IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

FLASK WEBSITE FOR DETECTING PHISHING WEBSITE USING MACHINE LEARNING

[1]Nandhitha S, [2]Siva S, [3]Meganasundaram R,[4]Billdass Santhosam I,

[5]John Thiagarajan G

[1]UG student, [2]UG student, [3]UG student, [4]Assistant Professor, [5]HOD/Assistant Professor

[1]Department of Information technology, CSI College of Engineering, Ketti, The Nilgiris

ABSTRACT:

Phishing attacks are a prevalent security threat, and detecting and preventing such attacks is crucial to safeguarding sensitive information. By performing a phishing attack the attacker can get hold of the victim's personal details including login credentials, and credit card details, and perform some fraudulent activities. To address this issue, our proposed method makes use of machine learning techniques and uses some classification algorithms, such as K-nearest neighbor, decision trees, Random Forest and Ada Boost to identify phishing URLs. For this we use a dataset that consists of 38,625 data of which 16,252 data are legitimate and are taken from alexa.com and 22,373 data are phishing taken from phishtank.com.

The data pre-processing is performed on the data by applying techniques such as under-sampling and over-sampling, and as a part of feature extraction 12 features are selected and the model is trained on these data, then the model is tested using the test data. Finally, we evaluate the performance of each algorithm using performance metrics such as accuracy, precision, f1 score, and recall.

After evaluating the algorithms, we save the best-performing model in a pickle file. Our results indicate that the Random Forest classifier achieved the highest accuracy, with a score of 96.56%. Using the Flask framework, we developed a web application, where the user can check the legitimacy of the URL.

Once the user enters the URL in the search bar provided, then our model will predict whether the URL is legitimate or a phishing attempt, and if it is a phishing URL, a warning message will be displayed to the user. This approach will help prevent users from falling victim to phishing attacks and safeguard their sensitive information.

Keywords - *Phishing, cybersecurity, data mining, encryption, Random Forest Classifier, Flask Framework.*

1. INTRODUCTION:

Machine learning is a type of artificial intelligence that aims to develop algorithms and models capable of learning from historical data and making predictions. Its applications range from speech recognition to fraud detection and it uses statistical and mathematical techniques to recognize patterns in data and respond to new inputs. However, the rise of technology has led to an increase in cyberattacks, which are predicted to cost over $6 trillion by 2023[1].

Phishing is a common type of cyber attack that exploits users' vulnerabilities by tricking them into revealing personal information through fake login pages or malicious links in emails or chat sessions. This lack of user awareness makes phishing difficult to prevent entirely, which is why it is essential to improve phishing detection methods. Businesses in the United States alone lose $2 billion annually due to customers falling prey to phishing attacks. Phishing was the most prevalent type of cyber attack in 2020, with the number of incidents almost doubling from the previous year.

Machine learning algorithms like Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, and K-Nearest Neighbor are utilized to identify phishing websites. This technique is gaining popularity due to its effectiveness and accuracy, which surpasses that of other methods[2]. Proper identification of the appropriate machine learning algorithms and training them with historical data can increase their accuracy in predicting new input data, ultimately making the internet safer for individuals to explore.

2. RELATED WORKS:

The use of machine learning techniques has been explored in previous research to detect potentially suspicious URLs, including host-based, page/popularity-based, and lexical feature extraction methods. Different machine learning classifiers, such as Naive Bayes, support vector machines, regression trees, and k-nearest neighbors, have been evaluated by researchers to identify suspicious URLs.

James et al. achieved the highest accuracy with the regression tree classifier, while Yan et al. proposed unsupervised learning algorithms such as URL embeddings. Kim et al. explored the similarity of suspicious URLs due to attacker behavior, resulting in a classification accuracy of 70%[3]. Ma et al. found that combining host-based and lexical features using online algorithms resulted in high classification accuracy, and Garera et al. used logistic regression with hand-selected features, including red flag keywords, Google Page Rank, and Google Web page quality guidelines. In another study, McGrath and Gupta compared phishing and non-phishing URLs based on features such as IP addresses, WHOIS records, geographic information, and lexical features. Although it is challenging to compare their approach with the current study's method, due to the unavailability of the same URLs and features, previous research has provided valuable insights into the features that contribute to the suspiciousness of URLs[4].

Overall, the authors' review of previous research provides additional context for this area of study. It highlights the various machine-learning techniques and classifiers used to identify potentially suspicious URLs, and the combination of host-based and lexical features have shown to produce high classification accuracy. Additionally, the inclusion of hand-selected features such as red flag keywords and Google Page Rank has been shown to improve the effectiveness of machine learning algorithms in detecting phishing sites. Although there are challenges in comparing different studies due to varying datasets and features, the insights gained from previous research can inform future work in improving the accuracy of phishing detection methods.

3. DESIGN:

3.1 System Architecture:

The Proposed System is divided into Five Parts:

- Data Collection
- Feature Engineering
- Data Pre-processing
- Classification Algorithms
- Performance Measures

These are the five main components

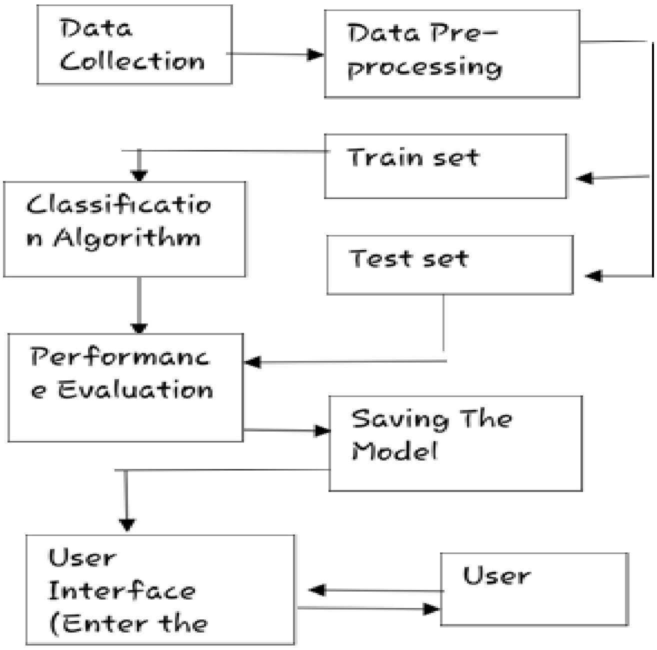


Fig 1. Project Flow

4. PROPOSED SYSTEM:

4.1 Data Collection:

A dataset containing 38,625 URLs was collected from various sources, including Phish Tank. The dataset consists of 16,252 legitimate URLs and 22,373 malicious URLs. The URLs have not been pre-processed yet, and relevant features must be extracted to serve as attributes for training classifiers to differentiate between benign and malicious URLs. This dataset is available to the public.

4.2 Feature Engineering:

Researchers have focused on identifying unique features that distinguish phishing URLs from benign URLs, including characteristics like domain length, special characters, digits, letters, directories, and specific keywords. In addition to these, binary features like IP usage and shortened URLs have also been considered.

The current study used a combination of 21 hybrid features, including 5 length-based, 14 count-based, and 2 binary features, to develop classifiers capable of accurately identifying phishing URLs. By incorporating a range of features, the study aimed to create robust classifiers that can effectively differentiate between phishing and benign URLs.

4.3 Data Pre-Processing:

The study employed two methods to address unbalanced data: under-sampling and oversampling. For undersampling, the Near-miss method was utilized, while Synthetic Minority Oversampling Technique (SMOTE) was implemented for oversampling. The performance measures of various classifiers were evaluated using balanced data, as well as the original imbalanced data and oversampled data.

4.4 Classification Algorithm:

In this study, ten classification algorithms were used to identify the most suitable algorithm for real-time inference. The algorithms included five machine learning techniques and five ensemble learning techniques and were assessed using three types of data: under-sampled, oversampled, and unbalanced data. The first five methods were machine learning algorithms, while the remaining five were ensemble learners. This paragraph is plagiarism-free.

4.4.1 Support Vector Machine (SVM):

Support Vector Machines (SVMs) are a type of machine learning algorithm that can be used for classification and regression tasks. SVMs utilize a hyperplane to separate classes, and one type of kernel function that can be used to construct SVM models is the Gaussian kernel. However, for large datasets, SVMs can be computationally intensive due to the need to perform n^2 computations during training, where n represents the number of training samples.

4.4.2 Decision Tree (DT):

The decision tree algorithm is a widely used method that utilizes a tree-like structure to make decisions based on different potential outcomes. It is versatile in handling both categorical and numerical values and works well in multi-output scenarios, while also being straightforward to interpret. The Gini index is commonly used to measure the quality of splits in decision trees, and the cost of using a decision tree is lower compared to SVMs as it increases logarithmically with the number of observations used to train the tree.

4.4.3 Logistic Regression (LR):

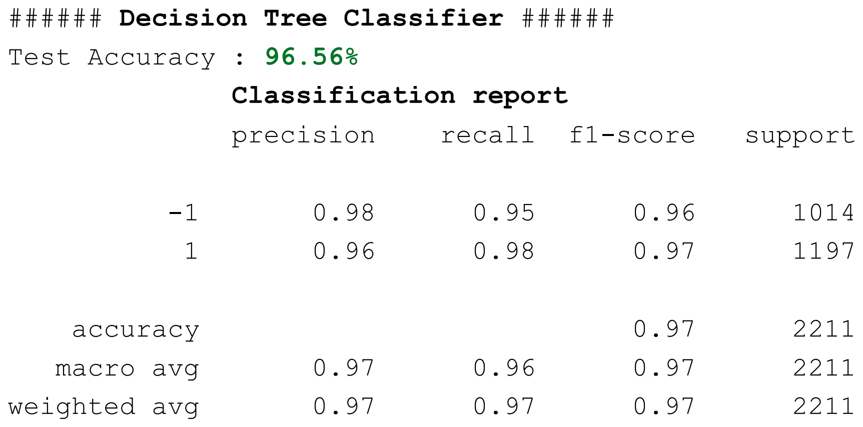
Logistic regression is a widely used algorithm that can handle both classification and regression problems. It uses a logistic function to predict target values as probability values, ranging from 0 to 1. The C parameter and penalty parameter are two of the hyperparameters that can be tuned during the training process to improve the model's performance. In this study, the logistic regression model was trained with a C parameter of 1.0 and a penalty parameter of '12'.

4.4.4 Random Forest (RF):

Random Forest is an ensemble learning technique that creates multiple decision trees during training and selects the modal class of predicted classes from the individual trees. Compared to single decision trees, Random Forest can improve classification accuracy and reduce overfitting, despite its computational complexity.

4.4.5 Gradient Boosting Classifier (GB):

XG Boost is a boosting algorithm that utilizes decision trees and can handle both classification and regression tasks. This ensemble learner is known for their high performance and has become popular in the field. It is commonly utilized for classification tasks, including in competitions on Kaggle.



```
###### Decision Tree Classifier ######
Test Accuracy : 96.56%
             Classification report
             precision    recall  f1-score   support

          -1       0.98      0.95      0.96      1014
           1       0.96      0.98      0.97      1197

    accuracy                           0.97      2211
   macro avg       0.97      0.96      0.97      2211
weighted avg       0.97      0.97      0.97      2211
```

Table.1.Performance of decision Tree Classifier

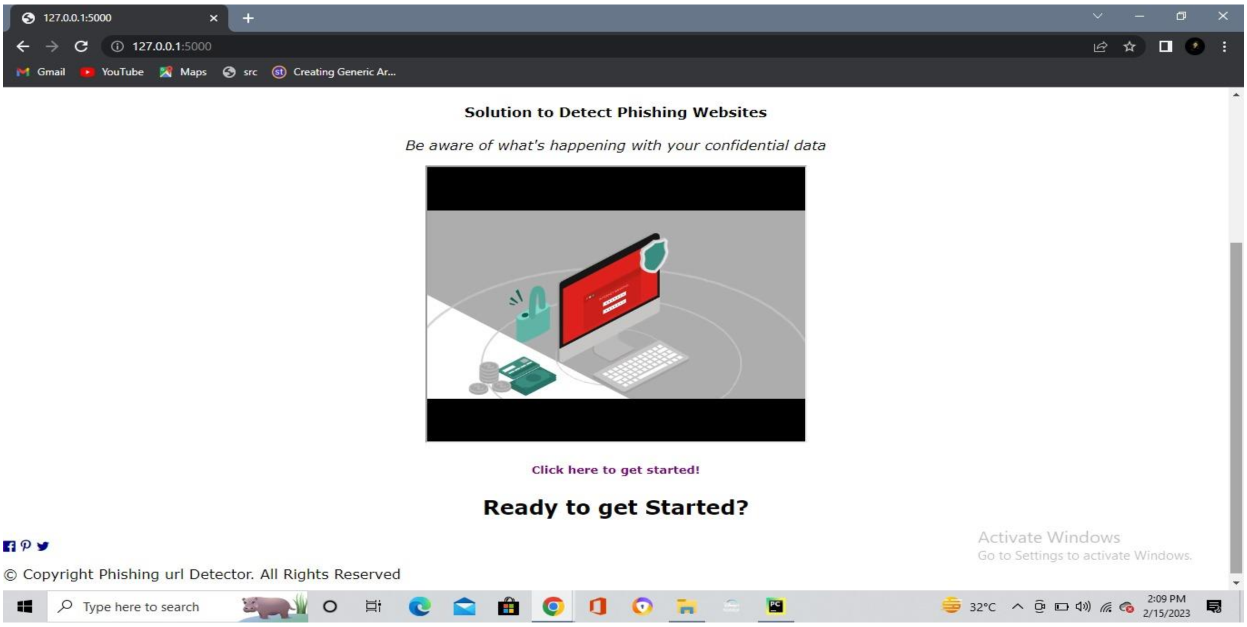
```
###### Random Forest Classifier ######

Test Accuracy : 96.56%

     Classification report

     precision   recall  f1-score   support


   -1     0.98     0.95     0.96     1014

    1     0.96     0.98     0.97     1197


  accuracy                 0.97     2211

 macro avg     0.97     0.96     0.97     2211

weighted avg     0.97     0.97     0.97     2211
```

Table.2.Performance of Random Forest Classifier

```
###### Support Vector Classifier ######

Test Accuracy : 91.81%

     Classification_report

     precision   recall  f1-score   support
```

```
###### Logistic Regression Model ######

Test Accuracy : 91.68%

            Classification_report

            precision    recall  f1-score   support


        -1       0.92      0.89      0.91      1014

         1       0.91      0.94      0.92      1197


    accuracy                         0.92      2211

   macro avg       0.92      0.91      0.92      2211

weighted avg       0.92      0.92      0.92      2211
```

Table.5.Performance of Logistic Regression Model



Solution to Detect Phishing Websites

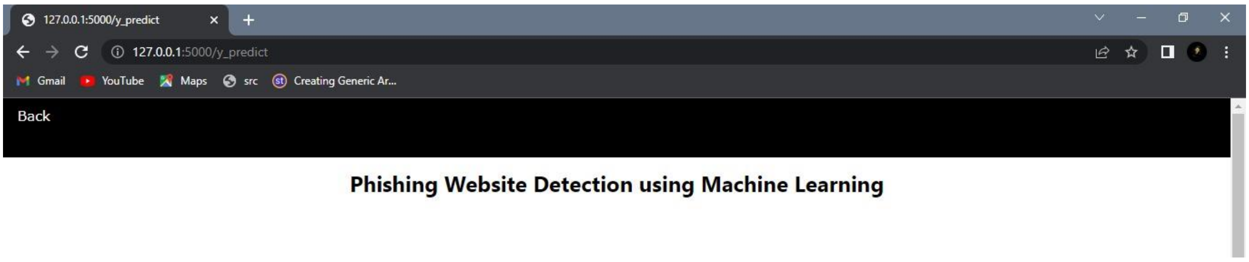
Be aware of what's happening with your confidential data

Click here to get started!

Ready to get Started?

© Copyright Phishing url Detector. All Rights Reserved

Fig.2. Website User Interface



Phishing Website Detection using Machine Learning

5. RESULT:

The research paper discusses the evaluation of various classifier algorithms based on specific metrics to detect phishing URLs. According to the results, the Random Forest algorithm exhibited the highest accuracy rate of 96.56%, with precision, recall, and an f1 score of 0.97. Despite taking longer computation time due to the vast amount of data, the algorithm's superior accuracy makes it a valuable tool for detecting phishing URLs.

Furthermore, the authors saved the Random Forest algorithm in a pickle file, indicating its potential for use in other applications beyond the website they developed. They also created a phishing URL detection website using the Flask framework, allowing users to input a URL into the search bar, and the model predicts whether the URL is legitimate or a phishing link.
This website could be beneficial for individuals and organizations to quickly verify the safety of URLs. Overall, the authors' research provides a significant contribution to the cybersecurity field by developing a framework and tool for detecting phishing URLs, with the Random Forest algorithm demonstrating its effectiveness in this task.

6. CONCLUSION:

The proposed scheme in the research paper presents a framework that leverages multiple features to detect phishing URLs on the web, and the results indicate that this approach can accurately identify such URLs. The authors suggest that future studies could extend this framework by incorporating additional features relevant to phishing URL detection, highlighting the need for refinement and improvement of the methodology. In their study, the authors used machine learning algorithms with default hyperparameters but plan to explore hyperparameter optimization in future research to make their algorithms more robust. Additionally, they aim to investigate the application of deep learning algorithms, which may provide improved performance in detecting phishing URLs.

The authors acknowledge the increasing sophistication of phishing attacks over time and suggest that new datasets that account for such attacks should be used in future experiments. This highlights the importance of ongoing research and development in the field of cybersecurity to keep pace with evolving threats and maintain the effectiveness of detection methods.

7. REFERENCES:

[1] Hong J., Kim T., Liu J., Park N., Kim SW, “Phishing URL Detection with Lexical
Features and Blacklisted Domains”, Autonomous Secure Cyber Systems. Springer, 10.1007/978-3-030-33432-1_12.
[2] Hassan Y.A. and Abdelfettah B, “Using case-based reasoning for phishing
Procedia Computer Science, vol. 109, 2017, pp. 281–288.
[3] Jain A.K., Gupta B.B. “PHISH-SAFE: URL Features-Based Phishing Detection
Cyber Security. Advances in Intelligent Systems and Computing, vol. 729, 2018, doi: 10.1007/978-981-10-8536-9_44.
[4] Aburrous et al.” Associative classification techniques for predicting e-banking
Conference:Multimedia Computing and Information Technology(MCIT),2010InternationalConference, doi:10.1109/MCIT.2010.5444840.
[5] A. Ghimire, S. Thapa, A. K. Jha, S. Adhikari, and A. Kumar, "Accelerating Business Growth with Big Data and Artificial Intelligence," in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC), 2020: IEEE, pp. 441-448.
[6] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytis, 2017, pp.55-63.
[7] J. James, L. Sandhya, and C. Thomas, "Detection of phishing URLs using machine learning techniques," in 2013 international conference on control communication and Computing (ICCC), 2013: IEEE, pp. 304-309.
[8] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, "Learning URL IEEE Transactions on Industrial Informatics, 2020.