STM JOURNALS

Journal of
Computer Technology & Applications

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 16, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoCTA

https://journals.stmjournals.com/jocta

Review JoCTA

Detection of Phishing Website URLs and Email/SMS Using Random Forest and Multinomial Naive Bayes

Sridevi Ravada[1,*], Harika Kasukurthi[2], Deekshitha Govindu[2], Sowmya Vara[2], Akshaya Enumukkala[2]

Abstract

Currently, phishing attacks via SMS/email and URL have become significant threat to cybersecurity, posing risks to both individuals and organizations alike. Phishing attacks typically involve the creation of fraudulent websites or the dissemination of deceptive emails and SMS messages to trick users into disclosing sensitive information such as passwords, credit card numbers or personal details. To respond to these attacks, we develop a robust system for the detection of phishing URLs and SMS/emails using machine learning techniques. For phishing website URL detection, we extracted various features from HTML content using Beautiful Soup, and then applied supervised learning algorithms such as Decision Trees, Random Forest, and XGBoost, and Multinomial Naïve Bayes to classify URLs as Legitimate or Phishing. We achieved promising results with the Random Forest, which demonstrated high accuracy in distinguishing between legitimate and phishing URLs. For email/SMS, TF-IDF Vectorization, Natural language preprocessing is used and then applied supervised learning algorithms such as Multinomial Naïve Bayes, support vector classifier (SVC), Random Forest, Decision Tree, AdaBoost and XGBoost. We achieved promising results with the Multinomial Naïve Bayes, which demonstrated high accuracy in distinguishing between spam and not spam.

Keywords: Phishing attacks, multinomial Naïve Bayes, TF-IDF vectorization, natural language preprocessing, random forest, beautiful soup

INTRODUCTION

Phishing attacks have emerged as a significant threat in recent years, rapidly escalating due to various factors. These malicious activities include deceiving individuals into disclosing sensitive information or installing harmful software like ransomware. Cybercriminals utilize phishing as a form of social engineering, manipulating people into sharing login credentials, financial details, or personal data. They often masquerade as trusted entities like banks or reputable companies, leveraging communication channels such as emails, phone cells (vishing), text messages (smishing), or counterfeit websites.

These deceptive tactics exploit human psychology, persuading recipients to click on malicious links, download harmful files, or disclose confidential information. Phishing websites, for instance, mimic legitimate sites to steal credentials or financial information easily. In contrast, spam mails are unsolicited bulk emails containing deceitful content or malicious attachments designed to exploit vulnerabilities.

Vishing and smishing tactics use voice calls or text messages to dupe victims into revealing personal data, capitalizing on emotions like fear or

***Author for Correspondence**
Sridevi Ravada
E-mail: srideviravada@gvpcew.ac.in

[1]Assistant Professor, Department of Information Technology, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India
[2]Student, Department of Information Technology, Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India

Received Date: October 25, 2024
Accepted Date: December 19, 2024
Published Date: December 31, 2024

Citation: Sridevi Ravada, Harika Kasukurthi, Deekshitha Govindu, Sowmya Vara, Akshaya Enumukkala. Detection of Phishing Website URLs and Email/SMS Using Random Forest and Multinomial Naive Bayes. Journal of Computer Technology & Applications. 2025; 16(1): 22–30p.

urgency for immediate responses. Both phishing and spam present considerable risks, including identity theft, financial loss, data breaches, and malware infections.

The goal is to create strong machine learning models for two essential tasks: identifying phishing websites and classifying SMS spam. The goal is to differentiate between legitimate and malicious websites using URL features and to identify spam messages accurately within SMS data.

LITERATURE SURVEY

Ahammad *et al*. conducted a study to evaluate how effectively machine learning algorithms can identify phishing URLs [1]. They highlighted the importance of specific features, such as domain age (the duration for which the domain has existed), URL length (the total character count of the URL), and the presence of HTTPS (a security protocol indicating whether the URL is secure). These features serve as critical indicators to differentiate phishing URLs from legitimate ones.

Salloum *et al*. explored the application of natural language processing (NLP) techniques for spam email detection [2]. Their approach involved breaking down email content into smaller parts through tokenization and analyzing the importance of words using TF-IDF (Term Frequency-Inverse Document Frequency). This method helps identify patterns in textual features, enabling the detection of spam emails based on the linguistic structure and word relevance.

Gualberto *et al*. emphasized the role of feature engineering in phishing and spam detection [3]. They explored strategies to create meaningful features from raw data, showcasing how proper feature selection and extraction can significantly enhance the accuracy of detection models. By optimizing features, the models can focus on the most relevant attributes, reducing noise and improving performance.

Mutalib *et al*. focused on the interpretability of machine learning models, which is crucial for understanding how decisions are made and building trust in these systems [4]. They provided insights into making complex algorithms more transparent. Liang *et al*., on the other hand, examined the robustness of models against overfitting, a condition where a model performs well on training data but poorly on new data [5]. They also highlighted the benefits of ensemble methods like Random Forest, which combine predictions from multiple models to improve accuracy and reliability in phishing and spam detection.

El Aassal *et al*. leveraged web scraping and HTML parsing techniques to extract features from phishing websites and spam emails [6]. These methods allow automated collection and analysis of web content, enabling the identification of patterns that distinguish phishing attempts and spam.

Harun *et al*. highlighted the importance of evaluating detection models using standardized datasets, such as the Phishing Websites Dataset and the Email Dataset [7]. These datasets provide a reliable benchmark, ensuring that models are tested under consistent conditions and their performance can be compared accurately across studies.

This body of work collectively demonstrates the advancements in phishing and spam detection through feature engineering, model interpretability, robustness, and the use of standardized datasets for evaluation.

PROPOSED SYSTEM

The proposed system aims to tackle two pressing challenges in today's digital sphere: discerning between spam and legitimate messages in SMS and email communications and pinpointing potentially harmful phishing URLs. Through advanced machine learning techniques, this system delivers real-time solutions, fortifying users' digital security and confidence [8].

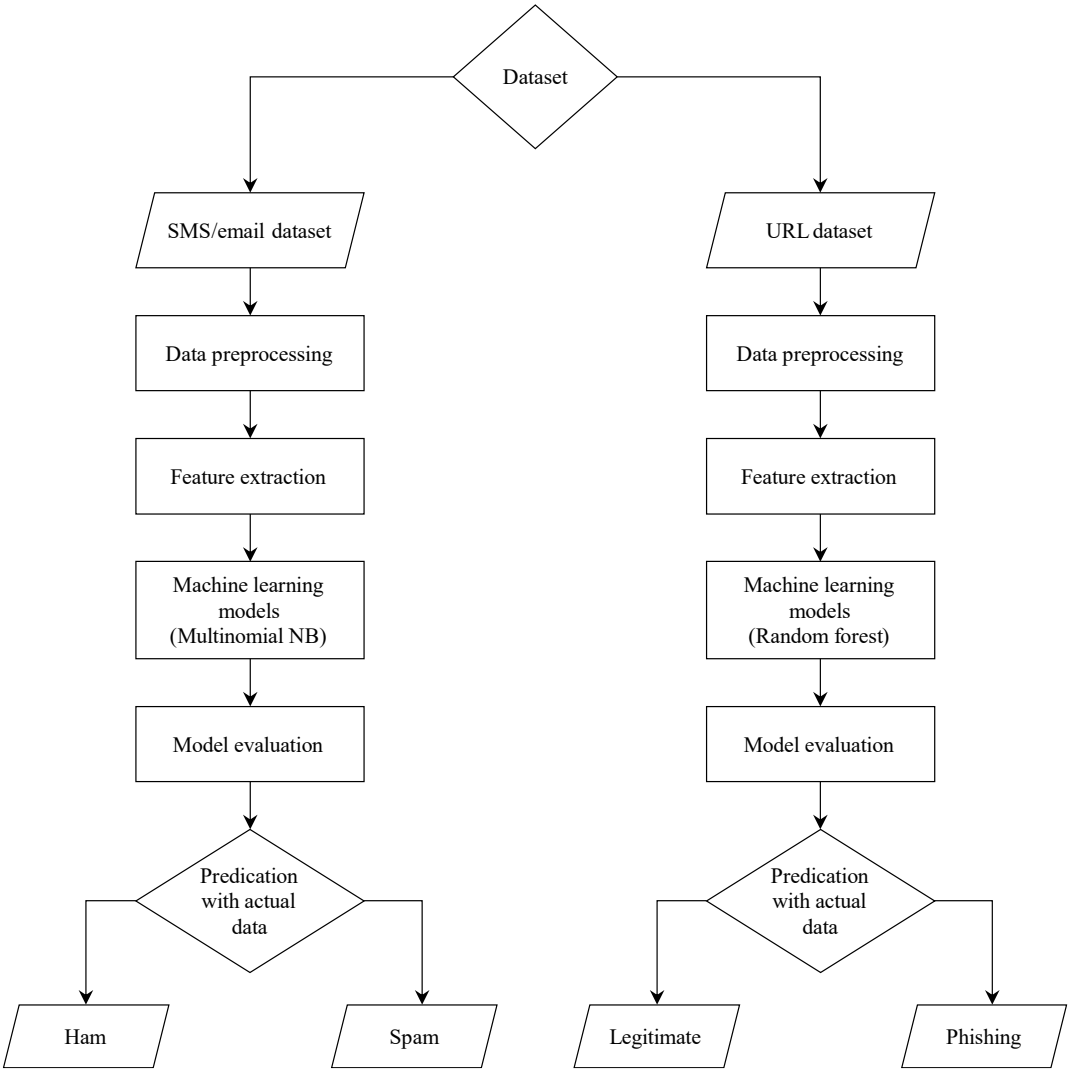

Figure 1. Proposed system.

To address the spam/ham classification task, the system relies on a trained model adept at distinguishing between authentic and spam messages. Users can input a message into the application to swiftly ascertain its classification, enabling them to filter out unwanted and potentially harmful content from their communication channels [9]. This feature proves especially valuable in environments with high message volumes, such as email inboxes or SMS notifications (Figure 1).

For email/SMS classification, the system employs a blend of Natural Language Processing (NLP) techniques, TF-IDF Vectorization, and Machine Learning (ML) models. Initially, text data undergoes preprocessing steps, including tokenization, stop word removal and stemming, to convert it into a numerical format, capturing textual features. ML models, including Support Vector Machines, Multinomial Naïve Bayes, Random Forest, XGBoost, AdaBoost and decision trees, are trained on this vectorized data to categorize messages as spam or non-spam. This approach harnesses NLP to extract meaningful features from text data and ML models to identify patterns for precise classification.

Beyond spam detection, the system provides functionality for detection of phishing URLs in web content. By scrutinizing the structure and content of URLs, it identifies suspicious links that could lead to phishing websites attempting to deceive users into disclosing sensitive information. Through a user-friendly interface, individuals can input a URL and promptly receive feedback on its legitimacy, aiding them in sidestepping online scams and fraudulent activities [10].

The system combines HTML content analysis with machine learning techniques to identify phishing URLs. It utilizes a python package is Beautiful Soup, to convert the input URL into HTML content. Through HTML parsing, it extracts features from the HTML content, examining for specific tags indicative of phishing characteristics commonly found in deceptive websites, such as <title>, <input>, <button>, <image>, <a>, among others. These features are the foundation for constructing a feature vector for each URL, representing its HTML structure and content. These vectors are then fed into machine learning models, including XGBoost, Multinomial Naïve Bayes , Decision Trees and Random Forests, to classify URLs as legitimate or phishing, thereby enhancing classification accuracy. Furthermore, the system offers performance visualization tools, providing insights into the effectiveness of the underlying machine learning models. Users can access metrics and graphs illustrating the models' performance in classifying spam/ham messages and phishing URLs, enabling them to assess the reliability and accuracy of system's predictions.

DATASET

SMS/Email

Data Preparation

The SMS/email spam dataset was collected in a CSV file from Kaggle, the dataset appears to be a collection of text messages, initially with five columns (v1, v2 and three unnamed columns) and 5572 rows.

Data Preprocessing

In the phase of data pre-processing, the dataset is collected and processed, to ensure better performance and detailed analysis.

- In general, the null values are identified from the dataset and removed.
- Next, the data type is converted into the desired format for ease of computation and performance.
- After preprocessing, it was reduced to 5169 rows with only two columns (target and text). In the initial dataset, there are additional columns (Unnamed 2, Unnamed 3 and Unnamed 4) that were likely empty or irrelevant, hence they were dropped during cleaning. The 'v1' column likely denotes the classification of messages (e.g., 'ham' for legitimate messages and 'spam' for unsolicited messages). The 'v2' column contains the actual text of the messages. During cleaning, irrelevant or missing columns were removed, leaving only the essential information for analysis: the target (which likely indicates whether the message is 'ham' or 'spam') and the text of the messages. This process resulted in the removal of 403 rows, likely due to missing or incomplete data [11].

Data Visualization

Figure 2 contains 87.4% of ham messages and 12.6% of spam messages.

Website URLs

Data Preparation

The Phishing URL dataset was collected in a CSV file from phishtank.org and tranco-list.eu, the dataset appears to be related to phishing URLs, containing 26,585 instances and 44 features.

Data Preprocessing

In the phase of data pre-processing, the dataset is collected and processed, to ensure better performance and detailed analysis.

- In general, the null values are identified from the dataset and removed.
- Next, the data type is converted into the desired format for ease of computation and performance.
- After preprocessing, 15,053 instances and 44 features were obtained. Each instance likely represents a URL, with the features representing various characteristics or attributes of those URLs. The presence of 1 or 0 in the classes indicates whether a URL is classified as legitimate (0) or phishing (1). This is a typical binary classification problem, where the goal is to predict whether a given URL is legitimate, or a phishing attempt based on its features.

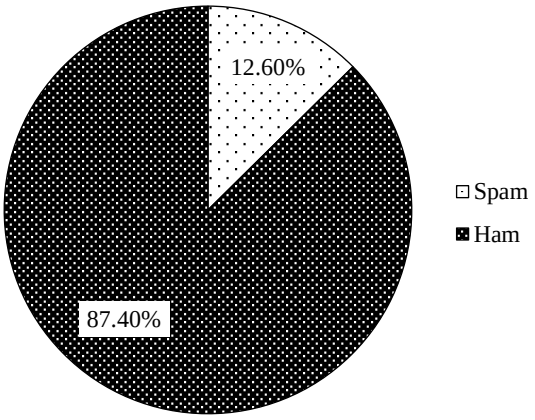

Figure 2. SMS/email spam dataset.

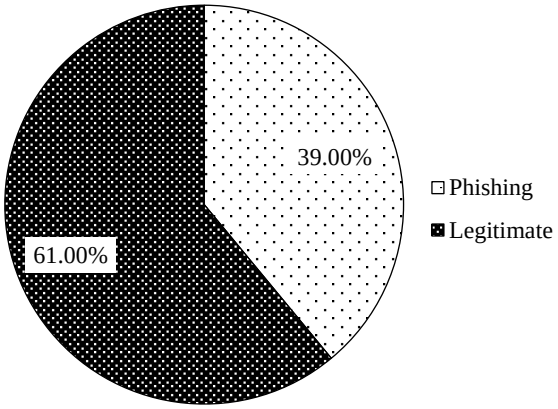

Figure 3. Website URLs dataset.

Data Visualization

Figure 3 contains 61% of legitimate URLs and 39% of phishing URLs.

ALGORITHMS USED

Random Forest

Random Forest is a widely recognized ensemble learning method commonly used in machine learning for addressing classification and regression challenges. It harnesses the strength of multiple decision trees to enhance both accuracy and robustness. The algorithm operates by constructing a forest of decision trees, each trained on a random subset of features [12]. During prediction, each tree independently forecasts the outcome, and the final result is obtained by aggregating these predictions—using voting for classification tasks and averaging for regression tasks, as shown in Figure 4.

One of the key benefits of Random Forest is its effectiveness in managing large datasets that feature high dimensionality and noisy data. It mitigates over fitting by introducing randomness during training, which leads to diverse trees that collectively make more accurate predictions. Furthermore, Random Forest offers insights into feature importance, allowing for the identification of which features have the greatest impact on the model's predictions [13].

Multinomial Naive Bayes

The Multinomial Naive Bayes algorithm is a probabilistic classifier commonly used in text classification tasks, particularly when dealing with features that represent word counts or frequency distributions. It functions according to Bayes' theorem and assumes that features are independent from one another [14]. In the realm of text classification, Multinomial Naïve Bayes calculates the probability of a document being associated with a specific class (e.g., spam or non-spam) given its feature values (e.g., word occurrences). It determines these probabilities by assessing the likelihood of observing each feature within documents of each class, in addition to the prior probabilities of each class as shown in Figure 5.

In the classification process, the algorithm identifies the class with the highest probability as the predicted class for the input document. Despite its simplicity and the assumption of feature independence, Multinomial Naïve Bayes frequently achieves strong performance in practice, particularly in tasks involving large text datasets, such as email classification or sentiment analysis [15].

METHODOLOGY

Collection of Historical Data

The first step involves gathering relevant data for training and testing the machine learning models. This may include datasets containing examples of legitimate and phishing URLs, as well as SMS messages labelled as spam or non-spam.

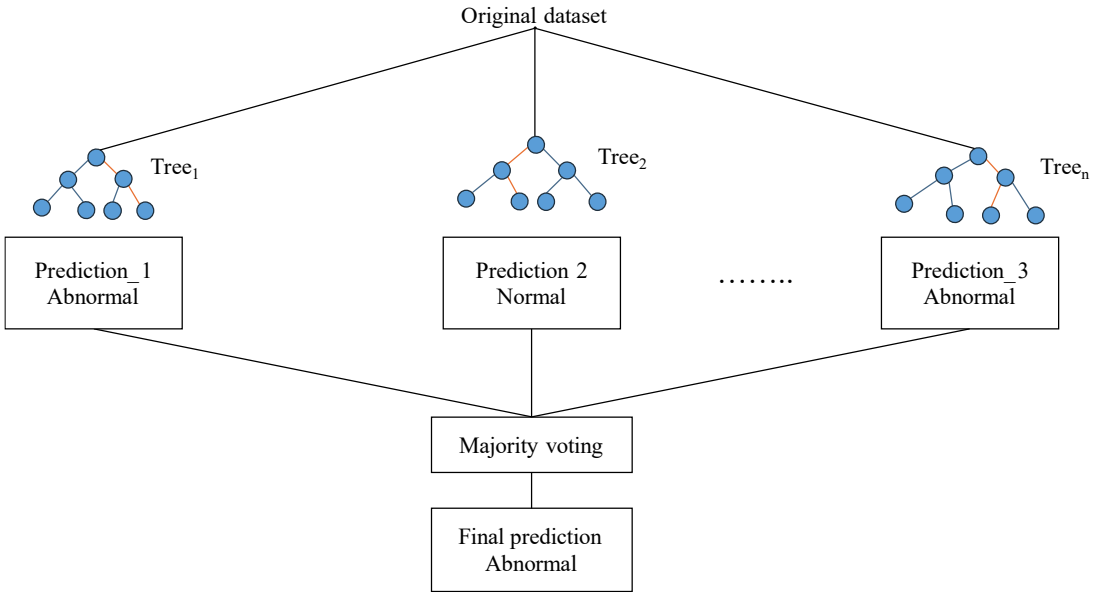

Figure 4. Architecture of random forest.

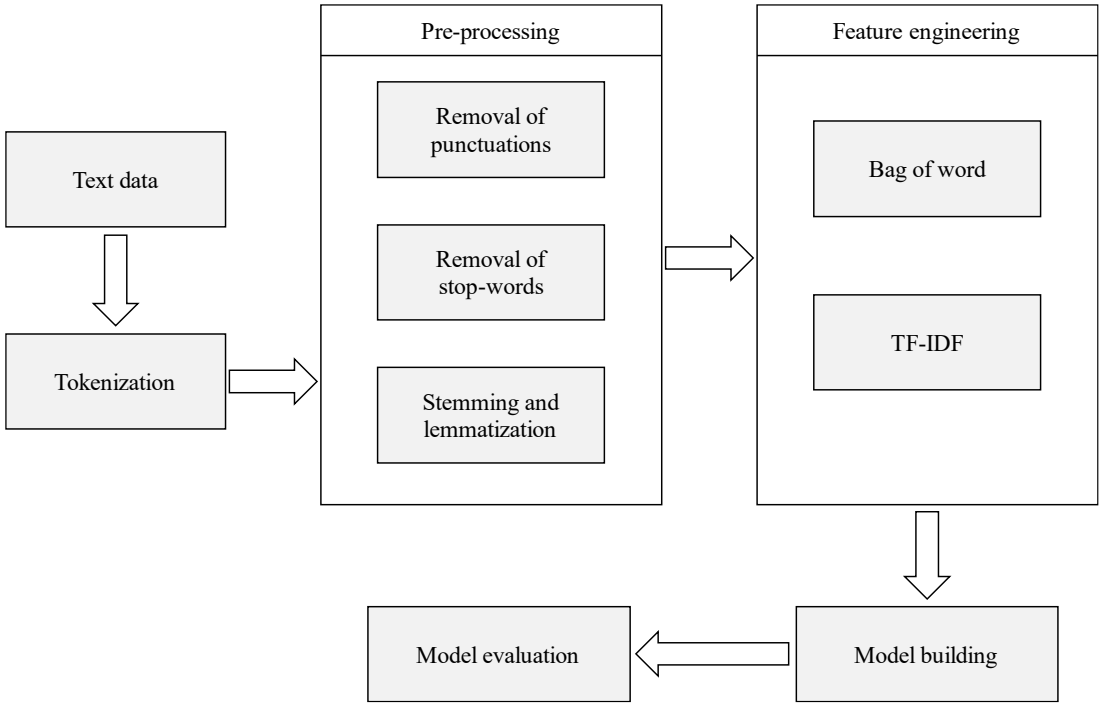

Figure 5. Architecture of multinomial naive bayes.

Data Preprocessing

The collected data needs to be pre-processed to prepare it for model training. This involves tasks like text normalization, tokenization, removing stop-words and punctuation, stemming and vectorization using techniques like TF-IDF vectorization (Term Frequency-Inverse Document Frequency).

Feature Extraction

Feature extraction like Beautiful Soup and NLP is a process where relevant features are identified from emails, URLs, or SMS messages to contribute to the detection of phishing or spam. This may include features like domain reputation, URL length, and presence of HTTPS, email header anomalies, and lexical analysis of content.

Model Training

The preprocessed data that has been collected is classified into training data (80%) and testing data (20%) then used to train machine learning models such as Random Forest, for phishing website detection and Multinomial naive bayes for SMS spam classification [16]. This step involves splitting the data into training and fitting the models to the training data, and then tuning hyper parameters such as n_estimators is 50 and random_state is 2 to optimize model performance.

Performance Evaluation

Following the training of the models, their performance is evaluated using metrics such as accuracy, precision, recall, and F1 score. This helps assess how well the models are performing in detecting phishing websites or spam messages.

RESULTS

The collected data is divided into training and testing datasets. Next, the information that has been retrieved from Kaggle and phishtank.org and tranco-list.eu is analysed, and then pre-processed. Random Forest, Multinomial Naive Bayes, SVC, Decision Tree, AdaBoost and XGBoost algorithms have been implemented as illustrated in the graphs given (Figures 6 and 7), and the prediction analysis is described as shown in Tables 1 and 2.

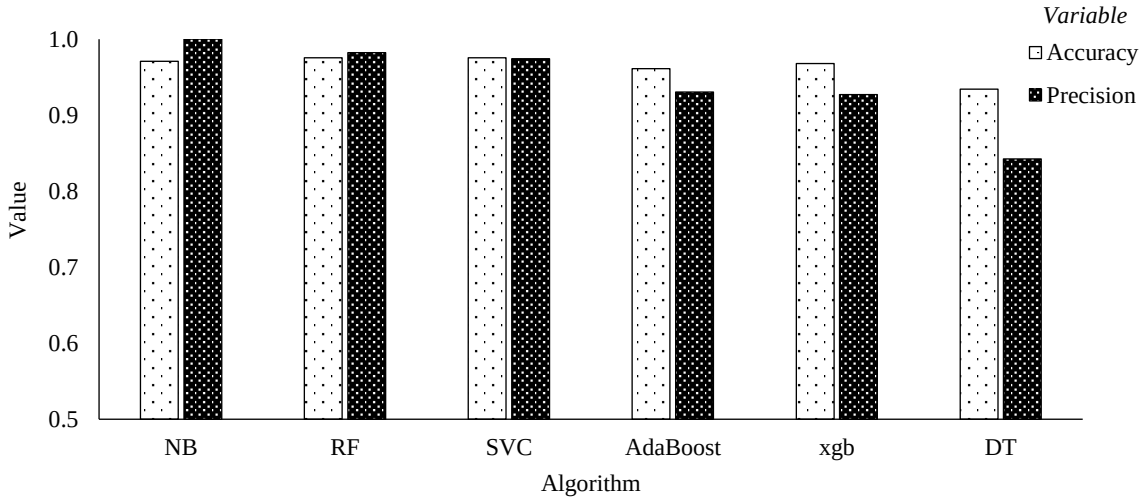

Figure 6. Performance of SMS/email.

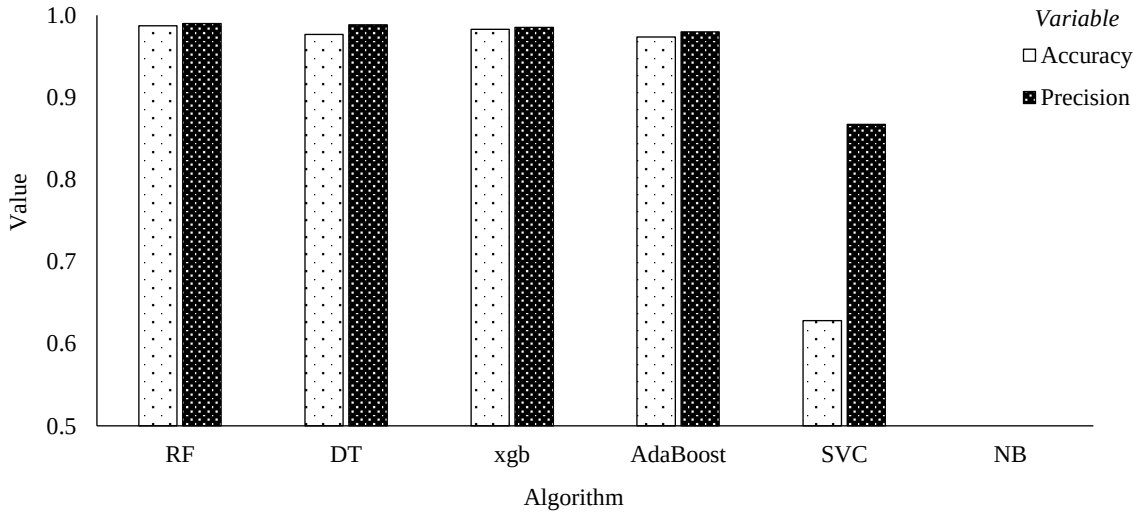

Figure 7. Performance of Website URLs.

Table 1. Classification report of multinomial naive bayes.

	Precision	Recall	F1-Score	Support
0	0.97	1.00	0.98	896
1	1.00	0.78	0.88	138
Accuracy			0.97	1034
Macro avg.	0.98	0.89	0.93	1034
Weighted avg.	0.97	0.97	0.97	1034

Table 2. Classification report of random forest.

	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	3187
1	0.99	0.98	0.98	2130
Accuracy			0.99	5317
Macro avg.	0.99	0.98	0.98	5317
Weighted avg.	0.99	0.99	0.99	5317

By comparing algorithms shown in Figures 6 and 7, multinomial naive bayes gives the highest accuracy and precision for SMS/email [16].

- *Accuracy*: (TP+TN)/(TP+TN+FP+FN)
- *Precision:* (TP)/(TP+FP)
- *F1-score:* (2TP)/(2TP+FP+FN)
- *Recall*: (TP)/(TP+FN)

CONCLUSION

The project aims to develop a robust method for Machine learning techniques like Random Forest and Multinomial Naive Bayes. By using advanced tools such as Beautiful Soup for HTML parsing and NLP for tokenization, removing stop-words, and stemming. The implementation involves a multi-step process including pre-processing, feature extraction, model training, deployment, validation, and evaluation. Results demonstrate improved accuracy in identifying legitimate or phishing for website URLs, and spam or not spam for SMS/email. The project emphasizes the potential of machine learning to combat cybercrimes like phishing and spam messages.

Future Scope

For SMS/email Spam Classification, integrating deep learning models such as recurrent neural networks (RNNs) or transformer-based architectures like BERT could improve the accuracy of spam detection by capturing complex patterns and contextual information in messages. Additionally, exploring ensemble methods or model stacking techniques to combine the predictions of multiple models could further enhance classification performance. For Phishing Website URL Detection, enhancing feature extraction methods with more sophisticated techniques like natural language processing (NLP) and Machine learning algorithms specifically designed for web content analysis could improve the system's ability to detect subtle indicators of phishing. Furthermore, incorporating user feedback mechanisms and continuous model retraining based on real-time data could enhance the system's adaptability and effectiveness in detecting evolving phishing threats.

REFERENCES

1. Ahammad SH, Kale SD, Upadhye GD, Pande SD, Babu EV, Dhumane AV, Bahadur MD. Phishing URL detection using machine learning methods. Adv Eng Softw. 2022 Nov 1; 173: 103288.
2. Salloum S, Gaber T, Vadera S, Shaalan K. Phishing email detection using natural language processing techniques: a literature survey. Procedia Comput Sci. 2021 Jan 1; 189: 19–28.
3. Gualberto ES, De Sousa RT, Vieira TP, Da Costa JP, Duque CG. The answer is in the text: Multi-stage methods for phishing detection based on feature engineering. IEEE Access. 2020 Dec 9; 8: 223529–47.

4. Mutalib NH, Sabri AQ, Wahab AW, Abdullah ER, AlDahoul N. Explainable deep learning approach for advanced persistent threats (APTs) detection in cybersecurity: a review. Artif Intell Rev. 2024 Nov; 57(11): 1–47.
5. Raghunathan A, Xie SM, Yang F, Duchi J, Liang P. Understanding and mitigating the tradeoff between robustness and accuracy. [Preprint]. arXiv:2002.10716. 2020 Feb 25. DOI: https://doi.org/10.48550/arXiv.2002.10716.
6. El Aassal A, Baki S, Das A, Verma RM. An in-depth benchmarking and evaluation of phishing detection research for security needs. IEEE Access. 2020 Jan 28; 8: 22170–92.
7. Harun NZ, Jaffar N, Kassim PS. Physical attributes significant in preserving the social sustainability of the traditional Malay settlement. In Reframing the Vernacular: Politics, Semiotics, and Representation. Cham: Springer International Publishing; 2020; 225–238.
8. Divakaran DM, Oest A. Phishing detection leveraging machine learning and deep learning: A review. IEEE Secur Priv. 2022 Jun 14; 20(5): 86–95.
9. Akanchha A. Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates. Thesis. Halifax, Nova Scotia: Dalhousie University; 2020. Available form https://dalspace.library.dal.ca/items/445ef57f-5c6b-4232-a05c-3f4073238a63
10. Liu DJ, Geng GG, Jin XB, Wang W. An efficient multistage phishing website detection model based on the CASE feature framework: Aiming at the real web environment. Comput Secur. 2021 Nov 1; 110: 102421.
11. Rao RS, Pais AR. Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput Appl. 2019 Aug; 31(1): 3851–73.
12. Cao Y, Han W, Le Y. Anti-phishing based on automated individual white-list. In Proceedings of the 4th ACM workshop on Digital identity management. 2008 Oct 31; 51–60.
13. Agarwal S, Kaur S, Garhwal S. SMS spam detection for Indian messages. In 2015 IEEE 1st International Conference on Next Generation Computing Technologies (NGCT). 2015 Sep 4; 634–638.
14. Crawford M, Khoshgoftaar TM, Prusa JD, Richter AN, Al Najada H. Survey of review spam detection using machine learning techniques. J Big Data. 2015 Dec; 2: 23(24p).
15. Radhakrishnan A, Vaidhehi V. Email Classification using Machine learning algorithms. Int J Eng Technol (IJET). 2017 Apr; 9(2): 335–40.
16. Hota HS, Shrivas AK, Hota R. An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique. Procedia Comput Sci. 2018 Jan 1; 132: 900–7.

