DETECTION OF PHISING WEBSITES USING MACHINE LEARNING

BY

ARNAB KUMAR DAS - 10200220008
SHUBHANKAR DAS - 10200220006
SAYAN BANERJEE - 10200220041
SARTHAK DEY - 10200220015

UNDER THE GUIDANCE OF : PROF. MALABIKA SENGUPTA

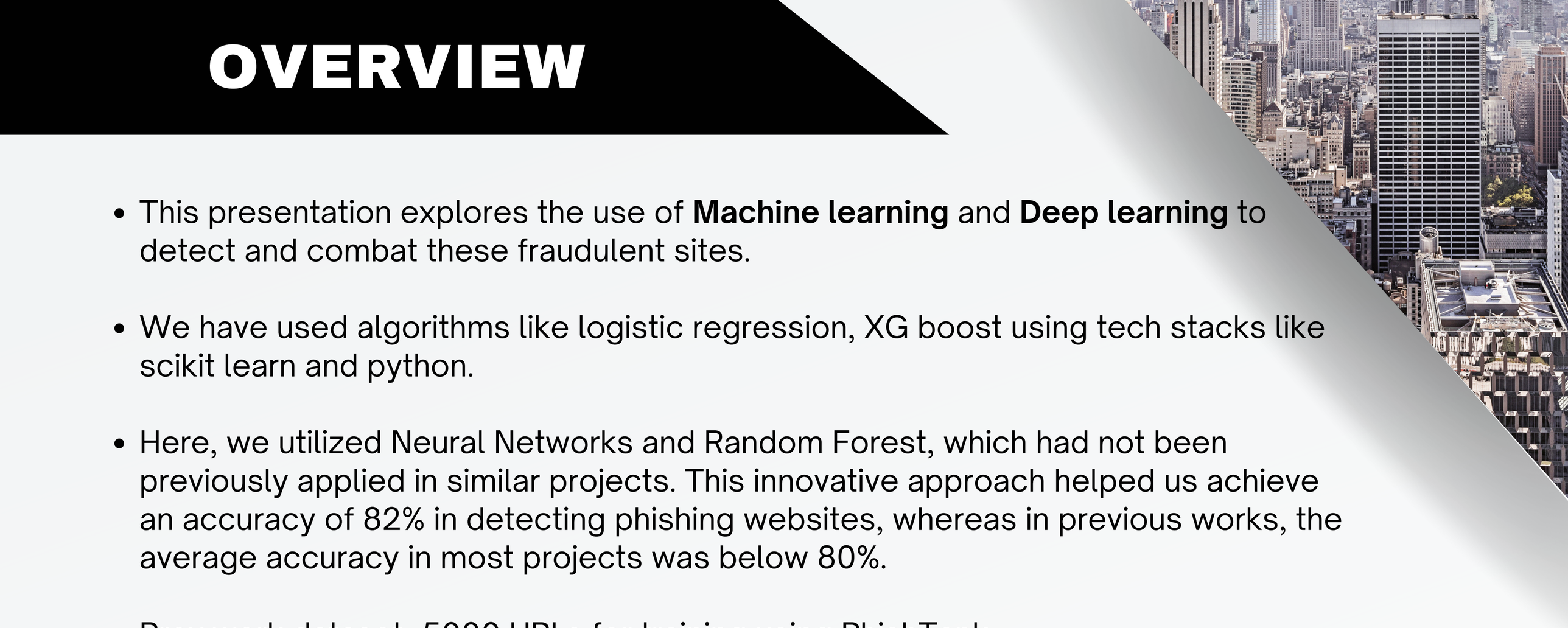
PROBLEM STATEMENT

Problem Statement: The increasing frequency of phishing attacks threatens online security, requiring a sophisticated solution to detect phishing websites accurately.

Objectives: Develop a high-accuracy machine learning model for phishing URL classification. Implement real-time detection to prevent access to malicious sites.

Outcome: A proactive solution that enhances online security for users and professionals.

OVERVIEW



- This presentation explores the use of **Machine learning** and **Deep learning** to detect and combat these fraudulent sites.

- We have used algorithms like logistic regression, XG boost using tech stacks like scikit learn and python.

- Here, we utilized Neural Networks and Random Forest, which had not been previously applied in similar projects. This innovative approach helped us achieve an accuracy of 82% in detecting phishing websites, whereas in previous works, the average accuracy in most projects was below 80%.

- Prev work dataset- 5000 URLs for training using PhishTank
  Our work dataset- 5,49346 URLs for training using Kaggle

- Implementation of the model: We have created a simple web application that is user-friendly, scalable, and efficient.

FACTORS TO CONSIDER FOR ANALYSIS

URL Based features

Evaluate the authenticity using features like number of dots, http/https present in url, length of url string.

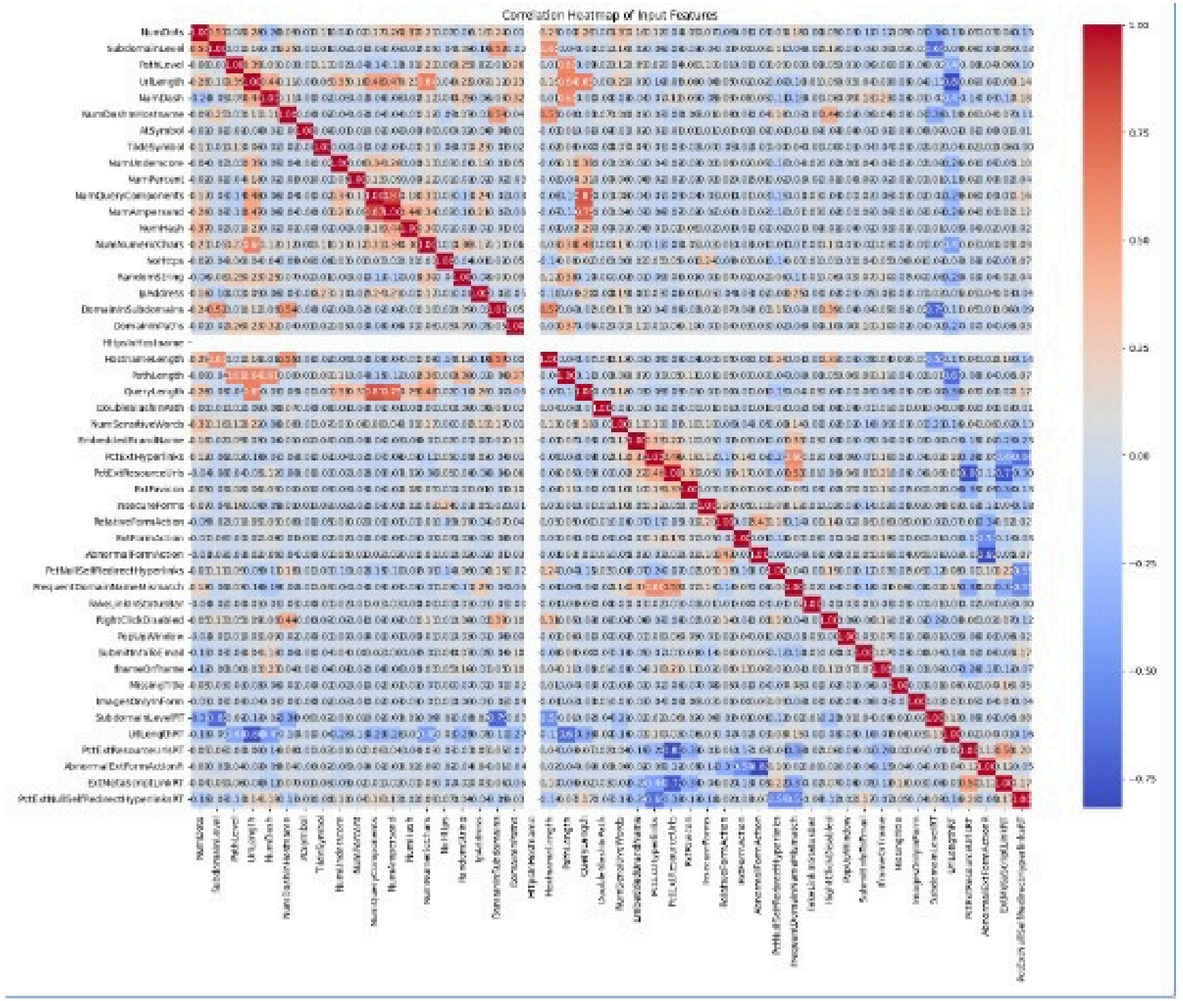
DNS Filter

Analyze the DNS filters used by the website.
Features- Domain age, IP address, Number of sub-domains.

HTML and JavaScript based features

Evaluate the authenticity using features like right click disable or not, Iframes, self redirecting hyperlinks.

CO-RELATION OF HEATMAP



LITERARY SURVEY

SL No	Author and Year	Aim	Main findings	Limitations
1	**Title:** Detecting phishing websites using machine learning technique **Author:** Ashit Kumar Dutta, 2021	The proposed framework employs RNN- LSTM to identify the properties Pm and Pl in an order to declare an URL as malicious or legitimate	The outcome of this study reveals that the proposed method presents superior results rather than the existing deep learning methods	The future direction of this study is to develop an unsupervised deep learning method to generate insight from a URL
2	**Title:** A systematic literature review on phishing website detection techniques **Authors:** Qabajeh et al., 2018	This review paper compares traditional anti-phishing methods, which includes raising awareness, educating users, conducting periodic training or workshop, and using a legal perspective. The Computerized anti-phishing techniques talk about list-based and machine-learning techniques	Machine Learning and rule induction are suitable to combat phishing due to their high detection rate and, more importantly, the easy-to-understand outcomes.	Sixty-seven studies were analyzed in work, and the research did not discuss Deep Learning techniques.
3	**Title:** Classification of Phishing Attack Solutions by Employing Deep Learning Techniques: A Systematic Literature Review **Authors:** Eduardo Benavides,2020	This systematic literature review aimed to evaluate various other scholars' proposals for identifying phishing attacks using Deep Learning algorithms	In conclusion, there is still a significant gap in the area of Deep Learning algorithms for phishing attack detection..	This work includes 19 studies, and only research articles on phishing and Deep Learning are considered in this study.
4	**Title:** Applications of deep learning for phishing detection: a systematic literature review **Author:** Catal et al., 2022	The work answers nine research questions. The main aim is to synthesize, assess, and analyses Deep Learning techniques for phishing detection.	According to this study, 42 studies applied Supervised ML algorithms out of 43 studies. The most used algorithm was DNN, and the best performance was given by DNN and Hybrid DL algorithms.	The work only discusses Deep Learning related studies for phishing detection.

COMPARISON TABLE

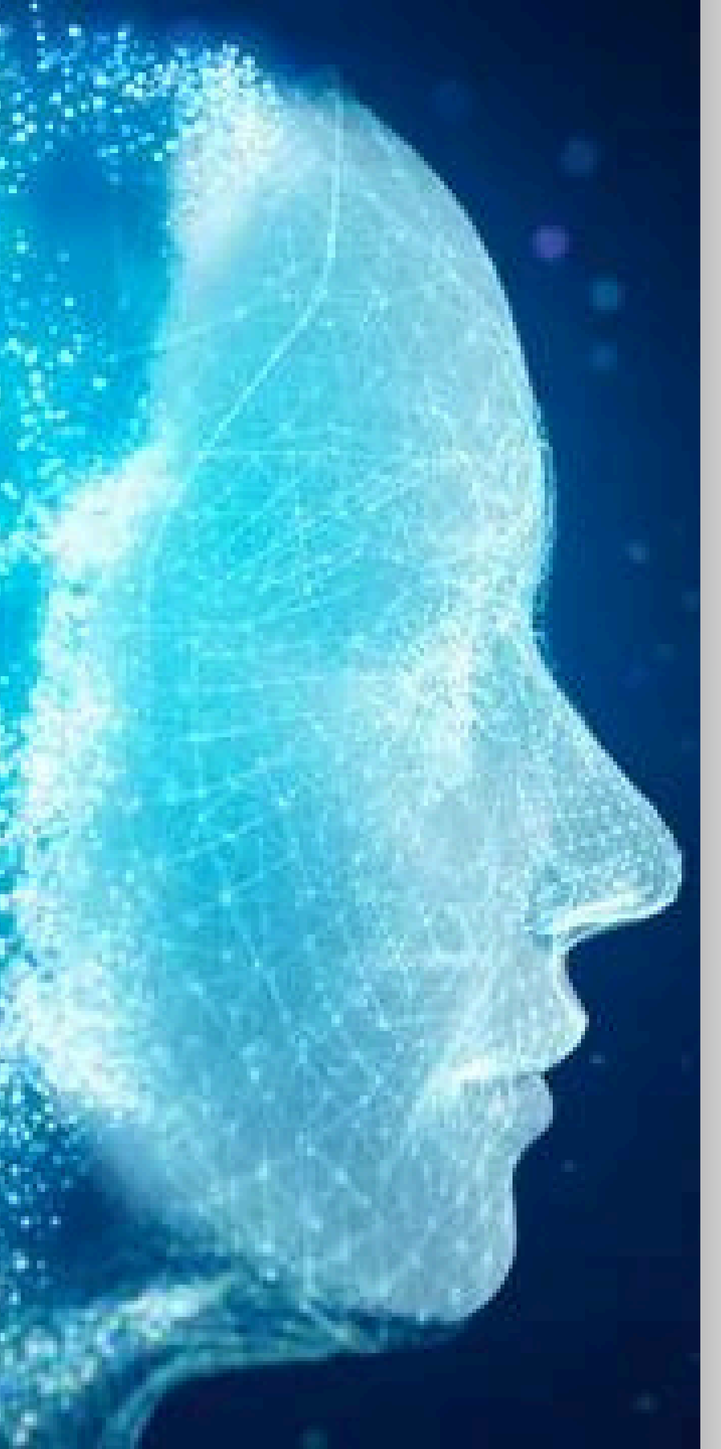
FACTORS	PREVIOUS WORK	OUR WORK
Dataset	5000 URLs for training using PhishTank.	5,49346 URLs for training using Kaggle.
Feature Extraction	8-10 features have been used like IP address in url, prefix or suffix, web traffic.	50+ features have been used like self redirecting hyperlink, right click disable or not, number of pop-ups coming.
Accuracy	As per our research the accuracy score of previous models is 80-90%.	In our model accuracy score is 93.85%.
Correlation	As per as our research concerns no one has yet analyzed the correlation among all the features.	In our work we have provided a data analysis of correlation among the features.

POTENTIAL OUTCOMES OF THE PROJECT

01 Enhanced Cybersecurity By effectively detecting and blocking phishing websites, users can reduce the risk of falling victim to identity theft, financial fraud, and other cybercrimes.

02 Time and Cost Savings Automated detection of phishing websites saves time and resources by minimizing manual efforts in identifying and reporting fraudulent sites.

03 User Education Through awareness campaigns and educational initiatives, users can become more knowledgeable about the risks associated with phishing and take proactive measures to protect themselves.



DATA SPLITTING

We are using a downloaded dataset from Kaggle. Dataset contains 5000 spam URLs and 5000 legitimate URLs and at the end we are splitting the dataset and using 80% of it as training data and 20% of it as testing data.

80%

TRAINING DATA

20%

TESTING DATA

IMPLEMENTATION

Below are the steps of how we plan on approaching the problem statement:

1.Define Objectives: We are building a website where we will collect links of suspected websites. Then we will check the sites using the machine learning algorithms. If the site is a phishing website, we'll add it in our database and then submit the report to organizations like APWG.

2.Data Collection: We collected our training and test data from the UCI phishing dataset that is publicly available.

3.Feature Extraction: Identify relevant features from the URLs that can help distinguish between phishing and legitimate websites. Features might include URL length, presence of HTTPS, domain age, and other relevant characteristics.

4.Data Preprocessing: Clean and preprocess the dataset. This involves handling missing values, encoding categorical variables, and scaling numerical features.

5.Split Data: Divide the dataset into training and testing sets. This allows you to train the model on one subset and evaluate its performance on unseen data.

6. Model Selection: We chose to start with the Random Forest classifier to work with. After reading various works on this field, many have approached the problems with this algorithms. We plan on starting off with random forest and testing other algorithms too on the way to determine which works best for our requirement.

7. Feature Selection: The difficulty arises when we must determine what are the most relevant features from a set and what combination of features give us near perfect classification accuracies. From the 30 features, we identified a few subsets.

8. Training: Train the model using the training dataset. The model will learn to distinguish between phishing and legitimate websites based on the provided features.

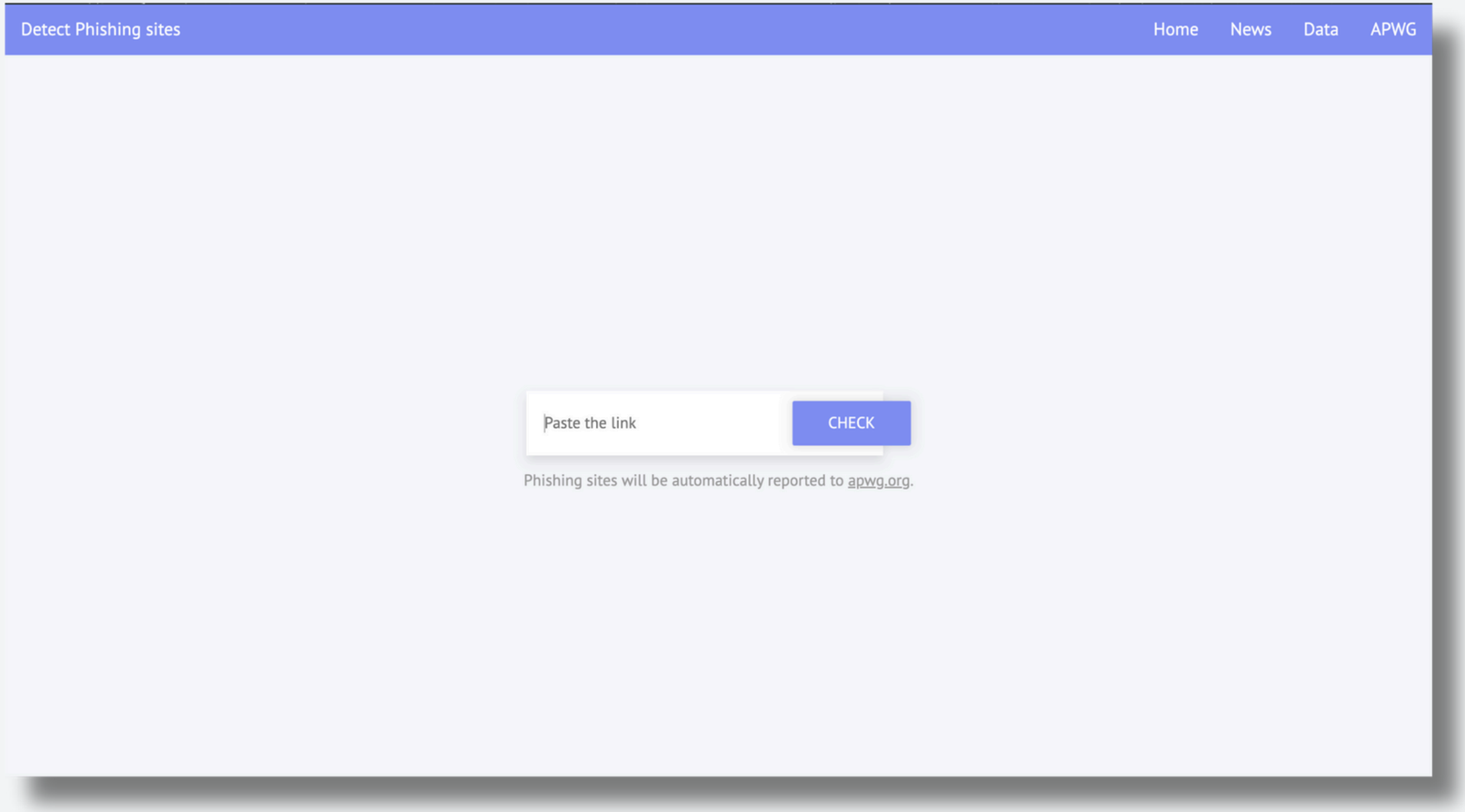
9. Testing: Evaluate the model's performance on the testing dataset. Use metrics such as accuracy, precision, recall, and Fscore to assess its effectiveness.

10.Hyperparameter Tuning: Optimize the performance of the model by tuning its hyperparameters.

11.Validation: Perform additional validation, such as cross-validation, to ensure the model's generalizability and robustness.

12.Deployment: Once satisfied with the model's performance, deploy it for real-time detection. This could involve integrating it into a web application, browser extension, or network security system.

User Interface

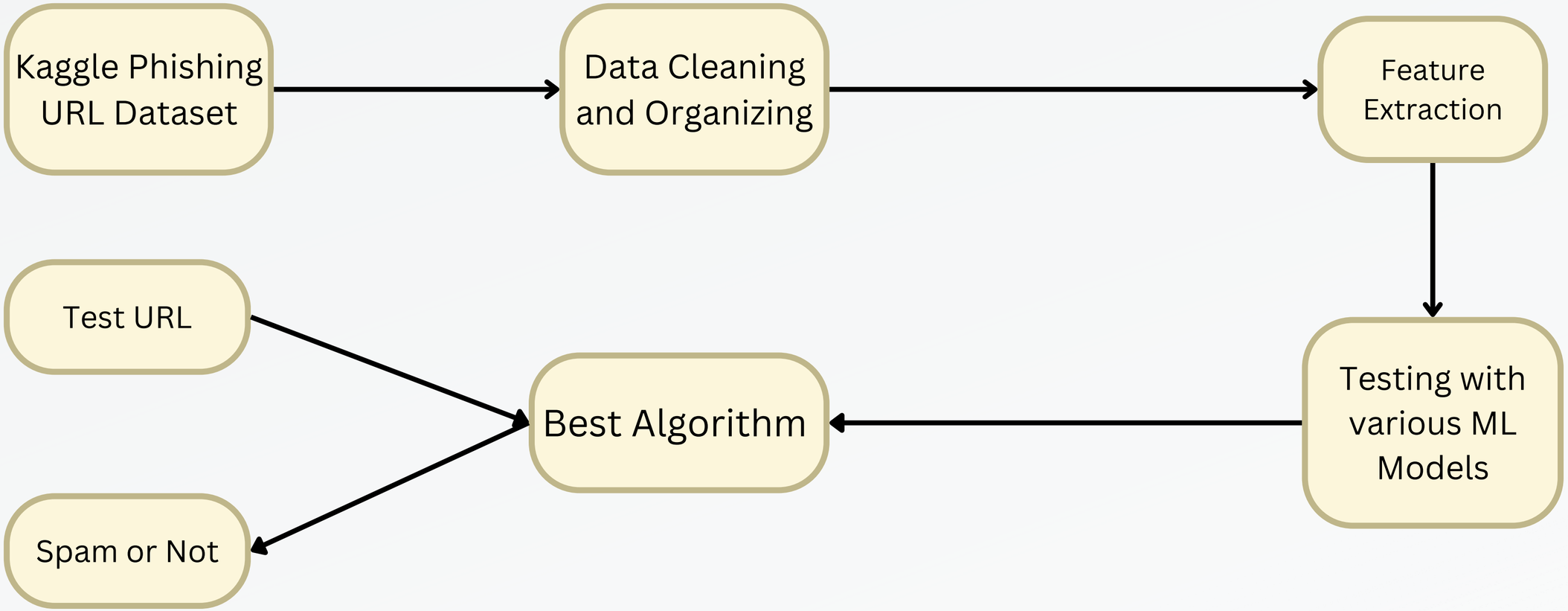


FINAL INPUT TO OUTPUT FEATURE

```
Provide URL: ↑↓ for history. Search history with c-↑/c-↓
```

```
Provide URL: http://secure.bankofamerica-login.com/login.php?redirect=http://bankofamerica.com
The predicted class for the given URL is: Spam
```

BLOCK DIAGRAM



EVALUATION METRICS

Performance Metrics of Logistic Regression

```
Training Accuracy: 94.74%
Testing Accuracy: 93.90%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       988
           1       0.93      0.95      0.94      1012

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000

Confusion Matrix:
[[919  69]
 [ 53 959]]
```

EVALUATION METRICS

Performance Metrics of XGBoost

```
Training Accuracy: 1.0
Testing Accuracy: 0.9895
Confusion Matrix:
[[ 974   14]
 [   7 1005]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       988
           1       0.99      0.99      0.99      1012

    accuracy                           0.99      2000
   macro avg       0.99      0.99      0.99      2000
weighted avg       0.99      0.99      0.99      2000
```

FUTURE WORK

Moving forward, our roadmap includes the following key milestones:

- API Integration: Seamlessly link the algorithms to the website through APIs, ensuring a user-friendly and responsive interface.
- Database Creation: Establish a robust database structure to store information on identified phishing websites, enhancing data management and analysis capabilities.
- Submission to APWG: Implement a systematic process to submit reports on identified phishing websites to organizations like APWG, contributing to the broader cybersecurity community.
- User Feedback Integration: Incorporate mechanisms for user feedback to continuously improve and refine the performance of our phishing detection system.
- Web Extension Development: Develop a web extension add-on to extend the reach of our phishing detection tool, providing users with added convenience and accessibility.

Thank You