IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detecting Phishing Websites Using Flask Web Interface As Well As Machine Learning

[1]Ms. Sanjana B Patil, [2]Mr. Kartik Kalal, [3]Mr. Swaroop R Shetty,

[4]Ms. Tanushree Shindhe, [5]Ms. Ashwini Garaddi,

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor

[12345]Department of Electronics & Communication Engineering,

[12345]KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, Karnataka, India

Abstract: Cybercriminals fabricate unauthorized copies of genuine websites and email accounts to obtain sensitive data. These emails frequently include real company logos and slogans. When users click on links supplied by these hackers, they inadvertently provide access to their confidential information, such as bank account details, personal login credentials, and photos. Even though Decision Trees and Random Forest algorithms are commonly utilized in current systems, their accuracy requires enhancement. The existing models also suffer from increased latency. Present systems do not have a dedicated user interface, and there is no comparison of different algorithms. When consumers open the emails or links provided, they are deceived into visiting a fraudulent site that imitates the legitimate company. The models are used to identify phishing websites based on URL centrality features and to determine and implement the best machine learning model. The Random Forest method will be employed to compare accuracy and results.

Key Words - Features, Machine Learning dataset, URL, Phishing.

I. INTRODUCTION

In today's world of digital connectivity, phishing attacks have become a major threat to internet users, putting their sensitive data at risk. Machine learning provides a potential remedy for these issues by providing a more sophisticated and adaptable approach to identifying phishing websites. By examining Even so, Random Forest and Decision Trees websites, Machine learning algorithms can precisely distinguish between legitimate and phishing sites. Among these algorithms, It has been demonstrated that the random forest is especially useful for identifying phishing websites.This algorithm uses an ensemble learning technique, which involves multiple decision trees to make predictions. Every decision tree is taught using a random subset of features and data, creating a diverse set of classifiers. The final prediction is made based on a majority vote among the decision trees. This ensemble method improves the model's overall accuracy and minimizes the risk of overfitting. Random forests are especially suitable for phishing detection as they can handle large datasets, manage missing data, and identify complex relationships between features. Furthermore, their non-parametric nature makes them resistant to assumptions about the underlying data distribution.

II. PROBLEM DESCRIPTION

The formula known as the random forest is an type of An approach to group learning that includes several decision trees generate predictions. Each tree is trained using a random selection of features and data, leading to a varied group of classifiers. The ultimate prediction is made based on the majority vote from these decision trees.This ensemble strategy improves the model's accuracy and minimizes the likelihood of overfitting. Random forests are particularly helpful in identifying phishing because they can process large datasets, handle missing information, and uncover intricate relationships between features. Furthermore, their non-parametric nature makes them less prone to errors due to incorrect assumptions about data distribution. Creating a scalable and efficient method to detect and prevent phishing attacks is essential for protecting internet users and their sensitive data. Current methods, which rely on manually created rules and blacklists, often fall short as phishing sites constantly change to avoid detection. Machine learning presents a potentially alternative, providing a flexible and data-driven way to recognize and categorize phishing websites.

III. LITERATURE SURVEY

In the realm of cybersecurity, phishing detection has gotten a lot of attention, which has led experts to investigate several clever strategies. Classifiers for machine learning, such as Support Vector Machines (SVM), Decision Trees, and Artificial Neural Networks, were assessed based on their capacity to identify phishing threats in a study by Abu-Nimeh et al. Their tests demonstrated that machine learning-based solutions might improve security and successfully reduce erroneous detections. In recent advancements, ensemble methods such as Random Forest and Gradient Boosting have gained popularity. Rao and Ali evaluated these techniques for phishing detection and observed improved classification results, especially in handling large and complex datasets. Based on these investigations, this research uses a Using the Random Forest approach, classify phishing sites quickly and effectively while ensuring higher detection accuracy. Preethi P. Pokal Ramadevi; K Akshaya, Sangamitra SD; Pritikha A P [6] It is crucial to have mechanisms that can determine phishing schemes as they happen a to protect users while they are online. Scientists have examined methods to improve d ata handling and enhance models along with the detection of phishing in utilizing machine learning techniques in real time. Vazhayil Anu In order to forecast the accuracy of classifying phishing URLs, Vinaya Kumar R, Soman KP et al. [7] concentrate on merging CNN with the Convolutoinal Neural Newtork, Long Short Term Memory. While LSTM gets sequential information, CNN helps find unique information between the characters. The distinctive character associations were found using CNN.

IV. URL-BASED DETECTION

URLs contain several elements—such as the protocol (e.g., http or https), domain name, subdomains, path, and query parameters—that can be individually analyzed for anomalies. Phishing websites commonly exploit these elements in creative ways. For example, attackers might use excessively long URLs to obscure suspicious elements, include misleading subdomains (e.g., paypal.login.account.com instead of paypal.com), or use an IP address in place of a domain name. Additionally, symbols like “@”, “-”, and “//” are often embedded to confuse users or redirect them to malicious destinations. Because it doesn't need viewing the actual text of the webpage, this method works well. It is rapid, effective, and perfect for systems that identify risks in real time because it just looks at the URL's text format. This tactic is frequently utilized in conjunction with machine learning in today's phishing detection systems.

URL, including how many subdomains there are, whether HTTPS is present, the age of the domain, and the use of URL shortening services like bit. which often help to hide the final destination. These characteristics are employed to teach a model for machine learning, like a Random Forest classifierthat gains knowledge of distinguish between legitimate and fraudulent URLs by identifying p trends in the instruction While URL-based detection has many benefits, such as little computational overhead and high speed, it also has drawbacks. It might have trouble identifying complex phishing attempts when the URL seems legitimate or has been designed to closely mimic a reliable website. The efficacy of this strategy alone is further diminished by the possibility that new phishing domains are not yet included in

current blacklists. Therefore, even while URL-based detection is essential for spotting phishing attacks, it is frequently combined with additional techniques like content analysis or behavior tracking to build a more complete and reliable protection system against changing online threats. Nevertheless, URL-based detection has drawbacks in addition to its benefits. It might have trouble with complex phishing efforts that make use of well constructed URLs or newly registered domains that aren't currently on the blacklist. Furthermore, certain phishing websites may have URLs that appear authentic yet fool users with misleading content or behavior that is impossible to detect with URL analysis alone.

IV. IMPLEMENTATION

The detection model in This study is built using the Random Forest technique, It is for its exceptional precision, resilience, and ability to handle noisy and unbalanced datasets.

The goal of this project is to create a system that analyzes URLs and detects This project's objective is to develop a system that. A Flask-built web interface allows users to gain access to the system by entering a URL and receiving immediate verification of its legitimacy.

Modules

- Data collection
- Data preprocessing
- Feature extraction
- Module selection
- Analysis

An explanation of the modules

Data preprocessing

Data preparation is the initial step in the implementation process. The model's training and testing dataset, which comprises a sizable collection of records from both authentic and fraudulent websites, was sourced from a reliable source. Every dataset entry includes a variety of URL-extracted information, including URL length, special character presence, domain registration length, and HTTPS usage. After that, the dataset is cleaned up by addressing any missing values and making sure any category features are suitably encoded into numerical form so the They can be processed more readily by a machine learning model.

Feature extraction

The most pertinent signs for phishing detection are then found through feature extraction. due to their strong correlation with phishing activity, features including "having IP address," "URL length," "prefix-suffix in domain," and "SSL certificate validity" were selected. These characteristics are essential for assisting the model in differentiating between trustworthy and dangerous websites. Techniques for the selection feature can also be accustomed to lower dimensionality and enhance the effectiveness and execution of the model.

Module training

Once the data is ready, The Random Forest classification system is trained using the labeled dataset. Random Forest is a method of ensemble learning that creates numerous decision trees throughout training and generates the mode of the classes (classification) of each individual tree. It reduces the likelihood of overfitting and ensures a more widely applicable model by averaging the results of multiple decision trees. The classifier gains knowledge from previous data. to identify patterns associated with phishing websites.

Evaluation

Following training, the model is assessed using common measures including F1-score, recall, accuracy, and precision. These metrics aid in assessing the model's performance in distinguishing between authentic and phishing websites.

Flask Web Interface

1. User Interface

Input: A text box for users to enter a URL.

Output: A message indicating whether the URL is likely phishing or legitimate.

2. Backend Functionality

Receives the URL input from the user.

utilizes the same feature extraction procedure as when training the model.

loads the Training of a Random Forest model

(phishing_model.pkl).

predicts whether the URL is legitimate.

brings the outcome back to the user interface.

Deployment

Lastly, Python frameworks like Flask or Django are accustomed to incorporate The model that was trained

Into an online application. When a user enters a URL

into the interface, the backend system utilizes the trained Using the Random Forest model, determine the

URL as either authentic or phishing after extracting features from the input. Determine, using the Random Forest model, the

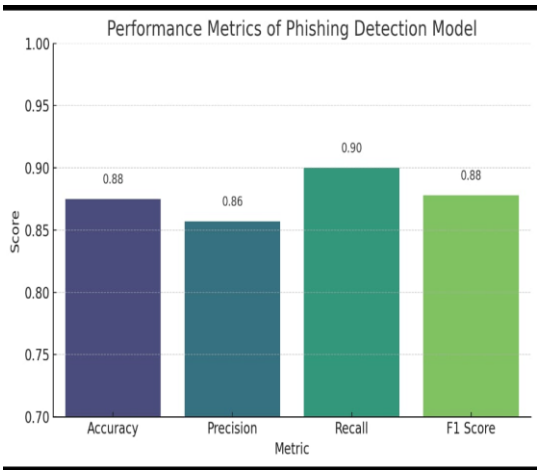
in real time by deploying it on cloud-based settings like Hugging Face Spaces or other platforms. By ensuring that the detection procedure is effective and easy to use, the overall implementation helps people and businesses stay safe from phishing scams.

V. RESULTS

Accuracy (87.5%):
Represents the overall correctness of the model. Out of all predictions, 87.5% were correct (phishing or legitimate).

Precision (85.7%):
Indicates how a large number of the websites flagged as phishing were actually phishing. High precision reduces false alarms.

Recall (90%):
Tells how many of the actual phishing sites were correctly identified. High recall means better phishing detection.



F1 Score (87.8%):
Precision and recall are balanced. When False negative and false positive results must be traded off, it is especially useful.

VI. FINAL RESULTS

This project effectively illustrates how to develop a system based on machine learning that

uses URL-level information to identify phishing websites. Utilizing the Random Forest classifier, the model demonstrated efficacy in detecting malicious URLs with a high accuracy of 87.5% a notably high recall of 90%.

In addition to demonstrating Machine learning's efficacy in cybersecurity, the project and incorporates this capability into an intuitive Flask web interface, enabling real-time phishing detection. Even with a small sample, the application of URL lexical features proved to be a simple and efficient method for spotting phishing attempts early on.

In order to help people and businesses stay safe online, this system serves as a scalable prototype that might be improved with more intricate features, bigger datasets, and real-time deployment on cloud platforms.

REFERENCE

[1] A. Jain and B. Gupta, "Phishing Detection: Analysis of Visual Similarity-Based Approaches," Information Security Journal & *Applications*, vol. 36, pp. 1–8, 2017.

[2] L. McCluskey, F. Thabtah, and M. Mohammad,

"Intelligent Rule-Based Phishing Websites Classification," *IET Information Security*, No. 3, vol. 8, pages.153–160, 2014.

[3] The International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 2012, pp. 517–522, "A Novel Method for Identifying Phishing Websites with Random Forest," by A. K. Jain and R. C. Joshi.

[4] R. Verma and K. Dyer, "On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers," in the Proceedings of the *ACM Computer Conference and Communications Security (CCS)*, 2015, pp. 1232–1243.

[5] F. Toolan and J. Carthy, "Feature Selection for Spam and Phishing Detection," in *eCrime Researchers Summit*, 2010, pp. 1–12.

[6] A. Almomani, B. A. B. Yassein, M. Al-Betar, and A. A. Awad, "An Enhanced Phishing Detection Model Using Random Forests," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 7, pp. 1440–1452, 2017.

[7] "UCI Machine Learning Repository – Phishing Websites Dataset", 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[8] T. Basnet, S. Sung, and M. Liu, "Rule-Based Phishing Attack Detection," in *Networking and Services (ICNS)*, 2012, pp. 533–538.

[9] Volume 10, Issue 8, pages 331–338 of The Journal of Advanced Applications of Computer Science, 2019. An empirical investigation of techniques for machine feature selection learning-based phishing website detection. A. F. M. Saiful Islam, F. Z. Rokhani, and M. A. Rahman.

[10] Phishing Rough Set Theory-Based Detection, Journal of International of Information and Computing Security, vol. 7, no. 3, pp. 270–284, 2015; H. S. Abdelhamid, A. Ayesh, and F. Thabtah.