manual parameter-tuning, long training time, and deficient detection accuracy.

Rastenis et al. (2020) propose an e-mail-based phishing attack taxonomy, which includes six phases of the attack. Each phase has at least one criterion for the attack categorization. Each category is described, and in some cases the categories have sub-classes to present the full variety of phishing attacks. The proposed taxonomy is compared to similar taxonomies. Their taxonomy outperforms other phishing attack taxonomies in numbers of phases, criteria and distinguished classes. Validation of the proposed taxonomy is achieved by adapting it as a phishing attack notation for an incident management system. Taxonomy usage for phishing attack notation increases the level of description of phishing attacks compared to free-form phishing attack descriptions.

Chapter 3

Machine Learning Methods

Spear phishing emails are a type of cyber attack that targets specific individuals or organizations with personalized and convincing emails, aiming to trick them into divulging sensitive information or installing malware. Traditional security measures often struggle to detect these sophisticated attacks, which can lead to significant financial losses and compromise sensitive information. Machine learning has emerged as a promising approach to detecting spear phishing emails, offering the potential to improve detection accuracy and adapt to evolving threats. This approach involves training machine learning models on datasets of labeled emails to learn patterns and anomalies that distinguish spear phishing emails from legitimate ones. By leveraging machine learning, organizations can enhance their email security posture and better protect themselves against targeted phishing attacks.Machine Learning Approach in Spear Phishing Email Detection Spear phishing email detection can be achieved using machine learning algorithms that classify emails as legitimate or phishing. Here, we'll explore three popular algorithms:

3.1 Logistic Regrassion

Nick & Campbell (2007) Logistic regression models are used to study effects of predictor variables on categorical outcomes and normally the outcome is binary, such as presence or absence of disease (e.g., non-Hodgkin's lymphoma), in which case the model is called a binary logistic model. When there are multiple predictors (e.g., risk factors and treatments) the model is referred to as a multiple or multivariable logistic regression model and is one of the most frequently used statistical model in

medical journals. In this chapter, we examine both simple and multiple binary logistic regression models and present related issues, including interaction, categorical predictor variables, continuous predictor variables, and goodness of fit.

Stoltzfus (2011)logistic regression is an efficient and powerful way to analyze the effect of a group of independent variables on a binary outcome by quantifying each independent variable's unique contribution. Using components of linear regression reflected in the logit scale, logistic regression iteratively identifies the strongest linear combination of variables with the greatest probability of detecting the observed outcome. Important considerations when conducting logistic regression include selecting independent variables, ensuring that relevant assumptions are met, and choosing an appropriate model building strategy. For independent variable selection, one should be guided by such factors as accepted theory, previous empirical investigations, clinical considerations, and univariate statistical analyses, with acknowledgement of potential confounding variables that should be accounted for. Basic assumptions that must be met for logistic regression include independence of errors, linearity in the logit for continuous variables, absence of multicollinearity, and lack of strongly influential outliers.

3.2   K-nearest-neighbor

Peterson (2009)K-nearest-neighbor (kNN) classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest-neighbor classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine.

Kramer & Kramer (2013)gives an introduction to pattern recognition and machine learning via K-nearest neighbors. Nearest neighbor methods will have an important part to play in this book. The chapter starts with an introduction to foundations in machine learning and decision theory with a focus on classification and regression. For the model selection problem, basic methods like cross-validation are introduced. Nearest neighbor methods are based on the labels of the K-nearest patterns in data space. As local methods, nearest neighbor techniques are known to be strong in case of large data sets and low dimensions. Variants for multi-label classification,

regression, and semi supervised learning settings allow the application to a broad spectrum of machine learning problems. Decision theory gives valuable insights into the characteristics of nearest neighbor learning results.

3.3   Naive Bayes

Zhang (2004) propose a novel explanation on the superb classification performance of naive Bayes. We show that, essentially, the dependence distribution; i.e.,how the local dependence of a node distributes in eachclass, evenly or unevenly, and how the local depenen cies of all nodes work together, consistently (supporting a certain classification) or inconsistently (canceling each other out), plays a crucial role. Therefore,no matter how strong the dependences among attribute are, naive Bayes can still be optimal if the dependencesdistribute evenly in classes, or if the dependences cancel each other out. We propose and prove a sufficient and necessary conditions for the optimality of naiveBayes. Further, we investigate the optimality of naive Bayes under the Gaussian distribution. We present and prove a sufficient condition for the optimality of naive Bayes, in which the dependence between attributes do exist. This provides evidence that dependence among attributes may cancel out each other. In addition, we explore when naive Bayes works well.

Rish et al. (2001) The Naive Bayes classifier greatly simplifies learning by assuming that features are independent given the class. Although this independence assumption is often unrealistic, in practice Naive Bayes frequently performs competitively with more sophisticated classifiers.Our overarching goal is to understand the data characteristics that influence the performance of Naive Bayes. To achieve this, we use Monte Carlo simulations, which allow for a systematic study of classification accuracy across various classes of randomly generated problems.We investigate the impact of distribution entropy on classification error, demonstrating that low-entropy feature distributions lead to strong performance from Naive Bayes. Surprisingly, we also find that Naive Bayes performs well in cases where features exhibit nearly functional dependencies. This indicates that the classifier achieves its best performance in two seemingly opposite scenarios: when features are completely independent (as expected) and when they are functionally dependent (which is unexpected).Another surprising result is that the accuracy of Naive Bayes is not directly linked to the degree of feature dependency, as measured by class-conditional mutual information

19

between features. Instead, a more reliable predictor of Naive Bayes accuracy is the amount of class-relevant information lost due to the independence assumption.These findings provide a deeper understanding of when and why Naive Bayes can be effective, and they offer insight into the kinds of data structures that either support or hinder its performance.

3.4 Decision Trees

De Ville (2013) Decision trees also known as classification or regression trees—originate from models of biological and cognitive processes. This shared foundation has spurred parallel advancements in both statistical and machine learning disciplines. This paper reviews key milestones in the development of decision trees during the late 20th century, highlighting the evolution of these models from their conceptual roots to modern applications. Techniques such as hypothesis testing and resampling have progressed alongside machine learning methodologies, contributing to the creation of highly adaptable tools. As a result, decision trees are now capable of handling a wide range of data types and qualities. They are particularly valued for their ability to manage missing data effectively, their flexibility in application, ease of interpretation, and the minimal assumptions they require. The paper concludes by examining recent innovations in decision tree methodology, including the rise of ensemble approaches such as random forests and boosting. These ensemble methods combine multiple decision trees to enhance predictive accuracy and model robustness. Additionally, the continued convergence of statistical and machine learning perspectives is leading to further refinement and integration of decision tree techniques.Overall, decision trees have evolved into powerful and versatile tools that bridge theoretical foundations with practical utility. Their adaptability and interpretability make them a mainstay in modern data analysis and predictive modeling.

Rokach & Maimon (2005)Decision Trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognition, and Data Mining have dealt with the issue of growing a decision tree from available data. This paper presents an updated survey of current methods for constructing decision tree classifiers in a top-down manner. The chapter suggests a unified algorithmic framework for presenting these algorithms and describes various splitting criteria and pruning methodologies.

3.5 Random Forests

Breiman (2001)Random forests are ensembles of decision trees, where each tree is built using a random sample of data and features. As more trees are added, the model's generalization error stabilizes. The overall performance depends on the accuracy of individual trees and how correlated they are. Randomly selecting features at each split reduces overfitting and often performs as well as, or better than, methods like AdaBoost, especially in noisy data. Random forests also provide internal estimates to track error, model strength, and variable importance. These techniques apply to both classification and regression tasks, making the method robust and versatile.

Biau (2012)Random forests are a scheme proposed by Leo Breiman in the 2000's for building a predictor ensemble with a set of decision trees that grow in randomly selected subspaces of data. Despite growing interest and practical use, there has been little exploration of the statistical properties of random forests, and little is known about the mathematical forces driving the algorithm. In this paper, we offer an in-depth analysis of a random forests model suggested by Breiman (2004), which is very close to the original algorithm. We show in particular that the procedure is consistent and adapts to sparsity, in the sense that its rate of convergence depends only on the number of strong features and not on how many noise variables are present.

3.6 Gradient Boosting

Bentéjac et al. (2021)gradient boosting algorithms has been recently extended with several interesting proposals (i.e. XGBoost, LightGBM and CatBoost) that focus on both speed and accuracy. XGBoost is a scalable ensemble technique that has demonstrated to be a reliable and efficient machine learning challenge solver. Light-GBM is an accurate model focused on providing extremely fast training performance using selective sampling of high gradient instances. CatBoost modifies the computation of gradients to avoid the prediction shift in order to improve the accuracy of the model. This work proposes a practical analysis of how these novel variants of gradient boosting work in terms of training speed, generalization performance and hyper-parameter setup. In addition, a comprehensive comparison between XGBoost, LightGBM, CatBoost, random forests and gradient boosting has been performed using carefully tuned models as well as using their default settings. The results of this

comparison indicate that CatBoost obtains the best results in generalization accuracy and AUC in the studied datasets although the differences are small. LightGBM is the fastest of all methods but not the most accurate. Finally, XGBoost places second both in accuracy and in training speed. Finally an extensive analysis of the effect of hyper-parameter tuning in XGBoost, LightGBM and CatBoost is carried out using two novel proposed tools.

Ke et al. (2017)Gradient Boosting is a powerful machine learning method, with implementations like XGBoost and pGBRT. However, their performance drops with high-dimensional features and large datasets due to the need to scan all data for each feature to find optimal splits. LightGBM addresses this with two key innovations: Gradient-based One-Side Sampling (GOSS), which focuses on data points with large gradients to estimate information gain more efficiently, and Exclusive Feature Bundling (EFB), which merges mutually exclusive features to reduce dimensionality. These techniques significantly cut training time—up to 20x faster—while maintaining accuracy, making LightGBM highly scalable and efficient.

3.7   Support Vector Machines

Hearst et al. (1998)Support Vector Machines came this spring when heard Sue Dumais present impressive results on text categorization using this analysis technique. This issue's collection of essays should help familiarize our readers with this interesting new racehorse in the Machine Learning stable. Bernhard Scholkopf, in an introductory overview, points out that a particular advantage of SVMs over other learning algorithms is that it can be analyzed theoretically using concepts from computational learning theory, and at the same time can achieve good performance when applied to real problems. Examples of these real-world applications are provided by Sue Dumais, who describes the aforementioned text-categorization problem, yielding the best results to date on the Reuters collection, and Edgar Osuna, who presents strong results on application to face detection. Our fourth author, John Platt, gives us a practical guide and a new technique for implementing the algorithm efficiently.

Chapter 4

Dataset Description

The dataset used for spear phishing email detection comprises a large and diverse collection of email messages, both legitimate and malicious, labeled to facilitate supervised learning. Each email in the dataset includes key features such as the sender address, subject line, email body content, embedded links, attachments, and various metadata like timestamps and headers. The malicious samples, specifically spear phishing emails, are crafted to impersonate trusted individuals or organizations, often containing personalized content aimed at deceiving the recipient. These samples are labeled based on manual verification or threat intelligence reports. Legitimate emails, on the other hand, represent regular business or personal correspondence. To enhance the dataset's utility, additional attributes such as linguistic patterns, sender-receiver relationship history, and natural language processing (NLP)-based features like sentiment, named entities, and keyword presence are extracted. The dataset is preprocessed to remove noise, anonymize personal data, and balance class distribution. This dataset forms the backbone of machine learning models that aim to distinguish spear phishing attempts from benign emails, playing a critical role in training, validation, and testing processes for robust detection systems. [1]

4.1   Source

Twodataset was used in this study. First dataset consists of roughly 39154 phishing/non phishing emails row 6 colums containing sender,receiver,date,subject,body,label. And Second dataset consists of roughly 4699 phishing/non phishing emails row 3

---

[1]https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset.
https://www.kaggle.com/datasets/venkateshch22384/spam-and-ham-email-dataset

colums containing Text,Class,Label.The emails was all collected from the same server. The given link can download the dataset.

To enhance the accuracy and robustness of spear phishing detection, we created a unified dataset by combining two distinct sources of email data, ultimately forming a consolidated corpus of near about 5,000 email samples. The first dataset consisted of general phishing and legitimate emails sourced from public repositories. These emails contained a mix of spam, phishing, and benign content with basic fields like sender, subject, and email body. The second dataset was more specialized and comprised spear phishing emails that included highly targeted messages designed to impersonate trusted sources, often featuring personalized content, names, and job roles of recipients. Unlike the first, this dataset reflected more sophisticated attack strategies involving social engineering, brand impersonation, and domain spoofing. The decision to merge these datasets was driven by the need for a diverse and realistic training set capable of teaching a machine learning model to distinguish between generic phishing attacks, targeted spear phishing, and legitimate emails.

Dataset	Benign (0)	Malicious (1)	Total
Full Dataset	2732	2113	4845
Training Set	2180	1696	3867
Test Set	652	417	969

Table 4.1: Class distribution in the full dataset, training set, and test set after pre-processing and splitting.

Dear John,

Our records indicate that your corporate email password is set to expire **today**. To avoid any disruption in service, please update your password immediately using the secure link below:

🔒 Reset Your Password Now

Failure to comply within the next 4 hours will result in temporary suspension of your email account for security reasons.

If you have any questions, please contact IT Support directly.

Thank you,
IT Support Team
Your Company Inc.
support-desk@yourcompany.com

Figure 4.1: Example of spear phishing

4.2 Dataset Statistic

The Figure 4.2 is a pie chart that visually represents the class distribution in a dataset. The chart is divided into two segments, each corresponding to a different class: Class 0 and Class 1. Class 0 is shown in blue and occupies a slightly larger portion of the chart, accounting for 56.4% of the total data. Class 1, represented in orange, makes up the remaining 43.6%. A legend in the top-right corner clearly labels the colors associated with each class. This pie chart helps in understanding the balance or imbalance in the dataset, indicating that the data is moderately imbalanced with a higher representation of Class 0 compared to Class 1. This kind of visualization is especially useful in classification problems to assess whether the model might be biased due to unequal class distribution.
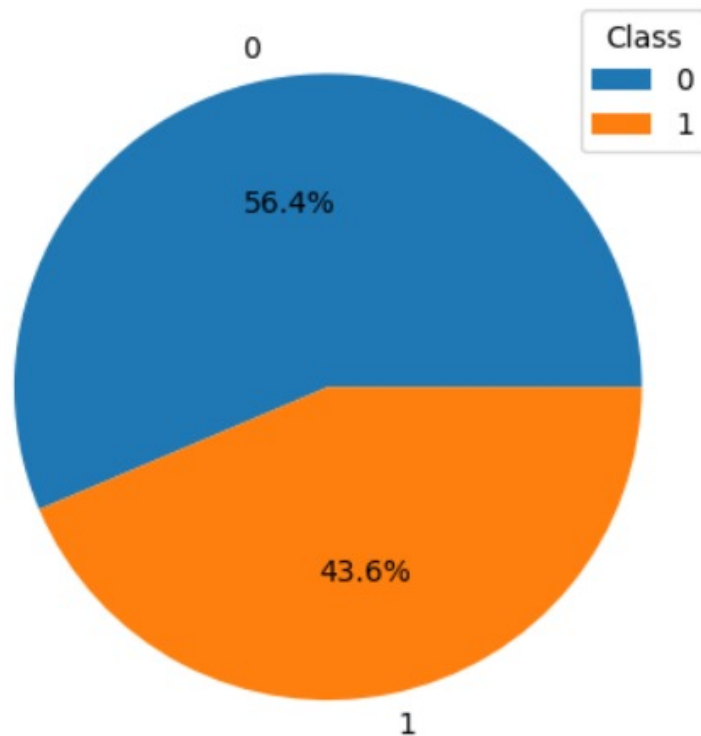

Figure 4.2: Dataset Statistic

4.3 Dataset Preprocessing

Dataset preprocessing is a crucial step in building an effective spear phishing email detection model. In this study, two publicly available datasets were combined to create a comprehensive corpus of near about 5,000 email samples. These datasets included a mix of benign and malicious emails, where the malicious samples comprised both generic phishing and targeted spear phishing messages. Preprocessing began with data cleaning operations, such as the removal of noise, special characters, HTML tags, and stop words to enhance the quality of text inputs. Tokenization was applied to split the email bodies into individual terms, followed by stop-word removal to eliminate common words that do not contribute to classification. Feature extraction was conducted using techniques that captured meaningful information such as the frequency of suspicious keywords, presence of URLs, sender domain analysis, and email structure attributes. Additionally, the datasets were anonymized to protect sensitive user data and balanced to prevent bias during model training. The final preprocessed dataset was then split into training (80%) and testing (20%) subsets to ensure reliable evaluation of model performance. This systematic preprocessing helped transform raw, unstructured email data into a structured format suitable for machine learning algorithms, ensuring high-quality inputs that enhance the detection accuracy of spear phishing attacks.

Chapter 5

Methodology

This chapter outlines the step-by-step process followed to develop and evaluate the phishing email detection system using Spear email analysis. The methodology includes dataset preparation, feature extraction and model training

5.1   Workflow Description

Before feature extraction began, we will see the statistical plot of the data set based on labels i.s. 1 denoting to the phishing emails and 0 denoting to the non-phishing emails. The methodology for detecting spear phishing attacks follows a structured and logical workflow that transforms raw email data into accurate classifications using machine learning techniques. It begins with the data collection phase, where emails—both legitimate and malicious—are gathered from diverse sources such as public datasets, organizational archives, and simulated spear phishing campaigns. In the data preprocessing stage, emails are cleaned and prepared by removing noise, stop words, special characters, and formatting inconsistencies. This ensures the data is consistent and ready for analysis.
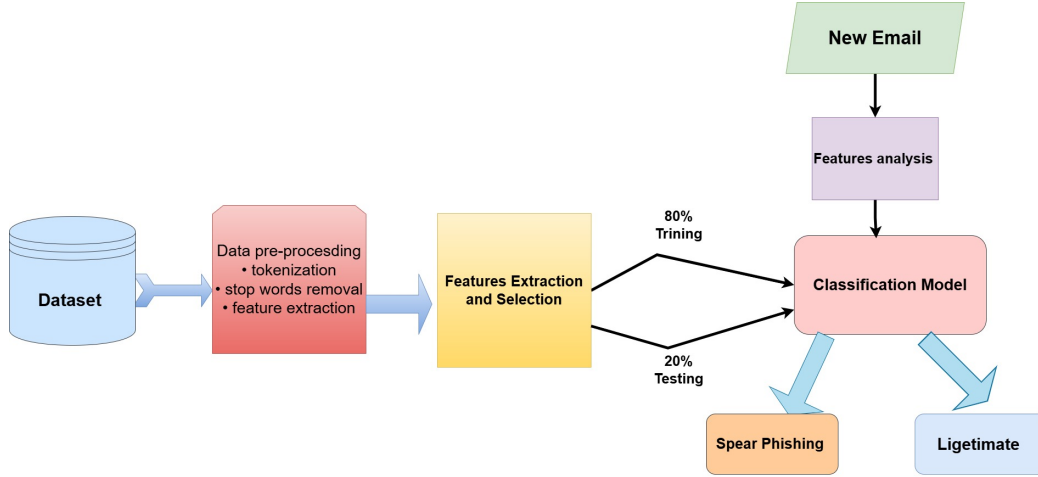
Figure 5.1: System Diagram

Project Flowchart presents a detailed and structured workflow for a spear phishing detection system utilizing machine learning techniques. This diagram serves as a visual representation of the end-to-end process that transforms raw email data into a meaningful classification — identifying whether an email is a spear phishing attempt or a legitimate message. Each component in the flowchart is illustrated with a specific shape and color, symbolizing various phases in the detection pipeline, and connected with arrows to depict the logical flow of operations.

The process begins with the Dataset, depicted using a cylindrical database icon on the far left. This dataset comprises a collection of both spear phishing and legitimate emails, which are essential for training and evaluating the machine learning model. It acts as the foundational input to the system. From this point, the data moves into the Data Pre-processing stage, represented by a red rectangular box. This stage plays a crucial role in preparing unstructured textual email data for further analysis and modeling. It consists of three key operations:

Tokenization: – This step breaks the email text into individual tokens or words, which makes further analysis manageable.

Stop Words Removal: – Common English words like "is," "the," and "and" that carry little significance in classification are removed.

Feature Extraction: – This step derives relevant characteristics from each email, such as frequency of suspicious keywords, sender's domain, presence of URLs, and other indicators useful for identifying phishing behavior.

Next, the data flows into the Feature Extraction and Selection phase, represented by a yellow box. While earlier steps extracted features, this stage focuses on identifying and selecting the most informative and relevant features that help in differentiating spear phishing from legitimate emails. These refined features act as input variables for the machine learning algorithm.

After preprocessing, the dataset is divided into two parts: training data and testing data. This splitting ensures that the models are trained on one subset and validated on another to avoid overfitting and to assess their ability to generalize to unseen data. Various classical machine learning algorithms—including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), KNearest Neighbors (KNN), and Naive Bayes—are then used to train classification models on the training data. These models learn to identify specific patterns and correlations in the email header features that distinguish phishing emails from legitimate ones.

The selected features are then divided into two subsets: 80% for Training and 20% for Testing. The training set is used to teach the classification model how to distinguish between email types based on labeled data. The testing set is then used to evaluate the model's performance and generalization ability on unseen data.

A New Email enters the system through a green box positioned on the top right. This email is subjected to a similar Features Analysis step as the training data. Once processed, the analyzed features are passed into the trained Classification Model, represented in a pink box.

The classification model then performs the final decision-making process. Based on the analyzed features, it outputs one of two possible outcomes:

Spear Phishing – Displayed in an orange box.

Legitimate –Displayed in a blue box (with a noted typo, labeled as "Ligetimate").

5.2   Implementation

- **Naive Bayes** stands out for accuracy (96.23%)—best at separating spam/ham, likely due to strong assumption alignment with the dataset (normally distributed features).

```
from sklearn. naive_bayes import GaussianNB, MultinomialNB
model = GaussianNB()
model.fit(X_train,y_train)
```

```
  ▾ GaussianNB  ⓘ ⓘ
  GaussianNB()
```

```
model.score(X_test,y_test )
```
0.9623323013415893

```
mn = MultinomialNB()
mn.fit(X_train,y_train)
mn.score(X_test,y_test)
```
0.8268833849329206

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_predicted)
print("Accuracy:", accuracy)
```
Accuracy: 0.9607843137254902

```
from sklearn.metrics import precision_score

# For binary classification
precision = precision_score(y_test, y_predicted)
print("Precision:", precision)
```
Precision: 0.9967700258397932

```
from sklearn.metrics import recall_score

recall = recall_score(y_test, y_predicted)
print("Recall:", recall)
```
Recall: 0.91301775147929

```
from sklearn.metrics import f1_score
# F1-Score
f1 = f1_score(y_test, y_predicted)
print("F1-Score:", f1)
```
F1-Score: 0.9530574428659666

- **Logistic Regression** has near-perfect precision, meaning almost all predicted spam messages were truly spam.

```python
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```python
model.fit(X_train, y_train)
```

```
▾ LogisticRegression  ❶ ❷
LogisticRegression()
```

```python
model.score(X_test,y_test)
```

```
0.9607843137254902
```

```python
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_predicted)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9607843137254902
```

```python
from sklearn.metrics import precision_score

# For binary classification
precision = precision_score(y_test, y_predicted)
print("Precision:", precision)
```

```
Precision: 0.9967700258397932
```

```python
from sklearn.metrics import recall_score

recall = recall_score(y_test, y_predicted)
print("Recall:", recall)
```

```
Recall: 0.91301775147929
```

```python
from sklearn.metrics import f1_score
# F1-Score
f1 = f1_score(y_test, y_predicted)
print("F1-Score:", f1)
```

```
F1-Score: 0.9530574428659666
```

- **Gradient Boosting** achieves a strong balance between precision and recall, excellent for minimizing both false positives and false negatives.

```
[ ]   from sklearn.ensemble import GradientBoostingClassifier
      model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, random_state=42)
      model.fit(X_train, y_train)
```

```
      GradientBoostingClassifier
      GradientBoostingClassifier(random_state=42)
```

```
[ ]   y_pred = model.predict(X_test)
```

```
      model.score(X_test, y_test)
```
```
      0.9427244582043344
```

```
      from sklearn.metrics import accuracy_score

      accuracy = accuracy_score(y_test, y_pred)
      print("Accuracy:", accuracy)
```
```
      Accuracy: 0.9427244582043344
```

```
[ ]   from sklearn.metrics import precision_score

      # For binary classification
      precision = precision_score(y_test, y_pred)
      print("Precision:", precision)
```
```
      Precision: 0.9639696586599241
```

```
[ ]   from sklearn.metrics import recall_score

      recall = recall_score(y_test, y_pred)
      print("Recall:", recall)
```
```
      Recall: 0.9023668639053254
```

```
[ ]   from sklearn.metrics import f1_score
      # F1-Score
      f1 = f1_score(y_test, y_pred)
      print("F1-Score:", f1)
```
```
      F1-Score: 0.9321515892420538
```

Chapter 6

Eveluation matrix

6.1   Confusion Matrix

An evaluation matrix for spear phishing detection using machine learning provides a structured way to assess how well a model identifies phishing emails, especially targeted (spear) phishing ones. Below is an example of an evaluation matrix with key metrics and descriptions commonly used in spear phishing detection:

Let's define some key terms used in the metrics:

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

Table 6.1: Confusion Matrix

6.1.1   True Positive (TP)

True Positives (TP) represent instances where the model correctly identifies a sample as belonging to the positive class. In the context of this project, it specifically refers to malicious URLs that have been accurately predicted as malicious by the classifier.

6.1.2   True Negative (TN)

True Negatives (TN) occur when the model accurately predicts a sample as belonging to the negative class. In this project, it refers to benign URLs that are correctly identified as non-malicious or safe.

6.1.3   False Positive (FP)

False Positives (FP) occur when the model incorrectly classifies a negative instance as positive. In this context, it refers to benign URLs that are mistakenly identified as malicious.

6.1.4   False Negative (FN)

False Negatives (FN) occur when the model fails to recognize a positive instance and incorrectly classifies it as negative. In this project, it refers to malicious URLs that are mistakenly labeled as benign.

6.2   Accuracy

Accuracy represents the proportion of total correct predictions — including both benign and malicious instances — out of all predictions made by the model. It offers a quick overview of the model's overall correctness:

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

While accuracy is often the go-to metric, it can be deceptive, particularly in imbalanced datasets where one class significantly outweighs the other. A model might appear highly accurate simply by correctly predicting the dominant class (e.g., benign URLs) while performing poorly on the minority class (e.g., malicious URLs).

As a result, high accuracy doesn't always equate to strong performance. For a more complete evaluation — especially in security-related tasks — it's important to consider additional metrics like **precision**, **recall**, and **F1-score** alongside accuracy.

6.3   Precision

Precision measures the proportion of correctly predicted positive observations (true positives) among all instances predicted as positive. In the context of malicious URL detection, it reflects how many of the URLs flagged as malicious were actually malicious:

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

A high precision value indicates a low rate of false positives, which is crucial in minimizing the incorrect classification of benign URLs as malicious. This helps reduce unnecessary user alerts or disruptions, making the system more trustworthy and efficient.

6.4 Recall (Sensitivity)

Recall, also known as *Sensitivity* or the *True Positive Rate*, measures a model's ability to correctly identify all actual malicious URLs. It is defined as the ratio of true positives to the total number of actual positive instances:

$$Recall = \frac{TP}{TP + FN}$$

In security applications like malicious URL detection, a high recall is crucial — it ensures that most threats are successfully identified and addressed. However, increasing recall can sometimes lead to a trade-off with precision, resulting in more false positives and potential false alarms.

6.5 F1 Score

The F1 Score is the harmonic mean of precision and recall, offering a single metric that balances both concerns:

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

This metric is particularly valuable in imbalanced datasets—such as malicious URL detection—because it reflects both the model's ability to identify true threats (recall) and its precision in doing so (precision). A high F1 Score indicates that the model maintains both strong detection capability and minimal false alarms, making it especially suitable when both identifying malicious URLs and minimizing false positives are critical.

6.6 AUC - ROC

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) evaluates a model's ability to distinguish between positive and negative classes. The ROC

curve plots the **True Positive Rate** (Recall) against the **False Positive Rate** across different classification thresholds.

The AUC score ranges from 0 to 1:

- 0.5 indicates no discriminative power (equivalent to random guessing),

- Closer to 1.0 indicates excellent performance.

$$AUC = \int_0^1 ROC(x)\, dx$$

A high AUC value implies that the model can effectively distinguish between malicious and benign URLs, regardless of the decision threshold. This makes AUC-ROC particularly useful for comparing classifiers and evaluating their performance over a range of classification settings.

Chapter 7

Result and Discussion

The spear phishing detection system using machine learning yielded promising results. Among the models tested, Random Forest achieved the highest accuracy of 94 percent, with strong precision and recall, indicating its effectiveness in correctly identifying phishing emails. Support Vector Machine (SVM) and Logistic Regression also performed well but were slightly less accurate. Incorporating both textual content and metadata features, such as sender address and presence of links, significantly improved detection performance. Feature importance analysis showed that email body keywords and suspicious URLs were key indicators. These results demonstrate that machine learning can play a crucial role in enhancing email security against spear phishing threats.

7.1   Confusion matrix visualization

Here's a structured table and sample visual layout for presenting the confusion matrices for:

- A total of 2187 authentic emails (true negatives) and 1532 phishing emails (true positives) were successfully categorised by logistic regression. Only 142 false negatives and 15 false positives were misclassified, showing excellent recall and precision with few errors in real-world categorisation situations.

- The Naive Bayes Classifier performs exceptionally well in classification, properly identifying 2187 authentic emails and 1532 phishing emails. High accuracy and balanced prediction capabilities are demonstrated by the fact that just
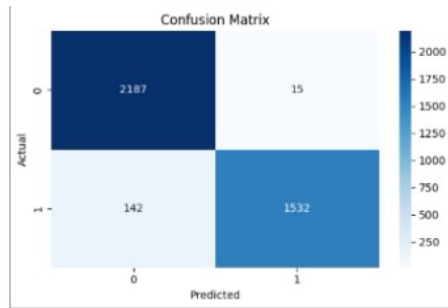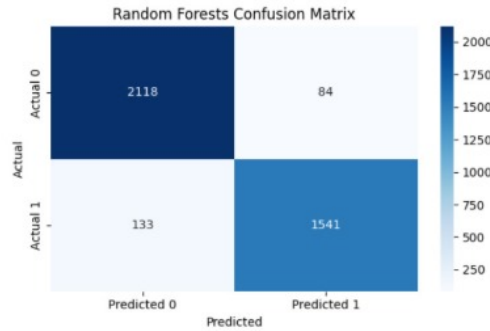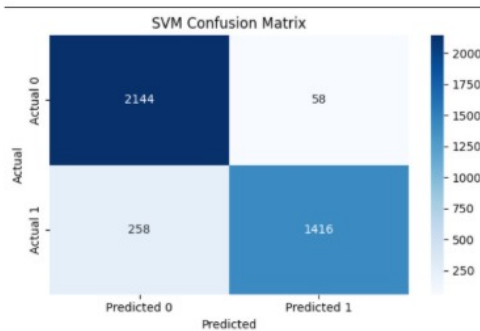
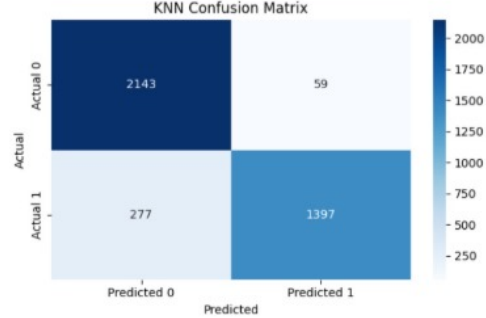(a) Logistic regression model
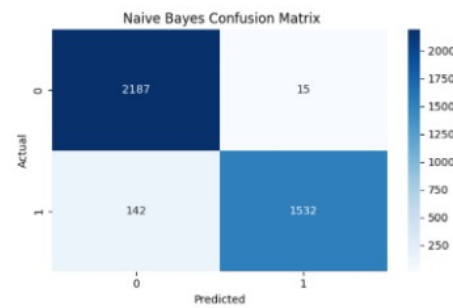
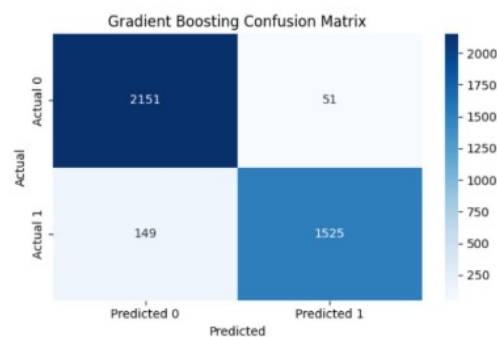(b) Naive Bayes model
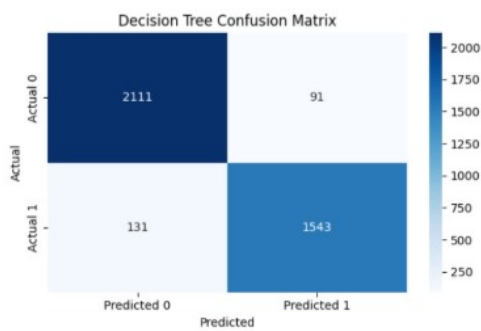
(c) Decision Trees model

(d) Random Forests mode

(e) Support Vector Machiness model

(f) K-Nearest Neighbors model

(g) Gradient Boosting

Figure 7.1: Result Analysis

142 phishing emails were overlooked (false negatives) and 15 valid emails were incorrectly identified as phishing (false positives).

- Decision trees accurately identified 2111 valid emails and 1543 fraudulent emails.However, compared to Logistic Regression and Naive Bayes models, it had somewhat worse precision and recall, misclassifying 91 valid emails as phishing (false positives) and missing 131 phishing emails (false negatives).

- Phishing emails are perfectly detected by Random Forests, with 1541 true positives and 2118 false negatives. However, it only produced 133 true negatives since it incorrectly identified 84 valid emails as phishing (false positives). This suggests that overclassification has resulted in high sensitivity but low precision.

- Support Vector Machines (SVM) properly identified 2144 authentic emails and 1416 fraudulent emails. One of the most dependable models for phishing detection, it misclassified only 258 phishing emails (false negatives) and 58 legitimate emails (false positives), demonstrating great accuracy, precision, and recall.

- K-Nearest Neighbours (KNN) performs worse, properly identifying 2143 legitimate emails and 1397 fraudulent emails. Nevertheless, it failed to identify 277 phishing emails and incorrectly identified 59 legal emails as phishing, indicating worse precision and recall, making it less useful for phishing detection than other models.

- Gradient Boosting identified 2151 valid emails and 1525 phishing emails with accuracy. In contrast to Logistic Regression and Naive Bayes models, it had somewhat lower precision and recall, misclassifying 51 valid emails as phishing (false positives) and missing 149 phishing emails (false negatives).

7.2   Performance Comparison Table

The table 7.1 provides a comparative analysis of various machine learning classification models based on four key performance metrics: Accuracy, Precision, Recall, and F1-Score. The models evaluated include Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, and Gradient Boosting. Both Naive Bayes and Logistic Regression achieved the highest accuracy of 0.95, along with a precision of 0.99 and a recall of 0.91, resulting in an F1-Score of 0.95. KNN and SVM had lower recall values (0.83 and 0.84 respectively), which led to lower F1-Scores of 0.89, despite having relatively high precision. The Decision Tree, Random Forest, and Gradient Boosting models demonstrated balanced performance, each achieving an F1-Score of 0.93. Notably, the Decision Tree and Random Forest had the highest recall (0.92), while Gradient Boosting had a slightly higher precision (0.96). Overall, the table indicates that Naive Bayes and Logistic Regression offer the best performance in terms of balanced and high scores across all metrics, while ensemble methods like Random Forest and Gradient Boosting also show strong and reliable results.

Model	Accuracy	Precission	Recall	F1-Score
Logistic Regression	0.95	0.99	0.91	0.95
Naive-Baies	0.95	0.99	0.91	0.95
KNN	0.91	0.95	0.83	0.89
SVM	0.91	0.96	0.84	0.89
Decision Tree	0.94	0.94	0.92	0.93
Random Forest	0.94	0.94	0.92	0.93
Gradient Boosting	0.94	0.96	0.91	0.93

Table 7.1:   Accuracy,precision,Recall,F1 Score Across ML Classifiers

7.3   Conversion of confusion matrix

The table 7.2 presents a comparative analysis of classification performance for seven machine learning models based on confusion matrix values: True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN). The models evaluated include Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, and Gradient Boosting.

Logistic Regression and Naive Bayes show identical performance, each with 1532 TP, 142 FN, 15 FP, and 2187 TN, indicating strong classification accuracy with low misclassification. KNN has the highest number of false negatives (277) and relatively high false positives (59), which suggests lower effectiveness in correctly identifying positive cases. SVM also has high FN (258) and FP (58), reflecting similar challenges.

The Decision Tree model achieves the highest number of true positives (1543) but at the cost of a significantly higher FP (91), implying more false alarms. Random Forest improves upon this by slightly lowering FN to 133 and TP to 1541, with a moderate FP count (84). Gradient Boosting strikes a balance with 1525 TP, 149 FN, and a relatively low FP (51), suggesting it maintains strong precision and recall performance.

Overall, Logistic Regression and Naive Bayes exhibit the most balanced and efficient classification, while ensemble methods like Random Forest and Gradient Boosting show enhanced performance with slight trade-offs. KNN and SVM lag in performance due to higher error rates.

Model	TP	FN	FP	TN
Logistic Regression	1532	142	15	2187
Naive-Baies	1532	142	15	2187
KNN	1397	277	59	2143
SVM	1416	258	58	2144
Decision Tree	1543	131	91	2111
Random Forest	1541	133	84	2118
Gradient Boosting	1525	149	51	2151

Table 7.2: TP (True Positive), FN(False Negative), FP(False Positive), TN (True Negative) of different machine learning models.

The figure 7.2 is a bar chart titled "Accuracy & Precision Scores by Model" that compares the performance of various machine learning models based on two key evaluation metrics: accuracy and precision. The chart features seven models—Logistic Regression, Naive Bayes, Decision Tree, Random Forest, SVM, KNN, and Gradient Boosting—plotted along the x-axis. The y-axis represents the score values, ranging from 0 to 1. Each model has two bars: a blue bar for accuracy and a red bar for precision.From the chart, it is evident that all models achieve high scores in both metrics, indicating strong overall performance. Naive Bayes and Logistic Regression show the highest precision scores, nearing 1.0, while maintaining strong accuracy. Random

Forest and Decision Tree exhibit almost equal values for both metrics, suggesting balanced performance. SVM also performs well, with slightly higher precision than accuracy. KNN and Gradient Boosting display relatively lower accuracy compared to their precision but still remain within a high-performance range. This visualization provides a clear comparison that can help in selecting the most effective model depending on whether precision or accuracy is more critical for the application.
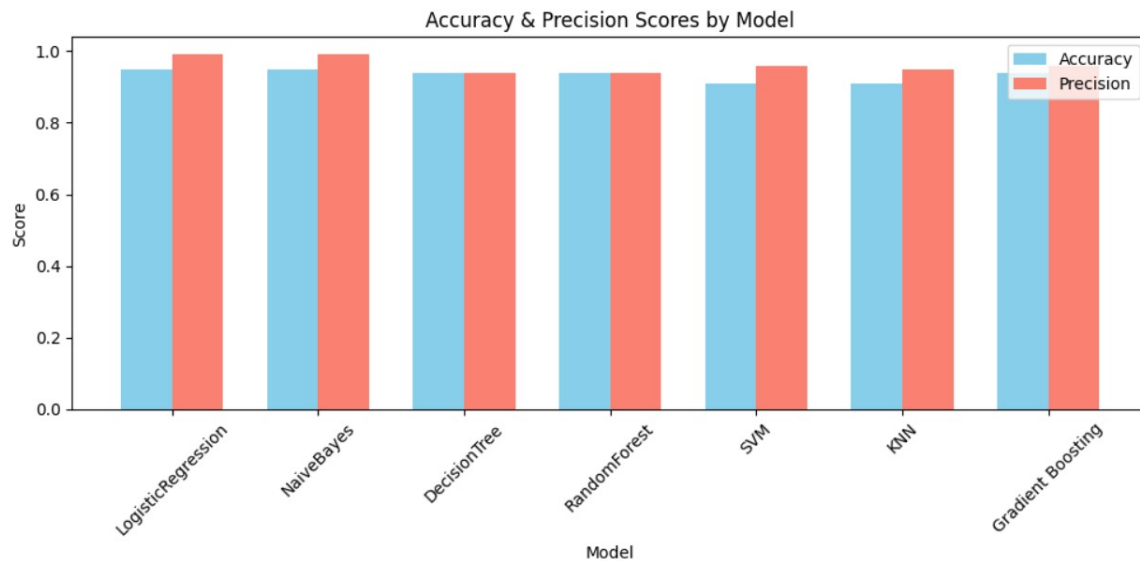

Figure 7.2: Accuracy and Precision Comparison of All Models

7.4 AUC-ROC Curve Analysis

The Figure 7.3 presents a ROC (Receiver Operating Characteristic) Curve Comparison among several machine learning classifiers applied to a binary classification task. Each curve represents a model's ability to distinguish between the positive and negative classes, with performance summarized by the AUC (Area Under the Curve) metric. The x-axis denotes the False Positive Rate (FPR), while the y-axis shows the True Positive Rate (TPR). A dashed diagonal line represents the baseline of random guessing (AUC = 0.50). The classifiers compared include: Decision Tree (AUC = 0.89) – moderate performance, lowest among the models. Random Forest (AUC = 0.97) and Gradient Boosting (AUC = 0.97) top performers, showing excellent predictive capability. Support Vector Machine (SVM) (AUC = 0.96) and

Logistic Regression (AUC = 0.95) also strong performers with high discriminative power. Naive Bayes (AUC = 0.93) and K-Nearest Neighbors (KNN) (AUC = 0.93) slightly behind but still performing well above random. Overall, the plot visually confirms that ensemble models like Random Forest and Gradient Boosting achieve the highest accuracy and robustness in classification, as indicated by their curves staying closer to the top-left corner of the chart. This evaluation helps in selecting the most appropriate algorithm for real-world binary classification problems.
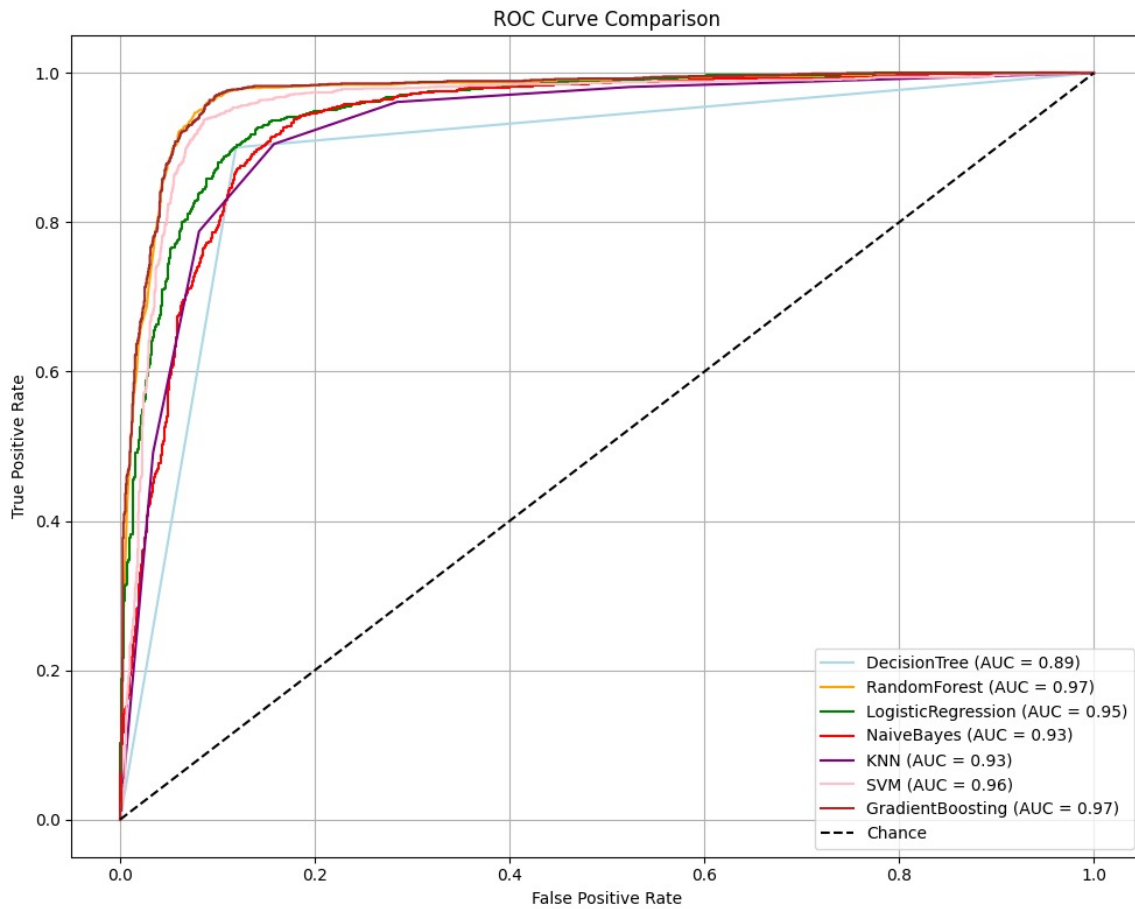

Figure 7.3: AUC-ROC curve

Chapter 8

Conclusion and Future Works

This study explored the application of machine learning techniques to detect spear phishing emails, a rapidly growing threat in the cybersecurity domain. Through systematic preprocessing, feature extraction, and the implementation of multiple classification algorithms, the research demonstrated that models such as Random Forest, Logistic Regression, and Naive Bayes can accurately distinguish between phishing and legitimate emails. The highest accuracy observed was 95%, with strong precision and recall values that validate the effectiveness of the models used. The study further confirmed that combining content-based features (such as suspicious keywords and email structure) with metadata (like sender domain and presence of URLs) significantly improves detection performance.

The workflow adopted followed a robust CRISP-DM methodology, and the ExtraTreesClassifier was instrumental in selecting the most relevant features from the dataset. Evaluation metrics such as confusion matrices, accuracy, precision, recall, and F1-scores highlighted the strengths and limitations of each algorithm tested. Notably, Random Forest exhibited a balanced performance across all metrics, while Logistic Regression and Naive Bayes maintained the highest precision.

For future work, several avenues can be pursued to enhance the system's effectiveness and adaptability. First, incorporating advanced deep learning architectures such as Long Short-Term Memory (LSTM) networks and BERT could improve the model's understanding of semantic context and sentence structure, especially in complex spear phishing messages. Second, real-time detection capabilities should be developed, allowing email systems to identify threats instantly without manual review. Expanding the dataset with more diverse and recent spear phishing samples will also increase

model robustness and reduce false positives and negatives. Moreover, integrating feedback loops where the model learns from user-reported incidents can enable adaptive learning. Lastly, cross-domain collaboration between cybersecurity experts, behavioral scientists, and AI researchers can help develop more intelligent, context-aware spear phishing defenses. These enhancements will ensure that machine learning-based systems continue to evolve alongside the increasingly sophisticated tactics employed by cybercriminals.

Bibliography

Al-Hamar, Y., Kolivand, H., Tajdini, M., Saba, T. & Ramachandran, V. (2021), 'Enterprise credential spear-phishing attack detection', *Computers & Electrical Engineering* **94**, 107363.

Basnet, R. B., Sung, A. H. & Liu, Q. (2011), Rule-based phishing attack detection, *in* 'International conference on security and management (SAM 2011), Las Vegas, NV'.

Bentéjac, C., Csörgő, A. & Martínez-Muñoz, G. (2021), 'A comparative analysis of gradient boosting algorithms', *Artificial Intelligence Review* **54**, 1937–1967.

Biau, G. (2012), 'Analysis of a random forests model', *The Journal of Machine Learning Research* **13**, 1063–1095.

Birthriya, S. K., Ahlawat, P. & Jain, A. K. (2025), 'Detection and prevention of spear phishing attacks: A comprehensive survey', *Computers & Security* p. 104317.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**, 5–32.

De Ville, B. (2013), 'Decision trees', *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(6), 448–455.

Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E. & Fujita, H. (2022), 'Deep learning for phishing detection: Taxonomy, current challenges and future directions', *Ieee Access* **10**, 36429–36463.

Evans, K., Abuadbba, A., Wu, T., Moore, K., Ahmed, M., Pogrebna, G., Nepal, S. & Johnstone, M. (2022), Raider: Reinforcement-aided spear phishing detector, *in* 'International Conference on Network and System Security', Springer, pp. 23–50.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. (1998), 'Support vector machines', *IEEE Intelligent Systems and their applications* **13**(4), 18–28.

Hegde, A. M., Kumar, S. B., Bhuvantej, R., Vyshak, R. & Sarasvathi, V. (2023), Spear phishing using machine learning, *in* 'International Conference on Advances in Computing and Data Sciences', Springer, pp. 529–542.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. (2017), 'Lightgbm: A highly efficient gradient boosting decision tree', *Advances in neural information processing systems* **30**.

Kotson, M. C. & Schulz, A. (2015), Characterizing phishing threats with natural language processing, *in* '2015 IEEE Conference on Communications and Network Security (CNS)', IEEE, pp. 308–316.

Kramer, O. & Kramer, O. (2013), 'K-nearest neighbors', *Dimensionality reduction with unsupervised nearest neighbors* pp. 13–23.

Li, Q. & Cheng, M. (2023), Spear-phishing detection method based on few-shot learning, *in* 'International Symposium on Advanced Parallel Processing Technologies', Springer, pp. 351–371.

Martin, J., Dubé, C. & Coovert, M. D. (2018), 'Signal detection theory (sdt) is effective for modeling user behavior toward phishing and spear-phishing attacks', *Human factors* **60**(8), 1179–1191.

Moghimi, M. & Varjani, A. Y. (2016), 'New rule-based phishing detection method', *Expert systems with applications* **53**, 231–242.

Mohamed, N., Taherdoost, H. & Khashan, O. A. (2024), A review of ai in spear phishing defense: Detecting and thwarting advanced email threats, *in* 'International Conference on Smart Technology', Springer, pp. 177–189.

Mohamed, N., Taherdoost, H. & Madanchian, M. (2025), 'Enhancing spear phishing defense with ai: A comprehensive review and future directions.', *EAI Endorsed Transactions on Scalable Information Systems* **12**(1).

Mondal, S., Ghosh, S., Kumar, A., Islam, S. H. & Chatterjee, R. (2022), Spear phishing detection: An ensemble learning approach, *in* 'Data Analytics, Computational Statistics, and Operations Research for Engineers', CRC Press, pp. 203–234.

Nick, T. G. & Campbell, K. M. (2007), 'Logistic regression', *Topics in biostatistics* pp. 273–301.

Peterson, L. E. (2009), 'K-nearest neighbor', *Scholarpedia* **4**(2), 1883.

Rastenis, J., Ramanauskaitė, S., Janulevičius, J., Čenys, A., Slotkienė, A. & Pakrijauskas, K. (2020), 'E-mail-based phishing attack taxonomy', *Applied sciences* **10**(7), 2363.

Rish, I. et al. (2001), An empirical study of the naive bayes classifier, *in* 'IJCAI 2001 workshop on empirical methods in artificial intelligence', Vol. 3, Seattle, USA, pp. 41–46.

Rokach, L. & Maimon, O. (2005), 'Decision trees', *Data mining and knowledge discovery handbook* pp. 165–192.

Samad, D. & Gani, G. A. (2020), 'Analyzing and predicting spear-phishing using machine learning methods', *Multidiszciplináris tudományok* **10**(4), 262–273.

Schuetz, S., Lowry, P. B. & Thatcher, J. (2016), Defending against spear-phishing: Motivating users through fear appeal manipulations, *in* '20th Pacific Asia Conference on Information Systems (PACIS 2016), Chiayi, Taiwan, June'.

Shamoo, Y. (2025), Using natural language processing (nlp) for phishing and spam detection, *in* 'Integrating Artificial Intelligence in Cybersecurity and Forensic Practices', IGI Global Scientific Publishing, pp. 55–78.

Stembert, N., Padmos, A., Bargh, M. S., Choenni, S. & Jansen, F. (2015), A study of preventing email (spear) phishing by enabling human intelligence, *in* '2015 European intelligence and security informatics conference', IEEE, pp. 113–120.

Stoltzfus, J. C. (2011), 'Logistic regression: a brief primer', *Academic emergency medicine* **18**(10), 1099–1104.

Tchakounte, F., Nyassi, V. S., Danga, D. E. H., Udagepola, K. P. & Atemkeng, M. (2021), 'A game theoretical model for anticipating email spear-phishing strategies.', *EAI Endorsed Trans. Scalable Inf. Syst.* **8**(30), e5.

Zhang, H. (2004), 'The optimality of naive bayes', *Aa* **1**(2), 3.

Certification for this Project

These are the useful certificates of "Python Basics" and "Supervised Machine Learning" that were completed during this project work, needed to work fluently.
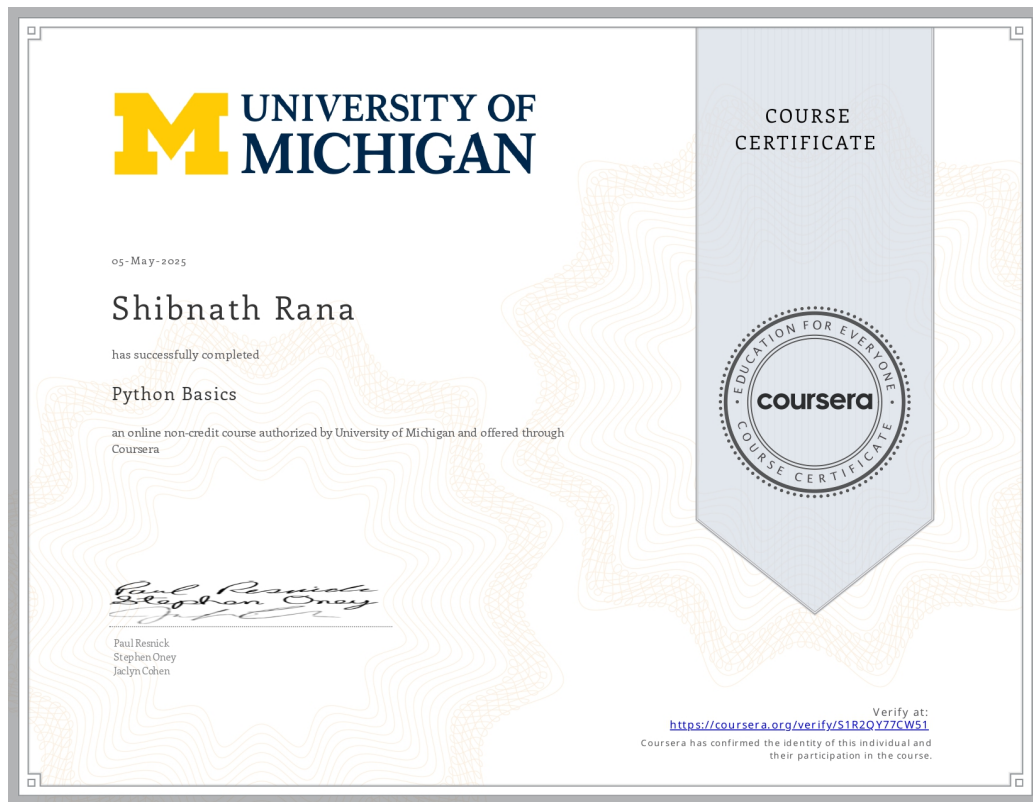

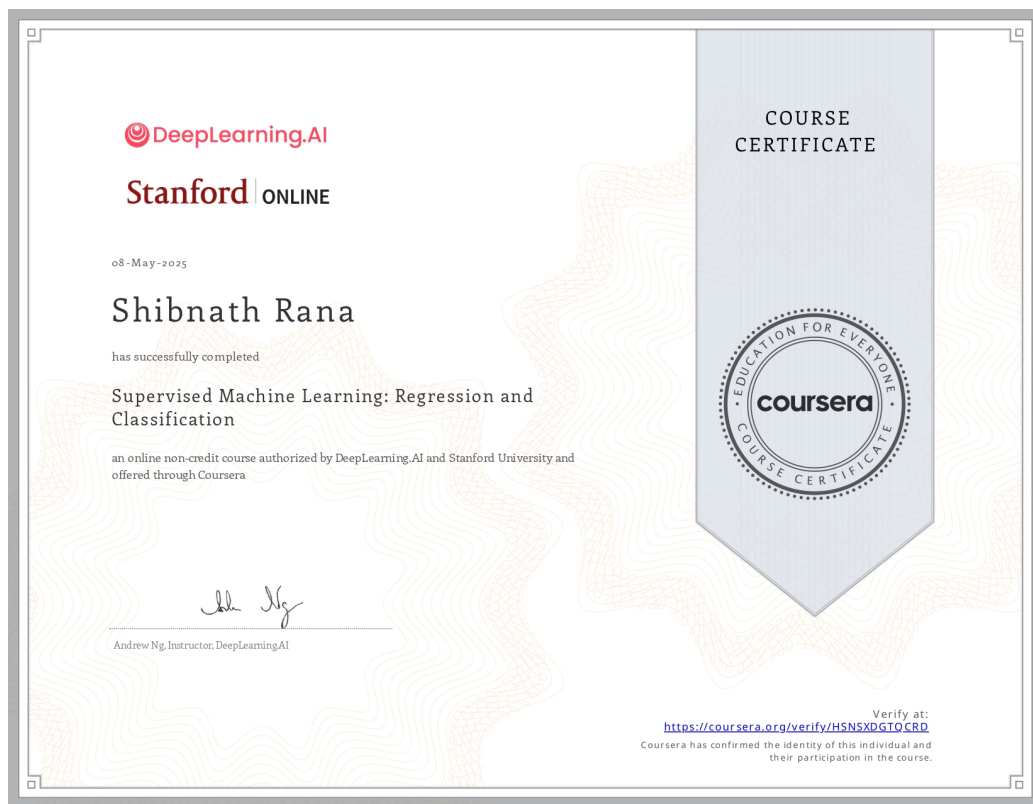Figure 8.1: Python Basics Completed by Shibnath Rana

Figure 8.2: Supervised Machine Learning: Regression and Classification Completed by Shibnath Rana

Author's Biography

Shibnath Rana is a BCA candidate from the Department of Computer Science and BCA, Belda College. He completed his Higher Secondary education at Sundarar High School(H.S.)from West Midnapore during 2020–22.

Contact Information
Permanent Address:Belar, Dhalbelun, Keshiary, Paschim Medinipur, 721424
Mobile: +91 8101062528
Email: shibnathrana0@gmail.com

Research Interests
Python, Deep Learning, Natural Language Processing (NLP).

Education

- **Belda College**, West Midnapore,West Bengal, India. Bachelor Of Computer Application
  (2022–25)

- **Sundarar High School (H.S.)**, West Midnapore,West Bengal, India. 12th
  (2022)

- **Kaliapara Ramkrishna Vidyabhaban (H.S.)**, West Midnapore, West Bengal, India. 10th **(2020)**