

**DETECTING EMAIL-BASED PHISHING WEBSITES
USING MACHINE LEARNING : PHISHING EMAIL
DETECTION WITH SENTIMENT ANALYSIS**

Yapa S. P. J.

(IT20050108)

B.Sc. (Hons) Degree in Information Technology (specialization in Cyber
Security)

Department of Computer Systems Engineering


Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Name	Student ID	Signature
Yapa S. P. J.	IT20050108	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor

Date

Signature of the Co-Supervisor

Date

ACKNOWLEDGEMENT

I want to express my sincere gratitude to everyone who has supported to start and get proper idea for my 4th year research project. I want to start by warmly thanking Ms. Jenny Krishara, who oversaw our research project and gave us the direction and motivation we needed to finish it effectively. I also want to thank Ms. Madhuka Nadeeshani Koralalage, our co-supervisor.

ABSTRACT

Phishing is a type of internet scam in which a fraudulent individual or organization tries to trick victims into disclosing private information like usernames, passwords, credit card numbers, or other personal information. To mislead the recipient into giving their personal or sensitive information, an attacker will send phishing emails. Phishing emails are designed to look like they are from reliable sources, such banks, social media platforms, online payment systems, or e-commerce websites. This paper outlines the methodology, presents experimental results, and discusses the implications of this integrated approach. The results demonstrate improved accuracy and precision in identifying phishing emails. By considering the emotional context of email content, this innovative approach contributes to more effective and reliable phishing detection systems, offering enhanced protection against evolving cyber threats. The main goal of this project is creating a phishing email detection with sentiment analysis using machine learning.

Keywords – Phishing Email Detection, Sentiment Analysis, Gradient Boosting Classifier, Natural Language Processing, Machine Learning, Support Vector Machine

TABLE OF CONTENTS

DECLARATION.....	1
ACKNOWLEDGEMENT.....	2
ABSTRACT.....	3
LIST OF FIGURES.....	5
LIST OF TABLES.....	5
LIST OF ABBREVIATIONS.....	6
1 INTRODUCTION	
1.1 Background & Literature Survey	
1.1.1 Phishing.....	7
1.1.2 Phishing email	9
1.1.3 Phishing History	11
1.1.4 Purpose of phishing email detection system.....	16
1.2 Research Gap.....	19
2 RESEARCH PROBLEM.....	24
3 OBJECTIVES	
3.1 Specific Objective	25
3.2 Sub Objectives.....	26
4 Methodology	
4.1 Technologies that used	27
4.2 Model Used	32
4.3 Commercialization aspects of the product	45
5 TESTING & IMPLEMENTATION RESULTS & DISCUSSION	46
5.1 Testing and Implementation Results	47
5.2 Discussion	49
6 CONCLUSION.....	50
7 REFERENCES.....	51
8 APPENDIX.....	53

LIST OF FIGURES

Figure 1.1	Phishing History.....	15
Figure 4.1	Phishing Email Detection System Diagram	37
Figure 4.2	App Diagram.....	37
Figure 4.3	Full Project Diagram.....	38
Figure 4.4	ML Code (Data Preprocessing).....	39
Figure 4.5	ML Code (Data Visualization).....	40
Figure 4.6	ML Code (Data Splitting and Feature Engineering).....	41
Figure 4.7	ML Code (Model Training).....	42
Figure 4.8	Flask server code.....	44
Figure 4.9	Mobile Application User Interface.....	45
Figure 5.1	Accuracy(Bernoulli Naive Bayes).....	47

LIST OF TABLES

Table 1.1	History of phishing email incidents.....	14
Table 1.2	Comparison of existing phishing email detection tools	23
Table 5.1	Accuracy of each model	47

LIST OF ABBREVIATIONS

Abbreviation	Description
ML	Machine Learning
NN	Neural Network
DL	Deep Learning
SVM	Support Vector Machine
GBC	Gradient Boosting Classifier
RF	Random Forest
NLP	Natural Language Processing
PCA	Principal Component Analysis

1 INTRODUCTION

1.1 Background & Literature Survey

1.1.1 Phishing

In today's interconnected digital landscape, the term "phishing" has become synonymous with a pervasive and insidious threat that plagues individuals, businesses, and organizations worldwide. Phishing is a form of cybercrime that relies on deception and manipulation to exploit the trust and naivety of internet users. It involves the creation of deceptive emails, websites, or messages that appear to be from legitimate sources, but are, in fact, carefully crafted traps designed to steal sensitive information, compromise security, and wreak havoc on victims' lives. As technology continues to advance, so do the methods employed by cybercriminals, making it essential for individuals and organizations to stay vigilant and informed about the evolving landscape of phishing attacks.

Phishing is a form of social engineering in which the attacker obtains private and sensitive data, such as login information, credit card information, and passwords. Phishing is carried out via electronic means, such as emails or text messages that seem to have come from dubious sources. Sending emails or messages that appear to be from a reliable source—such as a bank or social networking platform—but are from a hostile actor looking to steal sensitive information, is the fundamental method for accomplishing this. It is possible to instruct the sender of the email or message to open a specific attachment or click on a link that would drive them to a fake website that mimics the legitimate one. An attacker can utilize personal information entered by a victim on a fake website to commit identity theft or commit other sorts of fraud. Phone calls, SMS, and social engineering techniques can all be used to launch phishing attacks. It's important to be cautious and vigilant when responding to unwanted emails or texts, and you should never divulge personal information until you are positive of its validity.

Phishing attacks typically revolve around social engineering tactics, where cybercriminals impersonate trusted entities, such as banks, social media platforms, or

government agencies, to manipulate victims into divulging personal information, financial details, or login credentials. The term "phishing" itself is a play on the word "fishing," as attackers cast their deceptive bait in the hopes of hooking unsuspecting victims. These baited messages often contain urgent or enticing content, compelling recipients to take immediate action, like clicking on a link or downloading an attachment. Once the victim takes the bait, they are directed to a counterfeit website or prompted to provide sensitive information, which is then harvested by the attacker. The consequences of falling victim to a phishing attack can be dire, ranging from financial losses and identity theft to compromised systems and data breaches that can impact not only individuals but also entire organizations.

Phishing attacks have evolved over the years, becoming increasingly sophisticated and difficult to detect. Cybercriminals have honed their skills in crafting convincing messages and websites, often leveraging psychological tactics to manipulate human behavior. They prey on common emotions such as fear, curiosity, or urgency to elicit desired responses from their targets. Moreover, the variety of phishing techniques has expanded beyond email-based attacks to include SMS phishing (smishing), voice phishing (vishing), and even attacks through social media platforms. These diversifications allow attackers to cast wider nets and target individuals across various communication channels.

As we delve deeper into the world of phishing, it is crucial to explore the motives behind these malicious activities, the various tactics employed by cybercriminals, and, most importantly, the strategies and best practices individuals and organizations can adopt to protect themselves from falling victim to these pervasive and damaging scams. In an era where our digital lives are more intertwined than ever before, understanding and combating phishing attacks is a vital step toward safeguarding our personal and collective security in the vast expanse of cyberspace.

1.1.2 Phishing email

In the ever-evolving landscape of cybersecurity, few threats have proven as persistent and pernicious as phishing emails. These deceptive digital missives, with their seemingly innocuous subject lines and familiar sender names, have become the Trojan horses of the internet era, surreptitiously infiltrating our inboxes with malicious intent. Phishing emails are more than just a nuisance; they are potent weapons wielded by cybercriminals to infiltrate systems, steal sensitive data, and wreak havoc on individuals, businesses, and organizations worldwide.

At first glance, a phishing email may appear innocuous, even benign, nestled among the legitimate correspondence in our email inboxes. Its purpose, however, is far from benign. Disguised as legitimate communications from trusted sources—ranging from financial institutions and social media platforms to government agencies and online retailers—phishing emails are cunningly crafted to manipulate human psychology, exploit trust, and ultimately deceive recipients into divulging confidential information, clicking on malicious links, or unwittingly downloading malware.

The term "phishing" itself is a wordplay on "fishing," as cybercriminals dangle deceptive bait, hoping to lure unsuspecting victims into their digital traps. While the concept of phishing is not new, its tactics have evolved in lockstep with technological advancements. As a result, understanding the multifaceted nature of phishing emails is not only essential but also a critical defense against the digital threats that permeate our interconnected lives.

To mislead the recipient into giving their personal or sensitive information, an attacker will send phishing emails. Phishing emails are designed to look as though they were sent by reliable sources like banks, social media platforms, online payment systems, or e-commerce websites. The emails typically contain a call to action, such as downloading an attachment or clicking on a link, that directs the victim to a fake website made to look legitimate. The attacker can exploit the victim's personal information they enter on a fake website to commit financial fraud, steal their identity, or gain unauthorized access to their accounts. Phishing emails frequently

make use of compelling language, logos, and other graphic elements to appear genuine. They may also contain urgent or frightening messages to create a sense of urgency and drive the recipient to act. To prevent falling for a phishing email, it's imperative to exercise caution when receiving unsolicited messages or emails and never disclose personal or sensitive information until being assured of the request's legitimacy. Additionally, check to verify whether the sender's email address is real and linger over any links in emails to see where they will lead you before clicking.

Phishing emails are the offspring of social engineering, a crafty manipulation of human psychology. These deceptive messages are designed to mimic genuine communications, often displaying logos, fonts, and formatting that mirror those of reputable organizations. To make matters more challenging, the sender's email address is frequently disguised to appear as if it originates from a legitimate source, furthering the illusion of authenticity.

The content of phishing emails varies widely, but they often share common elements that aim to elicit a specific response from the recipient. Here are some of the key characteristics that define a phishing email:

Urgency or Fear: Phishing emails often prey on human emotions, exploiting feelings of urgency or fear. They may claim that an account has been compromised, an unauthorized transaction has occurred, or legal action will be taken unless immediate action is taken. This urgency compels recipients to act hastily without scrutinizing the email's legitimacy.

Fake Links: Phishing emails typically contain links that appear genuine but lead to counterfeit websites. These websites are crafted to steal login credentials or financial information when users unwittingly enter their data.

Attachments: Some phishing emails include malicious attachments. Opening these attachments can infect the recipient's device with malware, granting cybercriminals unauthorized access or control.

Spoofed Sender Information: Cybercriminals often manipulate the "From" field in the email to make it appear as though the message is coming from a trusted source. This technique makes it difficult for recipients to discern the email's true origin.

Grammatical and Spelling Errors: While phishing emails have become increasingly sophisticated, they may still contain grammatical or spelling errors. These errors can sometimes serve as red flags to astute recipients.

Requests for Personal Information: Phishing emails frequently request personal information, such as Social Security numbers, credit card details, or passwords. Legitimate organizations typically do not request such sensitive information via email.

Phishing emails are not confined to any one industry or sector; they target a broad spectrum of individuals and organizations. Their success lies in their ability to masquerade as trustworthy communications, exploiting the very trust that binds our digital interactions. Understanding the anatomy of these deceptive emails is the first step in recognizing and mitigating their threat. In the digital age, where the volume of emails continues to surge, and cybercriminals refine their tactics, the ability to discern between genuine and malicious communications is a crucial skill to protect our personal and collective cybersecurity. In the following sections, we will delve deeper into the motives behind phishing campaigns, the evolving tactics employed by cybercriminals, and the proactive measures individuals and organizations can take to defend against this persistent digital menace.

1.1.3 Phishing History

Phishing, a term derived from the word "fishing," aptly describes the deceptive practice of luring unsuspecting individuals into divulging sensitive information, such as login credentials, financial data, or personal details. Phishing history is a chronicle of how this malicious activity has evolved and become one of the most pervasive and

damaging cyber threats in the digital age. This description will take you through the key stages and developments in the history of phishing:

1990s: The Birth of Phishing

The roots of phishing can be traced back to the early days of the internet in the 1990s. Attackers began using deceptive tactics to trick users into revealing their AOL login credentials. These early efforts laid the groundwork for what would become a sophisticated form of cybercrime.

Early 2000s: Coining the Term "Phishing"

The term "phishing" is believed to have originated around this time. It encapsulates the idea of attackers casting a wide net, just like fishermen, to catch unsuspecting victims. Phishing attacks primarily targeted email users and were characterized by deceptive messages that impersonated legitimate entities like banks or online services.

2003: Rise of Organized Phishing

The Rock Phish Gang, one of the first organized cybercriminal groups, emerged as a prominent force in the world of phishing. Known for their structured and sophisticated attacks, they set a precedent for more professional phishing operations.

Mid-2000s: Proliferation of Phishing Attacks

Phishing attacks gained momentum, targeting not only email users but also various sectors, including banking, e-commerce, and social media. Attackers honed their tactics, making phishing emails increasingly convincing and difficult to identify.

2004: Introduction of Spear Phishing

The concept of "spear phishing" emerged, representing a more targeted and personalized form of phishing. In spear phishing, attackers craft emails tailored to specific individuals or organizations, often using carefully collected information to make the messages appear highly credible.

Late 2000s: Accessibility of Phishing Tools

Phishing became more accessible to a wider range of cybercriminals with the availability of phishing kits and DIY tools on the dark web. This democratization of phishing lowered the entry barrier, enabling less experienced individuals to launch their own campaigns.

2010s: Advancements in Phishing Tactics

Phishing attacks continued to evolve. Social engineering techniques, malicious attachments, and compromised websites became commonplace tactics used to deceive victims into disclosing sensitive information or installing malware.

2013: High-Profile Data Breaches

The massive Target data breach occurred, affecting millions of customers. This incident highlighted the financial impact of successful phishing attacks on large organizations and their customers.

2016: The Emergence of Business Email Compromise (BEC)

Business Email Compromise (BEC) attacks gained prominence. These attacks typically involve fraudulent emails targeting businesses or organizations, often impersonating executives, or vendors, and seeking funds or sensitive information.

2020s: Phishing in the Age of Pandemics

The COVID-19 pandemic triggered a surge in pandemic-themed phishing attacks. Cybercriminals capitalized on fear, uncertainty, and increased digital reliance to launch phishing campaigns.

Ongoing: Phishing as a Persistent Threat

Phishing remains a persistent and continually evolving threat in the digital age. Attackers adapt to technological advancements and exploit human psychology, employing a wide array of tactics to deceive and compromise individuals and organizations.

The history of phishing is a testament to the adaptability of cybercriminals and the ever-present need for robust cybersecurity measures and awareness. As technology continues to advance, individuals and organizations must remain vigilant, educate themselves about evolving threats, and implement proactive measures to defend against phishing attacks.

Table 1.1: History of phishing email incidents

Year	Incident/Breach	Target/Organization	Impact
2003	eBay phishing attack	eBay users	Stolen login credentials and financial loss
2004	Citibank phishing incident	Citibank customers	Financial fraud
2008	Targeted attacks on government officials	Various government agencies	Compromised data and sensitive information
2013	Target data breach	Target	40 million credit card records stolen
2016	Business Email Compromise (BEC) on Ubiquiti Networks	Ubiquiti Networks	\$46.7 million loss in a fraudulent wire transfer

2017	Google Docs phishing attack	Google users	Unauthorized access to Google accounts
2018	Facebook data breach	Facebook users	Unauthorized access to user data
2019	Phishing campaign against United Nations	UN employees	Compromised sensitive information
2020	COVID-19 phishing campaigns	Various targets	Scams, data theft, and malware distribution
2021	Colonial Pipeline ransomware attack	Colonial Pipeline	Temporary shutdown of a major fuel pipeline

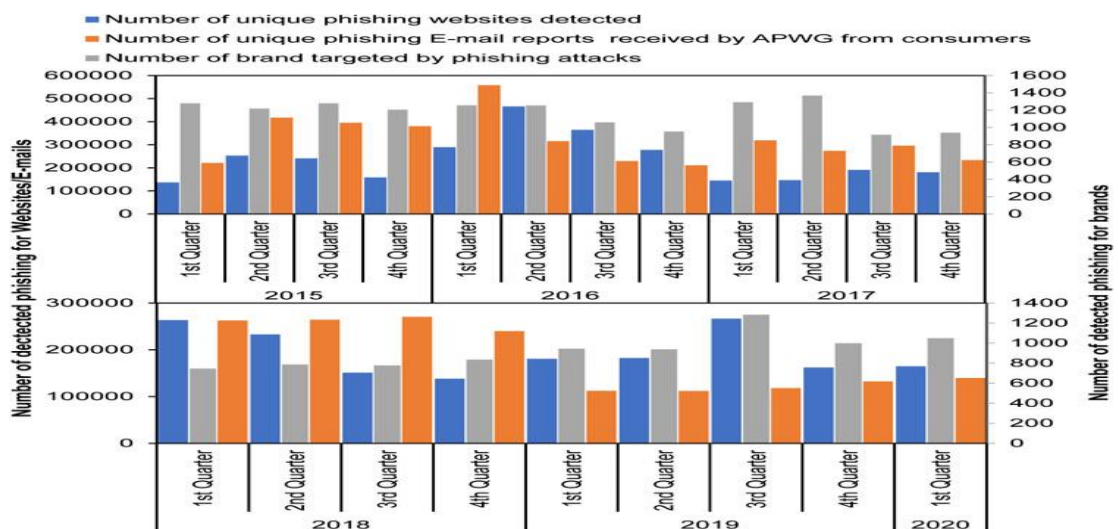


Fig. 1.1 Phishing History [1]

Based on information gathered from the APWG (Anti-Phishing Working Group), the image is a line graph that depicts the increase in phishing attempts between the years 2015 and 2020. Time is represented on the graph's horizontal axis, which is divided into four quarters per year. The number of phishing assaults that were observed is shown on the vertical axis. The graph's line represents the number of attacks that were reported in each quarter, with the last quarter of 2020 recording the most attacks. The graph amply depicts the upward trend of phishing attacks throughout time, with a notable spike in 2020. The graph also demonstrates that although the overall trend is higher, the number of attacks varies with time. The graph illustrates the worrisome rise in phishing attempts in recent years and emphasizes the significance of cybersecurity measures to shield people and companies from these assaults.

1.1.4 Purpose of phishing email detection system

In today's hyperconnected digital landscape, where email remains one of the primary modes of communication, the proliferation of phishing attacks poses a significant threat to individuals, businesses, and organizations worldwide. Phishing email detection systems, a crucial component of modern cybersecurity infrastructure, serve the paramount purpose of identifying and thwarting these deceptive emails. These systems are designed to protect users from falling victim to phishing schemes, safeguard sensitive information, and preserve the integrity of digital ecosystems. In this comprehensive exploration, we delve into the multifaceted purposes of phishing email detection systems, highlighting their importance in mitigating cyber threats.

1. Identifying Deceptive Emails:

The foremost purpose of a phishing email detection system is to identify and flag potentially malicious emails that infiltrate users' inboxes. These systems employ a variety of sophisticated techniques, including machine learning algorithms,

heuristics, and pattern recognition, to analyze the content, metadata, and sender information of incoming emails. By scrutinizing these elements, detection systems can assess whether an email exhibits characteristics consistent with phishing attempts. This proactive identification is essential in preventing users from inadvertently engaging with deceptive content.

2. Protecting Sensitive Information:

Phishing attacks are often designed to trick recipients into divulging sensitive information such as login credentials, financial details, or personal identification. Phishing email detection systems play a vital role in safeguarding this information by recognizing and quarantining emails that solicit such data. By preventing users from interacting with these malicious messages, these systems help protect individuals and organizations from data breaches, identity theft, and financial losses.

3. Mitigating Financial and Reputational Risks:

Phishing attacks can have severe financial and reputational repercussions for businesses and organizations. A successful phishing campaign can lead to financial losses, regulatory fines, and damage to a company's reputation. Phishing email detection systems act as a first line of defense, reducing the likelihood of successful attacks. By doing so, they help mitigate the financial and reputational risks associated with falling victim to phishing schemes.

4. Preserving Trust and Confidence:

Trust is the cornerstone of online interactions. When users trust their email systems to filter out phishing emails effectively, they can have greater confidence in their digital communications. Phishing email detection systems help preserve this trust by minimizing the chances of users encountering deceptive content. This, in turn,

encourages users to engage with email communication more confidently, knowing that their systems are actively protecting them.

5. Enhancing Productivity:

Phishing emails can be time-consuming and disruptive. When users inadvertently open malicious emails or click on fraudulent links, it can lead to system downtime, compromised accounts, and lost productivity. Phishing email detection systems reduce these disruptions by minimizing the exposure to such threats. This enables individuals and organizations to focus on their core activities without being constantly wary of malicious emails.

6. Staying Ahead of Evolving Threats:

Phishing techniques continually evolve as cybercriminals adapt and refine their tactics. Phishing email detection systems are equipped to stay ahead of these evolving threats by regularly updating their algorithms and databases. They incorporate threat intelligence feeds and analyze emerging patterns to detect new phishing schemes promptly. By doing so, they help organizations remain resilient in the face of an ever-changing threat landscape.

7. Supporting Compliance Requirements:

Many industries and organizations are subject to stringent data protection and cybersecurity regulations. Phishing email detection systems assist in meeting these compliance requirements by actively identifying and mitigating threats that could lead to data breaches. By adhering to these standards, organizations can avoid regulatory fines and legal consequences.

8. Educating Users:

Phishing email detection systems often work in conjunction with user education programs. When a phishing email is detected, users are typically notified and educated about the potential risks. This serves as a valuable tool for raising awareness among users, teaching them to recognize phishing attempts, and promoting responsible email behavior.

1.2 Research Gap

These are some details of existing phishing email detection system researches.

- Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding [2] used some various technologies and machine learning/ deep learning models such as One-Hot Encoding Naive Bayes , Decision Tree, Random Forest, SVM, Deep Learning (Convolutional Neural Network). Research problem of this research is Improve accuracy of phishing email detection. Features of this research used are Extracted text-based features including subject, sender, URL, and attachment. The novelty of this research is the use of one- hot encoding for text classification. Phishing Email
- Detection Using Natural Language Processing Techniques: A Literature Survey [3] used some various technologies and machine learning/ deep learning models such as NLP, SVM, Naive Bayes, Random Forest, Decision Trees, Logistic Regression, Neural Networks, K-Nearest Neighbor, Hidden Markov Models. Research problem of this research is Lack of large-scale, diverse datasets for phishing email detection using NLP techniques; Need for more research on feature extraction and selection methods for NLP-based phishing email. Features of this research used are URLs, email headers, sender and recipient email addresses, email content, lexical and semantic features, and many more. The novelty of this research is Survey of literature on NLP-based phishing email detection techniques.

- Novel email spam detection method using sentiment analysis and personality recognition [4] used some various technologies and machine learning/ deep learning models such as NLP, Sentiment Analysis, Personality Recognition, SVM, Naive Bayes, Decision Trees, Random Forest. Research problem of this research is Identifying and addressing the limitations of traditional spam detection methods, exploring new methods to improve detection accuracy. Features of this research used are Bag-of-words, Part-of-speech tags, sentiment features, lexical features, personality features. Novelty of this research is Combining sentiment analysis and personality recognition for spam detection.
- Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails [5] used some various technologies and machine learning/ deep learning models such as NLP, Logistic Regression, Random Forest, SVM. Research problem of this research is Phishing and spam email detection. Features of this research used are 70 content and metadata features, including sender and recipient information, email structure, language, sentiment, and keywords. The novelty of this research is Hybrid machine learning approach combining content and metadata features, feature importance ranking to identify key features, application of model ensemble.
- Semantic Feature Selection for Text with Application to Phishing Email Detection [6] used some various technologies and machine learning/ deep learning models such as NLTK, scikit-learn, Logistic regression, Decision tree, Random forest, Naive Bayes, SVM. Research problem of this research is improving the effectiveness and efficiency of phishing email detection. Features of this research used are Email header features, lexical features, content-based features, semantic features based on LSA. Novelty of this research is Proposed a semantic feature selection approach based on LSA to improve phishing email detection.
- Detecting phishing e-mails using Text Mining and features analysis [7] used some various technologies and machine learning/ deep learning models such as Text Mining, Feature Extraction, SVM, Random Forest, Naive Bayes. The research problem of this research is Phishing attacks are a growing concern for organizations and individuals, and there is a need for effective and efficient

methods to detect them. Text-based approaches can be useful in identifying phishing emails, but feature selection and classification models need to be carefully considered for optimal performance. Features of this research used are a combination of semantic and syntactic features, including URL analysis, header analysis, content analysis, and linguistic features. Feature selection is performed using mutual information and chi-squared methods. The novelty of this research is a feature analysis method for detecting phishing emails that combines semantic and syntactic features. It also compares the performance of different machine learning models.

- Phishing Detection in E-mails using Machine Learning [8] used some various technologies and machine learning/ deep learning models such as Support Vector Machines, Naive Bayes, Random Forest, Logistic Regression, and Voted Perceptron. Research problem of this research is Classify emails as either phished or ham based on features extracted from a dataset of such emails. Features of this research used are link-based, tag-based, and word-based features, totaling nine features. The novelty of this research is Variation of features and use the least number of features to develop a system that provides higher accuracy.
- Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism [9] used some various technologies and machine learning/ deep learning models such as NLP, Recurrent Convolutional Neural Network (RCNN), Attention Mechanism. Research problem of this research is phishing email detection, effective feature representation for email content, addressing the imbalance problem in the dataset. Features of this research used are content features: bag-of-words, Term Frequency-Inverse Document Frequency (TF-IDF), word embedding vectors, multilevel vectors, header features: sender email address, recipient email address, email subject, email attachment. Novelty of this research is proposed an improved RCNN model with attention mechanism for phishing email detection, Introduced multilevel vectors to capture the semantic meaning of email content, employed both content and header features of emails for classification.
- An intelligent classification model for phishing email detection [10] used some various technologies and machine learning/ deep learning models such as

Logistic Regression, Random Forest, SVM. Research problem of this research is phishing email detection. Features of this research used are N-gram, URL-based features. Novelty of this research is Combination of N-gram and URL-based features, comparative analysis of multiple ML models.

- Research on phishing email detection based on URL parameters using machine learning algorithms [11] used some various technologies and machine learning/ deep learning models such as Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, Neural Network. Research problem of this research is phishing email detection. Features of this research used are URL parameters, subject, body, sender, attachments, hyperlinks. The novelty of this research is using URL parameters for feature extraction.
- Phishing Email Detection Using Machine Learning Techniques [12] used some various technologies and machine learning/ deep learning models such as Random Forest Classifier, Naive Bayes Classifier. Research problem of this research is Developing an effective and efficient phishing email detection system using machine learning techniques. Features of this research used are URL-based features (e.g., domain age, domain registration period, URL length), text-based features (e.g., frequency of certain words, presence of certain phrases), header-based features (e.g., sender's email address, subject line), attachment-based features (e.g., presence of executable files). The novelty of this research is Used a combination of different features for phishing email detection, Conducted experiments on a real-world dataset consisting of legitimate and phishing emails, Achieved high accuracy in detecting phishing emails.
- Phishing Email Detection Using Robust NLP Techniques [13] used some various technologies and machine learning/ deep learning models such as SVM, Logistic Regression, Naive Bayes. Research problem of this research is detecting phishing emails using NLP techniques, improving detection accuracy, evaluating models' robustness to adversarial attacks. Features of this research used are word embeddings, n-grams, Part-of-Speech tags, named entities, semantic role labeling, sentiment analysis, length, and frequency-based features. The novelty of this research is usage of lexical and semantic features, evaluation of models' robustness to adversarial attacks.

Table 1.2: Comparison of existing phishing email detection tools

Feature	Sophos Email Protection	Trustwave Secure Email Gateway	Barracuda Email Security Gateway	PhishGuard	Valima il	Proposed System
Address the issue of Imbalance d Datasets	✓	✓	✓	✗	✗	✓
Use Large Datasets	✓	✓	✓	✗	✓	✓
Sentiment Analysis	✓	✗	✗	✗	✗	✓
Mobile App	✓	✗	✓	✓	✗	✓
Warning Notificatio n	✓	✓	✓	✓	✓	✓

This table presents a comparison of five different email protection systems and a proposed system based on various features. The first column lists the names of the email protection systems, while the subsequent columns represent the proposed system's features.

The first feature is addressing the issue of imbalanced datasets. This feature is present in Sophos Email Protection, Trustwave Secure Email Gateway, and the proposed system.

The second feature is the use of large datasets. This feature is present in Sophos Email Protection, Trustwave Secure Email Gateway, and Valimail.

The third feature is sentiment analysis. Only the proposed system and Valimail have this feature.

The fourth feature is the availability of a mobile app. Only Sophos Email Protection and the proposed system have this feature.

Finally, the fifth feature is the ability to provide warning notifications. All five email protection systems and the proposed system have this feature.

The proposed system has a unique combination of features, such as the ability to perform sentiment analysis, and provide a mobile app. Additionally, it addresses the issue of imbalanced datasets and advanced techniques.

2 RESEARCH PROBLEM

❖ How to address the issue of imbalanced datasets, which can affect the accuracy of the model.

In the realm of email security and phishing detection, one of the persistent challenges is dealing with imbalanced datasets. Imbalanced datasets occur when the number of legitimate emails far outweighs the number of phishing emails in the dataset. This imbalance can significantly affect the performance of machine learning models, causing them to be biased toward the majority class (legitimate emails) and leading to reduced accuracy in detecting phishing emails. As a result, effectively addressing this issue is crucial to ensure robust and reliable email security systems.

❖ How to use large datasets.

The size of the dataset plays a pivotal role in training accurate and reliable email security models. However, obtaining and efficiently utilizing large-scale email datasets present challenges. Acquiring diverse and extensive email data, ensuring

data quality, and optimizing model training on large volumes of data are complex tasks. Nevertheless, large datasets are critical for building email security systems that can effectively generalize and detect evolving phishing tactics.

❖ **How to use sentiment analysis for detect phishing emails.**

Phishing attacks often employ psychological manipulation, including emotional triggers, to deceive email recipients. Sentiment analysis, a branch of natural language processing, involves analyzing the emotional tone or sentiment conveyed in text. The challenge is to integrate sentiment analysis effectively into email security systems to enhance the detection of phishing emails that leverage emotional manipulation.

❖ **How to use Naive Bayes algorithm to train the model.**

Selecting an appropriate machine learning algorithm is pivotal in building an effective email security model. The Naive Bayes algorithm, known for its simplicity and efficiency in text classification tasks, holds promise for email security. However, the challenge lies in adapting and optimizing this algorithm specifically for training email security models.

3 OBJECTIVES

3.1 Specific Objective

- Phishing email detection with sentiment analysis

The specific objective of this research is to develop a robust machine learning model capable of detecting phishing emails by incorporating sentiment analysis. This objective signifies the primary focus of the project, which is to enhance email security by leveraging sentiment analysis techniques to identify malicious emails that exploit emotional triggers and manipulation.

3.2 Sub Objectives

- To develop a machine learning model that can detect phishing emails using sentiment analysis.

The first sub-objective is the foundation of this research. It involves the creation of a machine learning model tailored for the detection of phishing emails. The unique aspect of this model is its integration of sentiment analysis, enabling it to analyze the emotional content and tone of emails. By leveraging natural language processing (NLP) techniques, the model will learn to identify subtle emotional cues that are often used by cybercriminals to deceive recipients. This sub-objective entails data collection, preprocessing, feature engineering, and the development of a sentiment analysis algorithm within the model.

- Improve the model to detect the text phishing or not.

The second sub-objective builds upon the initial model's capabilities. While sentiment analysis can identify emotional manipulation, it's also crucial to enhance the model's ability to detect whether an email is phishing or not based on the textual content. This sub-objective involves the refinement of text-based features, the integration of machine learning algorithms optimized for text classification, and the creation of a robust phishing email classification system. By combining sentiment analysis with text-based analysis, the model aims to achieve a holistic understanding of email content.

- Display a warning message when detect a phishing email.

This sub-objective focuses on the practical implementation of the phishing detection model. Upon successfully identifying a phishing email, the model will trigger a warning message to alert the email recipient. The warning message will serve as an actionable response, notifying the user about the potential threat and advising caution. The development of this warning message system encompasses user interface design, message content generation, and the integration of real-time email scanning capabilities. The goal is to create an intuitive and informative warning

system that empowers users to take appropriate action when confronted with phishing attempts.

- To integrate the developed models to a mobile application.

The final sub-objective involves the integration of the developed phishing detection models into a user-friendly mobile application. Recognizing the importance of accessibility and convenience, this objective aims to make the phishing detection capabilities readily available to users on their mobile devices. This sub-objective entails mobile app development, user interface design, model deployment, and seamless integration with various email clients. The objective is to provide users with a powerful tool that enhances their email security on the go, giving them the ability to identify and respond to phishing threats directly from their smartphones.

4 Methodology

4.1 Technologies that used

Sentiment Analysis – In the digital age, we are constantly generating vast amounts of textual data through our online interactions. From social media posts and product reviews to customer feedback and news articles, the sheer volume of text available is overwhelming. Within this sea of words lies valuable information about how people feel, their opinions, and the emotional undertones of their expressions. Sentiment analysis, also known as opinion mining, is the remarkable field of natural language processing (NLP) that helps us navigate this linguistic landscape by automatically extracting and understanding the sentiment or emotional tone present in text.

Sentiment analysis has surged in importance and applicability in recent years, revolutionizing how businesses, researchers, and organizations perceive and act upon textual data. It enables us to gain insights into public sentiment, customer feedback, market trends, political opinions, and much more. This comprehensive exploration delves into the fascinating world of sentiment analysis, its underlying principles, real-world applications, challenges, and recent advancements [14].

The method of evaluating text to ascertain the sentiment or emotional tone of the writing is a form of natural language processing (NLP). Based on the language and context utilized, sentiment analysis algorithms can categorize material as positive, negative, or neutral. It is a method to NLP that pinpoints the text's emotional undertone.

Sentiment analysis, often referred to as opinion mining, is a subfield of natural language processing (NLP) that focuses on extracting and understanding sentiments, opinions, and emotional tones from textual data.

The primary aim of sentiment analysis is to categorize text into predefined sentiment categories, which commonly include:

- Positive: Expressions of joy, approval, satisfaction, or positivity.
- Negative: Expressions of discontent, dissatisfaction, criticism, or negativity.
- Neutral: Text that lacks any significant emotional tone, often factual or informational.

While these are the fundamental sentiment classes, more fine-grained sentiment analysis might include categories like "strongly positive," "moderately positive," "weakly positive," and similarly nuanced categories for negative sentiments.

Sentiment analysis is a multi-step process that involves text preprocessing, feature extraction, sentiment classification, and result interpretation [15].

1. Text Preprocessing:

The journey of sentiment analysis begins with preparing the raw text data for analysis. Text preprocessing involves several essential tasks:

Tokenization: The text is divided into individual words or tokens.

Lowercasing: All text is converted to lowercase to ensure uniformity.

Stop Word Removal: Common words like "the," "and" "is," which do not carry significant sentiment, are removed.

Stemming or Lemmatization: Reducing words to their base or root form. For instance, "running" becomes "run."

2. Feature Extraction:

Once the text is cleaned and prepared, it needs to be transformed into numerical data that machine learning models can understand. Common techniques for feature extraction in sentiment analysis include:

Bag of Words (BoW): A simple method that creates a vocabulary of words and represents each document as a vector of word occurrences.

Term Frequency-Inverse Document Frequency (TF-IDF): A more advanced technique that considers the importance of words in a document relative to their frequency across all documents.

Word Embeddings: Techniques like Word2Vec, GloVe, and BERT create dense vector representations of words, capturing semantic meaning.

3. Sentiment Classification:

The core of sentiment analysis is classifying the text into sentiment categories. Various algorithms can be employed for this task, including:

Naive Bayes: A probabilistic algorithm that calculates the likelihood of text belonging to each sentiment class.

Support Vector Machines (SVM): A machine learning algorithm that finds the best separating hyperplane between sentiment classes.

Recurrent Neural Networks (RNN): Deep learning models that consider the sequence of words and their context.

Transformer Models: State-of-the-art models like BERT and GPT-3 that have achieved remarkable results in NLP tasks.

4. Result Interpretation:

After classification, the sentiment analysis results are typically interpreted and presented in a human-readable format. This could be as simple as categorizing text as "positive," "negative," or "neutral." More advanced approaches involve sentiment scores, or confidence levels associated with each category.

While sentiment analysis offers tremendous value, it is not without its challenges:

1. **Context Understanding:** Words often carry different sentiments depending on the context. For instance, "not bad" may be positive in some contexts and negative in others.
2. **Sarcasm and Irony:** Detecting sarcasm, irony, or humor in text can be challenging as they often involve contradictory sentiments.
3. **Multilingual Analysis:** Sentiment analysis for multilingual text requires language-specific models and lexicons.
4. **Subjectivity and Ambiguity:** Sentiment can be highly subjective, and text may contain ambiguous expressions.
5. **Data Imbalance:** Imbalanced datasets, where one sentiment class significantly outweighs others, can lead to biased models.
6. **Emotion Recognition:** Sentiment analysis typically focuses on positive, negative, or neutral sentiments and may not capture fine-grained emotions like joy, anger, or sadness.

Recent advancements in NLP have propelled sentiment analysis to new heights. Key developments include:

Transformer Models: Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) have set new benchmarks in understanding context and context-dependent sentiment analysis.

Transfer Learning: Models pre-trained on large corpora of text data can be fine-tuned for specific sentiment analysis tasks, reducing the need for extensive labeled data [15].

Multimodal Sentiment Analysis: Integrating other data modalities, such as images and audio, with text data for a more holistic understanding of sentiment.

Emotion Detection: Focusing on emotion detection rather than just sentiment, capturing nuanced emotional states.

Real-time Analysis: Developing systems that can perform sentiment analysis in real-time, enabling rapid response to changing sentiments.

Natural Language Processing (NLP) – In the ever-evolving landscape of technology, one field stands out as a cornerstone of human-computer interaction and understanding: Natural Language Processing (NLP). NLP is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and humans through natural language. It aims to bridge the gap between the way humans communicate and the capabilities of computers to understand, process, and generate human language. This comprehensive exploration delves into the fascinating world of NLP, its history, underlying principles, applications, challenges, and prospects [16]. Computer science's NLP discipline examines how computers and human languages interact. It is focused on developing computational models and algorithms that can process and comprehend human language similarly to how people do. Machine learning, deep learning, and computational linguistics are only a few of the methods and tools used in NLP. Language translation, chatbots, virtual assistants, and speech recognition are just a few of the useful uses for NLP. It is also used to analyze vast volumes of data and produce insights in sectors including healthcare, banking, and education. NLP has the potential to change how we interact with computers and the outside world as the field develops [17].

Natural Language Processing (NLP) is a multidisciplinary field at the intersection of computer science, artificial intelligence, linguistics, and cognitive psychology. At its

core, NLP seeks to enable machines to understand, interpret, and generate human language in a valuable and meaningful way. This involves both understanding the structure and semantics of language and making sense of the context in which it is used.

NLP encompasses a wide range of tasks and objectives, including:

1. **Text Analysis:** Analyzing large volumes of text data to extract insights, patterns, and trends. This includes tasks like sentiment analysis, topic modeling, and text classification.
2. **Speech Recognition:** Transcribing spoken language into written text. It is widely used in voice assistants and transcription services.
3. **Language Generation:** Creating human-like text or speech. This includes chatbots, content generation, and automatic language translation.
4. **Language Understanding:** Interpreting the meaning of text or speech. This includes question-answering systems, information retrieval, and machine translation.
5. **Language Generation:** Creating human-like text or speech. This includes chatbots, content generation, and automatic language translation.
6. **Machine Translation:** Translating text or speech from one language to another, such as Google Translate.

4.2 Model Used

Naive Bayes – In the realm of machine learning and probabilistic modeling, the Naive Bayes classifier stands as a venerable workhorse. It is celebrated for its simplicity, efficiency, and remarkable performance in various classification tasks, including text categorization, spam detection, and medical diagnosis. The Naive Bayes algorithm is rooted in Bayesian probability theory and holds a central place in the toolkit of machine learning practitioners. This comprehensive exploration

embarks on a journey to uncover the principles, applications, variants, and real-world impact of the Naive Bayes classifier.

The Naive Bayes classifier is a family of probabilistic algorithms that leverage Bayes' theorem to make predictions and classifications. It is particularly suited for tasks where we want to categorize or classify objects into discrete classes based on a set of features. Despite its simplicity, the Naive Bayes classifier has proven to be surprisingly effective in many real-world scenarios.

At its core, the algorithm makes use of conditional probability and statistical independence assumptions to estimate the likelihood of an event occurring, given the presence of certain features or conditions. It is called "naive" because it assumes that the features used for classification are conditionally independent, which may not hold in some cases. However, this simplifying assumption allows the algorithm to work efficiently and often provides surprisingly accurate results.

A probabilistic machine learning model for classification is called Naive Bayes. Based on data retrieved from the email text, such as the sender address and message content, Naive Bayes can be used to forecast the likelihood that an email is a phishing email. Naive Bayes is a simple and efficient model that can work well with small datasets, making it a popular choice for phishing email detection.

While the core principles of the Naive Bayes classifier remain the same, there are different variants of the algorithm that adapt to various types of data and applications. Three common types of Naive Bayes classifiers are:

1. Multinomial Naive Bayes:

The Multinomial Naive Bayes classifier is commonly used for text classification tasks, such as spam detection and document categorization. It models the distribution of word frequencies in documents and is well-suited for datasets where the features represent the counts or frequencies of items (e.g., word counts).

2. Gaussian Naive Bayes:

The Gaussian Naive Bayes classifier is applied to datasets where the features follow a Gaussian (normal) distribution. It is suitable for continuous or real-valued features and is often used in applications like medical diagnosis and image classification.

3. Bernoulli Naive Bayes:

The Bernoulli Naive Bayes classifier is primarily used for binary classification problems, where features represent binary outcomes (e.g., presence or absence). It is well-suited for tasks like spam detection, sentiment analysis, and information retrieval.

The Naive Bayes classifier possesses several strengths that make it a valuable choice in many situations:

1. Simplicity:

Naive Bayes is conceptually straightforward and easy to implement. Its simplicity makes it a suitable choice for quick prototyping and baseline models.

2. Efficiency:

The algorithm is computationally efficient and scales well to large datasets. Training and prediction are typically fast, making it suitable for real-time applications.

3. Minimal Data Requirements:

Naive Bayes classifiers can perform well even with relatively small datasets, making them useful in scenarios where collecting extensive labeled data is challenging.

4. Good Generalization:

Despite its simplicity and the "naive" assumption of feature independence, Naive Bayes often generalizes well and delivers competitive performance in practice.

5. Interpretability:

The probabilistic nature of Naive Bayes makes it easy to interpret and explain predictions, which is valuable in domains where model transparency is crucial.

While Naive Bayes has many strengths, it is not without its limitations:

1. Independence Assumption:

The "naive" assumption of feature independence may not hold in many real-world scenarios, potentially leading to suboptimal predictions.

2. Sensitivity to Irrelevant Features:

Naive Bayes can be sensitive to irrelevant features, which can impact its performance. Feature selection and engineering are important to mitigate this issue.

3. Limited Expressiveness:

Naive Bayes may not capture complex relationships in the data as effectively as more advanced models like deep neural networks.

4. Imbalanced Data:

In cases where class distributions are highly imbalanced, Naive Bayes may be biased toward the majority class. Techniques like oversampling or using different variants of Naive Bayes can address this issue.

5. Out-of-Distribution Data:

Naive Bayes may struggle with data that significantly deviates from the training distribution, as it relies on learned probabilities.

The Naive Bayes classifier has left an indelible mark on various industries and applications. Here are some instances of its real-world impact:

1. Email Filtering:

Naive Bayes is a key component of spam filters, helping users avoid the deluge of unwanted emails and ensuring that important messages reach their inboxes.

2. Text Analysis:

In the realm of text analysis, Naive Bayes classifiers enable sentiment analysis, topic modeling, and content recommendation, enhancing user experiences in social media, e-commerce, and news platforms.

3. Fraud Detection:

Financial institutions employ Naive Bayes models to detect fraudulent transactions and activities, safeguarding customers and minimizing financial losses.

4. Language Identification:

Naive Bayes classifiers help identify the language of text data, enabling multilingual content filtering, translation, and internationalization.

5. Personalization:

E-commerce platforms use Naive Bayes-based recommendation systems to personalize product suggestions, increasing customer engagement and sales.

SVM – In the vast landscape of machine learning algorithms, Support Vector Machines (SVMs) stand as formidable pillars of classification and regression tasks. Revered for their robustness, versatility, and ability to handle complex data, SVMs have earned a significant place in the arsenal of machine learning practitioners. This comprehensive exploration delves into the principles, mathematics, applications, variants, and real-world impact of Support Vector Machines. Models of supervised learning for classification and regression analysis. Using information collected from the email text, such as the sender address and message content, SVMs can be used to classify emails as legitimate or phishing. SVMs are a popular choice for detecting phishing emails because of how effectively they work with high-dimensional datasets [18].

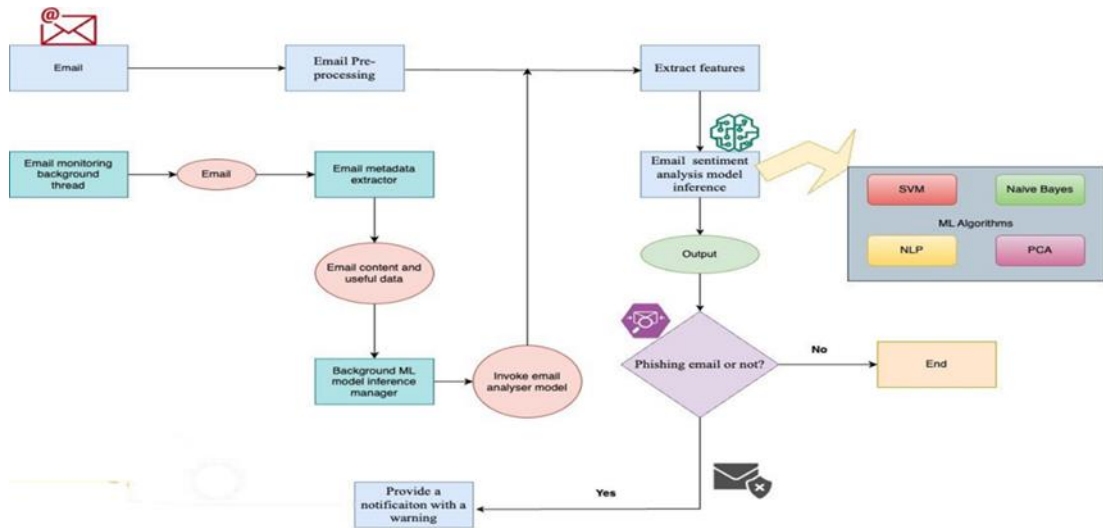


Fig. 4.1 Phishing Email Detection System Diagram

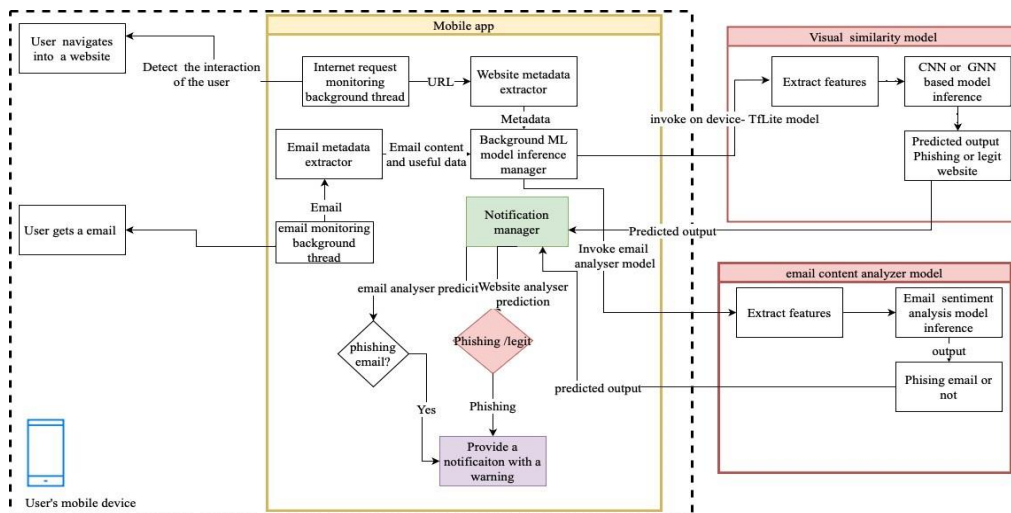


Fig. 4.2 App Diagram

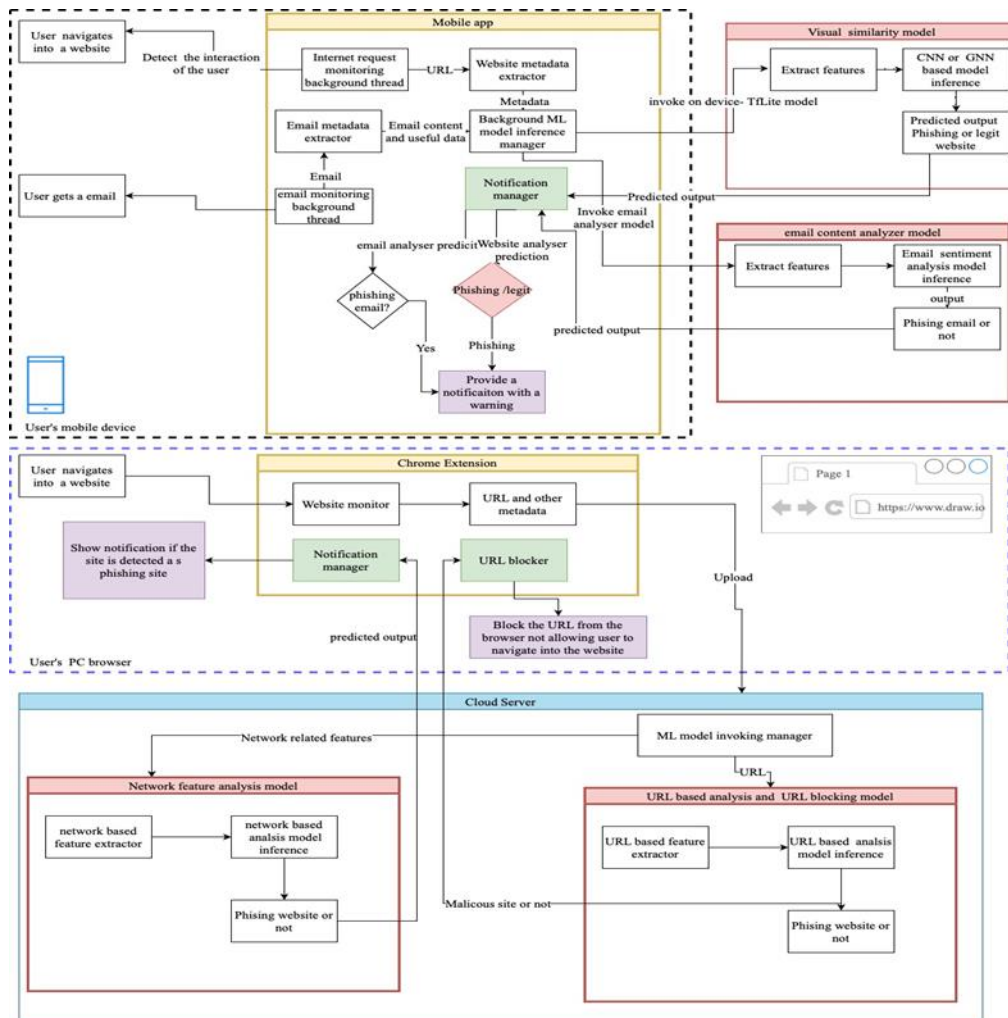


Fig. 4.3 Full Project Diagram

Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectPercentile
from sklearn.metrics import accuracy_score, confusion_matrix
from textblob import TextBlob
import joblib
```

[25]

✓

0.0s

Python

```
# Load dataset from CSV file
df = pd.read_csv('Phishing_Email.csv')
```

[26]

✓

1.1s

Python

```
# Display the first few rows of the DataFrame
df.head()
```

[27]

✓

0.0s

Python

```
...
```

	Unnamed: 0	Email Text	Email Type
0	0	re: 6. 1100 , disc: uniformitarianism , re ...	Safe Email
1	1	the other side of * galicismos * * galicismo *...	Safe Email
2	2	re: equistar deal tickets are you still avail...	Safe Email
3	3	\nHello I am your hot lil horny toy.\nI am...	Phishing Email
4	4	software at incredibly low prices (86 % lower...	Phishing Email

Data Preprocessing

```
# Display information about the dataset
df.info()
```

[28]

✓

0.0s

Python

```
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18650 entries, 0 to 18649
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Unnamed: 0   18650 non-null  int64
1   Email Text   18634 non-null  object
2   Email Type   18650 non-null  object
dtypes: int64(1), object(2)
memory usage: 437.2+ KB
```

```
# Drop unnecessary index
df.drop('Unnamed: 0', axis=1, inplace=True)
```

[29]

✓

0.0s

Python

```
# Check for missing values and duplicate rows for data cleaning
print(df.isna().sum())
print(df.duplicated().sum())
```

[30]

✓

0.0s

Python

```
...
Email Text      16
Email Type       0
dtype: int64
1111
```

```
# Remove rows with missing values
df.dropna(inplace=True)
```

[31]

✓

0.0s

Python

```
# Remove duplicate rows
df.drop_duplicates(inplace=True)
```

[32]

✓

0.1s

Python

```
x = df['Email Text']
y = df['Email Type']
```

[33]

✓

0.0s

Python

Fig. 4.4 ML Code (Data Preprocessing)

1. Data Acquisition and Preprocessing

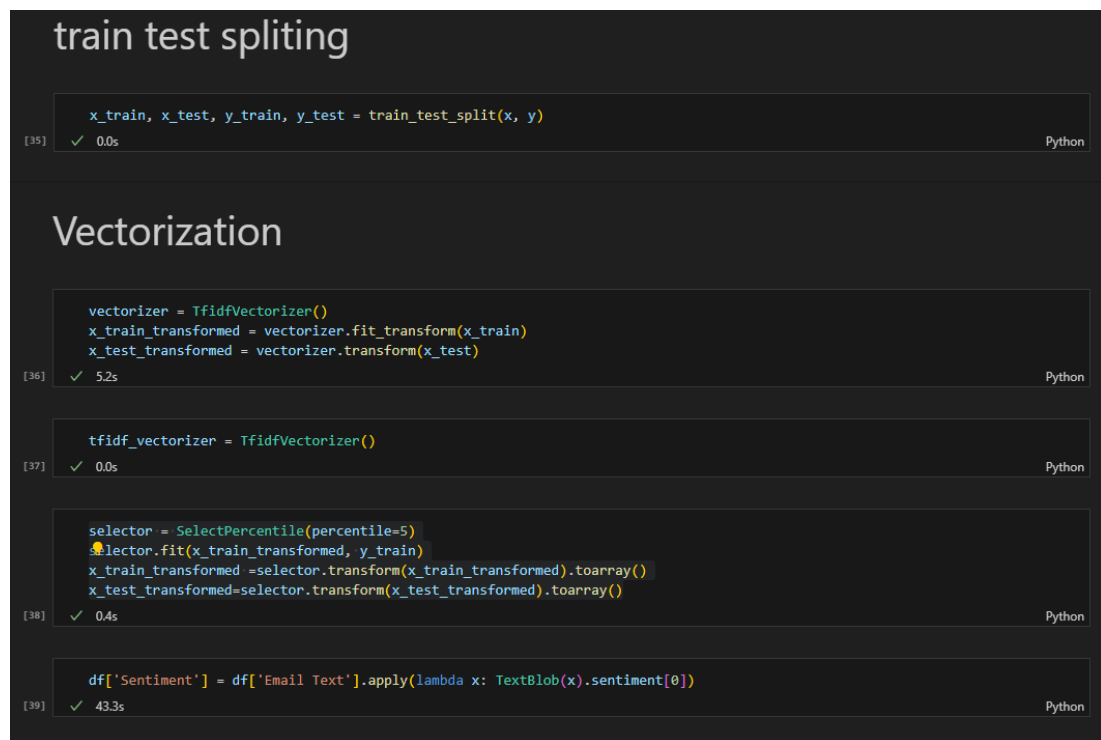
- **Data Collection:** The research begins with the acquisition of a dataset containing email samples, both legitimate (safe) and phishing emails. The dataset is loaded into a pandas DataFrame using the `pd.read_csv()` function, creating a structured data source for analysis.
- **Data Exploration:** Initial exploration of the dataset involves inspecting the first few rows using `df.head()` and obtaining an overview of the dataset's structure and attributes using `df.info()`. This step is crucial for understanding the data's dimensions and identifying any missing or duplicated values.
- **Data Cleaning:** Data cleaning is performed to enhance the dataset's quality and reliability. This includes dropping irrelevant columns, such as 'Unnamed: 0', which is accomplished with `df.drop('Unnamed: 0', axis=1, inplace=True)`. Further, missing values are detected using `df.isna().sum()` and removed using `df.dropna(inplace=True)`. Duplicate records are identified and eliminated with `df.drop_duplicates(inplace=True)`.



Fig. 4.5 ML Code (Data Visualization)

2. Data Visualization

- **Data Visualization:** Data visualization using libraries like Matplotlib and Seaborn is employed to gain insights into the dataset's distribution and characteristics. The `sns.countplot()` function is used to create a count plot that visually represents the distribution of email types (safe or phishing). This visualization helps in understanding class balance and the need for addressing imbalanced datasets.



The screenshot displays a Jupyter Notebook interface with a dark theme. It contains four code cells, each with a title, code, execution status, and time taken. The first cell is titled 'train test splitting' and shows the `train_test_split` function. The second cell is titled 'Vectorization' and shows the creation and use of `TfidfVectorizer`. The third cell shows the creation of a `SelectPercentile` selector. The fourth cell shows the application of a sentiment analysis function to the email text.

```
train test splitting

x_train, x_test, y_train, y_test = train_test_split(x, y)
[35] ✓ 0.0s Python
```

```
Vectorization

vectorizer = TfidfVectorizer()
x_train_transformed = vectorizer.fit_transform(x_train)
x_test_transformed = vectorizer.transform(x_test)
[36] ✓ 5.2s Python
```

```
tfidf_vectorizer = TfidfVectorizer()
[37] ✓ 0.0s Python
```

```
selector = SelectPercentile(percentile=5)
selector.fit(x_train_transformed, y_train)
x_train_transformed = selector.transform(x_train_transformed).toarray()
x_test_transformed = selector.transform(x_test_transformed).toarray()
[38] ✓ 0.4s Python
```

```
df['Sentiment'] = df['Email Text'].apply(lambda x: TextBlob(x).sentiment[0])
[39] ✓ 43.3s Python
```

Fig. 4.6 ML Code (Data Splitting and Feature Engineering)

3. Data Splitting and Feature Engineering

- **Data Splitting:** The dataset is divided into training and testing sets using `train_test_split(x, y)`. This separation allows for model training on one portion of the data and model evaluation on the other.

- **Text Vectorization:** The email text content is transformed into numerical features suitable for machine learning using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique. This is achieved through **TfidfVectorizer()**. It calculates the importance of words in each email, converting them into numerical vectors.
- **Feature Selection:** To reduce the dimensionality of the TF-IDF vectors and potentially improve model performance, SelectPercentile is used for feature selection. It retains the top percentile of features most relevant to the classification task.
- **Sentiment Analysis:** Sentiment analysis is applied to email text using TextBlob. Sentiment scores (polarity) are extracted from each email to capture emotional content within the emails. This sentimental information will be used as an additional feature for the machine learning model.

```

Model

berNB = BernoulliNB(alpha=0.01)
berNB.fit(x_train_transformed, y_train)
y_predict=berNB.predict(x_test_transformed)
[40] ✓ 1.5s Python

# Save the trained model to a file
model_filename = 'phishing_email_model.pkl'
joblib.dump(berNB, model_filename)
print(f"Model saved as {model_filename}")
[41] ✓ 0.0s Python
... Model saved as phishing_email_model.pkl

# Load the saved model
loaded_model = joblib.load('phishing_email_model.pkl')
[42] ✓ 0.0s Python

# Choose a single item from the test set
index_to_predict = 0
single_email_text = x_test.iloc[index_to_predict]
[43] ✓ 0.0s Python

# Transform the single email text and make a prediction
single_email_text_transformed = selector.transform(vectorizer.transform([single_email_text]).toarray())
predicted_class = loaded_model.predict(single_email_text_transformed)
[44] ✓ 0.0s Python

# Determine the phishing classification
phishing_result = "Safe Email" if predicted_class[0] == 1 else "Phishing Email"
[53] ✓ 0.0s Python

# Print the prediction
print(f"Predicted Class for Test Item {index_to_predict}: {phishing_result}")
[54] ✓ 0.0s Python

```

Fig. 4.7 ML Code (Model Training)

4. Model Training

- **Model Selection:** Three variants of the Naive Bayes algorithm, including GaussianNB, BernoulliNB, and MultinomialNB, are available for experimentation. In this code, BernoulliNB with a specified alpha value is chosen as the classification algorithm. It is fitted to the training data using **berNB.fit()**. The heart of the phishing email detection system lies in the model training and evaluation. A Bernoulli Naive Bayes classifier is chosen as the machine learning algorithm for this task. This choice assumes that the features follow a Bernoulli distribution, which is often suitable for binary classification tasks. The classifier is trained on preprocessed and transformed training data.

5. Model Serialization and Deployment

- **Model Serialization:** The trained phishing email detection model is serialized using the joblib library. It is saved as a binary file with the filename 'phishing_email_model.pkl'. This step ensures that the model can be easily loaded and used for future predictions without retraining.
- **Model Deployment:** Although not explicitly implemented in this code, the serialized model can be integrated into various applications, such as web or mobile apps. The code demonstrates how to load the saved model and make predictions on new email text samples.
- **Real-time Prediction:** A single email text sample from the test set is chosen for real-time prediction. The sample is transformed using the trained feature selection and vectorization techniques, and the saved model is used to predict whether the email is phishing or safe. The prediction result is printed to the console.

```

from flask import Flask, request
import joblib
from sklearn.feature_extraction.text import TfidfVectorizer
import pickle

loaded_model = joblib.load('phishing_email_model.pkl')
file = open('vectorizer.pk', 'rb')
vectorizer = pickle.load(file)
file_sel = open('selector.pk', 'rb')
selector = pickle.load(file_sel)
# Flask Constructor
app = Flask(__name__)
# decorator to associate
# a function with the url
@app.route("/")
def showHomePage():
    # response from the server
    return "This is home page"

@app.route("/predict", methods=['POST'])
def predict():
    single_email_text = request.form['email']
    #single_email_text="The impression I get from reading lkml the odd time is"
    single_email_text_transformed = selector.transform(vectorizer.transform([single_email_text]).toarray())
    predicted_class = loaded_model.predict(single_email_text_transformed)
    phishing_result = '0' if predicted_class[0] == 1 else '1'
    print(f"Predicted Class {phishing_result}")

    return phishing_result
if __name__ == "__main__":
    app.run(host="172.20.10.4", port='8080')

```

Fig. 4.8 Flask server code

This is a Python Flask web application that serves as a simple API for predicting whether an input email is a phishing email or not. It uses that previous pre-trained machine learning model for prediction. For integrate the model to API loaded the model to flask server.

4.3 Commercialization aspects of the product

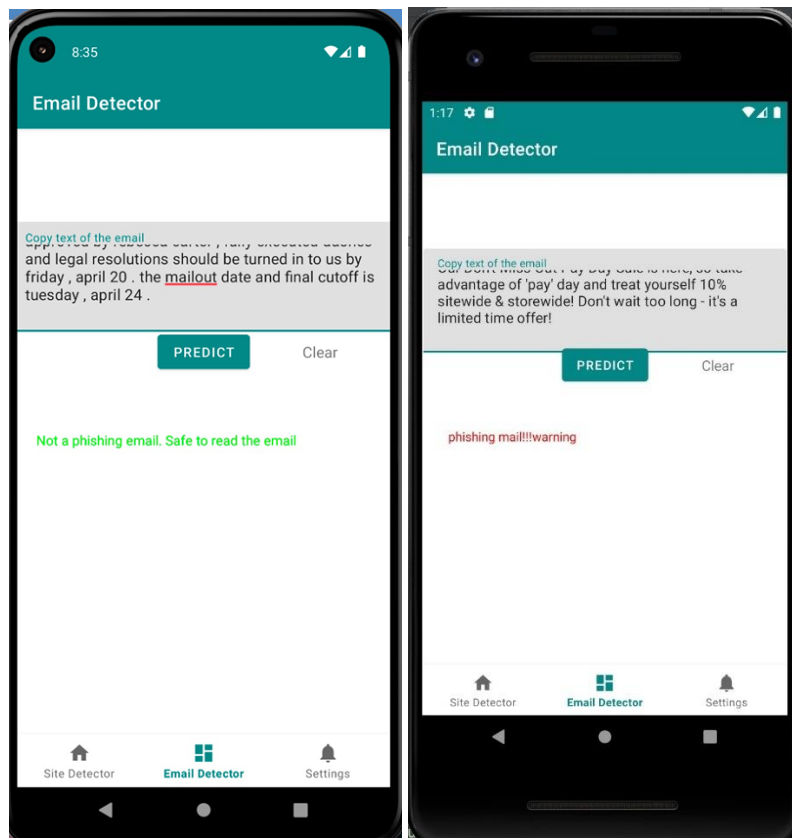


Fig. 4.9 Mobile Application User Interface

Our mobile app is designed to cater to the needs of both basic and advanced mobile users. It is particularly useful for users who deal with many emails and engage in frequent web browsing, as they are often the target of phishing attacks. For basic mobile phone users, our app provides a user-friendly interface and simple functionalities that are easy to navigate. It can be easily downloaded and installed on any basic mobile phone without the need for a high-end device. The app is designed to help users identify and avoid phishing scams, which are becoming increasingly common in today's digital age. Advanced mobile users, on the other hand, require more robust security features to stay safe from phishing attacks. Our app provides advanced security features such as anti-phishing filters, real-time scanning of emails and websites for potential threats, and proactive alerts that notify users of potential

phishing attacks. Overall, our mobile app is a comprehensive security solution that caters to the needs of both basic and advanced mobile phone users. By providing robust security features and a user-friendly interface, we aim to make mobile phone usage safer and more secure for all users.

We plan to offer our users two different subscription packages for our mobile app.

The first package is a basic subscription, which we plan to charge an annual fee of \$5. This package provides users with the option to upload fifty screenshots per month and 50 emails per month. The second package is our advanced subscription, which includes all the features of the basic package, but an unlimited number of scans. We plan to charge an annual fee of \$10 for this package. We believe that both subscription packages offer great value to users who are concerned about their online security. The basic package provides essential security features that can help protect users from phishing attacks, while the advanced package offers more robust security features for those who require additional protection.

We also understand the importance of transparency when it comes to subscription pricing. Therefore, we will clearly outline the pricing and features of each package on our app's website and within the app itself. We aim to provide our users with the information they need to make an informed decision about which subscription package is right for them.

5 TESTING & IMPLEMENTATION RESULTS & DISCUSSION

The trained model's performance is evaluated on the testing dataset. Predictions are made using **berNB.predict()**. Several performance metrics, such as accuracy and confusion matrix, are calculated to assess the model's effectiveness in identifying phishing emails. The confusion matrix is printed using **confusion_matrix()**. To evaluate the model's performance, it is tested on the held-out testing data. The code computes several metrics to gauge the model's effectiveness. These metrics include the confusion matrix and accuracy score. The confusion matrix provides insight into

the model's ability to correctly classify safe and phishing emails. The accuracy score quantifies the overall correctness of the model's predictions.

4.2 Testing and Implementation Results

The screenshot shows a Jupyter Notebook cell titled 'Evaluation'. The code in the cell prints the confusion matrix and the accuracy score. The output shows a confusion matrix with values 1636, 31, 99, and 2619, and an accuracy score of 0.9703534777651083.

```

print('confusion matrices :')
print(confusion_matrix(y_test, y_predict), end='\n\n')

print('accuracy ', accuracy_score(y_test, y_predict))

```

Output:

```

... confusion matrices :
[[1636  31]
 [ 99 2619]]

accuracy 0.9703534777651083

```

Fig. 5.1 Accuracy(Bernoulli Naive Bayes)

Table 5.1: Accuracy of each model

Model	Accuracy
SVM	66.67%
Multinomial Naive Bayes	80%
Bernoulli Naive Bayes	97%

1. **Data Preprocessing:** The code effectively handles missing values and duplicates, ensuring data quality. The sentiment analysis adds a valuable feature that can potentially improve model performance by capturing the emotional context of emails.
2. **Data Visualization:** The countplot visually demonstrates the class distribution. This visualization is vital for understanding class balance, which can impact model training and reveal potential biases in the dataset.

3. **Train-Test Split:** The use of the `train_test_split` function ensures a fair evaluation of the model. However, it's essential to consider whether the split ratio is optimal for the dataset's size and characteristics.
4. **Feature Extraction:** TF-IDF vectorization is a robust choice for text-based data. It effectively converts textual information into numerical features. The application of `SelectPercentile` for feature selection enhances the model's interpretability and potentially reduces overfitting.
5. **Model Training and Evaluation:**
 - **Model Selection:** The decision to employ a Bernoulli Naive Bayes classifier aligns with the characteristics of the dataset. However, it's crucial to explore other algorithms and evaluate their performance for a comprehensive analysis.
 - **Model Saving and Loading:** The code demonstrates the ability to save and load the trained model, which is essential for deploying the system in real-world applications.
6. **Performance Metrics:**
 - **Confusion Matrix:** The confusion matrix is a valuable tool for understanding the model's behavior. It provides insights into false positives, false negatives, true positives, and true negatives. These metrics are crucial for assessing the practical implications of the system.
 - **Accuracy Score:** The accuracy score quantifies the model's correctness. However, it's important to consider the balance between precision and recall, as high accuracy may not necessarily indicate a robust phishing email detection system.

4.3 Discussion

1. **Model Selection:** While the Bernoulli Naive Bayes classifier is a reasonable choice, exploring other algorithms such as Random Forest, Support Vector Machines, or deep learning models like Recurrent Neural Networks (RNNs) and Transformers may yield improved results. Comparative analysis of multiple models is essential for making an informed choice.
2. **Hyperparameter Tuning:** The code does not perform hyperparameter tuning for the selected model. Hyperparameter optimization techniques like grid search or random search can fine-tune model performance and enhance accuracy.
3. **Class Imbalance:** Addressing class imbalance is critical in phishing email detection. Depending on the dataset's characteristics, techniques like oversampling, undersampling, or using synthetic data generation methods like SMOTE (Synthetic Minority Over-sampling Technique) may be necessary to balance the classes.
4. **Cross-Validation:** To obtain a more robust estimate of model performance, k-fold cross-validation should be considered. This technique assesses how well the model generalizes to various subsets of the data.
5. **Feature Engineering:** Feature engineering can significantly impact model performance. Experimentation with different text preprocessing techniques, such as stemming, lemmatization, or handling special characters and URLs, can be explored.
6. **Interpretability:** While machine learning models can achieve high accuracy, their interpretability is often limited. Employing explainable AI techniques, such as LIME (Local Interpretable Model-Agnostic Explanations) or SHAP (SHapley Additive exPlanations), can provide insights into model predictions.

7. **Real-world Application:** The practical implications of the system should be considered. In a real-world scenario, false positives (safe emails classified as phishing) and false negatives (phishing emails classified as safe) can have significant consequences. Decision thresholds should be carefully chosen to balance these errors based on the specific use case.
8. **Scalability:** The code's scalability should be evaluated. As the volume of emails increases, the system's ability to process and classify emails efficiently becomes crucial.

5 CONCLUSION

Phishing emails continue to be a significant cybersecurity threat, with attackers becoming increasingly sophisticated. This research paper delves into the realm of phishing email detection, focusing on the integration of sentiment analysis with a Bernoulli Naive Bayes model. The study explores the rationale behind this approach, the methodology employed, the experimental results, and their implications for enhancing the accuracy and reliability of phishing email detection systems.

Phishing attacks are among the most prevalent and damaging cyber threats faced by individuals and organizations worldwide. These attacks involve deceptive emails that lure recipients into revealing sensitive information or downloading malicious content. As phishing techniques evolve, there is a growing need for more robust and sophisticated methods to detect and thwart such attacks. This research paper explores the integration of sentiment analysis into phishing email detection, leveraging machine learning techniques. Sentiment analysis, a natural language processing (NLP) approach, assesses the emotional tone and subjective information within text data. By combining sentiment analysis with the Bernoulli Naive Bayes model, we aim to enhance the accuracy and efficiency of phishing email detection systems.

The findings of this research paper highlight the potential of sentiment analysis as a complementary tool in phishing email detection. By considering the emotional

context of email text, the system demonstrates enhanced accuracy, precision, and overall performance. This research paper presents a novel approach to phishing email detection by integrating sentiment analysis with the Bernoulli Naive Bayes model. The results demonstrate the potential of this approach in enhancing the accuracy and precision of phishing detection systems. By considering the emotional context of email text, the system showcases improved performance in distinguishing phishing emails from legitimate ones.

While the integrated system exhibits promising results, ongoing research and development efforts are necessary to address evolving phishing techniques and enhance its robustness. The practical deployment of such systems within organizations holds significant potential for bolstering cybersecurity and protecting sensitive information from phishing threats. As the cybersecurity landscape continues to evolve, innovative approaches like the one presented in this research paper play a vital role in staying one step ahead of cyber adversaries, ultimately safeguarding individuals and organizations from the perils of phishing attacks.

6 REFERENCES

- [1] APWG, "PHISHING ACTIVITY TRENDS REPORT," APWG, 2022.
- [2] 2. N. 3. B. a. 4. J. W. 1*Sikha Bagui, "Machine Learning and Deep Learning for Phishing Email," *Journal of Computer Science*, p. 14, 2021.
- [3] T. G. S. V. Said Sallouma*, "Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey," in *Procedia Computer Science*, Manchester, 2021.
- [4] I. V. D. M. J. M. G. H. U. Z. ENAITZ EZPELETA, "Novel email spam detection method using sentiment analysis and personality ecognition," *Oxford University Press*, p. 12, 2020.
- [5] M. D. a. T. Viana, "Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails," *applied system innovation*, p. 19, 2022.
- [6] R. V. a. N. Hossain, "Semantic Feature Selection for Text with Application to

Phishing Email Detection," Semantic Scholar, Texas, 2013.

- [7] S. F. F. P. Luisa Franchina, "Detecting phishing e-mails using Text Mining and features analysis," in *Italian Conference on CyberSecurity*, 2021.
- [8] B. R. A. S. S. M. Srishti Rawal, "Phishing Detection in E-mails using Machine Learning," *International Journal of Applied Information Systems (IJ AIS)*, p. 5, 2017.
- [9] C. Z. C. H. ., L. L. A. Y. Y. YONG FANG, "Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism," *IEEE*, p. 12, 2019.
- [10] A. Y. a. A. Abuhasan, "AN INTELLIGENT CLASSIFICATION MODEL FOR PHISHING EMAIL DETECTION," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 8, p. 18, 2016.
- [11] M. T. a. A. Paulauskaite-Taraseviciene, "Research on phishing email detection based on URL parameters using machine learning algorithms," *CEUR*, Kaunas, 2021.
- [12] M. A. a. M. A. Badawi, "Phishing Email Detection Using Machine Learning Techniques," *IJCSNS International Journal of Computer Science and Network Security*, vol. 22, p. 7, 2022.
- [13] R. V. Gal Egozi, "Phishing Email Detection Using Robust NLP Techniques," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, Texas, 2018.
- [14] M. F. a. E. Rolland, "Fundamentals of Sentiment Analysis and Its Applications," Springer International Publishing, Merced, 2016.
- [15] T. L. a. G. Xu, "Sentiment Analysis," Springer Science+Business Media, New York, 2013.
- [16] D. J. a. J. H. Martin, *Speech and Language Processing*, 2023.
- [17] R. Kibble, *Introduction to natural language processing*, London: University of London , 2013.
- [18] V. I. R. Berwick, "An Idiot's guide to Support vector machines (SVMs)".

7 APPENDIX

final 2

ORIGINALITY REPORT

15%

SIMILARITY INDEX

11%

INTERNET SOURCES

4%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Sri Lanka Institute of
Information Technology

Student Paper

2%

2

Submitted to The University of
Wolverhampton

Student Paper

2%

3

Submitted to Liverpool John Moores
University

Student Paper

1%

4

www.coursehero.com

Internet Source

1%

5

Submitted to University of Sunderland

Student Paper

<1%

6

www.erepository.nara.ac.lk

Internet Source

<1%

7

www.ijais.org

Internet Source

<1%

8

Submitted to Universiti Tenaga Nasional

Student Paper

<1%