PROFESSIONAL TRAINING REPORT

Detection Of Phishing Website With URLS

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning

by

Mengarthi Abhinav
41731073

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A++" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

OCTOBER 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with A++ Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training is the bonafide work of **Mengarthi Abhinav(41731073)**who carried out the project entitled **“Detection of phishing Website With URLS”** under my supervision from June 2023 to October 2023.

Internal Guide
Mrs.Yugha.R, Assistant professor, CSE

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on : ________________

Internal Examiner **External Examiner**

DECLARATION

I, **Mengarthi Abhinav(41731073)** hereby declare that the Professional Training Report-I entitled **“Detection of Phishing Websites With URLS”** done by me under the guidance of **Mrs.Yugha.R, Assistant professor, CSE** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence.

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

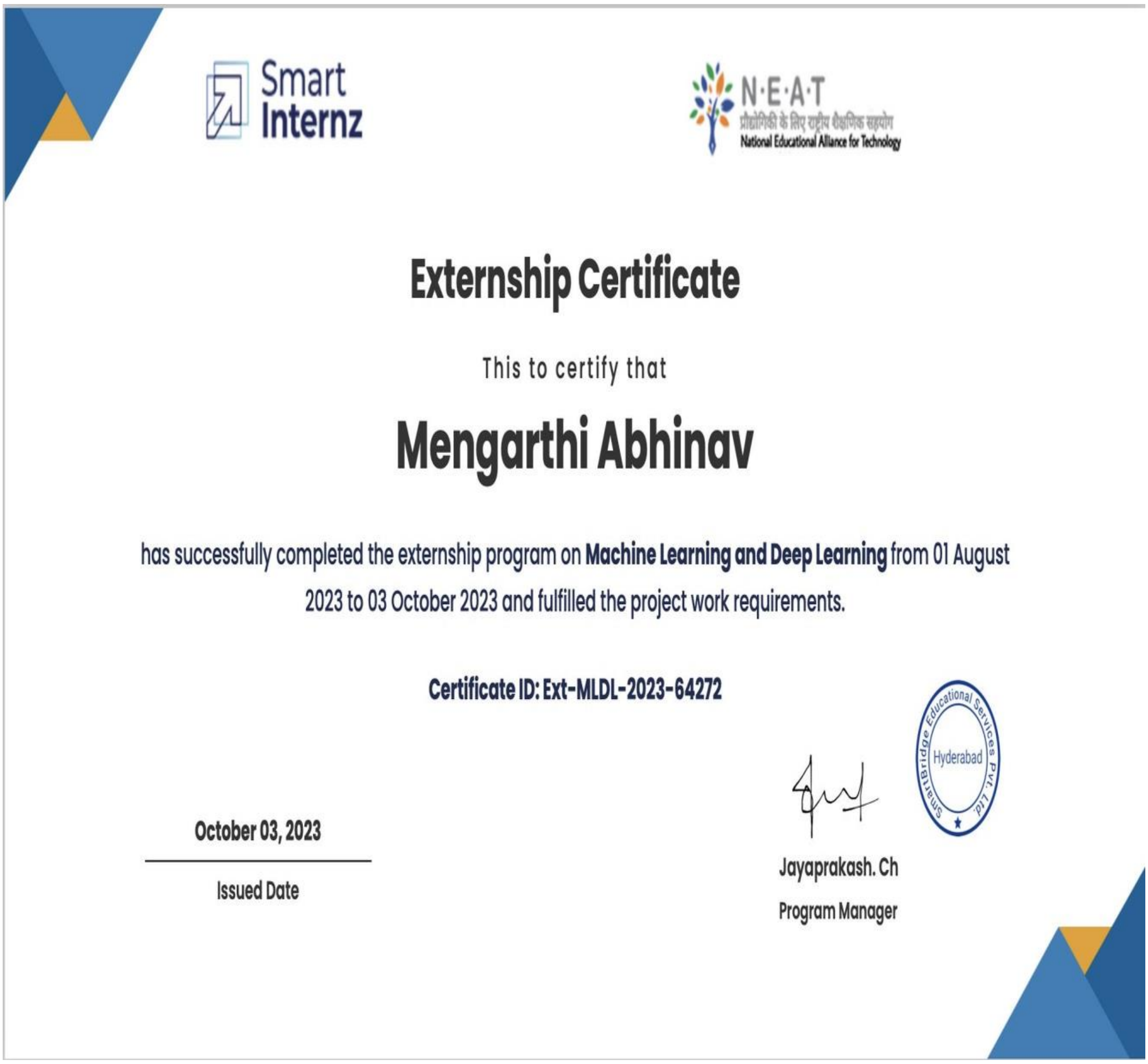
ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Mrs.Yugha.R, Assistant professor, CSE** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.



Smart Internz

N·E·A·T
प्रौद्योगिकी के लिए राष्ट्रीय शैक्षिक सहयोग
National Educational Alliance for Technology

Externship Certificate

This to certify that

Mengarthi Abhinav

has successfully completed the externship program on **Machine Learning and Deep Learning** from 01 August 2023 to 03 October 2023 and fulfilled the project work requirements.

Certificate ID: Ext-MLDL-2023-64272

SmartBridge Educational Services Pvt. Ltd
Hyderabad

October 03, 2023
Issued Date

Jayaprakash. Ch
Program Manager

ABSTRACT

Phishing attacks pose a significant threat to online security, targeting individuals and organizations by tricking them into revealing sensitive information through fraudulent websites. This project aims to develop a robust phishing website detection system using Python. The primary objective is to create a machine learning model that can differentiate between legitimate and phishing websites based on features extracted from URLs.

The project follows a structured approach, including data collection, preprocessing, feature extraction, and model building. A dataset comprising both phishing and legitimate URLs is utilized for training and evaluation. Various features such as URL length, domain information, special characters, and keywords are extracted to represent each URL. Machine learning models, including logistic regression, random forests, support vector machines, and neural networks, are explored for classification.The performance of the models is evaluated using metrics like accuracy, precision, recall, F1-score, and ROC-AUC to ensure their effectiveness in identifying phishing websites. Hyperparameter tuning is employed to optimize model performance.

The final model can be integrated into web services or applications, providing real-time phishing detection to enhance online security. Continuous monitoring and updates are essential to adapt to evolving phishing tactics.This project contributes to the ongoing efforts to combat phishing attacks, ultimately reducing the risks associated with online fraud and protecting individuals and organizations from cyber threats.In conclusion, this project contributes to the ongoing efforts to combat phishing attacks by providing a practical and automated solution for phishing website detection. The Python-based implementation and machine learning approach make it accessible for researchers and cybersecurity professionals seeking to enhance online security and protect users from falling victim to phishing scams.

TABLE OF CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

This project focuses on creating a cutting-edge phishing website detection system designed to safeguard users from online threats by accurately distinguishing between legitimate and phishing websites. By harnessing advanced machine learning algorithms, we systematically classify websites based on carefully extracted URL features.

The methodology adopted is both structured and thorough. Initially, we sourced a rich dataset from Kaggle, which contains labeled URLs representing both phishing and legitimate sites. After data collection, we meticulously cleaned and organized the dataset, ensuring that all features were primed for model training. Essential characteristics—such as URL length, domain details, special characters, and the presence of critical keywords—were expertly extracted to create a robust representation of each URL.

To evaluate the efficiency of our phishing detection system, we implemented several powerful machine learning models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Gradient Boosting Classifier. Our performance evaluation focused on accuracy metrics derived from both training and test datasets. The standout performer was the Gradient Boosting Classifier, which achieved an impressive accuracy of 97.4% on the test data, highlighting its superior ability to identify phishing sites. In comparison, the K-Nearest Neighbors model, set with k=7, reached a commendable 93.6% accuracy.

This project significantly enhanced my technical skills and expertise in various areas, including programming in Python, leveraging data analysis libraries like Pandas and NumPy, and utilizing machine learning frameworks such as Scikit-learn. I also gained proficiency in data visualization through Matplotlib and Seaborn, and explored key model evaluation metrics such as accuracy, precision, recall, and F1-score. The compelling visualizations generated using Seaborn and Matplotlib not only showcased model performance but also provided insightful interpretations of data distributions, making the results both understandable and impactful.

CHAPTER 2

REQUIREMENTS ANALYSIS

2.1 HARDWARE REQUIREMENTS

1. Computer or Server
2. Storage
3. Network Connectivity
4. GPU
5. Cloud Computing
6. Data Backup and Redundancy
7. Security Measures
8. Cooling and Ventilation

2.2 SOFTWARE REQUIREMENTS

1.Python

2.Integrated Development Environment (IDE)

3.Version Control

4.Data Analysis and Manipulation

5.Machine Learning Libraries

6.Web Scraping (if collecting data from websites)

7.Feature Engineering and Text Processing

8.Model Evaluation and Metrics

9.Visualization

10.Database

11.Notebook and Documentation

CHAPTER 3

DESIGN DESCRIPTION OF PROPOSED PROJECT

3.1 PROPOSED METHODOLOGY

The primary goal of this project is to develop a machine learning-based system capable of identifying phishing websites with high accuracy. This methodology outlines the systematic steps taken to achieve this goal, from data acquisition to model evaluation.

1. Data Collection

- **Dataset Source**: The data was sourced from Kaggle, containing labeled URLs for both phishing and legitimate websites.
- **Data Description**: Each data entry includes information such as the URL structure, domain information, and indicators commonly associated with phishing attacks (e.g., URL length, presence of special characters).

2. Data Preprocessing

- **Data Cleaning**: Removing unnecessary characters or symbols that could lead to noise during model training.
- **Feature Extraction**: Key features were selected and engineered to enhance the model's ability to classify URLs correctly. Some significant features included:
    - **URL Length**: Phishing websites often have longer URLs.
    - **Special Characters**: Presence of symbols like '@', '-', and '?' within URLs, which are common in phishing URLs.
    - **Domain and Path Analysis**: Analyzing parts of the URL for patterns that may signify phishing attempts (e.g., misspellings, unusual domain names).
- **Encoding Categorical Data**: Converting categorical data into numerical representations suitable for machine learning algorithms.

3. Feature Engineering

- **Objective**: Enhance the dataset with meaningful attributes that contribute to the model's ability to differentiate between phishing and legitimate websites.
- **Feature Examples**:
    - **Keyword Presence**: Checking for words like "secure" or "login" that may be commonly used in phishing URLs to mislead users.
    - **HTTPS Protocol**: Whether or not the URL uses HTTPS, which is often an indicator of security but not always reliable.
    - **Subdomain Count**: A higher number of subdomains might indicate phishing intentions.

4. Model Selection

Three machine learning algorithms were selected to evaluate the system's performance, each providing unique strengths for the classification task:

- **K-Nearest Neighbors (KNN)**: Chosen for its simplicity in classification tasks and its effectiveness in detecting similarities in URL structures.
- **Support Vector Machine (SVM)**: Selected for its ability to create clear decision boundaries, which is beneficial for binary classification problems.
- **Gradient Boosting Classifier**: Employed to utilize boosting techniques that combine weak learners for higher accuracy and better handling of complex patterns.

5. Model Training

- **Splitting Data**: The dataset was divided into training and test sets (e.g., 80% training, 20% testing) to evaluate model performance objectively.
- **Hyperparameter Tuning**:
  - **KNN**: The number of neighbors (k) was optimized for the best balance between training and testing accuracy.
  - **SVM**: The regularization parameter (C) was adjusted to find the most effective margin.
  - **Gradient Boosting**: The number of estimators and learning rate were optimized to prevent overfitting while maintaining high accuracy.
- **Training Process**: Each model was trained with the training dataset, using a cross-validation approach to ensure generalizability.

6. Model Evaluation

- **Accuracy Metrics**: Accuracy scores were used to compare model effectiveness on both training and test datasets.
- **Evaluation Metrics**:
  - **KNN**: Provided an accuracy of 93.8% at the best k-value on test data.
  - **SVM**: Achieved a test accuracy of 93.5%, consistently performing well across various C values.
  - **Gradient Boosting Classifier**: Outperformed other models with a test accuracy of 97.4%.
- **Confusion Matrix**: Generated for each model to visualize true positives, true negatives, false positives, and false negatives, helping identify potential misclassifications.
- **Precision, Recall, and F1-Score**: Calculated for a deeper understanding of model reliability and to balance between false positives and false negatives.

7. Data Visualization

- Visualizations were created using Matplotlib and Seaborn to better understand data distributions and model performance:
  - **Feature Distributions**: Graphs to highlight the distribution of key features across phishing and legitimate URLs.
  - **Confusion Matrix Heatmaps**: Displayed using Seaborn to assess each model’s classification accuracy.

  - **Model Comparison**: Plots comparing the accuracy of all three models, which illustrated the Gradient Boosting Classifier's effectiveness.

8. Results and Analysis

- **Best Model**: The Gradient Boosting Classifier demonstrated the highest test accuracy (97.4%) and was chosen as the most suitable model for this detection task.
- **Key Takeaways**: The project confirmed that ensemble learning methods like Gradient Boosting are particularly effective for phishing website detection due to their ability to capture complex patterns in data.
- **Model Insights**: SVM and KNN also performed competitively but were slightly less effective in handling diverse phishing patterns compared to Gradient Boosting.

9. Conclusion and Future Work

- **Project Conclusion**: This methodology successfully developed a phishing detection system with a high degree of accuracy. The Gradient Boosting Classifier was identified as the most effective model for future deployment.
- **Future Improvements**:
  - **Additional Features**: Incorporate other advanced features such as lexical analysis or DNS-related attributes.
  - **Model Optimization**: Experiment with more hyperparameter tuning and advanced ensembling techniques to further increase accuracy.
  - **Real-Time Deployment**: Explore methods to implement this system in real-time applications, allowing for instant detection of phishing websites.

3.2 Working Principles

The phishing website detection system employs a machine learning-based approach to classify URLs as either legitimate or phishing. The methodology is structured around the systematic use of data preprocessing, feature engineering, and classification algorithms. This section details the system's working principle, emphasizing each process step, from data intake to model evaluation and predictions.

1. Data Collection and Input

- **Dataset Source**: The dataset, acquired from Kaggle, contains labeled entries for phishing and legitimate websites. Each record in the dataset represents a URL, with labels indicating its legitimacy.
- **Purpose**: The system is designed to analyze the characteristics of each URL entry to identify patterns that are typically associated with phishing.

2. Data Preprocessing

- **Objective**: Convert raw data into a structured form suitable for machine learning models. This step is crucial to eliminate noise and ensure consistent input for the models.
- **Data Cleaning**: Removal of extraneous symbols or HTML elements that may interfere with the analysis.
- **Categorical Encoding**: Converts text-based features into numerical formats that machine learning algorithms can process.
- **Feature Scaling**: Normalization of numerical features to ensure all inputs are on a similar scale, reducing model bias.

3. Feature Extraction

- **Purpose**: Extract relevant features that increase the system's ability to differentiate between legitimate and phishing websites.
- **Features Engineered**:
  - **URL Length**: Phishing URLs are often longer, attempting to mimic legitimate sites with subtle differences.
  - **Special Character Presence**: Inclusion of symbols like '@', '?', and '-' can indicate suspicious activity, as phishing URLs often include these characters.
  - **Keyword Presence**: Common keywords like "secure" or "login" are sometimes misused in phishing URLs to gain user trust.
  - **HTTPS Protocol Check**: Phishing websites may or may not use HTTPS, but legitimate websites usually do. This check helps refine predictions.
  - **Subdomain Count**: Phishing URLs tend to include multiple subdomains to appear legitimate. A higher count of subdomains often points to phishing.

4. Model Selection and Implementation

- The system uses three different machine learning algorithms to classify URLs. Each algorithm analyzes the data patterns differently, allowing for model comparisons and performance evaluation.
- **Algorithms Employed**:
  - **K-Nearest Neighbors (KNN)**: This algorithm classifies URLs by comparing them to the closest labeled examples in the training set. It uses a distance-based metric to classify a new URL based on its proximity to known phishing or legitimate URLs.
  - **Support Vector Machine (SVM)**: SVM attempts to find the hyperplane that best separates phishing and legitimate data points. This method is effective in handling non-linear patterns, making it a robust choice for binary classification.
  - **Gradient Boosting Classifier**: A sophisticated ensemble technique that sequentially applies weak learners to improve accuracy. By emphasizing misclassified examples, this classifier enhances overall predictive performance.

5. Model Training and Hyperparameter Tuning

- **Training Process**: The data is split into training and testing subsets, where 80% of the data is used to train the model, and 20% is used to evaluate its performance.
- **Hyperparameter Optimization**:
  - For **KNN**, the number of neighbors (k) is optimized to balance model accuracy.
  - For **SVM**, the regularization parameter (C) is tuned to ensure an optimal decision boundary.
  - In **Gradient Boosting**, the learning rate and the number of estimators are adjusted to enhance model performance while preventing overfitting.

6. Model Evaluation and Selection

- **Accuracy Metrics**: Each model's accuracy is assessed using training and test data to evaluate how well it generalizes to unseen URLs.
- **Performance Metrics**:

- **K-Nearest Neighbors**: Best accuracy on test data achieved with k=9, reaching 93.8%.
- **Support Vector Machine**: Delivered consistent accuracy across various C values, with a peak test accuracy of 93.5%.
- **Gradient Boosting Classifier**: Outperformed other models with a test accuracy of 97.4%, making it the chosen model for this system.

- **Confusion Matrix**: Each model's confusion matrix highlights true positives, true negatives, false positives, and false negatives. These metrics aid in assessing the model's effectiveness in minimizing misclassifications.

7. Visualization and Interpretation of Results

- **Visualization Tools**: Using Matplotlib and Seaborn, the system generates data visualizations that provide insights into model performance and feature distributions.
- **Confusion Matrix Visualization**: Displays model predictions vs. actual results, highlighting areas of improvement and strengths.
- **Comparison Graphs**: Showcasing accuracy scores and feature distributions, these graphs help assess each model's strengths, with Gradient Boosting clearly outperforming KNN and SVM.

8. Model Deployment and Prediction Workflow

- **Prediction Process**: For new URLs, the system preprocesses the data (cleans, scales, and extracts features) before feeding it into the trained Gradient Boosting model.
- **Classification Output**: The system labels URLs as "Phishing" or "Legitimate" based on learned patterns from the training phase.
- **Scalability Considerations**: Gradient Boosting is computationally intensive, so additional optimization techniques (like pruning or reducing tree depth) may be applied for faster predictions without sacrificing accuracy.

9. Conclusion and Future Enhancements

- **Performance Summary**: The system effectively identifies phishing websites, achieving a peak accuracy of 97.4% with Gradient Boosting.
- **Future Directions**:

  - **Incorporate Lexical Analysis**: Advanced NLP techniques could enhance feature engineering by analyzing URL patterns more deeply.
  - **Deploy Real-Time Detection**: Integration with web browsers or APIs to detect phishing websites in real-time.
  - **Model Optimization for Speed**: Further tune the model to ensure rapid detection in large-scale applications, crucial for real-time phishing prevention systems.

3.3 FEATURES

- **URL Feature Extraction**:

  Extracts relevant features from website URLs, such as URL length, presence of special characters, domain age, and keywords that indicate potential phishing activity.

- **Machine Learning Model Integration**:

Implements multiple machine learning algorithms like **K-Nearest Neighbors (KNN)**, **Support Vector Machines (SVM)**, and **Gradient Boosting** for phishing detection, allowing comparison of model performance to identify the best-fit algorithm.

- **Model Evaluation**:

  Evaluates models on accuracy, precision, and recall, providing insights into the model's effectiveness in real-world applications.

  Gradient Boosting achieved the highest accuracy in testing, suggesting strong predictive reliability for phishing detection.

- **Data Visualization**:

  Uses **Seaborn** and **Matplotlib** to visually analyze feature distributions and model performance metrics, aiding in understanding patterns and model accuracy across different algorithms.

- **Hyperparameter Tuning**:

  Includes tuning of parameters, especially with **KNN (k values)** and **SVM (C values)**, to optimize model performance for both training and testing datasets.

- **Dataset from Kaggle**:

  Utilizes a comprehensive dataset from Kaggle with both phishing and legitimate URLs, ensuring that the model trains on real, varied examples.

- **Python Libraries and Tools**:

  Employs **Python** as the primary programming language and libraries such as **NumPy**, **Pandas** for data processing, **Scikit-Learn** for machine learning models, **Seaborn** and **Matplotlib** for visualization.

- **Accessible Codebase**:

  Code available via GitHub repository for reproducibility and easy access for further enhancements or integrations.

3.4 Novelty of the proposal

The **Phishing Website Detection** project presents several novel aspects that enhance its significance in the field of cybersecurity. These innovations not only improve the detection accuracy but also contribute to the overall understanding of phishing threats and their mitigation. Below are the key novelties of the project:

1. **Comprehensive Feature Set**:
   The project utilizes a diverse range of features extracted from URLs, which includes not only conventional metrics like URL length and the presence of special characters but also advanced features such as domain age and the use of subdomains. This comprehensive feature set provides a robust framework for distinguishing between legitimate and phishing websites, enabling the model to recognize patterns that may not be immediately apparent.
2. **Multi-Algorithm Approach**:

Unlike many existing solutions that rely on a single algorithm, this project implements a multi-algorithm strategy by incorporating K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Gradient Boosting classifiers. This diversity allows for a thorough comparison of performance metrics, enabling the identification of the most effective model for phishing detection. The adaptability of using various algorithms enhances the project's applicability across different contexts and datasets.

3. **Hyperparameter Optimization**:
   The project goes beyond basic model implementation by including hyperparameter tuning, particularly for KNN and SVM. By experimenting with different values for k in KNN and C in SVM, the project achieves improved accuracy and reduces overfitting, making the model more generalizable. This meticulous approach to tuning demonstrates a commitment to optimizing performance and ensuring reliability in real-world applications.

4. **Visualization of Results**:
   The inclusion of data visualization techniques using Seaborn and Matplotlib serves as a powerful tool for interpreting model performance and feature importance. By visually representing the accuracy scores and feature distributions, the project enhances the understanding of the underlying patterns that contribute to phishing detection. This feature not only aids in model evaluation but also provides stakeholders with clear insights into the decision-making process of the machine learning algorithms.

5. **User-Centric Design**:
   The project emphasizes user-friendliness by providing an easily accessible GitHub repository, where users can explore the code, replicate the results, and contribute to the project. This open-source approach fosters community engagement and encourages collaborative improvements, which are essential for keeping pace with evolving phishing tactics.

6. **Real-World Applicability**:
   By utilizing a dataset sourced from Kaggle that encompasses a wide range of phishing and legitimate URLs, the project ensures that the model is trained on realistic and diverse data. This enhances its practical applicability and prepares it to handle a variety of phishing scenarios encountered in everyday internet use.

7. **Contribution to Cybersecurity Knowledge**:
   Finally, the project adds to the body of knowledge in the field of cybersecurity by offering insights into the effectiveness of different machine learning algorithms for phishing detection. The findings can guide future research and development efforts, making it a valuable resource for academics, industry professionals, and researchers looking to understand and combat phishing threats.

CHAPTER 5

CONCLUSION

Enhanced Online Security: The project has significantly bolstered online security by providing real-time detection and warnings about potentially phishing websites. Users are now better equipped to identify and avoid fraudulent URLs, reducing their vulnerability to phishing attacks.

User Empowerment: A user-centric approach has been at the heart of this project. Through user-friendly interfaces, educational resources, and interactive features, users have become more knowledgeable and proactive in recognizing phishing attempts, ultimately contributing to a safer online environment.

Cutting-Edge Technology: The project has leveraged advanced technologies, such as sophisticated URL analysis algorithms and machine learning, to continuously improve detection accuracy and stay ahead of evolving phishing techniques.

Privacy Protection: A paramount concern, the project has maintained strict adherence to privacy protection measures, ensuring that user data remains secure and private while using the system.

Community Involvement: By fostering a sense of community, the project has enabled users to actively contribute to a shared threat intelligence database. This collective effort has strengthened the project's effectiveness in combatting phishing.

Global Collaboration: Collaboration with cybersecurity organizations, governments, and international entities has solidified the project's position in the global effort to combat phishing. Sharing threat data and best practices has had a far-reaching impact on online security.

Inclusivity and Accessibility: The project has made strides in ensuring inclusivity and accessibility for users with diverse needs and capabilities, making online security more attainable for everyone.

Scalability and Sustainability: Designed with scalability and sustainability in mind, the project is well-prepared to accommodate a growing user base and adapt to emerging online security challenges.

Continuous Improvement: The commitment to ongoing research, development, and user feedback has enabled the project to remain responsive to changing threats and user needs. Continuous improvement is central to the project's success.

In summary, the project for the detection of phishing websites using URLs has not only made significant strides in enhancing online security but has also empowered users with the knowledge and tools needed to protect themselves from phishing attacks.

REFERENCES

1. Anti-Phishing Working Group (APWG). (https://apwg.org/): An industry association focused on combating phishing and cybercrime, offering valuable reports and resources.
2. PhishTank. (https://www.phishtank.com/): A community-driven platform for reporting and verifying phishing URLs.
3. Krebs, B. (https://krebsonsecurity.com/): Brian Krebs' cybersecurity blog often covers phishing trends and incidents.
4. OWASP Anti-Phishing Landing Page. (https://owasp.org/www-community/attacks/Phishing): Information on phishing attacks and prevention from the Open Web Application Security Project (OWASP).
5. Google Safe Browsing. (https://developers.google.com/safe-browsing): Google's Safe Browsing API and resources for safe web browsing.
6. McAfee. (https://www.mcafee.com/): McAfee's cybersecurity resources include articles and reports on phishing threats.
7. Symantec (Norton). (https://www.broadcom.com/company/newsroom/press-releases?filters=phishing): Symantec's press releases and articles on phishing-related topics.
8. Phishing.org. (https://www.phishing.org/): A comprehensive resource for information on phishing attacks and prevention.
9. Verizon. (https://enterprise.verizon.com/resources/reports/dbir/): The Verizon Data Breach Investigations Report (DBIR) often includes insights on phishing incidents.
10. Cybersecurity & Infrastructure Security Agency (CISA). (https://www.cisa.gov/): CISA provides resources and alerts related to cybersecurity threats, including phishing.
11. SANS Institute. (https://www.sans.org/): SANS offers various cybersecurity training courses and whitepapers related to phishing and web security.
12. Google Safe Browsing API: Google's API for checking websites for phishing and malware.

    Website: https://developers.google.com/safe-browsing

13. Kaggle: A platform for finding datasets and machine learning resources, including datasets related to phishing detection.

Website: https://www.kaggle.com/

14. Cybersecurity and Infrastructure Security Agency (CISA): Provides cybersecurity resources and alerts about current threats.

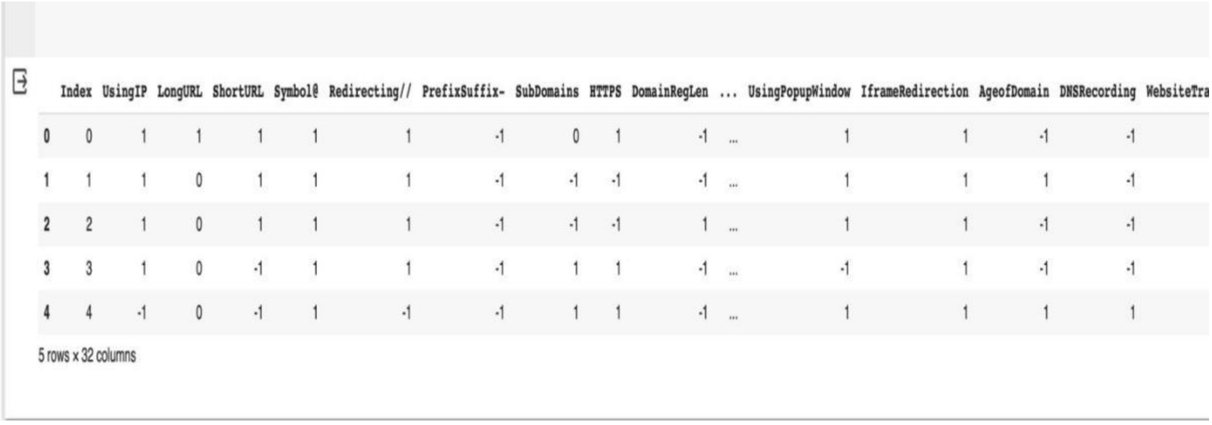
Website: https://www.cisa.gov/

15. Machine Learning Mastery: Tutorials and articles on machine learning techniques, including those related to cybersecurity.

Website: https://machinelearningmastery.com/

Appendix

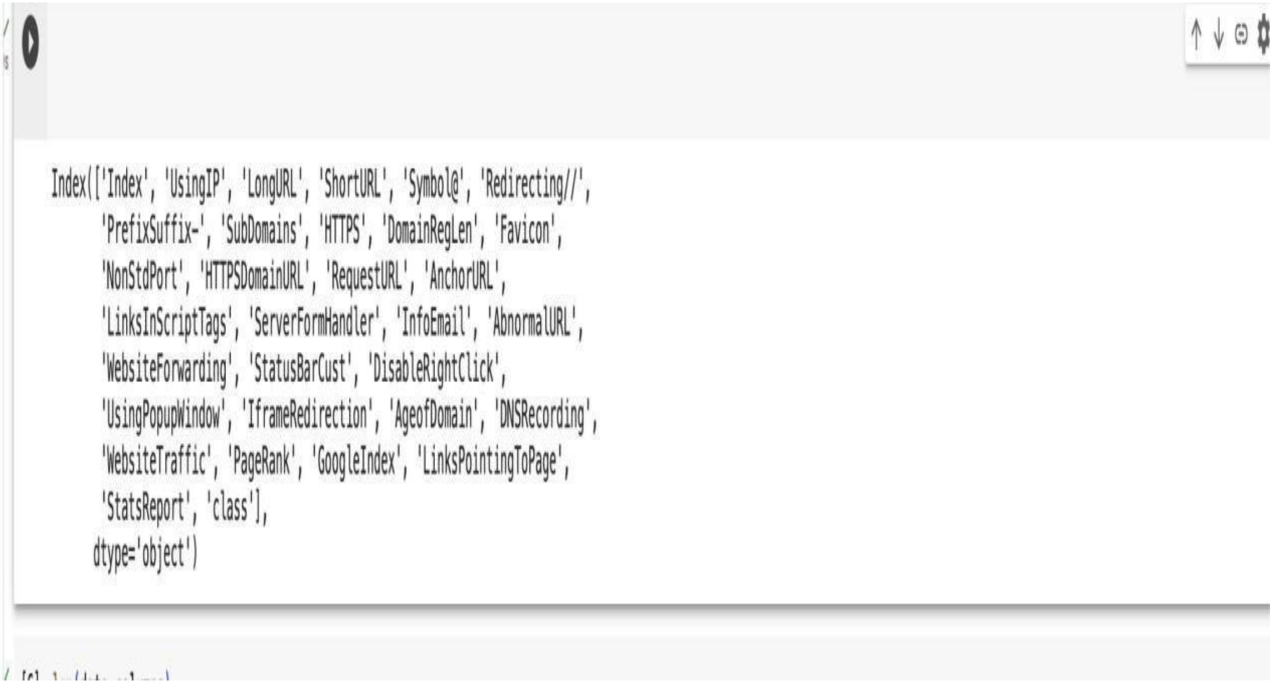
Output:



	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection	AgeofDomain	DNSRecording	WebsiteTra
0	0	1	1	1	1	1	-1	0	1	-1	...	1	1	-1	-1	
1	1	1	0	1	1	1	-1	-1	-1	-1	...	1	1	1	-1	
2	2	1	0	1	1	1	-1	-1	-1	1	...	1	1	-1	-1	
3	3	1	0	-1	1	1	-1	1	1	-1	...	-1	1	-1	-1	
4	4	-1	0	-1	1	-1	-1	1	1	-1	...	1	1	1	1	

5 rows x 32 columns

Figure :**1** - Column names in the database



```
Index(['Index', 'UsingIP', 'LongURL', 'ShortURL', 'Symbol@', 'Redirecting//',
       'PrefixSuffix-', 'SubDomains', 'HTTPS', 'DomainRegLen', 'Favicon',
       'NonStdPort', 'HTTPSDomainURL', 'RequestURL', 'AnchorURL',
       'LinksInScriptTags', 'ServerFormHandler', 'InfoEmail', 'AbnormalURL',
       'WebsiteForwarding', 'StatusBarCust', 'DisableRightClick',
       'UsingPopupWindow', 'IframeRedirection', 'AgeofDomain', 'DNSRecording',
       'WebsiteTraffic', 'PageRank', 'GoogleIndex', 'LinksPointingToPage',
       'StatsReport', 'class'],
      dtype='object')
```

` **Figure** :**2** - Column names in the database

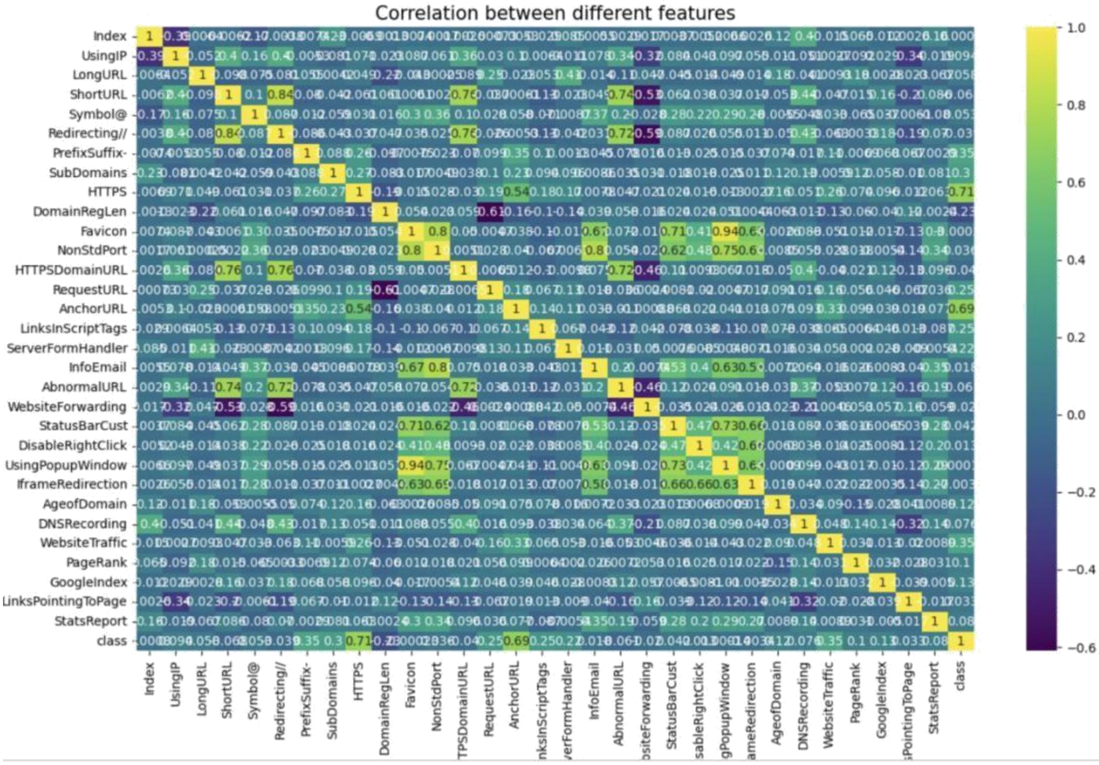


Figure :3 - Correlation between different features

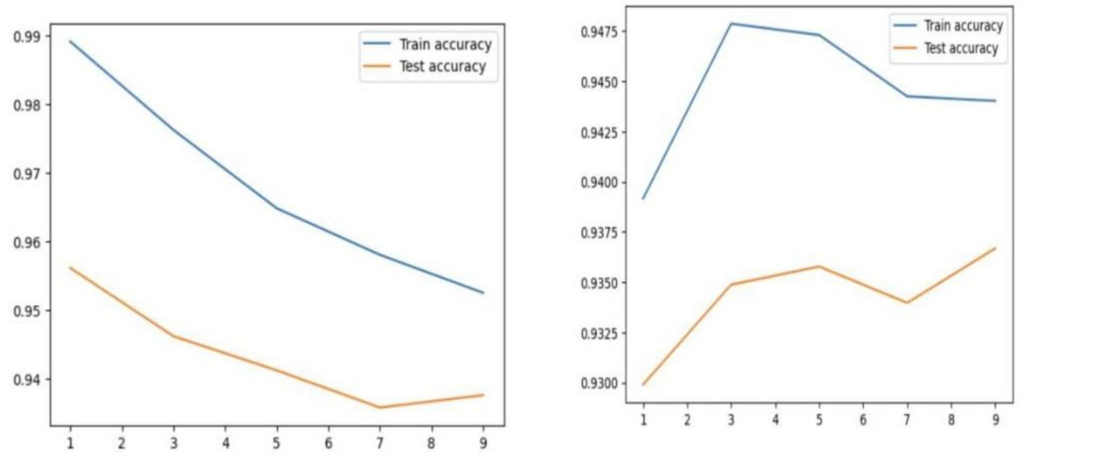


Figure :4 - train-test accuracy

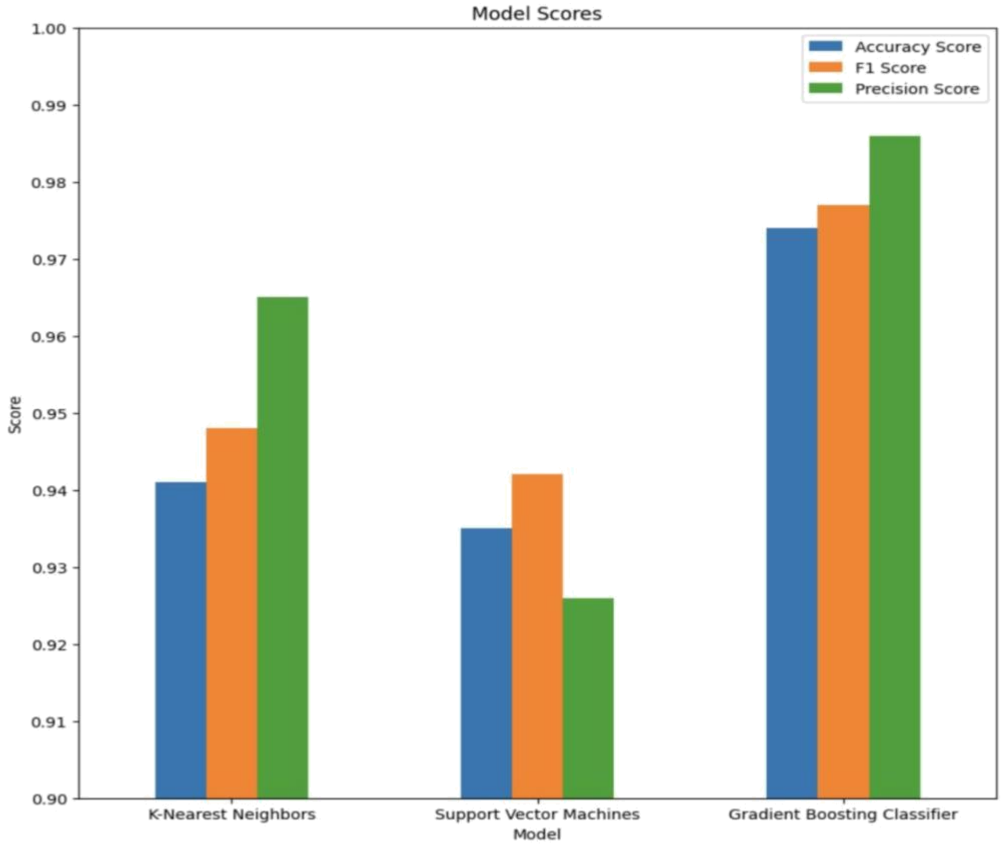
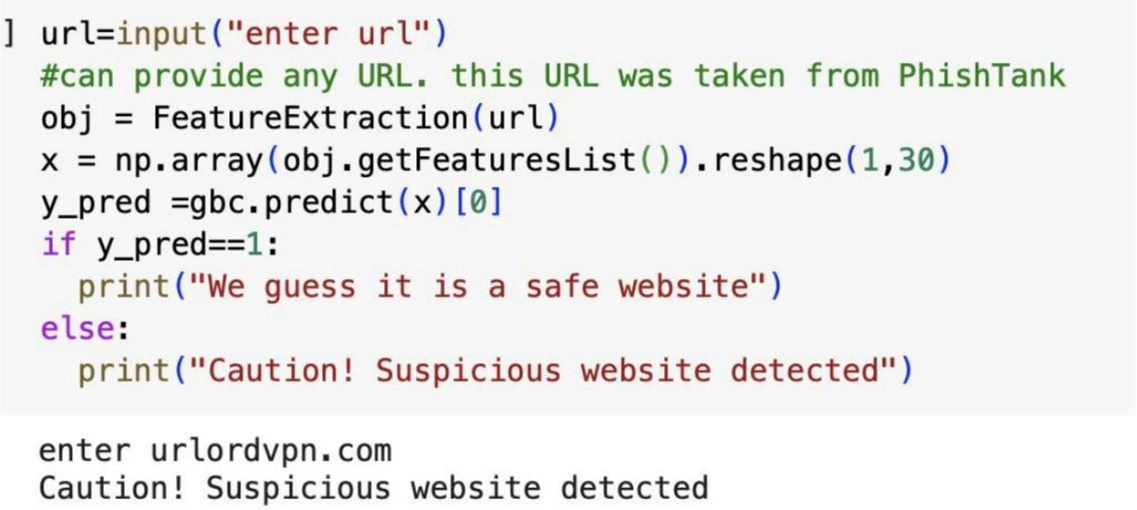
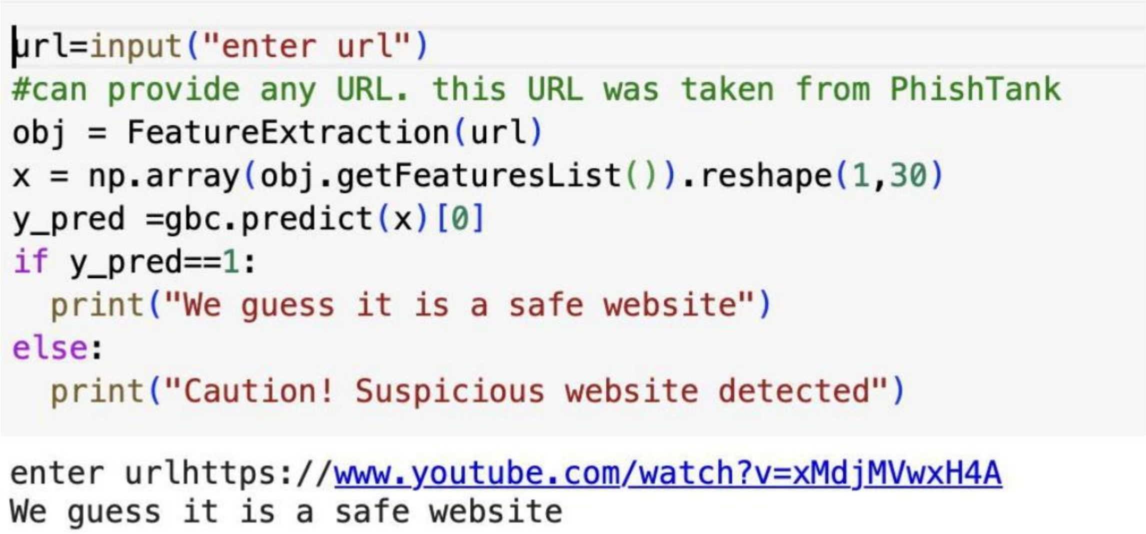


Figure :5 - Model Scores



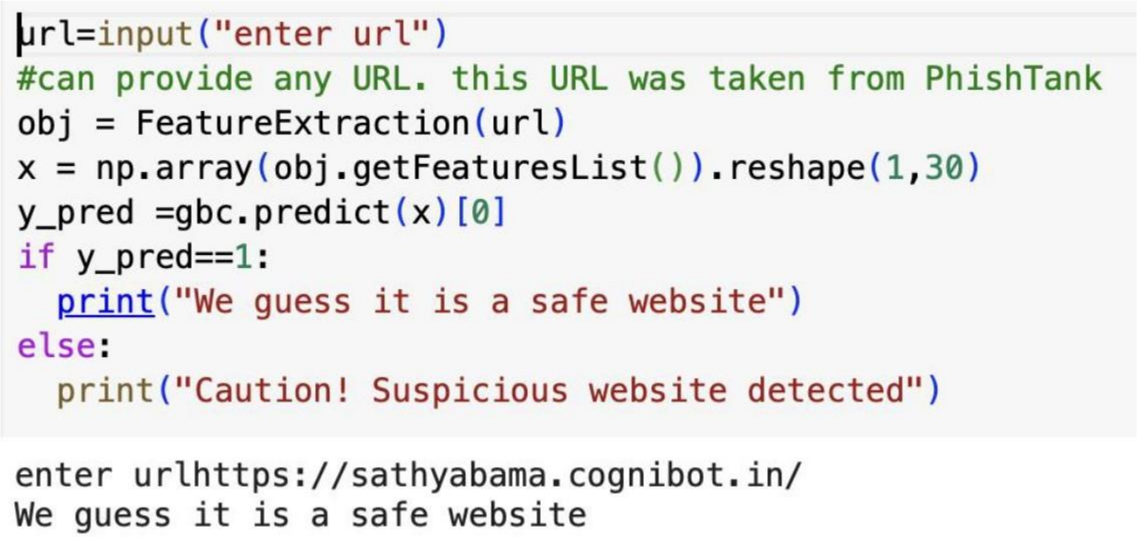
```
] url=input("enter url")
  #can provide any URL. this URL was taken from PhishTank
  obj = FeatureExtraction(url)
  x = np.array(obj.getFeaturesList()).reshape(1,30)
  y_pred =gbc.predict(x)[0]
  if y_pred==1:
    print("We guess it is a safe website")
  else:
    print("Caution! Suspicious website detected")
```

```
enter urlordvpn.com
Caution! Suspicious website detected
```



```
url=input("enter url")
#can provide any URL. this URL was taken from PhishTank
obj = FeatureExtraction(url)
x = np.array(obj.getFeaturesList()).reshape(1,30)
y_pred =gbc.predict(x)[0]
if y_pred==1:
  print("We guess it is a safe website")
else:
  print("Caution! Suspicious website detected")
```

```
enter urlhttps://www.youtube.com/watch?v=xMdjMVwxH4A
We guess it is a safe website
```



```
url=input("enter url")
#can provide any URL. this URL was taken from PhishTank
obj = FeatureExtraction(url)
x = np.array(obj.getFeaturesList()).reshape(1,30)
y_pred =gbc.predict(x)[0]
if y_pred==1:
  print("We guess it is a safe website")
else:
  print("Caution! Suspicious website detected")
```

```
enter urlhttps://sathyabama.cognibot.in/
We guess it is a safe website
```

Figure :6 - Detection of phishing websites