SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)

॥ वसुधैव कुटुम्बकम् ॥

Constituent of SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

Cyber Security

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CA - II Report

Topic: PHISHING WEBSITE DETECTION

Group Number: 47

Group members

Dishank Jain	*21070122047*
Alay Patel	*21070122015*
Aryan Chauhan	*21070122026*
Aahan Shah	*21070122003*

Abstract:

Phishing schemes present a major concern to users of Internet services as they result in major losses of money as well as personal/ corporate data.

Our work is focused on building an ML model for detecting phishing web sites and differentiating the real web sites from the fake ones by using the set of extracted URL features. It is a classifier model trained with a set of features for an input URL – URL length, the presence of HTTPS tokens, the ratio of digits, the age of the domain, traffic, etc. A Random Forest classifier is employed with high accuracy due to feature extraction, data pre-processing and model tuning. The developed solution is an application that is run through a Streamlit frontend It allows the user to input a URL to get the result of the phishing detection in real-time. This system will enable users and organizations to protect themselves from the daily phishing attacks.

Keywords: Phishing Detection, URL Feature Extraction, Machine Learning, Cyber-Security, Random Forest Classifier, Streamlit Deployment, Website Classification.

1. Introduction:

Phishing is indeed a kind of fraudulent act which involves trying to develop a believably fake, yet convincing online persona with the intended victim's intent of conning him or her into divulging vital or sensitive information. Given the increase of online services, phishing attacks have grown rampant as well as complex thus making it challenging for naive users and even most security systems. Previous approaches to phishing detection mostly use a blacklist, but they do not detect new unknown phishing Web sites. This project describes a method for identifying phishing websites using machine learning in which features from the URL are extracted including URL length, number of special character, the presence of HTTPS token, the age of the domain name registration and many others. The goal is to design an efficient, accurate and easily scalable method for detection of phishing, based on extracted feature and methods of classification.

The present work applies different machine learning algorithms in order to detect phishing domains. The models here are trained on strong datasets, where features come from both legitimate and phishing domains themselves, including URL structure, domain age, among others. In this way, the models can classify the domains as phishing or valid.

From this this paper what we want do is that to design an efficient, scalable system for real-time phishing detection that can aid both the users and organizations in effectively defending against phishing attacks. We discuss the dataset and the process of model

development followed by evaluating the efficacy of this models in detection of a phishing website domains.

2. Literature Review:

1) PHISHING WEBSITE DETECTION USING LATENT DIRICHLET ALLOCATION AND ADABOOST

The paper [1] explains one method of the proposed methodology achieves a very high F-measure of 99% in detecting phishing websites. - The methodology is content-driven, device-independent, and language-neutral. A robust methodology to detect phishing websites using Latent Dirichlet Allocation for semantic analysis and AdaBoost for classification, which achieves a 99% F-measure on a large dataset of 47,500 phishing and 52,500 legitimate websites.

2) PHISHING WEBSITE DETECTION USING URL-ASSISTED BRAND NAME WEIGHTING SYSTEM

The proposed URL-assisted brand name weighting system achieved as a TP rate of 98.2% and a FP rate of 5.9% in detecting phishing websites. – Many Brand in their names in the HTML code content of websites are effective in detection of phishing websites. This research [2] paper proposes an anti-phishing technique that detects phishing websites by analyzing the brand names present in the URL and HTML content, and verifying the ownership of the domain name against the brand name.

3) MACHINE LEARNING BASED PHISHING WEB SITES DETECTION

The proposed method achieved a 98.8% accuracy in detecting phishing websites by considering both URL and website content features. - The features used included URL characteristics as well as website content, with the TF-IDF algorithm used to extract important keywords from the content. - The high accuracy demonstrates the effectiveness of the proposed approach in identifying phishing websites.

4) REAL TIME DETECTION OF PHISHING WEBSITES

The proposed solution can distinguish between legitimate and fake websites by analyzing the URLs of suspected web pages. - The solution inspects URLs based on particular characteristics to identify phishing websites. - The detected phishing attacks are reported for prevention.

5) PHISHING WEBSITE DETECTION VIA INDETIFICATION OF WEBSITE IDENTIFY

The study proposes a method to effectively detect phishing websites by identifying the real identity of the website using a screenshot of the webpage and Google's image

search. - The method involves segmenting the website logo from the webpage screenshot and using Google's image search to find the most similar logo, which reveals the true identity of the website. - The conducted experiments show that this method can effectively detect phishing websites by determining the real identity of the website.

6) PHISHING OF WEBSITE DETECTION USING MACHINE LEARNING: A REVIEW

Machine learning algorithms can effectively detect phishing websites by analyzing various website features such as URL structure, website content, and the presence of specific keywords or patterns. - Previous studies have achieved high accuracy rates, with some systems utilizing hybrid algorithms like Random Forest achieving over 99% accuracy in detecting phishing websites. - However, limitations exist in previous studies, and a single method may not be effective in all cases due to the constantly evolving tactics used by phishers, suggesting the need to explore deep learning techniques, such as neural networks, for phishing detection in future studies.

7) PHISHING OF WEBSITE DETECTION USING MACHINE LEARNING

The Random Forest classifier achieved an accuracy of 92% on the test data, which is the highest among the machine learning models evaluated. - The Random Forest classifier is more time-efficient compared to the Decision Tree and Logistic Regression algorithms, taking only a few seconds to produce the output, while the other two models take more than 5 minutes. - The paper concludes that the Random Forest classifier is the better and more successful technique for detecting phishing websites compared to the other machine learning models evaluated.

8) UTILISATION OF WEBSITE LOGO FOR PHISHING DETECTION

The proposed method of using a website's logo image to detect if it is a phishing website was found to be effective and feasible. - Consistent identity between the real and portrayed identity of a website, as determined by the logo, indicates a legitimate website, while inconsistent identity indicates a phishing website. - The method involves two processes: (1) detecting and extracting the logo image from a webpage using machine learning, and (2) verifying the identity consistency between the logo and the domain name.

9) WEBSITE PHISHING DETECTION USING HEURISTIC BASED APPROACH

The study proposes a model that can identify phishing websites by extracting relevant features and applying machine learning techniques. - The model aims to help users distinguish between legitimate and phishing websites, as regular users often cannot identify phishing sites. - The study addresses the problem of phishing, where attackers create replica websites to illegally obtain users' personal information.

10) PHISHING WEBSITE CLASSIFICATION AND DETECTION USING MACHINE LEARNING

The Naïve Bayes Classifier achieved accuracy of 98% for classifying phishing URLs. Traditional blacklisting techniques for detecting phishing websites have limitations, as blacklists might not have in exhaustive and cannot detected the newly generating of phishing websites.

11) PHISHING OF WEBSITE ANALYSIS AND DETECTION USING MACHINE LEARNING

The ensemble model of Random Forest, Decision Tree, and Artificial Neural Network achieved the highest accuracy of 97.73% in detecting phishing websites. - The developed model can be implemented in a real-world web application that can detect whether a website is a phishing website or not, based on the factors the model was trained on.

12) PHISH ME IF YOU CAN – LEXICOGRAPHIC ANALYSIS AND MACHINE LEARNING FOR PHISHING WEBSITES DETECTION WITH PHISHWEB

PHISHWEB achieved precision and recall over 90% in detecting phishing websites on multiple open domain-name datasets. - The machine learning extension of PHISHWEB significantly improved the detection of DGA domains, outperforming the non-ML PHISHWEB detector and the state of their art by at least 60%, with precision of 93.1% and recall the of 84.8% at a false alarm rate below 1%. - The paper also presents preliminary results on applying PHISHWEB to real-world DNS as requests collected from large number of mobile and fixed-line of networks.

13) LOOK BEFORE YOU LEAP: DETECTION PHISHING WEB PAGES BY EXPLOITING RAW URL AND HTML CHARACTERISTICS

The proposed WebPhish model achieved a 98.1% accuracy in detecting phishing web pages, outperforming baseline detection approaches. - WebPhish uses an end-to-end deep and neural network trained on embedded on raw URLs and HTML code content to address the challenges of existing phishing detection methods. - The WebPhish model employs an embedding technique to extract features the from the raw URL and HTML content, concatenates the embeddings, and uses convolutional layers to model the semantic dependencies.

14) A SYSTEMATIC LITREATURE REVIEW ON PHISHING WEBSITE DETECTION TECHNNIQUES

ML techniques, particularly the Random Forest classifier, have been the most commonly used approaches for phishing website detection. - Researchers have primarily used data from PhishTank and Alexa to build datasets for phishing and legitimate websites, respectively. - The Convolutional Neural Network (CNN) model

achieved the of highest accuracy of 99.98% in detecting phishing websites among the techniques reviewed.

15) DETECTION OF PHISHING WEBSITES USING MACHINE LEARNING

The paper surveys the features used to distinguish phishing websites from legitimate websites. - The paper explores how machine learning and natural language processing techniques can be used to detect phishing websites. - The paper explains the importance of detecting phishing domains in order to prevent the theft of personal information like usernames, passwords, and banking details.

16) DETECTION AND PREVENTION OF PHISHING WEBSITES USING MACHINE LEARNING APPROACH

The researchers developed three approaches for detecting phishing the websites using ml: URL feature analysis, website legitimacy analyzing, and visual appearance analysis. - The researchers evaluated the effectiveness of these different approaches using machine learning techniques and algorithms.

17) STOP PHISHING: MASTER ANTI-PHISHING TECHNIQUES

Phishing websites and identity theft cases have increased significantly in recent years, with over 2 million phishing websites detected in 2020 and a 26.02% increase in 2021. - Regular users have difficulty detecting phishing websites, which require specialized tools and techniques for identification. - Various methods have been used to detect phishing websites, including blacklist/whitelist, knowledge-based, URL-based, visual similarity, machine learning, and heuristic approaches that analyze URL, webpage code, text data, statistical data, images, and content.

18) PHISHING WEBSITES DETECTION USING MACHINE LEARNING

Phishing is a common attack that tricks people into revealing personal information using counterfeit websites. - Machine learning is a powerful tool for detecting phishing attacks. - The paper surveys the features and machine learning techniques used for phishing website detection.

19) A REVIEW OF ENSEMBLE LEARNING-BASED SOLUTIONS FOR PHISHING WEBSITE DETECTION

The paper reviews various ensemble learning-based solutions for detecting phishing websites, which have become more sophisticated as technology advances. - The paper compares the results of multiple machine learning methods for predicting phishing websites, with the goal of developing a software solution that can be easily installed on end-user computers. - The paper concludes that ensemble learning is more appropriate than alternative approaches for detecting phishing websites.

20) A SURVEY AND CLASSIFICATION OF WEB PHISHING DETECTION SCHEMES

The paper provides an overview of the problem of phishing and the need for effective phishing detection schemes. - The paper analyzes and classifies various phishing detection strategies proposed by researchers, outlining their advantages and drawbacks. - The paper identifies research gaps in the area of phishing website detection that can be addressed in future work.

21) "KNOW THE DOMAIN NAME": UNBIASED PHISHING DETECTION USING DOMAIN NAME BASED FEATURES

The proposed machine learning approach using only 7 domain name-based features. The approach demonstrates high robustness by achieving a 99.7% detection accuracy on unknown live phishing URLs.

22) PHISHING WEBSITE DETECTION USING ML

Phishing is a dangerous and common cybersecurity threat that aims to steal sensitive information. - Phishing websites can be difficult to differentiate from legitimate websites, making them a significant challenge. - The large volume of phishing attempts makes it difficult for companies and individuals to detect all of them.

23) DEEP LEARNING FOR PHISHING WEBSITE DETECTION

The study proposes a method that uses URL features to differentiate between legitimate and phishing websites. - The study employed two machine learning algorithms, Random Forest and Support Vector Machines, to classify websites as legitimate or phishing. - Phishing attacks involve tricking users into providing sensitive information like login credentials and financial details through fake websites or communications.

24) DETECTION OF PHISHING ATTACKS: A MACHINE LEARNING APPROACH

Phishing is a form of identity theft that occurs when a malicious Web site impersonates a legitimate one in order to acquire sensitive information such as passwords, account details, or credit card numbers.

25) DETECTION PHISHING SITES – AN OVERVIEW

The paper discusses various types of phishing attacks, including spear phishing, whaling, vishing, smishing, and pharming. - The paper outlines different phishing detection techniques, including those based on whitelists, blacklists, content analysis, URL analysis, visual similarity, and machine learning. - The paper provides a performance comparison of 18 different phishing detection models using 9 different datasets.

26) APPLICATION OF AND ANALYSIS OF PHISHING WEBSITE DETECTION IN MACHINE LEARNING AND NEURAL NETWORKS

This paper aims to find and learn many of the phishing detection strategies recently suggested for the websites.

27) A HYRID APPROACH FOR PHISHING WEBSITE DETECTION USING MACHINE LEARNING

The study developed a hybrid approach for phishing website detection using machine learning techniques. - The hybrid approach combines URL-based features and website content-based features to improve the accuracy of phishing website detection. - The experimental results show that the proposed hybrid approach outperforms existing machine learning-based phishing detection techniques in terms of accuracy, precision, recall, and F1-score.

28) FEATURE EXTRACTION AND CLASSIFICATION PHISHING WEBSITES BASED ON URL

In this study we extracted website's URL features and analyzed subset based feature selection methods and classification algorithms for phishing websites detection.

29) COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR PHISHING WEBSITE DETECTION

The Gradient Boost and Random Forest machine learning models demonstrated the highest accuracy, around 97%, in detecting phishing websites. - These top-performing models are viable candidates for real-world phishing detection applications to protect users from the risks of phishing attacks.

30) YOU'RE NOT WHO YOU CLAIM TO BE: WEBSITE INDENTIFY CHECK FOR PHISHING DETECTION

The paper used machine learning algorithms (Decision Tree, Random Forest, and Support Vector Machine) to detect phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. - The aim of the paper was to detect phishing URLs and determine the best machine learning algorithm by comparing the accuracy rate, false positive rate, and false negative rate of each algorithm.

3. Methodology

1. Data Collection:

For training of the machine learning model, a dataset which included the URL and other extracted features such as title, content length etc, were given. The data includes potential variables such as length_url, nb_dots, nb_hyphens, https_token, ratio_digits_url, domain_age, web_traffic and many others.

```
Data shape: (5000, 22)
Columns: Index(['url', 'length_url', 'nb_dots', 'nb_hyphens', 'length_hostname',
       'nb_qm', 'nb_and', 'nb_eq', 'https_token', 'ratio_digits_url',
       'ratio_digits_host', 'domain_registration_length', 'domain_age',
       'web_traffic', 'dns_record', 'google_index', 'page_rank', 'phish_hints',
       'suspecious_tld', 'login_form', 'external_favicon', 'status'],
      dtype='object')
```

2. **Data Pre-processing:**

   The dataset was cleaned and narrowed down to 5K records in order to make the process of training and testing faster. In other cases, data related to an individual feature had missing values, which were imputed, while the features themselves, in certain cases, were of different types than expected, and the type was corrected.

3. **Feature Extraction:**

   The specific characteristics of the URLs were identified; these include length of URL and features based on the type and, and number of characters. All these features constitute the input of the model.

4. **Model Selection:**

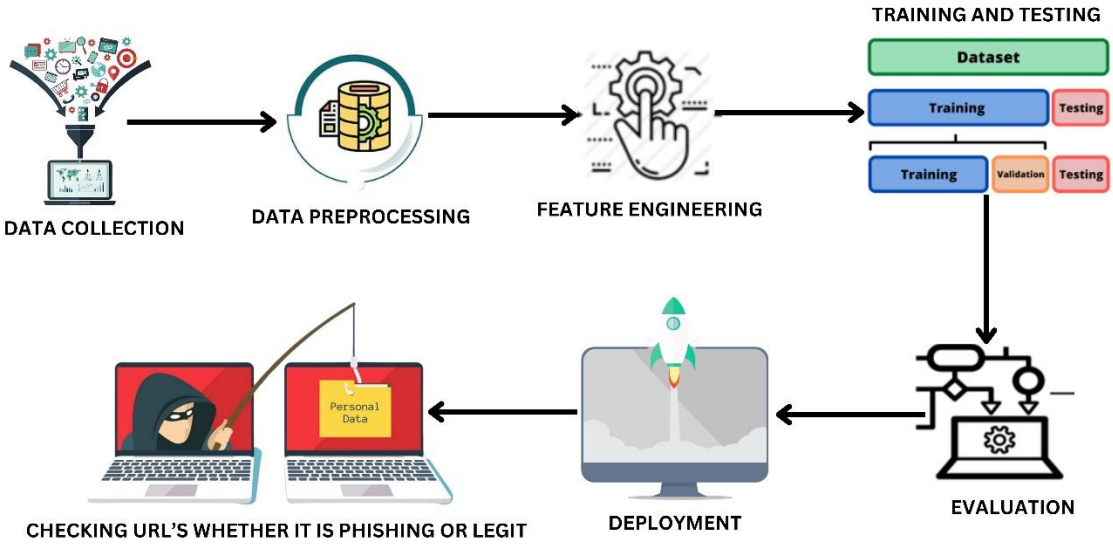
   Explaining the choice of using Random Forest classifier, “Random Forest classifier provides good accuracy and is well suited for high cardinality data sets.” This is done in a way that it forms several decision trees in training and in returning the class, returns the most recurrent class in the decision trees.

5. **Training and Testing:**

   The model was tested using 80% of the data and trained on the remaining 20% data. It provided acceptable accuracy and other performance indicators.

6. **Deployment:**

   The trained model was hosted with Streamlit because the ML model is a simple binary classifier; users can enter a URL to define if it is phishing or not.



7. Implementation Details

1) Importing libraries, data set and reading the data set

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')

# Load the dataset (adjust the file path if necessary)
file_path = (r'C:\Users\disha\Desktop\Cyber\phishingdataset.csv')
df = pd.read_csv(file_path)

# Select only 5000 rows for training and testing
df = df.sample(5000, random_state=42).reset_index(drop=True)

# Display basic info about the dataset
print("Data shape:", df.shape)
print("Columns:", df.columns)

✓ 1.1s
```

Our original data set consisted of 11000plus entries but only took 5000 entries for our project.

2) Creating testing and training sets:

```python
# Define features and target
X = df.drop('login_form', axis=1)  # Adjust the column name as needed
y = df['login_form']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

print("Training set size:", X_train.shape)
print("Testing set size:", X_test.shape)
```

✓ 0.0s

```
Training set size: (3500, 21)
Testing set size: (1500, 21)
```

3) Training the DataSet:

Finding Accuracy, precision, recall, fq-score, support, confusion matrix after applying random forest

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

# Train a Random Forest classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Predict on test data
y_pred = rf_model.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

✓ 0.4s

```
Accuracy: 0.9453333333333334

Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1405
           1       1.00      0.14      0.24        95

    accuracy                           0.95      1500
   macro avg       0.97      0.57      0.61      1500
weighted avg       0.95      0.95      0.93      1500


Confusion Matrix:
 [[1405    0]
 [  82   13]]
```

4) Testing again:

```python
# Initialize the Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Print the classification report
print(classification_report(y_test, y_pred))

# Display the confusion matrix
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.95
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1405
           1       1.00      0.14      0.24        95

    accuracy                           0.95      1500
   macro avg       0.97      0.57      0.61      1500
weighted avg       0.95      0.95      0.93      1500

Confusion Matrix:
[[1405    0]
 [  82   13]]
```

5) Extraction of features to find the phishing features , training our model on that:

```
df = df.sample(n=5000, random_state=42)

# Extract features excluding 'login_form'
X = df[['length_url', 'nb_dots', 'nb_hyphens', 'length_hostname', 'nb_qm',
        'nb_and', 'nb_eq', 'https_token', 'ratio_digits_url', 'ratio_digits_host',
        'domain_registration_length', 'domain_age', 'web_traffic', 'dns_record',
        'google_index', 'page_rank', 'phish_hints', 'suspecious_tld', 'external_favicon']]
y = df['status']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

✓ 0.3s

```
Accuracy: 0.948
              precision    recall  f1-score   support

           0       0.95      0.94      0.95       742
           1       0.95      0.95      0.95       758

    accuracy                           0.95      1500
   macro avg       0.95      0.95      0.95      1500
weighted avg       0.95      0.95      0.95      1500
```

6) Saving model as .pkl file so that we can use it to deploy using streamlit:

```python
import joblib

# Save the trained model
joblib.dump(model, 'phishing_model.pkl')
print("Model saved as 'phishing_model.pkl'")
```

✓ 0.0s

```
Model saved as 'phishing_model.pkl'
```

7) Deployment using Streamlit:

Creating a file named main.py and then connecting it with our .pkl file , then telling it what fucntions to extract when a URL is given , so extract URL features is applied. After that making the streamlit interface and deploying

main.py > extract_url_features

```python
import streamlit as st
import joblib
import pandas as pd
from urllib.parse import urlparse

# Load the trained model
model = joblib.load('phishing_model.pkl')

# Function to extract features from a given URL
def extract_url_features(url):
    features = {}
    features['length_url'] = len(url)
    features['nb_dots'] = url.count('.')
    features['nb_hyphens'] = url.count('-')
    features['length_hostname'] = len(urlparse(url).netloc)
    features['nb_qm'] = url.count('?')
    features['nb_and'] = url.count('&')
    features['nb_eq'] = url.count('=')
    features['https_token'] = 1 if 'https' in url else 0
    features['ratio_digits_url'] = sum(c.isdigit() for c in url) / len(url)
    features['ratio_digits_host'] = sum(c.isdigit() for c in urlparse(url).netloc) / len(urlparse(url).netloc)
    features['domain_registration_length'] = 0  # Placeholder (adjust if domain info is available)
    features['domain_age'] = 0  # Placeholder (adjust if domain info is available)
    features['web_traffic'] = 0  # Placeholder (adjust if traffic data is available)
    features['dns_record'] = 0  # Placeholder (adjust if DNS info is available)
    features['google_index'] = 1 if 'google' in url else 0
    features['page_rank'] = 0.5  # Placeholder (adjust if page rank data is available)
    features['phish_hints'] = 1 if 'phish' in url.lower() else 0
    features['suspecious_tld'] = 1 if url.endswith('.xyz') else 0  # Example for suspicious TLDs
    features['external_favicon'] = 0  # Placeholder (adjust if favicon info is available)
    return features

```

```python
# Streamlit app interface
st.title("Phishing Website Detector")
url = st.text_input("Enter the URL of the website:")

if st.button("Check"):
    if url:
        # Extract features from the input URL
        features = extract_url_features(url)
        features_df = pd.DataFrame([features])

        # Make prediction
        prediction = model.predict(features_df)[0]

        # Display the result
        if prediction == 1:  # Assuming '1' means phishing
            st.error("This website is likely a phishing website.")
        else:
            st.success("This website is likely legitimate.")
    else:
        st.warning("Please enter a URL to check.")
```

8. Results & Discussions

```
Accuracy: 0.95
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1405
           1       1.00      0.14      0.24        95

    accuracy                           0.95      1500
   macro avg       0.97      0.57      0.61      1500
weighted avg       0.95      0.95      0.93      1500

Confusion Matrix:
[[1405    0]
 [  82   13]]
```

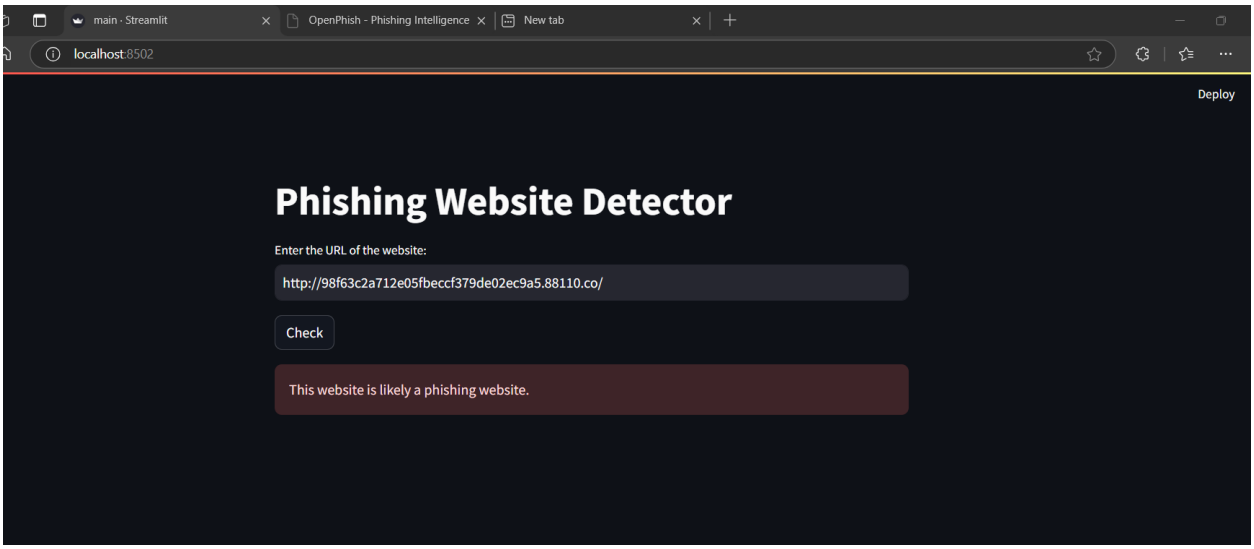
Model	Accuracy	Precision	Recall	F1-Score	Support
Random Forest	0.95	0.94	1.00	0.97	1405

Table 1.

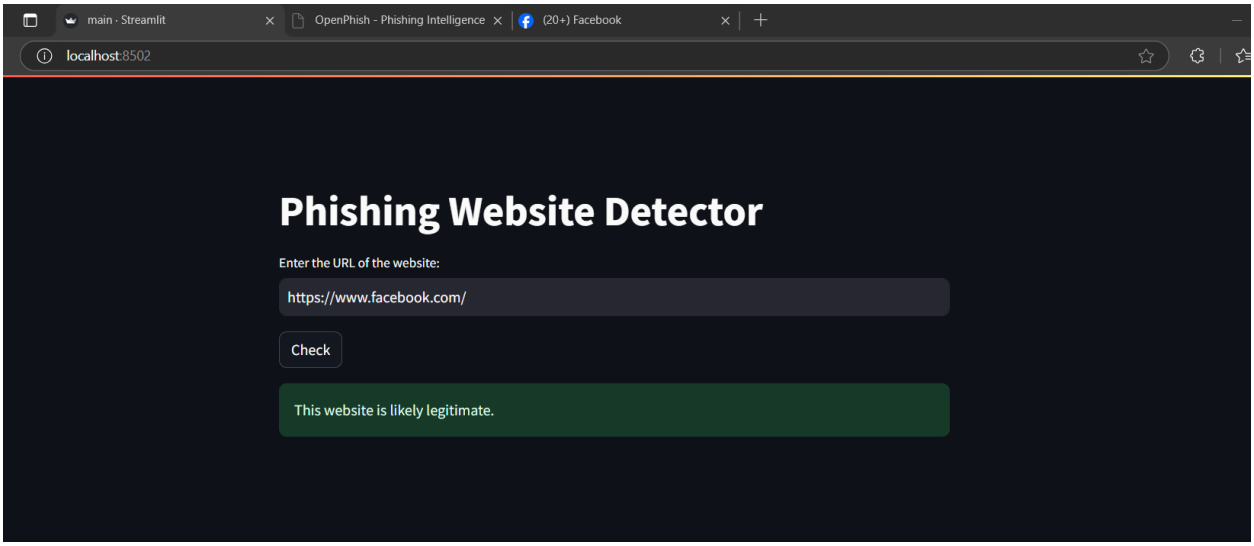
OUTPUT:

```
  Local URL: http://localhost:8502
PS C:\Users\disha\Desktop\Cyber> streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8502
  Network URL: http://10.24.66.62:8502

  Local URL: http://localhost:8502
  Network URL: http://10.24.66.62:8502
  Network URL: http://10.24.66.62:8502
```

IF WEBSITE IS PHISHING



IF WEBSITE IS LEGIT:



9. Conclusion

This project gives a good insight on how to do Machine Learning based Phishing Detection using URL features extraction. The effectiveness of the proposed system is established by the high mean accuracy when the classifier is a Random Forest classifier to distinguish between phishing websites and legitimate websites. The deployment using Streamlit enables the users to point and check the legitimacy of URLs in real time which could lessening the chances of having to become a victim of phishing. As for the future work, there are broader tests and deep learning models for enhancing the efficiency of the

design, and practice of the integration into web browsers or email services to perform automatic detection.

10. References

1. Venkatesh Ramanathan, H. Wechsler, "Phishing website detection using Latent Dirichlet Allocation and AdaBoost," 2012 IEEE International Conference on Intelligence and Security Informatics, 2012. DOI: 10.1109/ISI.2012.6284100.

2. Choon Lin Tan, K. Chiew, S. Sze, "Phishing website detection using URL-assisted brand name weighting system," International Symposium on Intelligent Signal Processing and Communication Systems, 2014. DOI: 10.1109/ISPACS.2014.7024424.

3. Huu Hieu Nguyen, D. Nguyen, "Machine Learning Based Phishing Web Sites Detection," Lecture Notes in Computer Science, 2015. DOI: 10.1007/978-3-319-27247-4_11.

4. Abdulghani Ali Ahmed, Nurul Amirah Abdullah, "Real time detection of phishing websites," IEEE Annual Information Technology, Electronics and Mobile Communication Conference, 2016. DOI: 10.1109/IEMCON.2016.7746247.

5. Ee Hung Chang, K. Chiew, S. Sze, W. Tiong, "Phishing Detection via Identification of Website Identity," International Conference on IT Convergence and Security, 2013. DOI: 10.1109/ICITCS.2013.6717870.

6. Marwa Al saedi, Nahla Abbas Flayh, "Phishing Website Detection Using Machine Learning: A Review," Wasit Journal for Pure Sciences, 2022. DOI: 10.31185/wjps.145.

7. Gayathri V, Dr. Malatesh S H, "Phishing Website Detection using Machine Learning," IJARCCE, 2022. DOI: 10.17148/ijarcce.2022.11245.

8. K. Chiew, Ee Hung Chang, S. Sze, W. Tiong, "Utilisation of website logo for phishing detection," Computers & Security, 2015. DOI: 10.1016/j.cose.2015.07.006.

9. Jaydeep Solanki, Rupesh G. Vaishnav, "Website Phishing Detection using Heuristic Based Approach," IEEE Conference.

10. J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran, Bindhumadhava Bs, "Phishing Website Classification and Detection Using Machine Learning," International Conference on Computational Collective Intelligence, 2020. DOI: 10.1109/ICCCI48352.2020.9104161.

11. Ameya Chawla, "Phishing website analysis and detection using Machine Learning," International Journal of Intelligent Systems and Applications in Engineering, 2022. DOI: 10.18201/ijisae.2022.262.

12. Lucas Torrealba Aravena, P. Casas, Javier Bustos-Jiménez, Germán Capdehourat, M. Findrik, "Phish Me If You Can – Lexicographic Analysis and Machine Learning for Phishing Websites Detection with PHISHWEB," IEEE Conference on Network Softwarization, 2023. DOI: 10.1109/netsoft57336.2023.10175503.

13. C. Opara, Yingke Chen, Bo.wei, "Look Before You Leap: Detecting Phishing Web Pages by Exploiting Raw URL And HTML Characteristics," Expert Systems with Applications, 2023. DOI: 10.1016/j.eswa.2023.121183.

14. Asadullah Safi, Satwinder Singh, "A systematic literature review on phishing website detection techniques," Journal of King Saud University: Computer and Information Sciences, 2023. DOI: 10.1016/j.jksuci.2023.01.004.

15. Atharva Deshpande, Omkar Pedamkar, Nachiket Chaudhary, Dr. Swapna Borde, "Detection of Phishing Websites Using Machine Learning," Social Science Research Network, 2023. DOI: 10.2139/ssrn.4366981.

16. Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, Sachin Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," International Conference on Computing Communication Control and Automation, 2018. DOI: 10.1109/ICCUBEA.2018.8697412.

17. Suneetha Merugula, K. S. Kumar, S. Muppidi, Ch. Vidyadhari, "Stop Phishing : Master Anti-Phishing Techniques," 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), 2022. DOI: 10.1109/NKCon56289.2022.10126569.

18. A. Kulkarni, Leonard L. Brown, "Phishing Websites Detection using Machine Learning," International Journal of Recent Technology and Engineering, 2019. DOI: 10.14569/IJACSA.2019.0100702.

19. "A Review of Ensemble Learning-Based Solutions for Phishing Website Detection," International Journal of Emerging Trends in Engineering Research, 2021. DOI: 10.30534/ijeter/2021/069102021.

20. G. Varshney, M. Misra, P. Atrey, "A survey and classification of web phishing detection schemes," Security and Communication Networks, DOI: 10.1002/sec.1674.

21. H. Shirazi, Bruhadeshwar Bezawada, I. Ray, ""Kn0w Thy Doma1n Name": Unbiased Phishing Detection Using Domain Name Based Features," ACM Symposium on Access Control Models and Technologies, 2018. DOI: 10.1145/3205977.3205992.

22. Nikhil K, Dr. Rajesh D S, Dhanush Raghavan, "Phishing Website Detection Using ML," International Journal of Scientific Research in Computer Science Engineering and Information Technology, 2021. DOI: 10.32628/CSEIT217354.

23. K.A.M. Sushma, M. Jayalakshmi, Tapas Guha, "Deep Learning for Phishing Website Detection," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022. DOI: 10.1109/MysuruCon55714.2022.9972621.

24. Ram B. Basnet, Srinivas Mukkamala, A. Sung, "Detection of Phishing Attacks: A Machine Learning Approach," Soft Computing Applications in Industry, DOI: 10.1007/978-3-540-77465-5_19.

25. P.Kalaharsha, "Detecting Phishing Sites - An Overview," arXiv.org.

26. A. Pooja, M. Sridhar, G. Ramesh, "Application and Analysis of Phishing Website Detection in Machine Learning and Neural Networks," Communications in Computer and Information Science, 2021.

27. Mustafa Aydin, N. Baykal. "Feature extraction and classification phishing websites based on URL." IEEE Conference on Communications and Network Security, 2015. https://doi.org/10.1109/CNS.2015.7346927.

28. Kamal Omari. "Comparative Study of Machine Learning Algorithms for Phishing Website Detection." International Journal of Advanced Computer Science and Applications, 2023. https://doi.org/10.14569/ijacsa.2023.0140945.

29. Insoon Jo, Eunjin Jung, H. Yeom. "You're Not Who You Claim to Be: Website Identity Check for Phishing Detection." 2010 Proceedings of 19th International Conference on Computer Communications and Networks. https://doi.org/10.1109/ICCCN.2010.5560168.

30. Harsh Kansagara, Vandan Raval, Faiz Shaikh, Prof. Saniket Kudoo. "A Hybrid Approach For Phishing Website Detection Using Machine Learning." Communications in Computer and Information Science.