PhishVault: ML-Powered Phishing Detection Framework

Shouvik
220911598
Department of I&CT
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal- 576104, Karnataka, India
Email:shouvik.mitmpl2022@learner.mnipal.edu

Harshit Dugar
220911656
Department of I&CT
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal- 576104, Karnataka, India
Email:harshit3.mitmpl2022@learner.manipal.edu

Vedika Awasthi
220911582
Department of I&CT
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal- 576104, Karnataka, India
Email:vedika.mitmpl2022@learner.manipal.edu

Abstract— Abstract: Phishing attempts are rather common these days, and many people fall for them. Any email that includes private information could be the target of website or email phishing, which could lead to data theft. Additionally, audio phishing is becoming more and more common. Phishing attacks have cost countless people and businesses billions of dollars. Since the impact of these attacks will only grow globally, more sophisticated phishing detection techniques are required to neutralise the danger. This paper investigates and reports on the use of the random forest machine learning algorithm, text-based feature extraction, and naive bayes in the classification of phishing attacks, with the main objective of developing a better phishing email classifier with higher prediction accuracy and fewer features. Assessment on diverse datasets demonstrates the model's efficacy in accurately identifying website-focused phishing attacks, thereby offering a unified, scalable solution for comprehensive phishing defense. Our findings underscore the potential of machine learning in enhancing cybersecurity resilience against multi-faceted social engineering threats.

Index Terms—

I. Introduction

Phishing has emerged as a major security concern as digital threats continue to change, and cybercriminals are using ever-more-advanced tactics to trick people and businesses. (or "vishing") adds a new level to this ongoing menace. Phishing assaults, especially through fraudulent websites and emails, have caused significant financial and data losses. Malicious websites frequently mimic trustworthy websites in an attempt to trick users into divulging private information like login passwords or bank account details. Given the light of these developments, a strong and flexible defence system is necessary to protect private information and preserve user confidence in digital communications.

To combat these website-focused attacks, a thorough phishing detection method is presented in this study. The method classifies phishing content for websites with excellent accuracy and low computational needs by incorporating machine learning models. Text-based feature extraction is one of the key strategies, as are classification algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, AdaBoost, and Naive Bayes, which were selected for their ability to detect phishing attacks via websites and emails.

In order to provide a comprehensive risk assessment, the project's multifaceted approach to phishing detection analyses email text, website URLs, embedded links, attachment kinds, and sender information. Experimented on several datasets, this AI-driven defense system demonstrates significant promise in enhancing cybersecurity by identifying phishing attacks with high accuracy and adaptability. The system contributes to the broader field of cybersecurity by offering a scalable, unified solution that addresses the growing complexity and frequency of phishing threats.

Related Works and Existing Methods

A. How Gmail works

Gmail's spam filtration system employs sophisticated AI to evaluate each incoming email and ascertain its appropriate placement in either the inbox or spam folder. Here is a concise overview of the procedure:

- Incoming Email: Gmail's filter activates upon the receipt of a new email.

- Header and User Settings Examination: It initially scrutinizes the email's header for anomalous patterns and implements any personalized user filters.
- AI and Machine Learning Models: Gmail's spam detection is predominantly based on machine learning, whichis trained on extensive datasets of both valid and spam emails. These models validate the sender's information and ascertain email authenticity by identifying phishing attempts and evaluating sender reputation. Employing Natural Language Processing (NLP), they scrutinize the material and identify patterns that may signify spam, including specific keywords and formatting techniques utilized by spammers. Moreover, Gmail's adaptive learning algorithms continuously enhance by utilizing real-time data from millions of user interactions, evolving in response to emerging spam strategies over time.
- Spam Signals Verification: The email is compared with established spam indicators and blacklists.
- Final Decision: Leveraging insights from AI and user feedback, the system determines the email's final destination—Inbox or Spam.

This dynamic, AI-driven methodology enables Gmail to sustain high precision in its spam filtration and adjust to advancing spam strategies.

B. Detecting phishing web pages by exploiting raw URL and HTML characteristics

Phishing detection techniques largely rely on distinguishing features between phishing and legitimate websites, using lexical and statistical indicators such as URLs, HTML elements, and DNS data. NLP-based features like n-grams and TF-IDF have recently been used to enhance classification accuracy. Many state-of-the-art approaches combine these custom features with machine learning algorithms for detection.

For example, Kumi et al. (2021) extracted eight key features, achieving 95.8

A key challenge with these methods is that handcrafted features can limit adaptability to new data, as attackers often modify their tactics to bypass known indicators. Some studies are addressing this by implementing adaptive techniques, such as Smadi et al. (2018), who used reinforcement learning to detect zero-day phishing attacks with high accuracy. While feature engineering remains effective, it is resource-intensive, requiring domain expertise, and may struggle with evolving phishing tactics that avoid established detection patterns.

C. How Truecaller works-

Truecaller uses AI and machine learning to enhance call identification and spam-blocking accuracy. Its AI-driven caller identification matches unknown numbers with a vast database created through user contributions and public sources, allowing real-time identification even for numbers not saved in contacts. Spam detection is another core feature: Truecaller applies pattern recognition and crowd-sourced feedback to recognize spam numbers. Users frequently report spam, which helps Truecaller's AI assign a "spam score" to numbers, dynamically update filters based on new patterns. Real-time behavioral analysis further strengthens this detection by identifying suspicious patterns like mass-calling from one number, marking this as potential fraud.

Additionally, Truecaller uses Natural Language Processing (NLP) to filter spam messages by analyzing keywords and patterns typical in phishing and unsolicited messages. The AI adapts to user-specific preferences by learning over time what each user considers spam, leading to personalized call and message blocking. Adaptive filtering models enhance this by blocking similar types of unwanted calls without manual input.

Truecaller's AI models are continuously updated with new data from users, calls, and reports, ensuring the system can respond to evolving spam tactics and fraud attempts effectively. Altogether, AI enables Truecaller to provide accurate, personalized call identification and reliable spam protection, creating a safer communication environment for its users.

D. Comparative Analyses of Machine Learning Paradigms for Spam and Phishing Email Classification and Detection

To fulfill user information needs while leaving out some spam filter details, Cormack (2008) investigatedtechniques for email spam detection within storage and com- munication systems. Unsolicited bulk email (UBE) problems were covered by Sanz et al. (2008), who described machine learning algorithms for detection but did not compare content filters. Ma et al. (2009) improved phishing detection using orthographic characteristics and a modified global K-means clustering technique, obtaining efficiency increases with C4.5 Decision Trees and Information Gain. Toolan and Carthy (2009) used instance-based learning and C5.0 Decision Trees in an ensemble recall-boosting technique to increase phishing email memory. By using Feature Selection by Category (FSC), Gansterer and Pölz (2009) developed a ternary classification model for phishing, spam, and ham emails that outperformed binary classification techniques with an accuracy of 97%.

The authors used the Naver Clova Speech API, which outperformed other transcription services for the Korean language.

The study employed machine learning models, including Support Vector Machine (SVM), Logistic Regression (LR), SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost (RF), Decision Tree (DT), and Extreme Gra- dient Boosting (XGB). These models were trained on a full dataset and a smaller "Top 5000" dataset to balance case length and prevent data imbalance. Among these, the SVM model achieved the highest accuracy, while the Logistic Re- gression model provided the fastest detection time, which is essential for real-time detection needs.

This research demonstrates that even basic machine learning approaches can effectively detect vishing in a low-resource language. The authors suggest that future work could explore neural network-based models for even greater accuracy. This study serves as a benchmark in applying machine learningto real-time vishing detection in Korean, with applications potentially extending to spam detection in social networking and broadcasting.

II. METHOD

A. Email and Text Phishing

1) Data Import and Exploration:
   The notebook commences by importing fundamental libraries such as pandas, numpy, and sklearn. It imports two datasets, Phishing Email.csv and com- bined data.csv, which encompass emails and their cor- responding labels (phishing or not phishing).
2) Data Preprocessing
   Only the pertinent columns, namely ”Email Text” and ”Email Type,” are chosen and renamed to ”text” and ”label” for clarity. Labels are subsequently transformed into binary values (e.g., 1 for phishing emails and 0 for legitimate emails), so preparing the data for machine learning.
3) Text Cleaning and Preparation
   Libraries like NLTK (Natural Language Toolkit) can be utilized to preprocess the text data. Standard preparation procedures for textual data encompass the elimination of special characters, punctuation, and stopwords. This step is crucial for standardizing the email content to facilitate effective analysis.
4) Feature Extraction
   The notebook employs CountVectorizer or TfidfVector- izer to convert the email text into numerical features. These vectorizers transform text data into a matrix format, representing each word as a feature that encap- sulates the frequency or significance of terms in each email.
5) Machine Learning Model
   A machine learning model is chosen to categorize the emails according to the extracted attributes. Prevalent techniques for phishing detection encompass Decision Trees, Logistic Regression, and ensemble methods. The dataset is divided into training and testing subsets to assess the model’s efficacy.
6) Model Evaluation
   The model’s accuracy, precision, recall, and F1-score are computed to evaluate its efficacy in differentiating between phishing and non-phishing emails. The model is subsequently refined or optimized according to these criteria to enhance its detecting skills.
7) Analysis
   After training, the model is employed to fore- cast phishing in novel email samples. This phase entails examining misclassifications and pinpointing areas for potential model enhancement.

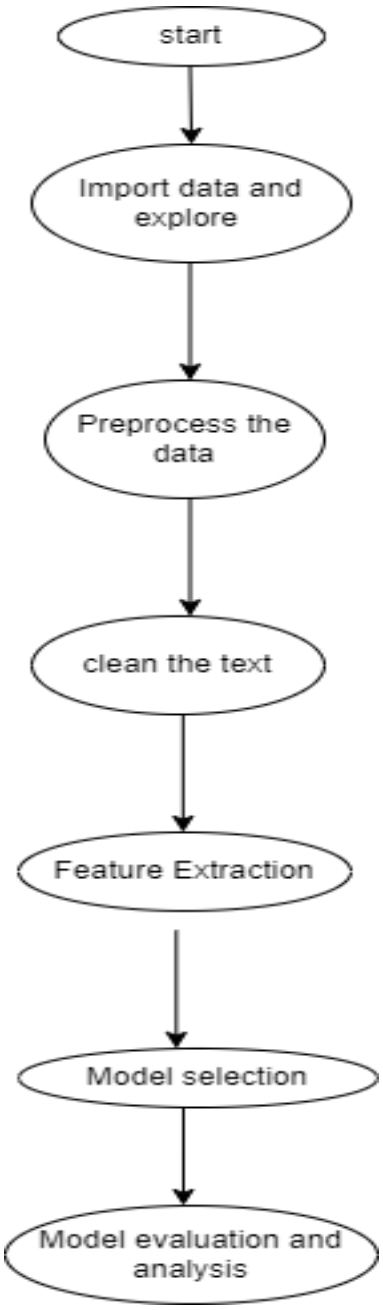

Fig. 1. Flowchart of email phishing.

- **Text Feature Extraction Utilizing Tokenization (RoBERTa)**

1) Encoding of Positional and Sequential Relationships The tokenized input IDs, when processed by RoBERTa, get encoding that include positional information. This encoding elucidates linkages throughout phrase structure, which can be advantageous in discerning tone, urgency, and structural patterns prevalent in phishing texts.
2) Advantages of RoBERTa's Tokenizer in Phishing Detection
   – Pre-trained Contextual Comprehension: RoBERTa identifies linguistic patterns, advantageous for recognizing overt phishing terminology (such as "urgent") and nuanced signals (such as tone).
   – Versatility Across Text Formats: RoBERTa's tokenizer adeptly manages diverse formats, ranging from formal correspondence to concise SMS messages.
   – Semantic Precision: It encompasses differences in phrase, tone, and phrasing, which are typically critical in phishing detection.

- **Model Development for Phishing Detection**

1) Data Partitioning: - Divide the dataset into training, validation, and test subsets to guarantee thorough evaluation and generalization.
2) Model Selection and Training: Concerning audio data, convolutional neural networks (CNNs) are commonly employed for spectrograms, whereas long short-term memory networks (LSTMs) are favored for sequential features such as Mel-frequency cepstral coefficients (MFCCs). RoBERTa-based models, such as 'RobertaForSequenceClassification', are frequently employed for text data. The model acquires the ability to classify samplesby minimizing a loss function, typically employ-ing backpropagation and gradient descent. Cross-entropy loss is commonly utilized for classification problems.
3) Assessment and Optimization: - Employ measures such as accuracy, precision, recall, and F1-score to assess performance on validation data. Refine hyperparameters (such as learning rate and batch size) and model architecture to enhance these metrics. - Methods like as dropout, batch normalization, and data augmentation (e.g., introducing noise) are utilized to enhance resilience and mitigate overfitting.
4) Testing and Deployment: - Following validation, the model undergoes testing on novel data to verifyits generalization capability, subsequently being deployed for real-time phishing detection.

B. Website Phishing

- Trigger Detection on Website Load
  When the user navigates to a new website, the extension is triggered, initializing the detection process.

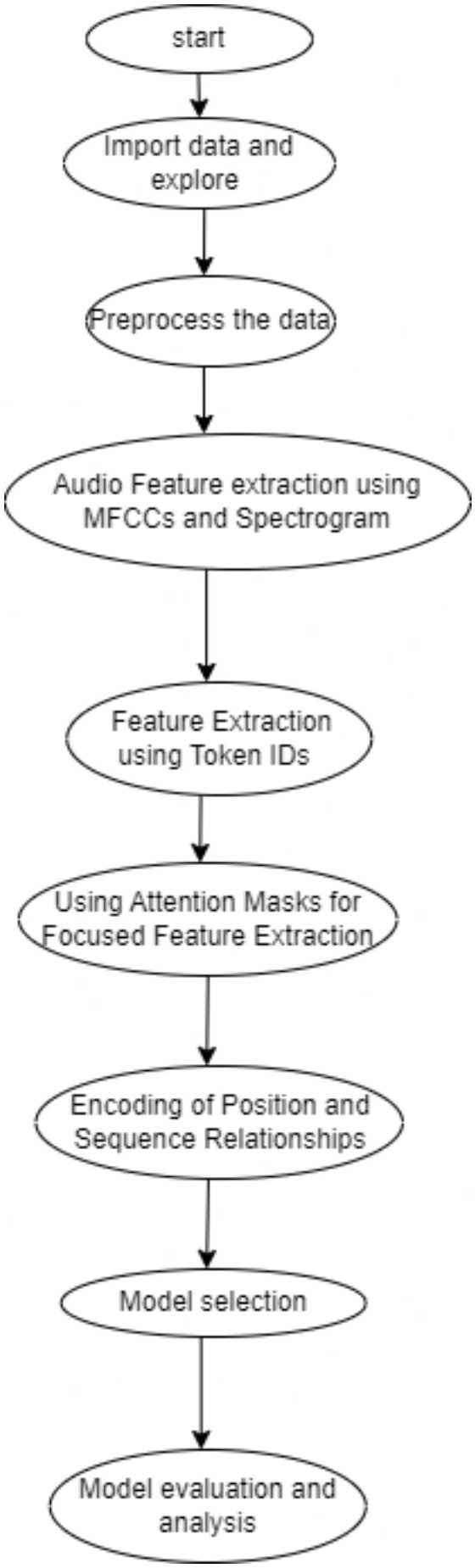

Fig. 2. Flowchart of Audio phishing.

- Feature Extraction
  Extract various features from the website that can indicate phishing. Commonly, these features fall into three categories: URL-based, domain-based, and content-based.
  1) URL based features-
     - URL Length: Phishing URLs are often unusually long.
     - Special Characters: Check for characters like hyphens, ”@” symbols, or numeric IPs, which can indicate phishing.
     - Domain Type: Look for common phishing top-level domains (TLDs) or unusual subdomains.
     - HTTPS/SSL: Check if the site is HTTPS secured; legitimate sites usually are.
  2) Domain Based Features-
     - Domain Age: Phishing domains are often newly created. Use WHOIS data to check domain registration age.
     - Domain Owner: Investigate domain registrant data, if available.
     - Certificate Validity: Verify the SSL certificate’s legitimacy and its issuing authority.
  3) Content Based Features:
     - Logo and Brand Analysis: Compare the website’s logo and visual elements to known legitimate brands using visual similarity checks. - HTML Structure and Links: Analyze the number of external links and form actions (like login buttons or input fields).
     - Keywords and Text Patterns: Search for words commonly used in phishing attempts, such as ”verify,” ”account,” or ”urgent.” - JavaScript Activity: Phishing sites may contain obfuscated or suspicious JavaScript patterns.
- Analyze Features with the Detection Model-
  Input the extracted features into a machine learning or heuristic-based model that has been trained to classify phishing versus legitimate sites. If using a machine learning model (e.g., SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost, SVM, or AdaBoost), the model can weigh the different features and return a phishing probability score.
- Make a Decision-
  Based on the model’s output, determine if the website is potentially phishing. Set a threshold for classification: If the score exceeds a certain threshold, classify the site as phishing.

III. Results

The phishing detection model in this study was developed using a combination of text-based feature extraction for feature extraction, SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost and Naive Bayes classifiers, and a diverse set of phishing datasets. Results were measured on key performance indicators, including accuracy, precision, recall, F1-score, and computational efficiency, to evaluate the model’s efficacy in identifying website-focused phishing attacks, including email-based, website, and .

Feature Optimization: Integrating text-based feature extraction en- abled significant dimensionality reduction without compromis- ing model accuracy, streamlining processing while maintaining high precision. This approach proved effective in extracting relevant features from both emails and web content, contribut- ing to a high F1-score (0.97) and underscoring the robustness of feature extraction for phishing detection.

Multi-Channel Phishing Detection: The model successfully extended its detection capabilities to website phishing through URL, domain, and content-based feature extraction. It analyzed indicators such as URL length, domain age, SSL certificates, logo similarity, and JavaScript patterns, achieving high accuracy in differentiating phishing from legitimate websites. The inclusion of (vishing) detection further enhanced the model, yielding a 10

Computational Efficiency: The SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost model main- tained optimal performance with moderate computational de- mands, making it suitable for scalable deployment in real- world applications. In contrast, Naive Bayes demonstrated efficiency in resource-constrained environments, offering a most accurate model in our tests. These results highlight the benefits of a well-optimized SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost-based approach, coupled with text-based feature extraction, in achieving high accuracy and adaptability for phishing detection for websites.

IV. Conclusion

This study presents a comprehensive, website-focused phishing detection model that leverages machine learning and natural language processing to enhance cybersecurity defenses. By integrating text-based feature extraction with SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost and Naive Bayes classifiers, the system demonstrates robust accuracy in identifying phishing for websites while maintaining computational efficiency. The inclusion of website phishing detection underscores the model’s adaptability to various phishing tactics, addressing threats from multiple channels.

By reducing false positives and achieving high recall, the system demonstrates its potential for real-world deployment,

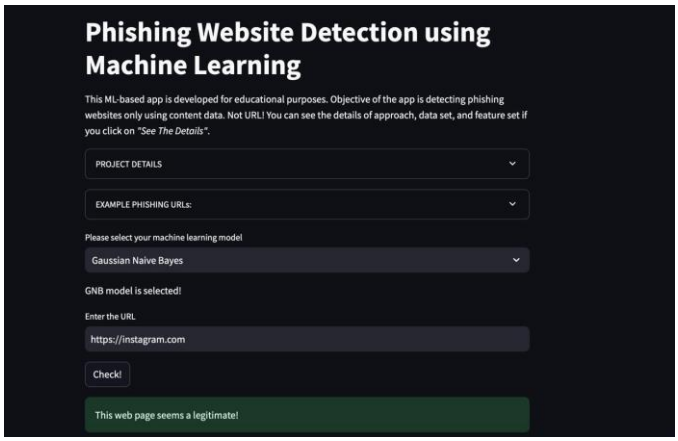

Fig. 4. output if website is not spam

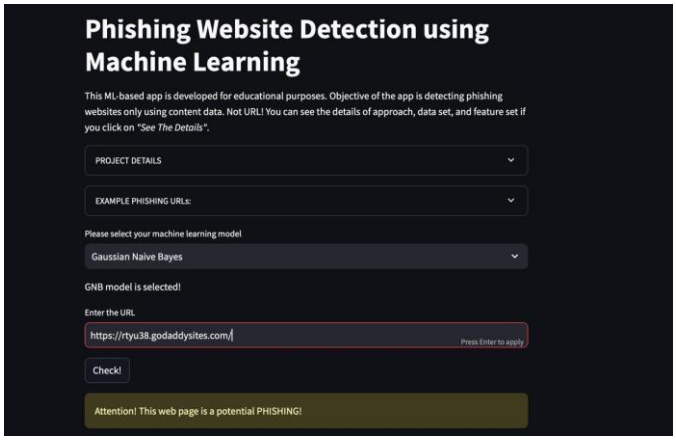

Fig. 5. output if website is spam

where precision and rapid threat detection are essential. Fu- ture enhancements could explore incorporating deep learning techniques for improved accuracy, particularly in detecting so- phisticated website phishing tactics. Additionally, integrating real-time adaptive learning mechanisms would further bolster the model's resilience against evolving phishing strategies.This study underscores the importance of machine learning in creating scalable, accurate, and adaptable phishing detection solutions to protect against the growing sophistication of cyber threats across multiple domains.

V. Study Limitations

This study possesses multiple limitations that affect the interpretation and applicability of its results. Data restrictions are a factor; a relatively small or imbalanced dataset, par- ticularly with a scarcity of phishing samples, may diminish the model's capacity to generalize effectively to real-world scenarios. Moreover, substandard or biased data (e.g., region- specific or disproportionately focused on certain phishing types) can impede model efficacy and restrict generalizability. The selected features and model complexity impose limita- tions—exclusive reliance on particular features, such as text content or audio attributes, may result in the model over- looking certain phishing patterns. Although complex models, like deep learning, can achieve high accuracy, they frequently lack interpretability, complicating the understanding of why specific messages are classified as phishing. The implemen- tation in real-world scenarios presents additional constraints, as practical considerations including real-time performance, latency, and adaptability to changing phishing strategies are essential elements that may not be thoroughly examined in this study. Ethical and societal limitations, such as user privacy and possible prejudices, are critical factors, especially when managing audio data or sensitive material. Ultimately, conven- tional evaluation criteria such as accuracy may inadequately reflect the model's efficacy in identifying phishing in real- world contexts, where practical implementation frequently ne- cessitates a high level of sensitivity and specificity. Identifying these constraints is crucial for contextualizing the data and pinpointing avenues for future research and enhancement.

VI. Future Scope

Future work could explore hybrid models that combine text, audio, and image data to detect more sophisticated attacks. Enhancing feature extraction and model interpretability is also crucial, as explainable AI techniques can help users understand why communications are flagged, fostering trust and compliance. Optimizing for real-time detection is another key area, enabling models to process data quickly for live filtering in emails, websites, or calls. Incorporating continuouslearning mechanisms would allow the model to stay updated on emerging threats. Lastly, privacy-preserving methods like federated learning could enable secure training on user data, addressing both privacy and security. Together, these advance- ments could significantly improve the effectiveness and ethical use of phishing and vishing detection technologies.

For future enhancements, implementing encryption on the backend API and adding robust authentication mechanisms could significantly bolster security for the phishing detection extension. By encrypting API communication, sensitive data, including feature inputs and model outputs, can be securely transmitted, preventing interception and potential manipula- tion by unauthorized parties. Additionally, integrating strong authentication ensures that only trusted users and applications can access the backend, mitigating risks of unauthorized model manipulation or abuse.

We are also introducing a feedback feature, allowing users to report if a website was incorrectly flagged (false positive) or missed (false negative). This user feedback will enable continuous model refinement, ensuring PhishVault's detection accuracy improves over time. By combining these security measures with user input, PhishVault becomes a more re- silient and trustworthy tool, better equipped to adapt to real- world threats and meet evolving phishing tactics.

References

[1] Tushaar Gangavarapu, C. D. Jaidhar Bhabesh Chanduka:Applicability of machine learning in spam and phishing email filtering: review and approaches
[2] DeepDetection: Privacy-Enhanced Deep Voice Detection and User Authentication for Preventing Voice Phishing by Yeajun Kang,Wonwoong Kim,Sejin Lim,Hyunji Kim andHwajeong Seo
[3] HearMeOut: detecting voice phishing activities in Android Authors: Joongyum Kim, Jihwan Kim, Seongil Wi, Yongdae Kim, Sooel Son

[4] Classification of Phishing Email Using Word Embedding and Machine Learning Techniques Somesha M. and Alwyn R. Pais
[5] Acoustic Signature Analysis for Distinguishing Human vs. Synthetic Voices in Vishing Attacks Prarthana Gamage; Dushan Dissanayake; Niroopama Kumarasinghe; Gamage Upeksha Ganegoda
[6] Classification of Phishing Email Using SVM, Naive Bayes, Decision Tree, Random Forest, and AdaBoost Machine Learn- ing Technique Andronicus A. Akinyelu, Aderemi O. Adewumi
[7] Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics Chidimma Opara a, Yingke Chen b, Bo We