# Table of Contents

## Executive Summary

One type of internet security issue that targets human weaknesses instead of software flaws is phishing websites. It can be defined as the practice of luring internet users in order to get private data, including passwords and usernames.This research aims to train deep neural networks and machine learning models using a dataset designed to identify phishing websites. The necessary URLs and website content-based attributes are taken from a dataset that contains both phishing and authentic URLs. Every model's performance level is quantified and contrasted.

## Introduction/Background

Phishing is a malicious practice used to detect malicious links in email, text messages, and other online sources. It is used to steal sensitive information such as usernames, passwords and financial information of the victims.The users may have to suffer from long-term financial loss in case of stealing credentials. Phishing is the most common attack to steal a user's confidential information through social engineering. Moreover, successful phishing attacks can result in compromising the entire system with severe consequences.

In order to deal with the escalating phishing attacks, there is a need to develop a tool to prevent phishing attacks in the near future. The tool will help to detect phishing URLs and prevent the attack within efficient response time. Moreover, this tool will raise awareness among users to identify phishing attacks.

.

## Specification

**Data Collection:**

The collection of phishing URLs comes from the open-source PhishTank service. This site offers an hourly updated collection of phishing URLs in several formats, including csv and json. To train the machine learning models, a dataset of 500 randomly selected phishing URLs is gathered.

The authentic URLs can be obtained from the University of New Brunswick's publicly available datasets. A variety of benign, spam, phishing, malware, and defacement URLs are included in

this dataset. The benign url dataset is taken into consideration for this study out of all of these types. To train the ML models, 500 randomly selected valid URLs are gathered from this dataset.

**Feature Extraction:**

The features in the category shown below were taken from the URL data:

Address Bar-based Functionalities

➔ Eight traits are extracted in this category.

Features based on Domains

➔ Three features have been retrieved in this category.

Features based on Javascript and HTML

➔ Four features have been retrieved in this category.

Thus, 14 features in all are taken out of the 1000 URL dataset.

**Training & Models**

The data is split into 80-20, or 800 training samples and 200 testing samples, prior to beginning the ML model training. This is obviously a supervised machine learning task based on the dataset. Classification and regression are the two main categories of supervised machine learning problems.

Since the input URL is categorized as either legitimate (0) or phishing (1), this data set falls under the classification problem category. The following supervised machine learning models (classification) were taken into consideration for this project's dataset training:

➔ Decision Tree

➔ Random Forest

➔ Support Vector Machines

➔ Logistic Regression

The test dataset is used to evaluate each model after it has been trained on the dataset.

## Design

The web-based phishing detection and prevention tool follows the Model-View-Controller (MVC) architecture. In this type of architecture, model represents the business logic which includes the different models trained on the dataset. View represents the HTML pages which are to be shown to the user so that the user can interact with the models. Lastly, Controller represents the view functions which are accessed through routes defined in the Flask application. In the business logic component of the tool, models are trained and saved which are then accessed when the user wants to classify a URL as either phishing or legitimate. The user interfaces are attached in the 'Description of final prototype/section' section.

## Implementation and Testing

1. **Feature Extraction:**

   Before getting to specifics of machine learning models, it is essential to extract the features first because the quality of the features directly impacts the model's ability to make accurate predictions. To increase the performance of the model, a total of 14 features were extracted from each URL.

   a. IP Address Check:
   b. @ Symbol Presence Check:
   c. URL Length Check:
   d. URL Depth Calculation:
   e. Redirection Check:
   f. HTTPS Domain Check:
   g. Dash in Domain Check:
   h. Iframe Presence Check:
   i. Status Bar Check:
   j. Right-Click Disable Check:
   k. Web Forward Check:
   l. Website Traffic Recognition Check:
   m. Domain Expiry Calculation:
   n. WHOIS Information Retrieval:
   o. Domain Age Calculation:

2. **Model Training:**

Now that the relevant features have been extracted from the given URL, it's time to train the machine learning models so that their results can be compared. For this purpose, four machine learning models were used: Logistic Regression, Support Vector Machines, XGBoost, Random Forest and Decision Trees to compare the results. The models were trained on a labeled dataset in order to learn patterns between the above extracted features. The dataset was first splitted into training and testing sets using a 80-20 ratio. Then, the model was trained using the scikit-learn library. The sample code for dataset splitting and Decision Tree model training is attached below:

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
random_state=42)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

The same code template was used to train Logistic Regression, SVM and Random Forest.

### 3. Model Testing:

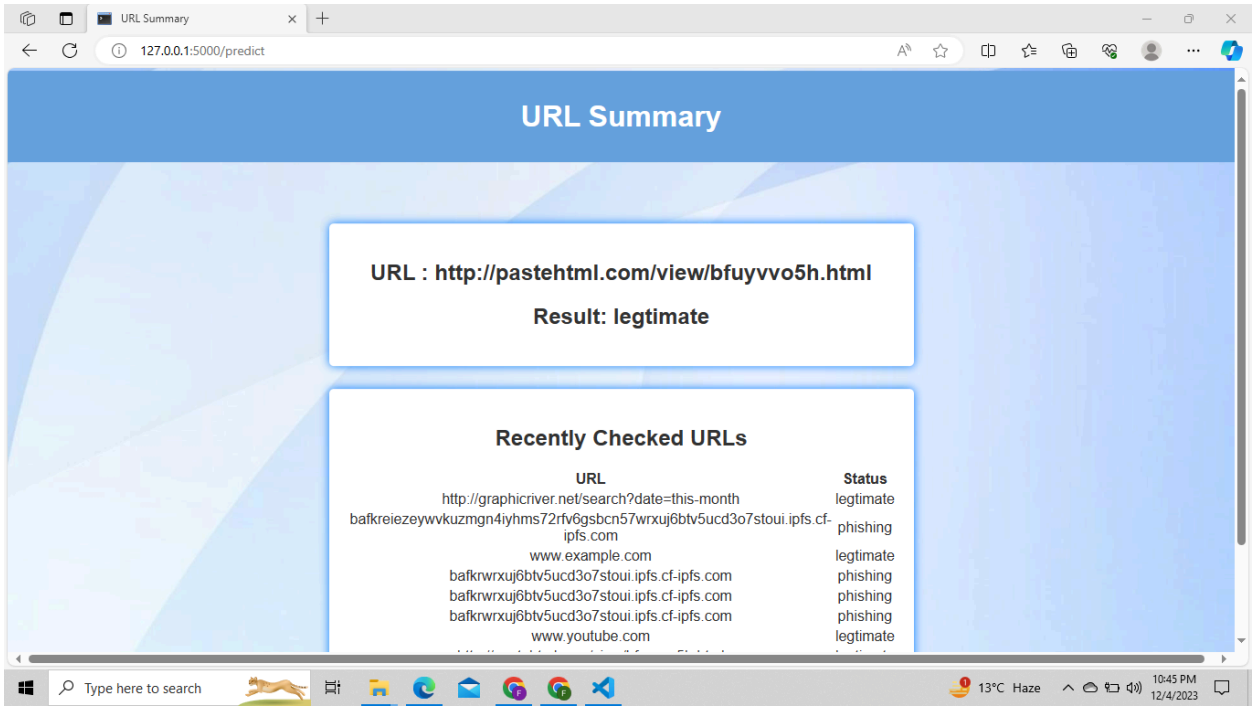
After training the models, the next important step is model testing. In order to test the models, the test set was used. The models were compared on the basis of train and test accuracies, precision, recall and F1 score. The code snippet and test result are attached below.

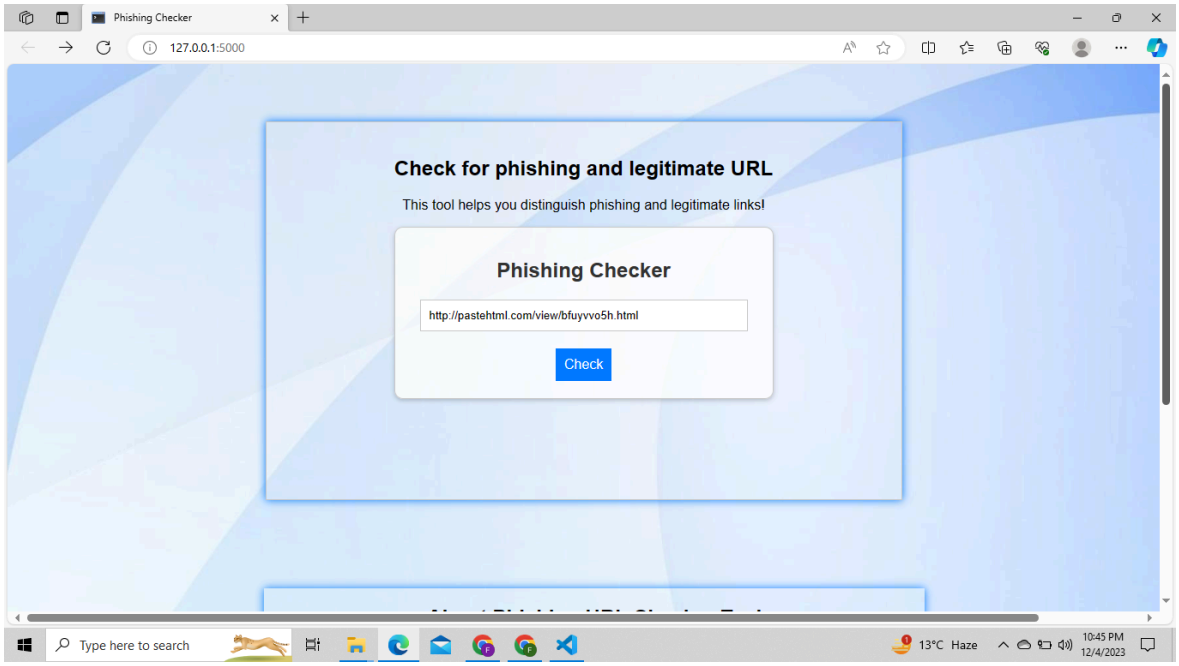
```
# Make predictions on the test set
y_test_pred = model.predict(X_test)
accuracy_test = accuracy_score(y_test, y_test_pred)
print(f"Training Accuracy: {accuracy_train * 100:.2f}%")
print(f"Test Accuracy: {accuracy_test * 100:.2f}%")

precision_dt = precision_score(y_test, y_test_pred)
recall_dt = recall_score(y_test, y_test_pred)
f1_dt = f1_score(y_test, y_test_pred)
print(f"Decision Tree Precision: {precision_dt:.2f}, Recall: {recall_dt:.2f}, F1
Score: {f1_dt:.2f}")
```

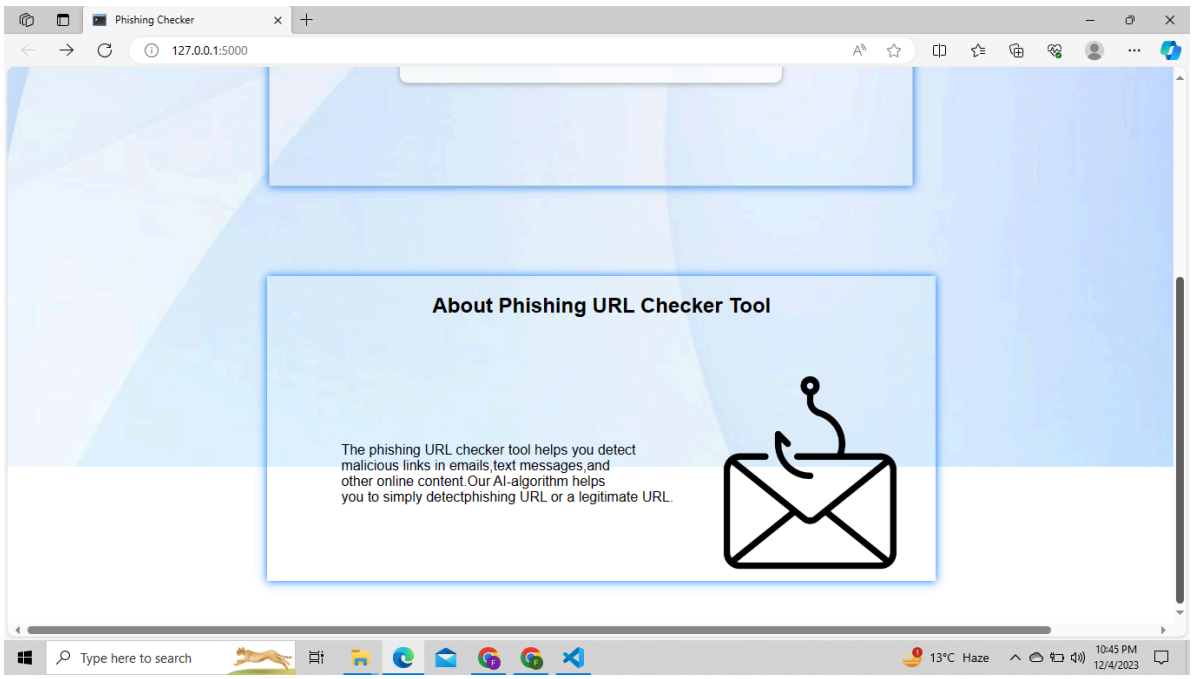
| | Logistic Regression | SVM | Random Forest | Decision Tree |
|---|---|---|---|---|
| **Train Accuracy** | 87.59% | 86.84% | 92.86% | 100.0% |
| **Test Accuracy** | 90.00% | 88.00% | 91.00% | 91.00% |
| **Precision** | 1.00 | 1.00 | 0.96 | 0.85 |
| **Recall** | 0.79 | 0.74 | 0.84 | 0.99 |
| **F1 Score** | 0.88 | 0.85 | 0.90 | 0.91 |

## Description of the final prototype/product

After a thorough comparison of the performance of the four machine learning models, the best performers came out to be Random Forest as it displayed an overall good balance between precision and F1 score and Decision Tree. The Decision Tree model was incorporated in our web based phishing detection and prevention tool. The user enters a URL which needs to be categorized as phishing or legitimate. The pre-trained model is used to make the prediction and the result is displayed to the user on the website.


URL Summary
127.0.0.1:5000/predict
URL Summary
URL : http://pastehtml.com/view/bfuyvvo5h.html
Result: legtimate
Recently Checked URLs
URL
Status
http://graphicriver.net/search?date=this-month
legtimate
bafkreiezeywvkuzmgn4iyhms72rfv6gsbcn57wrxuj6btv5ucd3o7stoui.ipfs.cf-ipfs.com
phishing
www.example.com
legtimate
bafkrwrxuj6btv5ucd3o7stoui.ipfs.cf-ipfs.com
phishing
bafkrwrxuj6btv5ucd3o7stoui.ipfs.cf-ipfs.com
phishing
bafkrwrxuj6btv5ucd3o7stoui.ipfs.cf-ipfs.com
phishing
www.youtube.com
legtimate
Type here to search
13°C Haze
10:45 PM
12/4/2023




# Literature Review

### Detection and Prevention of Phishing Websites using Machine Learning Approach

Gopi.K 1, Sri Lakshmi.S2, Ramya.P3, Pavan Kumar.R4, Ventkateswarlu swami.M5

## Introduction

Users of the internet are seriously at risk from phishing attacks. These attacks entail the creation of phony websites that imitate trustworthy websites, like those operated by banks or online merchants. Phishing attacks aim to deceive people into disclosing sensitive information, like credit card numbers or passwords.

It has been demonstrated that machine learning is a useful tool for identifying phishing websites. This research suggests a three-pronged method of using machine learning to identify phishing websites: visual appearance-based analysis, website legitimacy check, and URL-based analysis. After that, a machine learning model that combines these three strategies is presented in the paper. A dataset comprising authentic and phishing websites is used to train the model.

The model's performance is evaluated by the authors, who discover that it can identify phishing websites with a high degree of accuracy.

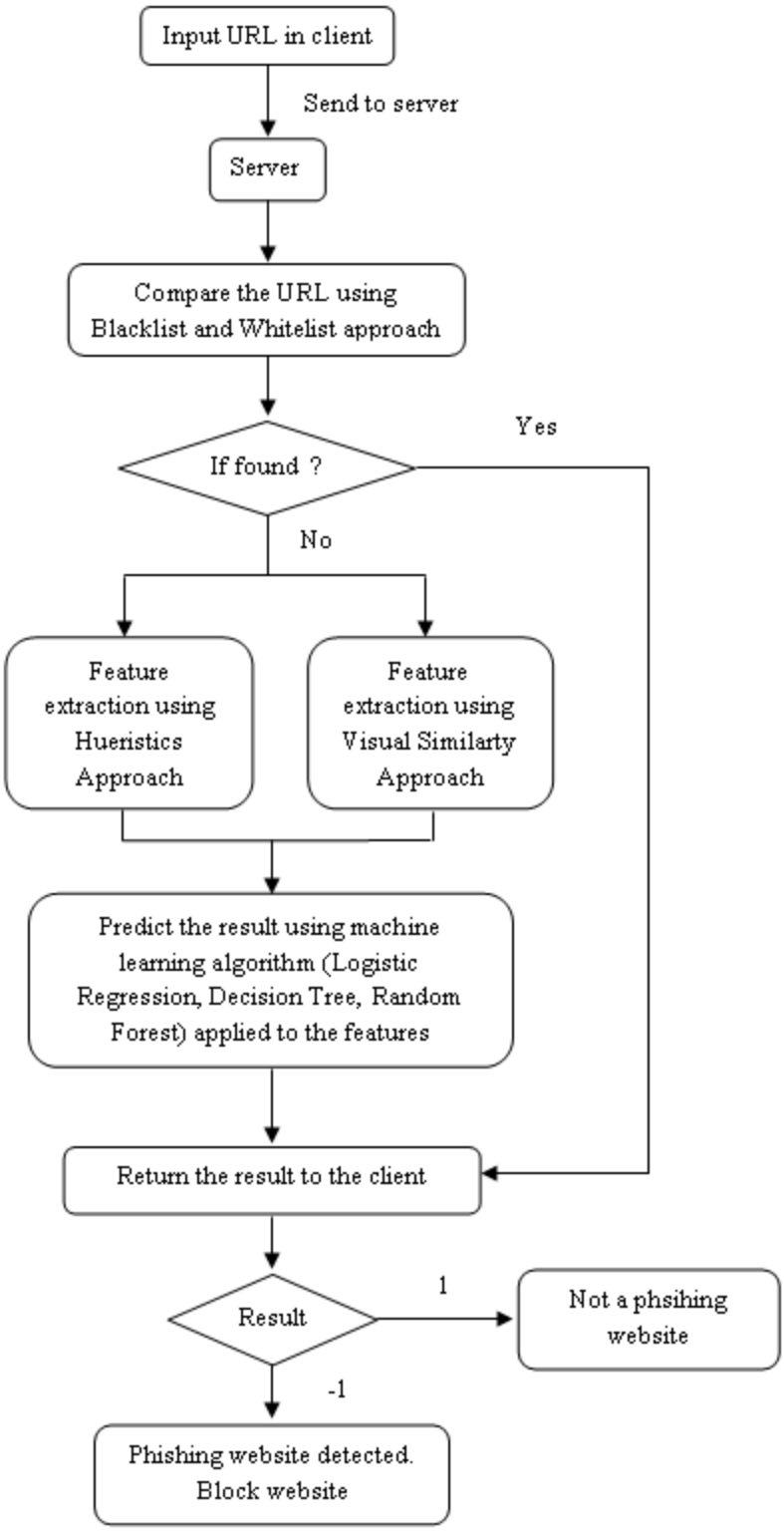
**Related Work**

Several research works have suggested machine learning-based methods for identifying phishing websites. While some of these studies concentrate on website content analysis, others concentrate on URL-based analysis. A few studies have also suggested hybrid strategies that incorporate website content analysis and URL-based analysis.

A hybrid approach combining URL-based analysis, website legitimacy check, and visual appearance-based analysis is proposed in the paper by Gopi et al. It has been demonstrated that this method is more thorough than earlier methods and is more successful at identifying phishing websites.

**Results & Findings**

The following figure shows the flow of the proposed system:


Input URL in client
Send to server
Server
Compare the URL using Blacklist and Whitelist approach
If found ?
Yes
No
Feature extraction using Hueristics Approach
Feature extraction using Visual Similarty Approach
Predict the result using machine learning algorithm (Logistic Regression, Decision Tree, Random Forest) applied to the features
Return the result to the client
Result
1
Not a phsihing website
-1
Phishing website detected. Block website

The following table shows comparison of different models on the basis of True Negative, True Positive, False Positive and False Negative and the accuracy achieved by each model:

TABLE I. CONFUSION MATRIX RESULTS

| Algorithm | TN | TP | FP | FN | Accuracy |
|---|---|---|---|---|---|
| Logistic Regression | 6447 | 2287 | 325 | 17 | 96.23 % |
| Decision Tree | 6393 | 2341 | 326 | 16 | 96.23 % |
| Random Forest | 6392 | 2374 | 297 | 13 | 96.58 % |

**Drawbacks**

Despite the fact that this system offers good accuracy for all above mentioned models, it has its shortcomings. A small amount of false positive and false negative results are picked up by the system. By adding far richer features to feed the machine learning algorithm, which would produce much higher accuracy, these drawbacks can be completely eliminated.

**Conclusion**

The paper makes a significant contribution to the field of phishing detection. The proposed approach is comprehensive and effective, and the machine learning model shows promise for real-world applications. The paper also provides a valuable overview of related work on machine learning-based phishing detection.

**Detection of Phishing Websites using Machine Learning Approach**

Kahksha, Sameena Naaz*

Department of Computer Science and Engineering, School of Engineering Sciences and Technology, New Delhi – 110062, India

## Introduction

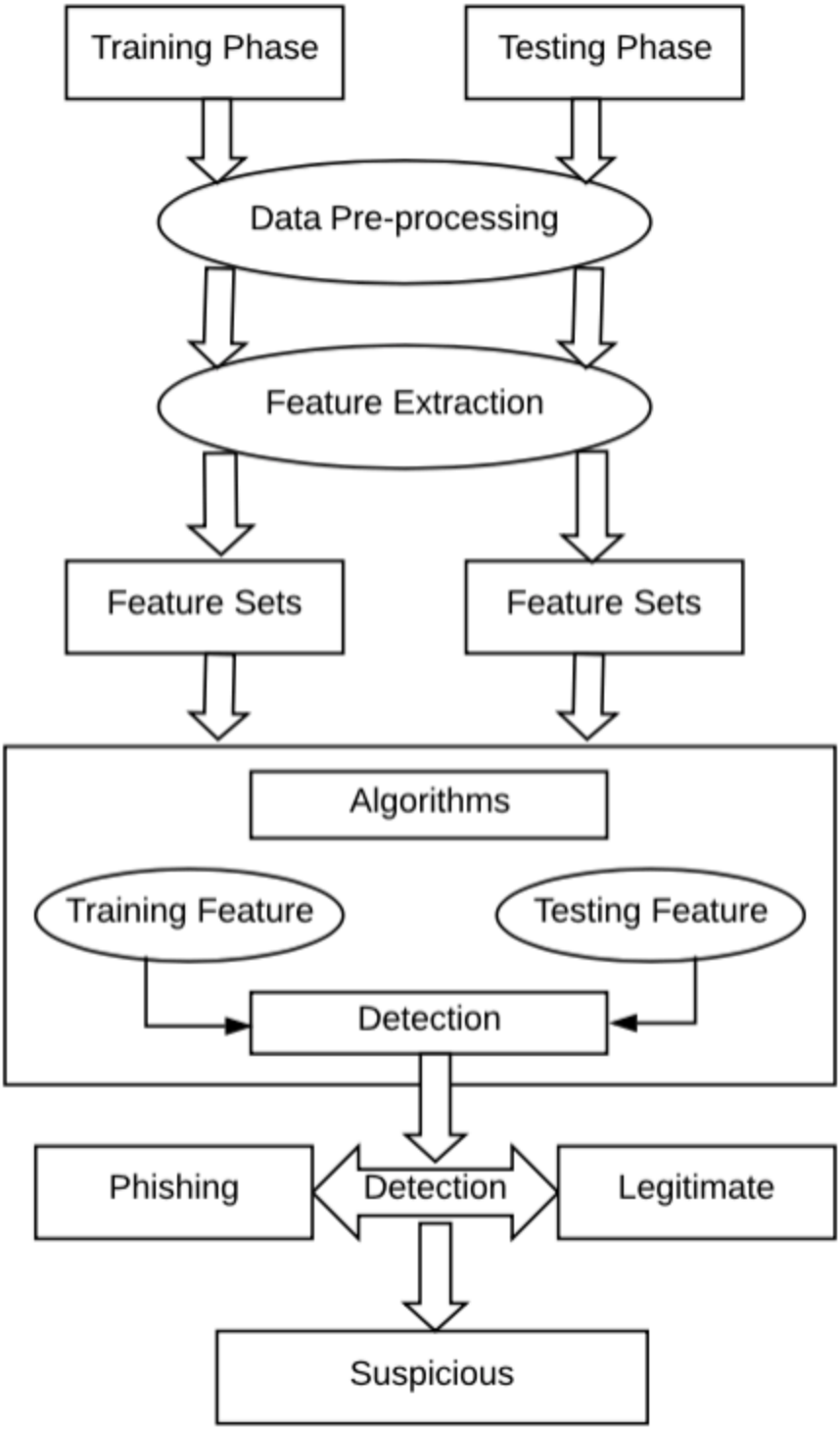
Phishing attacks are a serious risk to internet users because they try to fool them into disclosing personal information. One method that has shown promise for identifying phishing websites is machine learning. In order to detect phishing websites, this paper suggests a machine learning-based method that makes use of URL features, website content features, and visual similarity features.

## Related Work

Machine learning methods have been investigated in a number of studies for phishing detection. A few concentrate on URL-based analysis, looking at elements like URL length and questionable keywords. Others analyze the content of websites to find keywords, frames, and resemblances to well-known phishing websites. Hybrid methods combine content analysis with URL analysis to improve detection.

## Results & Findings

The following figure shows the flow of the proposed system:




The following table shows comparison of different models on the basis of accuracy and some other parameters achieved by each model:

| | DT | RF | NN | LM |
|---|---|---|---|---|
| **Accuracy** | 90.4% | 95.7% | 90.70% | 92.10% |
| **TPR** | 93.2% | 96.1% | 84.00% | 93.80% |
| **TNR** | 88.7% | 95.2% | 97.90% | 90.00% |
| **Precision** | 83.2% | 93.7% | 94.00% | 92.00% |
| **F-Measure** | 87% | 94% | 90.40% | 92.80% |
| **Error Rate** | 9.5% | 4.3% | 9.2% | 8.0% |

**Conclusion**

This paper stands out by incorporating visual similarity features alongside URL and content features. This comprehensive approach captures the diverse characteristics of phishing websites, leading to more accurate detection. Additionally, the proposed machine learning model demonstrates high accuracy in distinguishing phishing websites from legitimate ones.

**Future Directions**

As phishing tactics continue to evolve, further research is warranted to enhance the robustness and adaptability of machine learning-based phishing detection systems. Real-time detection mechanisms that can promptly identify new phishing websites are also crucial for effective protection.

### Machine Learning Technique Detects Phishing Sites based on Markup Visualization

**Introduction**

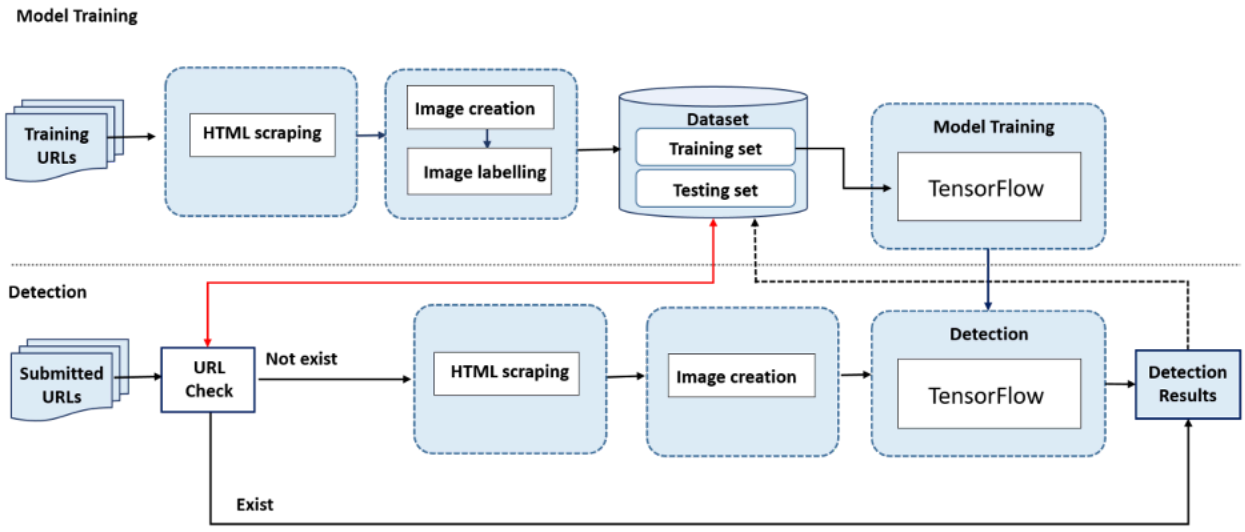
Phishing website detection accuracy and speed can be increased with the use of machine learning models that are trained on the visual representation of website code.

Researchers studying security at the Universities of Portsmouth and Plymouth in the United Kingdom have published a paper supporting this claim. The current detection techniques are either too slow or too inaccurate, and the researchers want to fix these issues. The researchers' method turns web page markup and code into images by using libraries for "binary visualization." They produced a dataset of authentic and phishing images of websites using this technique.

A machine learning model was then trained on the dataset to distinguish between phishing and genuine websites based on variations in their binary visualization. The trained model is applied to the target webpage's code after it has been transformed through binary visualization. The researchers employed MobileNet, a neural network optimized for resource-constrained devices rather than cloud servers, to accelerate the model's performance. In order to prevent drawing unwarranted and disproportionate conclusions, the system also progressively compiles a list of trustworthy and phishing websites.

**Results & Findings**

The following figure shows the flow of the proposed system:



Experiments conducted by the researchers revealed that the model's accuracy in identifying phishing websites reached 94%. Furthermore, because it makes use of a very small neural network, it can operate on user devices and deliver results almost instantly.

## Technology Review

Here is a broad overview of the most recent advancements in phishing detection and prevention technology:

1. AI and machine learning (ML):

Solutions: To detect phishing attempts, machine learning algorithms examine patterns in emails, websites, and user behavior.

Challenges: False positives and negatives continue to be a problem, and adversarial attacks have the potential to manipulate ML models.

2. Email Filtering and URL Analysis

Solutions: Heuristics, blacklists, and whitelists are used by sophisticated email filtering systems.

Challenges: Conventional filters may be circumvented by zero-day threats and polymorphic attacks.

3. Advances in Multi-Factor Authentication (MFA):

Solutions: By adding an additional degree of protection, MFA lessens the impact of compromised credentials.

Challenges: The reluctance of users to embrace multi-factor authentication and possible weaknesses in particular applications.

4. Web Browser Protection:

Solutions: Modern browsers come with built-in security tools, like alerts for websites that might be harmful.

Challenges: Users may disregard warnings and malicious websites can still avoid detection.

**Following are some of the general problems that makes it difficult to detect and prevent phishing attacks:**

- ➔ Spear phishing: Targeted attacks that modify phishing content to target particular people or companies.
- ➔ Zero-Day Attacks: These pose a threat to conventional signature-based systems by taking advantage of undiscovered vulnerabilities.
- ➔ Insider Threats: Malevolent actions carried out by employees of a company.

## Summary & Conclusions

Cybersecurity requires both phishing detection and prevention, and technology is constantly evolving to meet new problems. In this constantly changing environment, a comprehensive strategy incorporating technology, user education, and ongoing adaptation to new threats is crucial.

## Future Work

The field of phishing detection is always changing because attackers are coming up with new tricks. The main goal of future work should be to create machine learning models that can adapt to the changing landscape of phishing attacks. Furthermore, research in the future should concentrate on creating real-time systems for phishing website detection.

## References/Bibliography

Dickson, B. (2021, September). Machine learning technique detects phishing sites based on markup visualization. Retrieved from https://portswigger.net/daily-swig/machine-learning-technique-detects-phishing-sites-based-on-markup-visualization

Gopi.K 1, S. L. (2020, March). Detection and Prevention of Phishing Websites using Machine Learning. Retrieved from www.irjet.net

Kahksha Jalal, S. N. (2019). Detection of phishing website using machine learning approach.