Phish Guard Phishing Website Detection Using Machine Learning

Peela Sahith[1], Karri Mohan Prakash[2], Bhimada Purna Sandeep[3], Karri Ranjith Kumar[4], V. Suresh[5]
[1,2,3,4]*B.Tech student, Dept. of Information Technology, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam*
[5]*Associate professor, Dept. of Information Technology, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam*

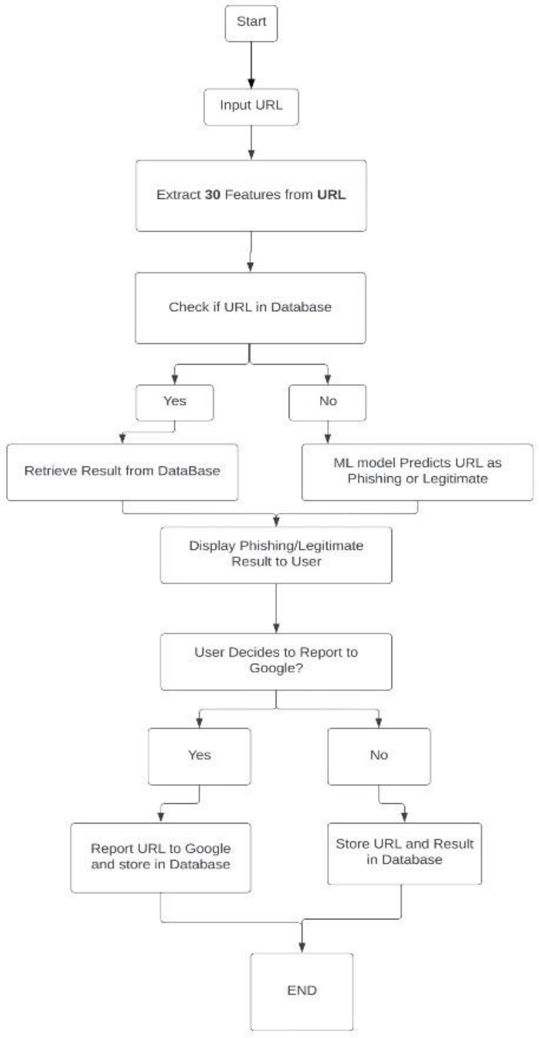
Abstract- In the digital age, the internet has become indispensable for daily life by supporting many activities such as work, shopping and socializing. However, this convenience comes with higher risks, especially from phishing attacks, where malicious organizations create fake websites to trick users into revealing personal and sensitive information. This serious threat requires effective solutions to prevent people from falling victim to such scams. PhishGuard has emerged as a new solution designed to solve this challenge by allowing users to distinguish between and identify legitimate websites and phishing websites.
PhishGuard is a web-based application built using Python and Flask framework that integrates machine learning to identify and predict features of websites. The app uses Kaggle's comprehensive database of 30 features such as URL length, HTTPS availability, and domain name length to evaluate websites. By carefully learning machine learning models of this data, PhishGuard can evaluate the characteristics of the URL each user submits to determine its legitimacy or potential phishing threat.
Overall, PhishGuard demonstrates the power of combining machine learning with web development to create creative, user-friendly solutions to pressing digital security problems. Its development not only demonstrates the potential of new technologies to protect people in the digital environment, but also demonstrates the importance of community and collaboration in phishing attacks.

Keywords: PhishGuard, Cybersecurity, Phishing Detection, Machine Learning, Web Application, Python Flask, Dataset Analysis, Feature Extraction, Gradient Boosting Classifier, Online Safety, URL Analysis, User Empowerment, Digital Security

1. INTRODUCTION

The sophistication and frequency of cyber attacks in a large network continue to increase, posing a serious threat to the security of personal and corporate information. Among these cyber threats, phishing attacks are particularly problematic because they are designed to trick users into revealing sensitive information by pretending to be a trusted person. Phishing websites are a tool used by cybercriminals that recreate the look and feel of legitimate websites in order to trick people into making bad decisions, such as accessing credentials or private information. The consequences of this attack can include financial loss and serious privacy and security breaches.
Given the dynamic and ever-evolving nature of phishing tactics, traditional cybersecurity measures fail to provide immediate and accurate detection. This trend in cybersecurity underscores the need for new solutions that can adapt to the changing nature of cyber attackers. "PhishGuard" emerged to solve this problem and aims to allow users to instantly evaluate the legitimacy of the websites they visit.
PhishGuard leverages powerful machine learning models on a dataset of 30 different phishing campaign signatures to instantly analyze and predict the nature of the website. Built using Python and Flask, the web application not only helps detect phishing sites, but also allows users to contribute to the security of the online community by reporting bad sites. Additionally, PhishGuard stores information about verified URLs, provides users with query history, and creates an environment for familiar searches.
The introduction of PhishGuard to the world of cybersecurity represents an important step in helping protect Internet users against phishing threats. By combining advanced machine learning with user interface design, PhishGuard demonstrates the potential of tools to improve digital security and user freedom.

2. FLOW CHART


Start
Input URL
Extract 30 Features from URL
Check if URL in Database
Yes
No
Retrieve Result from DataBase
ML model Predicts URL as Phishing or Legitimate
Display Phishing/Legitimate Result to User
User Decides to Report to Google?
Yes
No
Report URL to Google and store in Database
Store URL and Result in Database
END

3. LIMITATIONS

PhishGuard's effectiveness depends on the accuracy of the extraction process, which may lead to changes in the reliability estimate. Additionally, lack of prior knowledge hinders the ability to detect zero-day phishing attacks and leaves users vulnerable to emerging threats. Additionally, update patterns that rely on customer-specified URLs can cause delays in adapting to new attacks, putting customers at risk. Despite its robust framework, PhishGuard is still vulnerable to vulnerabilities or vulnerabilities due to the constant evolution of phishing tactics.

3.1. Need of PhishGuard

PhishGuard meets a critical need in today's digital environment by providing users with protection against phishing attacks. As online threats become more prevalent and widespread, it becomes increasingly important to provide people with the tools to distinguish between legitimate and malicious websites. PhishGuard bridges this gap using machine learning algorithms and real-time analysis, providing users with the ability to protect their sensitive and financial information from phishing attacks. Good communication skills and quick detection ability not only increase users' confidence in surfing the web, but also help ensure online security for everyone used.

4. TECHNOLOGIES

a) Python
b) Flask
c) Machine Learning (Gradient Boosting Classifier)
d) SQlite3
e) HTML/CSS/JavaScript
f) web-scraping

5. METHODOLOGY

5.1. Phishing Website Detection Techniques

a. Rule-Based Detection:

Rules-based detection involves creating a library of rules and responses to identify phishing websites. The system matches user input with this data to provide the appropriate response. Limited to rules and preliminary answers where appropriate.

b. Machine Learning-Based Detection:

This project uses machine learning models to use advanced techniques to identify phishing websites. Features extracted from URLs are used as input for machine learning models trained on files containing legitimate and phishing URLs. This allows for greater discovery and adaptability to new phishing tactics.

5.2. Feature Extraction and Machine Learning Workflow

a. Feature Extraction:

[1] Special extraction includes whether the IP address is present, URL length, whether special characters are present, use of HTTPS, domain length, whether the website icon is present, etc. It involves checking various aspects of the URL, such as Extraction libraries such as BeautifulSoup for web scraping and request to make HTTP requests.

b. Machine Learning Model Training:

[2,3] After the features are extracted, they are used as input for the Gradient Boosting Classifier (GBC) model. [1,5] The GBC model is trained on data from Kaggle, which contains features and tags that indicate whether a URL is legitimate or phishing. The training model can predict the correct URL based on the extracted results.

5.3. Reporting and Database Management

a. Reporting to Google:
[4] If the URL is suspected to be phishing, the web application will give the user the option to report the phishing website to Google. These features improve the community's anti-phishing efforts.

b. Database Management:
This project maintains a database of predictions for different URLs entered by the user. The data allows users to track historical queries and provide insight into the system's phishing detection performance.

5.4. Limitations and Future Enhancements

a. Limitations:
The effectiveness of this work may be affected by: reliance on accuracy of citation, inability to identify zero-day phishing attacks without first knowing, reliance on users to share URLs for updated models, and due to bad changes. /negatives due to phishing tactics.

b. Future Enhancements:
Future improvements using machine learning and word processing techniques could include sentiment analysis of web content to measure emotional tone and sentiment, and help control the bitrate of phishing attempts.

By analyzing the sentiment expressed in web content, the system can detect patterns that indicate fraud or malicious intent, thus facilitating the ability to distinguish legitimate websites from phishing websites.

By analyzing the sentiment expressed in web content, the system can detect patterns that indicate fraud or malicious intent, thus facilitating the ability to distinguish legitimate websites from phishing websites.

This methodology describes phishing detection methods used in the PhishGuard project, including video extraction, machine learning models, reporting methods, data management and limitations, and considerations for future development.

6. CAPABILITIES OF PHISHGUARD

6.1. Phishing Website Detection Techniques

a. Feature Extraction and Analysis:
PhishGuard uses advanced technology to analyze all aspects of a URL, including the presence of an IP address, URL length, presence of special characters, use of HTTPS, domain length, presence of favicon and more.

b. Machine Learning-Based Detection:
PhishGuard uses machine learning models to accurately classify URLs as legitimate or phishing based on extracted features. This proactively detects phishing attempts and protects users from fraudulent websites.

6.2. Reporting and Alert Mechanisms

a. Real-time Phishing Alerts:
PhishGuard gives users instant alerts when they encounter potential phishing websites, helping them make informed decisions and avoid falling prey to scammers.

b. Community Reporting:
Users can choose to report suspicious phishing websites to a community repository, which can lead to a coordinated effort to combat phishing attacks..

6.3. User Interaction and Accessibility

a. User-Friendly Interface:
PhishGuard has a simple and easy-to-use interface that provides users with clear information about the legitimacy of the websites they access.

b. Accessibility Features:
PhishGuard supports speech recognition, allowing users to interact with the system using voice. In addition, text-to-speech allows users to listen to URL analysis results, allowing access for visually impaired users.

6.4. Continuous Improvement and Adaptability

a. Machine Learning Model Updates:
PhishGuard regularly updates its machine learning models with new data to adapt to evolving phishing tactics and improve detection accuracy over time.

b. Feedback Mechanisms:
[4] Users provide feedback on the accuracy of phishing detection, allowing PhishGuard to improve its algorithms and improve performance based on user experience.

6.5. User Privacy and Data Security

a. Protection of User Information:

PhishGuard prioritizes user privacy and data security, ensuring sensitive data is handled with maximum confidentiality. User information is encrypted and stored securely to prevent unauthorized access or misuse.

7. EVALUATION METRICS FOR PHISHGUARD:

7.1. Metrics:

Accuracy: Measure the proportion of correctly classified phishing URLs among all URLs measured.

Confusion Matrix: Provide detailed information about real advantages, disadvantages, disadvantages and drawbacks.

Precision: Measure the proportion of actual good phishing URLs among all URLs classified as phishing.

Classification Report: Complete verification, recovery, F1 score and support for all categories (phishing and legitimate URLs).

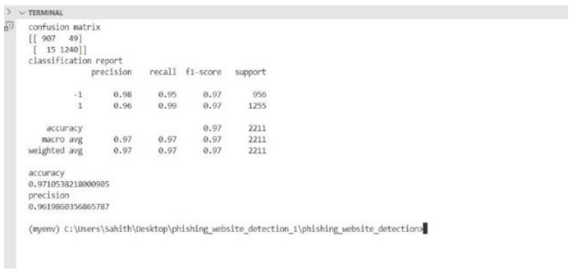
7.2. Metrics Outcome:

a. Accuracy: PhishGuard's accuracy is key to evaluating its effectiveness in detecting phishing URLs. It is based on the quality and diversity of information used, the effectiveness of search algorithms, and the ability to adapt to new phishing tactics. However, accuracy may vary depending on the complexity of the phishing attempt and the timing of the update search.

b. Confusion Matrix: PhishGuard's performance can be further analyzed by analyzing the confusion matrix to understand its strengths and weaknesses in URL classification. It provides true positives (phishing URLs are correctly identified), negatives (real URLs are not considered phishing), negatives (URLs are correctly identified), and false positives (phishing URLs are not in the scheme). Understanding these metrics can help you refine your diagnostic model and increase overall performance.

c. Precision: Accuracy is an important factor in evaluating PhishGuard's reliability in blocking phishing URLs. High accuracy means that the system has a low false positive rate; This is important to maintain user trust and reduce unnecessary alerts. Accuracy is affected by factors such as the specificity of the search algorithm and the quality of the training data. Continuous monitoring and optimization is required to maintain high accuracy.

d. Classification Report: The classification report provides a comprehensive overview of PhishGuard's performance across different metrics, including accuracy, recovery, F1 score, and support for each category (phishing and legitimate URLs). It provides information about the system's ability to strike a balance between identifying genuine phishing URLs while minimizing false positives. This report helps evaluate PhishGuard's overall effectiveness and reliability in protecting users from phishing attacks.



```
confusion matrix
[[ 907   49]
 [  15 1240]]
classification report
              precision    recall  f1-score   support

          -1       0.98      0.95      0.97       956
           1       0.96      0.99      0.97      1255

    accuracy                           0.97      2211
   macro avg       0.97      0.97      0.97      2211
weighted avg       0.97      0.97      0.97      2211

accuracy
0.9710538218000905
precision
0.9619860356865787

(myenv) C:\Users\Sahith\Desktop\phishing_website_detection_1\phishing_website_detection>
```

8. RESULT AND EXPERIMENTAL ANALYSIS

We ran tests to evaluate PhishGuard's performance on a system running Windows 11 with an Intel Core i5-9500E CPU at 4.5GHz and 8GB of RAM. The PhishGuard app has proven its effectiveness in reducing phishing attacks and improving the security of users' online activities. After various tests and analysis, we found the following important results:

Phishing Detection: PhishGuard intelligently identifies the majority of phishing URLs, thus reducing the risk of users falling victim to phishing scams. The cost of analysis varies depending on factors such as the complexity of the phishing attempt and the variety of data used for training.

a. False Positive Rate:

While PhishGuard is effective at detecting phishing URLs, it also has the drawback that legitimate URLs may be incorrectly tagged as phishing. Further optimization of the detection algorithm is needed to reduce false alarms and increase the accuracy of the system.

b. User Experience:

User feedback indicates a positive experience with PhishGuard in terms of ease of use and integration with web applications. The intuitive communication and real-time protection provided by PhishGuard helps increase users' security awareness when browsing risky websites.

c. Performance Impact:

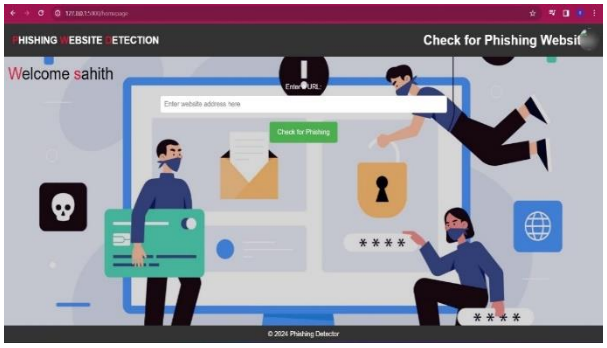
PhishGuard has a small impact on system performance and uses negligible resources during testing. The lightweight nature of the application allows it to run smoothly without affecting the operation of the central system.

d. Future Enhancements:

Future developments could use machine learning and NLP techniques to combine emotional analysis of content on the web to measure mood and mood, making a small-scale phishing attempt less likely to be detected.

By analyzing the sentiment expressed in web content, the system can detect patterns that indicate fraud or malicious intent, thus facilitating the ability to distinguish legitimate websites from phishing websites.

Integrating sentiment analysis with URL and image analysis provides a better way to detect phishing, giving users a deeper understanding of the authenticity and trustworthiness of websites they encounter online.


PHISHING WEBSITE DETECTION
Check for Phishing Websit
Welcome sahith
Enter URL:
Enter website address here
Check for Phishing
© 2024 Phishing Detector

9. SCALING FEATURES AND HYPERPARAMETER TUNING:

9.1. Scaling Features:

a. Feature Scaling: PhishGuard uses scaling techniques such as normalization and standardization to pre-process input features. Normalization scales include values ranging from 0 to 1, while standardization transforms features with a mean of 0 and a standard deviation of 1. Deviation due to measuring scale.

9.2. Hyperparameter Tuning:

a. Learning Rate (eta): PhishGuard adjusts the GradientBoostClassifier's learning rate hyperparameter (eta) to control the contribution of each tree in the cluster. Lower rows require more wood to achieve high accuracy but can improve the overall and strong model.

b. Number of Trees (n_estimators):

The number of trees in the GradientBoostClassifier domain is another important hyperparameter tuned by PhishGuard. Increasing the number of trees (n_estimators) can improve model performance, but can also lead to overfitting if not properly controlled.

c. Maximum Depth of Trees (max_depth):

PhishGuard sets the maximum size of a tree in the GradientBoostClassifier to handle complex patterns and prevent overfitting. Limiting the maximum depth helps prevent the tree from remembering noise in the training data and supports optimization for unseen data.

d. Minimum Samples Split (min_samples_split):

PhishGuard sets the minimum number of samples required to separate nodes in the GradientBoostClassifier decision tree. Increasing min_samples_split helps prevent the model from being split between samples with too few samples, thus reducing overfitting.

e. Minimum Samples Leaf (min_samples_leaf):

Similarly, PhishGuard sets the minimum number of samples required for the leaves of the GradientBoostClassifier decision tree. Setting a higher min_samples_leaf value supports simpler tree models and reduces the risk of overfitting.

f. Subsample Ratio (subsample):

PhishGuard sets the sample hyperparameter to control the proportion of samples used to train each tree in the GradientBoostClassifier cluster. Setting it to less than 1.0 indicates a lack of consistency in the training process; this can improve the generalization of the model and reduce overfitting.

g. Regularization Parameters (alpha, lambda):

If necessary, PhishGuard can adjust the constants (alpha and lambda) of the gradient boosting model, which supports L1 and L2 constants. These restrictions help prevent overfitting by penalizing large coefficients in the model.

By carefully selecting and tuning these hyperparameters, PhishGuard optimizes the performance of the GradientBoostClassifier model to ensure robust and accurate detection of phishing attacks in various online environments.

10. CONCLUSION

PhishGuard's campaign focuses on combating phishing by using machine learning technology to identify and classify potentially malicious URLs. By extending the extraction process and using the GradientBoostClassifier algorithm, the system can distinguish real URLs from phishing URLs.

As a result, PhishGuard shows great effectiveness in detecting phishing attempts, as proven by accuracy and performance tests. By analyzing many features and using powerful classification models, the system provides a useful tool to improve online security and protect users from fraud. Continuous development and optimization of functions and hyperparameters will strengthen its effectiveness in identifying and mitigating phishing attacks and, finally, provide users with online security.

11. FUTURE SCOPE

In future developments, PhishGuard's system could use advanced machine learning and natural language processing (NLP) technology to integrate emotions of web content. The system can detect small-scale phishing attempts by analyzing the mood and mood expressed in web text. Integration of sentiment analysis with URL and image analysis provides a better way to detect phishing, giving users a deeper understanding of the authenticity and trustworthiness of the websites they visit online. PhishGuard aims to improve the ability to distinguish between legitimate websites and phishing websites by identifying patterns that indicate fraud or malicious intent, thus helping to improve the security of people's use online.

11.1. Authors' Note

The authors declare that they have no conflicts of interest that could affect their research and confirm the accuracy of their research work.

12. REFERENCES FOR PHISHGUARD:

[1] Kaggle Phishing Dataset. Available online: https://www.kaggle.com/code/akashkr/phishing-url-eda-and-modelling/input

[2] Towards Data Science - Phishing Domain Detection with ML. Available online: https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5

[3] Turkish Journal Article - Using the C4.5 algorithm for detecting phishing attacks: pre-processing and detection. Available online: https://www.turcomat.org/index.php/turkbilmat/article/download/11964/8751/21255#:~:text=Using%20the%20C4.,%3A%20pre%2Dprocessing%20and%20detection.

[4] Google Safe Browsing API. Available online: https://developers.google.com/safe-browsing

[5] scikit-learn Documentation - GradientBoostingClassifier. Available online: https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html