Received: 07-08-2025
Accepted: 17-09-2025

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Detecting Email Phishing Using Machine Learning Algorithms

[1] Samuel M Orokpo, [2] Oluwasegun Ishaya Adelaiye, [3] Adamu Suliaman Usman
[1] 14 lugsmore Lane, Saint Helens, England, United Kingdom
[2] Department of Cybersecurity, Bingham University, Abuja, 961105, Nigeria
[3] Department of Computer Science, Bingham University, Abuja, 961105, Nigeria

Corresponding Author: **Oluwasegun Ishaya Adelaiye**

Abstract

Email phishing remains a persistent cybersecurity threat, undermining communication systems and exploiting unsuspecting users. This study proposes a machine learning based approach to detect phishing emails using the CRISP-DM methodology. A dataset of 18,650 emails obtained from Kaggle was analyzed with multiple algorithms, including Naïve Bayes and Random Forest, to evaluate their effectiveness in distinguishing between legitimate and malicious messages. Performance was assessed using accuracy, precision, recall, and F1-score. The Random Forest model achieved the highest performance with 98.30% accuracy, 0.96 precision, 1.00 recall, and an F1-score of 0.90. In contrast, Naïve Bayes produced a lower accuracy of 53.80% but achieved perfect recall, highlighting the importance of comparing algorithms to uncover their respective strengths and limitations. Future work will focus on refining the proposed solution and extending its applicability to address region-specific challenges, particularly in Nigeria, where phishing scams such as "yahoo yahoo" are increasingly prevalent. This research contributes to the advancement of robust email security strategies against evolving phishing threats.

Keywords: Phishing Email, Anomaly Detection, Machine Learning, Naïve Bayes, Random Forest, Email Filtering, CRISP-DM, Email Classification

1. Introduction

Email phishing, a prevalent form of cyber threat, involves deceptive practices where attackers use fraudulent emails to trick individuals into divulging sensitive information, such as login credentials, financial details, or personal data (Baki and Verma 2022) [7]. The term "phishing" draws its roots from the analogy of fishing, where attackers cast a wide net in the hope of luring unsuspecting victims. These phishing emails often mimic legitimate communication from reputable sources, creating a false sense of trust and urgency. The attackers employ various tactics, including social engineering, to manipulate recipients into taking actions that serve the malicious objectives, such as clicking on malicious links or downloading infected attachments. Phishing attacks have become increasingly sophisticated over time, adapting to technological advancements and exploiting the human element which is the weakest link in cybersecurity (Desolda *et al*. 2022; Pandey, Singh and Pal 2023) [13, 31].

The success of email phishing attacks is often attributed to the fact that they exploit common human vulnerabilities, such as trust and curiosity. Cybercriminals meticulously craft emails to appear genuine, utilizing familiar logos, sender names, and language to deceive recipients. Phishing attacks are versatile, ranging from broad-scale campaigns targeting a large number of users to highly targeted spear-phishing attacks tailored for specific individuals or organizations. The consequences of falling victim to email phishing can be severe, leading to financial losses, identity theft, unauthorized access to sensitive information, and even the compromise of entire systems. As a result, combating email phishing has become a critical aspect of cybersecurity, demanding ongoing research and technological advancements to stay ahead of the evolving tactics employed by malicious actors (Desolda *et al*. 2022) [13].

This paper leverages a machine learning approach to enhance the detection of email phishing attempts. Traditional rule-based systems often struggle to keep pace with the evolving tactics employed by phishing attackers. The integration of machine learning techniques aims to address these challenges by training models to discern patterns and characteristics indicative of phishing behaviour. By utilizing historical data and applying various algorithms, the system can learn to identify subtle nuances in emails that may escape conventional rule-based filters.

The contributions to knowledge from this paper is that the results indicate the superior performance of Random Forest in comparison to Naïve Bayes. It is noteworthy that the study by Rashid, Mahmood, and Nisar (2020) [33] reported an accuracy of 95.66%, but specific values for precision, recall, and F1-Score are not available. The contrasting outcomes emphasize the importance of choosing a robust algorithm, with Random Forest emerging as a promising solution for effective email phishing detection.

The motivation for studying email phishing detection in Nigeria is rooted in the need to mitigate the impact of these scams on individuals and businesses. By developing effective and tailored phishing detection mechanisms, it becomes possible to enhance cybersecurity defences, protect personal and financial information, and ultimately contribute to the overall digital resilience of the Nigerian population.

2. Literature Review

Phishing is a rampant cyber-attack technique that relies on social engineering to trick individuals into revealing sensitive information. This deceptive tactic often involves cybercriminals masquerading as trusted entities, such as legitimate websites, well-known companies, or government agencies. Their primary objective is to obtain confidential data like passwords, credit card numbers, and personal or financial details (AL-Otaibi and Alsuwat 2020) [4].

One common form of phishing is email phishing, where attackers send deceptive emails that appear to originate from reputable sources to gain unauthorized access to sensitive data or install a malware (Baki and Verma 2022) [7]. These emails may contain links to fraudulent websites or malicious attachments designed to deceive recipients into sharing their sensitive information. A more targeted version of this is spear phishing, where attackers customize their messages to specific individuals or organizations, making them more convincing and tailored (Aleroud, Abu-Shanab and Al-Aiad 2020) [2].

A recently modern trend in recent years was the emergence of Phishing-as-a-service (Phaas) on the dark web, marking yet another notable shift in the world of phishing attacks. This commercialization of phishing enabled even less technically skilled individuals to engage in these deceptive practices (Brunken, Buckmann and Hielscher 2023; Sabo, Black and Sarno 2023) [10, 34]. Simultaneously, the 2010s witnessed the rise of Business Email Compromise (BEC) attacks, which posed a particularly severe threat to businesses. In these sophisticated schemes, attackers frequently impersonated high-level executives or trusted vendors to extract sensitive information or funds. This tactic, commonly referred to as CEO fraud, relied on the psychological manipulation of employees, convincing them to take actions that compromised their companies (Cross and Gillett 2020; Al-Musib, Al-Serhani and Humayun 2021) [12, 3].

Phishing remains an enduring and pervasive cybersecurity threat. However, there exist a range of effective mitigation methods and best practices that both individuals and organizations can implement to fortify their defenses against these deceptive attacks. User education and training, Advanced email filtering solutions, multi-factor authentication (MFA), Dedicated anti-phishing software, ensuring secure website, implementing email authentication protocols, Regular software updates, phishing simulations, and reporting mechanisms contribute to a comprehensive defense strategy (Morakinyo 2021; Niamathulla and Bhalothia 2022; Klint 2023) [26, 29, 22]. The details of a few popular existing mitigation techniques are presented in the following subsections.

Numerous researchers have dedicated extensive efforts to devise solutions addressing the security vulnerabilities linked with phishing attacks. The realm of security challenges has grown more complex as attackers continually develop increasingly sophisticated methods. Phishing attacks have endured for well over a decade and continue to yield varying degrees of success.

Tan *et al.* (2023) [36], introduced a hybrid method that combines visual and textual identity cues, resulting in an impressive accuracy of 98.6%. This approach excels in reducing false positives and is a promising technique for identifying phishing websites. However, it relies on visual elements like logos, potentially limiting its applicability to websites lacking such visuals, and faces challenges in handling dynamically generated or non-English character-based content.

Kumar *et al.* (2023) [23] utilized a different route by focusing on features extracted from TLS 1.2 and TLS 1.3 traffic for phishing detection, achieving accuracy rates ranging from 93.63% to 95.40%. The approach is effective when dealing with encrypted traffic but might experience reduced performance when HTTPS encryption is not in use.

Catal *et al.* (2022) [11], conducted a comprehensive review of supervised deep learning models, providing a valuable overview of existing techniques. However, a significant portion of the studies they analyzed lacked feature selection algorithms, which could impact their efficiency and generalizability. The reliance on the same dataset across most studies raises concerns about the representativeness of real-world scenarios.

Zhu *et al.* (2023) [45] in their work, introduce a lightweight model combining CNN, BiLSTM, and attention mechanisms, yielding impressive results with an accuracy of up to 99.02%. Despite the strong performance, this approach employs a fixed population size and may introduce computational constraints when executed 100 times. The paper also focused primarily on accuracy and recall metrics may limit the model's comprehensive evaluation.

Obaid *et al.* (2021) [30] work, a Random Forest algorithm is utilized to classify and detect phishing sites, achieving an impressive accuracy of 96.91%. This method outperforms existing machine learning-based models, but it is important to conduct a more in-depth analysis, particularly concerning false positives and false negatives, in addition to focusing solely on accuracy and error rates.

Rashid, Mahmood and Nisar (2020) [33] adopted a Support Vector Machine (SVM) classifier, achieving an accuracy of 95.66% in distinguishing between phishing and legitimate websites while using just 22.5% of the functionality. This approach excels in performance but also requires a more detailed examination of false positives and false negatives. Like the previous approach, it places a significant emphasis on accuracy without considering other critical metrics such as precision, recall, or F1 score.

These approaches display varying strengths and limitations in the field of phishing detection. While many excel in accuracy, they should conduct more comprehensive evaluations, validations and consider a broader set of metrics to ensure robust and reliable performance in real-world scenarios. Each approach offers unique insights and methodologies, underlining the importance of selecting the most suitable technique for specific phishing detection needs.

3. Methodology

The CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology outlines a flexible, six-phase framework to guide data mining and analytics projects. Developed through an industry consortium, it provides a business-focused, iterative approach to ensure initiatives extract true value from data. The CRISP-DM methodology to be used in this project is presented diagrammatically in Fig 1.

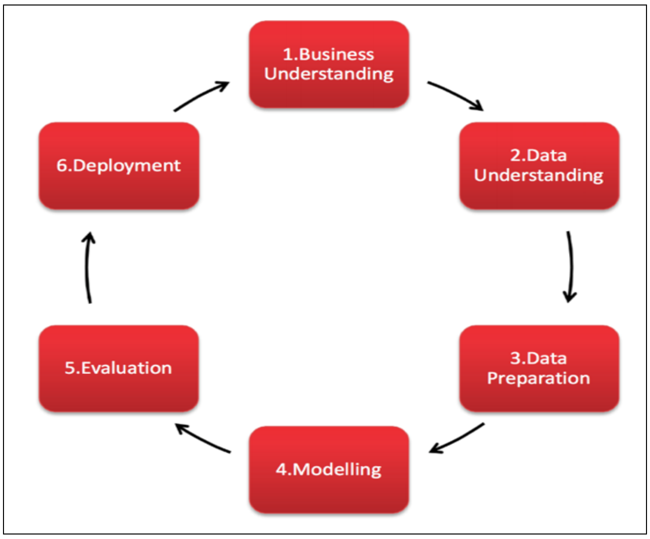


Fig 1: CRISP DM Methodology

Modelling encompasses selecting, testing, and optimizing analytical modelling approaches to uncover patterns and insights within prepared data. Guided by business needs outlined earlier, modelling requires an iterative process of refinement to achieve optimal analytical performance.

The Evaluation phase specifically assesses model performance and alignment with defined objectives. The modelling process and outputs are reviewed holistically to determine their effectiveness in meeting business needs. This stage serves as a critical go/no-go checkpoint within the project lifecycle.

Deployment, the final phase, operationalizes models and insights by embedding them within business processes to enhance decision-making. Planning the deployment approach precedes implementation, monitoring, and maintenance of the solution. This phase focuses on enabling adoption across the organization.

Spanning these six phases, CRISP-DM also emphasizes foundational principles like iteration, collaboration, and documentation. The nonlinear nature of the methodology allows revisiting previous stages to leverage new learnings. Tight collaboration between data scientists, analysts, and domain experts enhances context throughout. Comprehensive documentation enhances transparency, reproducibility, and oversight.

The CRISP-DM methodology delivers an adaptive framework to guide data mining efforts toward business impact. It provides a blueprint for extracting value from data assets by aligning analytical efforts with organizational objectives.

4. Model Training

The system implementation process follows the CRISP-DM Methodology, providing a structured and systematic approach to developing an email phishing detection system. Leveraging a dataset comprising 18,650 emails categorized as phishing email or safe email, the business understanding phase defines the problem of enhancing email security by accurately identifying and filtering out phishing emails. The data understanding phase explores the dataset, highlighting the email message content as the independent variable and the label indicating phishing or safe status as the dependent variable. In the subsequent data preparation phase, the dataset is cleaned, preprocessed, and split into training and testing sets, with the removal of a dummy column to mitigate noise. The modeling phase involves the selection of machine learning algorithms such as Naive Bayes, Support Vector Machines, and Random Forest for email classification, followed by the training of these models using the prepared dataset. The evaluation phase assesses model performance using metrics like accuracy and confusion matrix, leading to the deployment of the trained. The iterative nature of CRISP-DM allows for continuous refinement and improvement.

The implementation focuses on creating a robust and efficient email phishing detection system, emphasizing the importance of accurate classification to enhance email security. By following the CRISP-DM Methodology, the process ensures a comprehensive understanding of the data, effective model training, and rigorous evaluation. The implementation stages are presented below.

The process begins with uploading the dataset to R studio. This is achieved by using the following code.

```
data<read.csv("Phishing_Email.csv",fileEncoding='latin
1')
```

This code imports the dataset in a csv file and sets the file encoding to Latin so as to avoid encoding mismatch. Due to the processing capacity and system resource management the dataset is reduced from 18,650 rows to 500 rows. This is achieved using the following code.

```
data <- data[sample(nrow(data),500),] #Reduce dataset size to 500 due to low system resources
dim(data)#View dataset dimensions
```

Beginning with the conversion of text to lowercase, the code sequentially removes punctuation, numbers, and common English stop words. The stemming process further simplifies words to their root forms, contributing to text normalization. Whitespaces between words are then eliminated. This series of preprocessing steps transforms unstructured email text into a structured format, enhancing the suitability of the text for subsequent analyses, such as feature extraction and machine learning model training. The displayed corpus output provides a summarized view of the processed text, reflecting the impact of the applied NLP techniques on the dataset.

```
## <<VCorpus>>
## Metadata: corpus specific: 0, document level (indexed): 0
## Content: documents: 500
```

R code is then used to generate a Document-Term Matrix (dtm) from the pre-processed corpus using the DocumentTermMatrix function from the tm library. This matrix represents the frequency of terms (words) within each document (email) in the corpus. Each row corresponds to a document, and each column represents a unique term in the entire corpus. The values in the matrix indicate the frequency of each term within the respective documents. This dtm is a crucial step in the natural language processing pipeline, as it transforms the textual data into a structured format suitable for machine learning models, allowing for further analysis and feature extraction.

The information includes the count of non-sparse and sparse entries, indicating that the matrix is highly sparse, with a sparsity of 99%. The maximal term length in the matrix is specified as 2244, and the weighting scheme used is term frequency. Following this, the code removes sparse terms by eliminating those with a sparsity greater than 99.9% from the dtm using the removeSparseTerms function. The resulting dimensions of the modified dtm are displayed, revealing that the sparsity reduction has reduced the number of terms to 13,236 while retaining 500 documents in the matrix. This process helps mitigate the impact of overly common or rare terms, optimizing the matrix for subsequent analyses.

A function named convert_count that takes a numeric vector, converts values greater than 0 to 1 and assigns labels "No" and "Yes" using a factor with levels 0 and 1. Subsequently, the apply function is utilized to apply this conversion to each column of the Document-Term Matrix (dtm), resulting in a binary representation of term occurrences stored in the dataset. The next step involves converting this binary matrix into a data frame named data1. The code calculates the frequency of each term in the dtm using colSums and sorts the terms in decreasing order. The display of the least used words by showing the tail of the sorted frequency is presented, specifically the 10 terms with the lowest frequency in the corpus. This process aids in identifying and exploring the least common terms in the dataset, providing insights into potential outliers or unique aspects of the text data. The output is presented below.

```
## zudenken zum zunaechst zusammenhang
zussammen zustaendig
## 1 1 1 1 1 1
## zweck zweigenbaum zymg zzn
## 1 1 1 1
```

Using ggplot2 library to create a bar graph to visualize the frequency of words in the dataset. The code first constructs a data frame and then filters the data to include only words with frequencies exceeding 150. The resulting data frame is used to create a bar graph where the x-axis represents words reordered by frequency in decreasing order, and the y-axis represents the frequency of each word. This visualization presented in Fig 2 aids in identifying and exploring the most frequently occurring words in the dataset, offering insights into potential keywords or significant terms within the corpus.

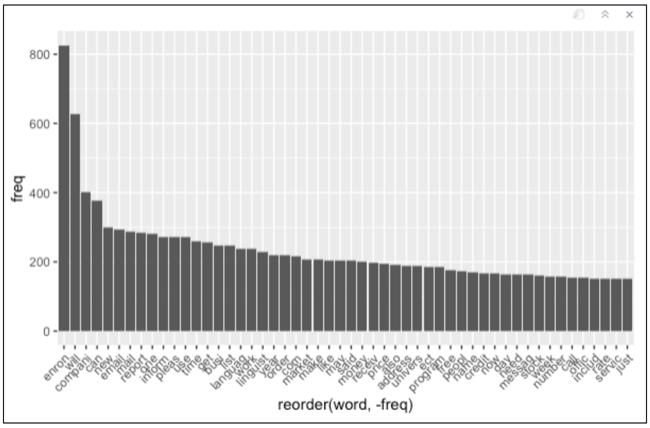


Fig 2: Bar graph Showing Word Frequency in Data frame

The intriguing observation that enron emerges both as the most and least used word in the dataset, with a frequency range of 150 words and above, suggests a nuanced pattern in its occurrence. This duality could stem from specific documents where enron plays a central role, contributing significantly to the high-frequency category. Conversely, in documents where it appears infrequently, it contributes to the least used category. The discrepancy might be influenced by factors such as document-specific usage, document length, and the contextual significance of enron within the dataset.

Incorporating rigorous data pre-processing techniques, the target column, Email.Type, was appended to the derived dataset, denoted as data1. This augmentation follows the meticulous breakdown of email messages into their simplest and most meaningful structural format, as accomplished in the preceding code. The Email.Type column serves as a critical label, distinguishing between phishing and safe emails, thus providing a categorical context to the processed feature set. To gain insight into the structural attributes of the appended column, the str(data1$Email.Type) function was employed, shedding light on its data type and overall composition. This meticulous fusion of processed features and corresponding labels fortifies the dataset for subsequent analyses, such as machine learning model training and performance evaluation, contributing to the robustness of the investigative framework.

```
data1$Email.Type = data$Email.Type #duplicate label
and attach to created data1 dataframe
str(data1$Email.Type)#show the structure
```

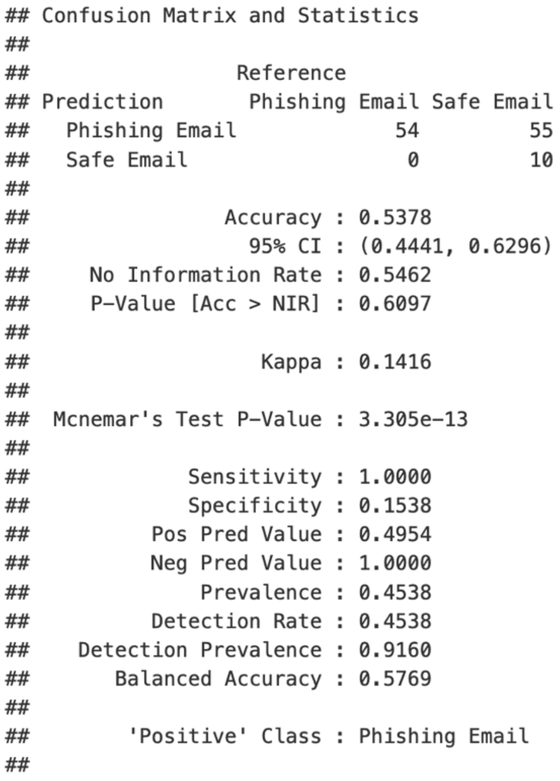
In preparing the data for training the dataset is divided into two parts 75% for training and 25% for testing

```
set.seed(123) #set seed to be able to track process
split = sample(2,nrow(data1),prob = c(0.75,0.25),replace
= TRUE) #Split dataset using 25% for testing and 75%
for training
train_set = data1[split == 1,] #store training data in
train_set
test_set = data1[split == 2,] #store testing data in test_set
```

In the preparation phase for model training, an essential step involves partitioning the dataset into distinct subsets allocated for training and testing purposes. To ensure reproducibility and traceability of the process, the random seed was set to 123. Employing a stratified sampling approach, the dataset was systematically divided into two segments: 75% for training and 25% for testing. This division was accomplished using the sample function, assigning probabilities of 0.75 and 0.25 to the respective subsets, while allowing for replacement to ensure a robust representation of the data. The training data was isolated and stored in the variable train_set, while the testing data found its place in the variable test_set. This strategic partitioning lays the foundation for the subsequent stages of model development and evaluation, enabling a comprehensive assessment of the performance of the model on unseen data.

In building a predictive model for email type classification, the Naïve Bayes algorithm was employed, leveraging the caret library for efficient model development and assessment. The model fitting process, executed within a repeated cross-validation framework repeated 10 times with 3 repeats, exhibited efficiency with a total elapsed time of approximately 8.41 seconds. Subsequently, the trained model was applied to the designated test set, and the resulting predictions were evaluated using a confusion matrix.

The confusion matrix provides a comprehensive overview of the performance of the model, illustrating the number of correctly and incorrectly classified instances. In this context, the matrix reveals that the model correctly identified 54 instances of phishing emails and 10 instances of safe emails. However, it also misclassified 55 phishing emails as safe and did not identify any safe emails as phishing see Fig 3.



```
## Confusion Matrix and Statistics
##
##                    Reference
## Prediction          Phishing Email Safe Email
##   Phishing Email                54         55
##   Safe Email                     0         10
##
##                  Accuracy : 0.5378
##                    95% CI : (0.4441, 0.6296)
##       No Information Rate : 0.5462
##       P-Value [Acc > NIR] : 0.6097
##
##                     Kappa : 0.1416
##
##  Mcnemar's Test P-Value : 3.305e-13
##
##               Sensitivity : 1.0000
##               Specificity : 0.1538
##            Pos Pred Value : 0.4954
##            Neg Pred Value : 1.0000
##                Prevalence : 0.4538
##            Detection Rate : 0.4538
##      Detection Prevalence : 0.9160
##         Balanced Accuracy : 0.5769
##
##          'Positive' Class : Phishing Email
##
```

Fig 3: Training Results for Naive Bayes

The overall accuracy of the model stands at 53.78%, with a sensitivity of 100% for phishing emails and a specificity of 15.38% for safe emails. The Kappa statistic, measuring the agreement between predicted and actual classifications, is computed at 0.1416, suggesting a slight level of agreement. The results further indicate that the model tends to be more sensitive in detecting phishing emails than specific in identifying safe emails, as evidenced by a higher sensitivity and lower specificity. The model is further visualized on a confusion matrix as presented in Fig 4.

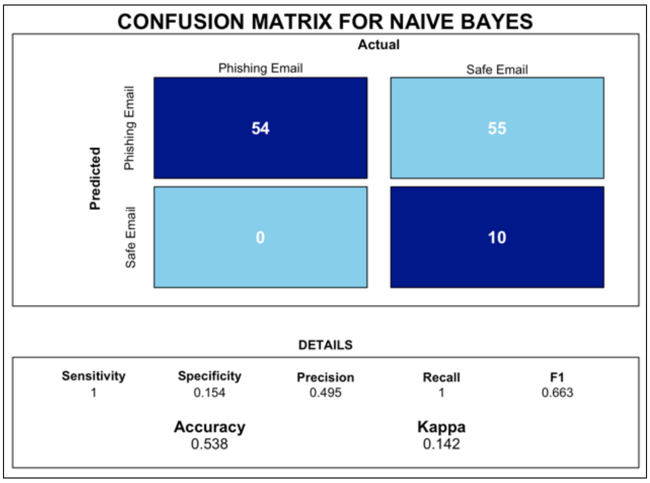
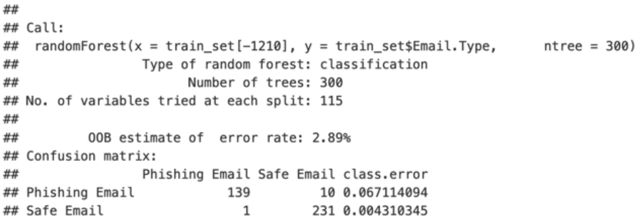


Fig 4: Graphical Results for Naive Bayes

The second algorithm, Random Forest was applied to construct a classification model, utilizing a forest ensemble comprising 300 trees. The trained random forest classifier exhibited high efficiency, achieving an out-of-bag (OOB) estimate of the error rate at approximately 2.89%. The confusion matrix derived from the training results showcases the accuracy of the model in classifying instances into phishing emails and safe emails. The model demonstrated a class error of 0.0671 for phishing emails and 0.0043 for safe emails.

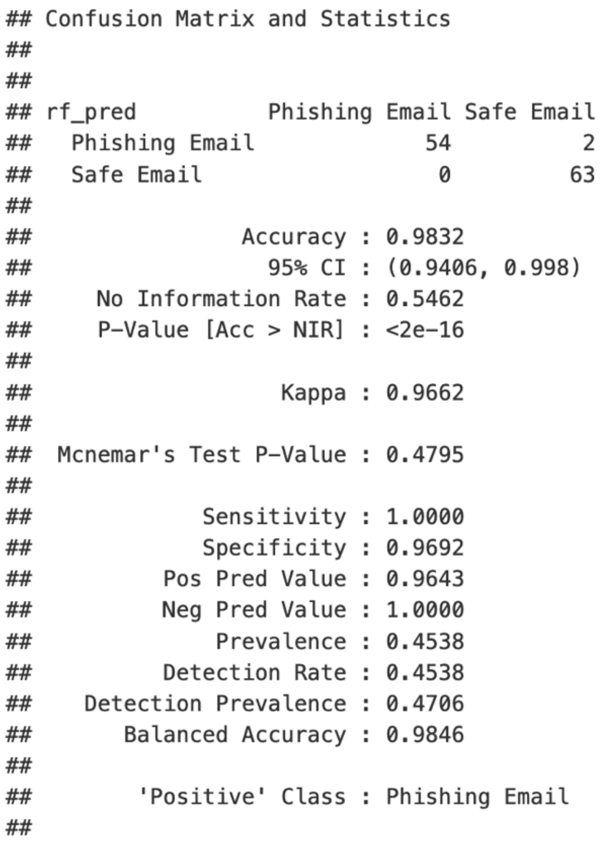
Upon applying the trained random forest model to the test set, the confusion matrix and associated statistics unveiled impressive performance metrics. The model accurately identified 54 instances of phishing emails and 63 instances of safe emails, with zero misclassifications for both categories. The overall accuracy of the random forest model stands at 98.32%, with a sensitivity of 100% for phishing emails and a specificity of 96.92% for safe emails. The Kappa statistic, reflecting the agreement between predicted and actual classifications, is high at 0.9662, indicating a high level of agreement.

These results underscore the efficacy of the random forest model in accurately classifying email types, demonstrating a substantial improvement over the previous Naive Bayes approach. The high accuracy, balanced sensitivity and specificity, and minimal class errors points our approach as a promising tool for email classification tasks within the context of phishing detection. The results are presented in Figure 5 and 6.



```
##
## Call:
##  randomForest(x = train_set[-1210], y = train_set$Email.Type,      ntree = 300)
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 115
##
##         OOB estimate of  error rate: 2.89%
## Confusion matrix:
##                Phishing Email Safe Email class.error
## Phishing Email            139         10 0.067114094
## Safe Email                  1        231 0.004310345
```

Fig 5: Training Results for Random Forest



```
## Confusion Matrix and Statistics
##
##
## rf_pred          Phishing Email Safe Email
##   Phishing Email             54          2
##   Safe Email                  0         63
##
##                Accuracy : 0.9832
##                  95% CI : (0.9406, 0.998)
##     No Information Rate : 0.5462
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9662
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.9692
##          Pos Pred Value : 0.9643
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4538
##          Detection Rate : 0.4538
##    Detection Prevalence : 0.4706
##       Balanced Accuracy : 0.9846
##
##        'Positive' Class : Phishing Email
##
```

Fig 6: More Training Results for Random Forest

The function draw_confusion_matrix was created to visualize the confusion matrix and associated result report for the random forest model. The function utilizes the base graphics system in R to generate a graphical representation of the confusion matrix. The layout is divided into two parts: the confusion matrix itself and additional details regarding precision, recall, F1-score, and accuracy. The rectangles within the matrix represent the predicted and actual classifications for phishing emails and safe emails. The color-coded rectangles enhance the visual representation, where blue represents phishing emails, and light blue represents safe emails as presented in Fig 7.

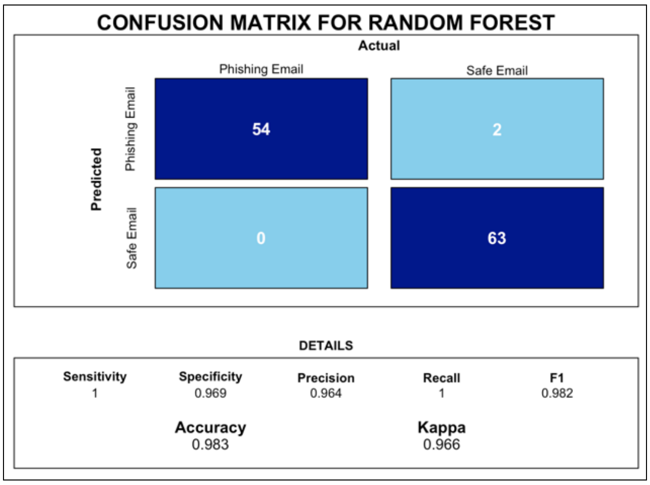


Fig 7: Graphical Results for Random Forest

The numerical values within the rectangles correspond to the counts of instances in each category. The details section provides information on precision, recall, F1-score, and accuracy for both phishing and safe email classifications. The graphical representation aims to offer a comprehensive overview of the performance of the model, aiding in the interpretation of classification results.

5. Results

The pervasive nature of email spamming, evolving from a mere nuisance to a significant threat in the interconnected digital landscape, underscores the need for effective email phishing detection mechanisms. This project delves into the realm of machine learning algorithms, specifically Naive Bayes and Random Forest, to tackle the challenge of identifying and categorizing phishing emails. Grounded in the CRISP DM methodology, the study explores a dataset containing 18,650 emails categorized as phishing or safe, employing a natural language processing pipeline for data pre-processing. This section evaluates the performance of the chosen algorithms against the backdrop of existing research, emphasizing accuracy, sensitivity, specificity, and the intricacies of confusion matrices. The objective is not only to showcase the efficacy of the proposed approach but also to contribute valuable insights and advancements to the field of email phishing detection, addressing the escalating sophistication of spam tactics in the contemporary digital landscape.

Table 1: Evaluation of Model Results

Algorithm	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	53.80%	0.50	1.00	0.663
Random Forest	98.30%	0.96	1.00	0.9
Rashid, Mahmood and Nisar (2020) [33]	95.66%	NA	NA	NA

The evaluation of the machine learning algorithms, Naïve Bayes and Random Forest for email phishing detection yields distinct performance metrics as presented in Table 6.1. Naïve Bayes exhibits an accuracy of 53.80%, with precision at 0.50, recall at 1.00, and an F1-Score of 0.663. In contrast, Random Forest demonstrates a significantly higher accuracy of 98.30%, with precision reaching 0.96, recall at 1.00, and an impressive F1-Score of 0.9. These results indicate the superior performance of Random Forest in comparison to Naïve Bayes. It is noteworthy that the study by Rashid, Mahmood, and Nisar (2020) [33] reported an accuracy of 95.66%, but specific values for precision, recall, and F1-Score are not available. The contrasting outcomes emphasize the importance of choosing a robust algorithm, with Random Forest emerging as a promising solution for effective email phishing detection.

6. Conclusion

Phishing attacks, especially in the form of email phishing, continue to pose a significant threat to individuals and organizations, as attackers employ increasingly sophisticated techniques to deceive users. Recognizing the limitations of traditional rule-based systems, this project explores the machine learning to develop a robust solution for phishing attack detection in emails. Spearheaded by a comprehensive literature review on existing machine learning techniques, the project aimed to address the dynamic nature of these attacks by leveraging labelled datasets, evaluating various classification algorithms, and utilizing key metrics like accuracy, precision, recall, and F1-score for performance assessment.

The CRISP-DM methodology which was adopted facilitates a structured and iterative approach, ensuring that the project aligns with business goals and extracts maximum value from data. The results obtained from the evaluation showcase the high accuracy and effectiveness of the Random Forest algorithm in detecting phishing emails, with notable precision and F1-score values.

The proposed machine learning model emerges as a promising solution in the ongoing battle against phishing attacks, showcasing the potential to significantly reduce risks across diverse contexts. The continuous adaptation of the model to evolving threats and the emphasis on interpretability underscore its value as a scalable and robust defence mechanism.

7. References

1. Abdul M, *et al*. Analysis the Types and Impacts of Phishing Attacks on Internet Users. Journal of Global Business and Social Entrepreneurship. 2023; 9(26) [online]. Available from: http://www.gbse.my/V9%20NO.26%20(JAN%202023)/Paper-325-.pdf [Accessed 5 Nov 2023].
2. Aleroud A, Abu-Shanab E, Al-Aiad A. An examination of susceptibility to spear phishing cyber attacks in non-English speaking communities. Journal of Information Security and Applications. 2020; 55:p102614. Available from: https://www.sciencedirect.com/science/article/pii/S2214212620307791 [Accessed 5 Nov 2023].
3. Al-Musib N, Al-Serhani F, Humayun M. Business email compromise (BEC) attacks. Materials Today: Proceedings, 2021 [online]. Available from: https://www.sciencedirect.com/science/article/pii/S2214785321027425 [Accessed 5 Nov 2023].
4. Al-Otaibi A, Alsuwat E. A study on social engineering attacks: Phishing attack. Int. J. Recent Adv. Multidiscip. 2020; 7(11):6374-6380. Available from: http://ijramr.com/sites/default/files/issues-pdf/3407.pdf [Accessed 5 Nov 2023].
5. Atimorathanna D, Ranaweera T. NoFish; Total anti-phishing protection system. In: 2020 2nd International Conference on Advancements in Computing (ICAC). IEEE, 2020, 470-475. Available from: https://ieeexplore.ieee.org/abstract/document/9357145/ [Accessed 8 Nov 2023].
6. Bakalo V. Phishing as the most Common Cyber Threat. Тези Доповідей, 2023, p33. Available from: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/12768/1/%D0%97%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA_%D1%82%D0%B5%D0%B7_2023%20%282%29.pdf#page=33 [Accessed 5 Nov 2023].
7. Baki S, Verma R. Sixteen Years of Phishing User Studies: What Have We Learned? IEEE Transactions on Dependable and Secure Computing. 2022; 20(2):1200-1212. Available from: https://ieeexplore.ieee.org/abstract/document/9713733/ [Accessed 5 Nov 2023].
8. Bell S, Komisarczuk P. An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. ACM International Conference Proceeding Series, 2020.
9. Bilogrevic I, *et al*. A machine-learning based approach to privacy-aware information-sharing in mobile social networks. Pervasive and Mobile Computing. 2016; 25:125-142. Available from: https://www.sciencedirect.com/science/article/pii/S1574119215000231 [Accessed 20 Dec 2023].
10. Brunken L, Buckmann A, Hielscher J. {“To} Do This Properly, You Need More {Resources”}: The Hidden Costs of Introducing Simulated Phishing Campaigns. In 32nd USENIX Security Symposium (USENIX Security 23), 2023, 4105-4122. Available from: https://www.usenix.org/conference/usenixsecurity23/presentation/brunken [Accessed 5 Nov 2023].
11. Catal C, *et al*. Applications of deep learning for phishing detection: A systematic literature review. Knowledge and Information Systems. 2022; 64(6):1457-1500.
12. Cross C, Gillett R. Exploiting trust for financial gain: An overview of business email compromise (BEC) fraud. Journal of Financial Crime. 2020; 27(3):871-884.
13. Desolda G, *et al*. Human factors in phishing attacks: A systematic literature review. ACM Computing Surveys (CSUR). 2022; 54(8):1-35. Available from: https://dl.acm.org/doi/abs/10.1145/3469886 [Accessed 20 Dec 2023].
14. Falade P. Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks. Arxiv.Org. 2023; 9(5):185-198. Available from: https://arxiv.org/abs/2310.05595 [Accessed 7 Nov 2023].
15. Furnell S, Millet K, Papadaki M. Fifteen years of phishing: Can technology save us? Computer Fraud & Security. 2019; 7:11-16. Available from: https://www.magonlinelibrary.com/doi/abs/10.1016/S1361-3723%2819%2930074-0 [Accessed 7 Nov 2023].
16. Gangavarapu T, Jaidhar C, Chanduka B. Applicability of machine learning in spam and phishing email filtering: Review and approaches. Artificial Intelligence Review. 2020; 53:5019-5081. Available from: https://link.springer.com/article/10.1007/s10462-020-09814-9 [Accessed 7 Nov 2023].
17. Ghazi-Tehrani A, Pontell H. Phishing evolves: Analyzing the enduring cybercrime. The New Technology of Financial Crime, 2022, 35-61. Available from: https://www.taylorfrancis.com/chapters/edit/10.4324/9781003258100-3/phishing-evolves-analyzing-enduring-cybercrime-adam-kavon-ghazi-tehrani-henry-pontell [Accessed 5 Nov 2023].
18. Governance RG-R. undefined, 2020. Comparing definitions of data and information in data protection law and machine learning: A useful way forward to meaningfully regulate algorithms? Wiley Online Library. 2022; 16(1):156-176. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/rego.12349 [Accessed 20 Dec 2023].
19. Hr MG, *et al*. Development of anti-phishing browser based on random forest and rule of extraction framework. Cybersecurity. 2020; 3(1).
20. Jain AK, Gupta BB. A survey of phishing attack techniques, defence mechanisms and open research

challenges. Enterprise Information Systems. 2022; 16(4):527-565.

21. Jampen D, *et al*. Don't click: Towards an effective anti-phishing training. A comparative literature review. Human-Centric Computing and Information Sciences. 2020; 10(1):1-41.
22. Klint R. Cybersecurity in home-office environments: An examination of security best practices post Covid, 2023 [online]. Available from: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1779054 [Accessed 7 Nov 2023].
23. Kumar M, *et al*. Machine learning models for phishing detection from TLS traffic. Cluster Computing, 2023, 1-15. Available from: https://link.springer.com/article/10.1007/s10586-023-04042-6 [Accessed 10 Nov 2023].
24. Manasa S, *et al*. Securing online bank transactions from phishing attacks using MFA and secure session key. Indian Journal of Science and Technology. 2015; 8(2):123-126. Available from: https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2015/Issue-Supplementary-2/Article17.pdf [Accessed 8 Nov 2023].
25. Mirsky Y, Lee W. The Creation and Detection of Deepfakes. ACM Computing Surveys. 2021; 54(1).
26. Morakinyo Oghenerhona Emmanuel. A secure bank login system using a multi-factor authentication, 2021 [online]. Available from: http://ir.mtu.edu.ng/jspui/handle/123456789/277 [Accessed 7 Nov 2023].
27. Moul KA. Avoid phishing traps. Proceedings ACM SIGUCCS User Services Conference, 2019, 199-208.
28. Nadeem M, *et al*. Phishing Attack, Its Detections and Prevention Techniques. International Journal of Wireless Security and Networks. 2023; 1(2):13-25. Available from: https://www.researchgate.net/profile/Syeda-Zahra-43/publication/374848676_Phishing_Attack_Its_Detections_and_Prevention_Techniques/links/6532592b1d6e8a70703fa896/Phishing-Attack-Its-Detections-and-Prevention-Techniques.pdf [Accessed 5 Nov 2023].
29. Niamathulla S, Bhalothia M. COVID-19: The Impact on Global Cyber Security Infrastructures in Organizations. In Cybersecurity Crisis Management and Lessons Learned From the COVID-19 Pandemic, 2022, 43-66. Available from: https://www.igi-global.com/chapter/covid-19/302220 [Accessed 7 Nov 2023].
30. Obaid AJ, *et al*. An adaptive approach for internet phishing detection based on log data. Pen.Ius.Edu.Ba. 2021; 9(4):622-631. Available from: http://pen.ius.edu.ba/index.php/pen/article/view/2398 [Accessed 10 Nov 2023].
31. Pandey M, Singh M, Pal S. Detection of Phishing Website Using Intelligent Machine Learning Classifiers Check for updates. In: Soft Computing and Signal Processing: Proceedings of 5th ICSCSP 2022. ICSCSP, 2023, p21. Available from: https://books.google.com/books?hl=en&lr=&id=-bjHEAAAQBAJ&oi=fnd&pg=PA21&dq=phishing+fishing&ots=YhXWJjRG2D&sig=qWou9loPjPMI-Z94wbJLDqY5QzI [Accessed 5 Nov 2023].
32. Pienta D, Thatcher JB, Johnston A. Protecting a whale in a sea of phish. Journal of Information Technology. 2020; 35(3):214-231.
33. Rashid J, Mahmood T, Nisar M. Phishing detection using machine learning technique. In: 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH). Institute of Electrical and Electronics Engineers Inc, 2020, 43-46. Available from: https://ieeexplore.ieee.org/abstract/document/9283771/ [Accessed 10 Nov 2023].
34. Sabo KE, Black J, Sarno DM. Developing IMPAWSTER: Improving Meaningful Phishing Awareness With Simulated Training and Email Roleplay. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2023 [online]. Available from: http://journals.sagepub.com/doi/10.1177/21695067231192197 [Accessed 5 Nov 2023].
35. Sharma P, Dash B, Ansari M. Anti-phishing techniques-a review of Cyber Defense Mechanisms. International Journal of Advanced Research in Computer and Communication Engineering, 2022, p2007. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4335354 [Accessed 8 Nov 2023].
36. Tan C, *et al*. Hybrid phishing detection using joint visual and textual identity. Expert Systems with Applications. 2023; 220:p119723. Available from: https://www.sciencedirect.com/science/article/pii/S0957417423002245 [Accessed 10 Nov 2023].
37. Tatang D, Zettl F, Holz T. The evolution of DNS-based email authentication: measuring adoption and finding flaws. ACM International Conference Proceeding Series, 2021, 354-369.
38. Venkatesh N, Tejaswini V, Soumya G. Malicious URL Detection Using Machine Learning. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 2023; 14(2):537-552. Available from: https://www.turcomat.org/index.php/turkbilmat/article/view/13684 [Accessed 5 Nov 2023].
39. Vollmer S, *et al*. Artificial intelligence and the value of transparency. Springer, 2020 [online]. Available from: https://link.springer.com/article/10.1007/s00146-020-01066-z [Accessed 20 Dec 2023].
40. Wassermann S, Meyer M, Goutal S. Targeted Attacks: Redefining Spear Phishing and Business Email Compromise. Arxiv.Org, 2023. [online]. Available from: https://arxiv.org/abs/2309.14166 [Accessed 5 Nov 2023].
41. Wayal GS, Bhandari V. An Analytical Research Based on Mitigating Review Spam on E-Commerce Sites. Hkclxb.Cn. 2023; 43:543-560. Available from: https://www.hkclxb.cn/article/view/2023/543.pdf [Accessed 5 Nov 2023].
42. Wolf CT. Professional Identity and Information Use: On Becoming a Machine Learning Developer. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019; 11420 LNCS:625-636.
43. Wood T, Basto-Fernandes V, Boiten E. Systematic Literature Review: Anti-Phishing Defences and Their Application to Before-the-click Phishing Email Detection, 2022. Arxiv.Org [online]. Available from: https://arxiv.org/abs/2204.13054 [Accessed 8 Nov 2023].
44. XU J. Biometrics in FinTech: A Technological Review.

Future And FinTech, The: Abcdi and Beyond, 2022 [online]. Available from: https://books.google.com/books?hl=en&lr=&id=mWF2EAAAQBAJ&oi=fnd&pg=PA361&dq=MFA+mobile+device,+a+fingerprint+scan,+a+smart+card,+or+even+a+retinal+scan&ots=lyV4vywlzC&sig=PlFdMmyxhgYYOyZSSeHissAnNTA [Accessed 8 Nov 2023].

45. Zhu E, *et al*. CCBLA: A Lightweight Phishing Detection Model Based on CNN, BiLSTM, and Attention Mechanism. Cognitive Computation. 2023; 15(4):1320-1333.