**Detection Of Malicious Attack In Internet Using Machine Learning**

**Suchetha N V**

**Chapter 1**

**Introduction**

* 1. **Project Introduction**

The internet is an essential platform for sharing information, conducting business, and communicating in the linked world of today. However, this pervasive connectedness exposes consumers to a range of online dangers, such as phishing attempts. Phishing is a tactic used by attackers to get people to divulge private information like usernames, passwords, or financial information by impersonating trustworthy organisations.

On the internet, there are several malicious assaults that include phishing, the dissemination of malware, cross-site scripting (XSS), SQL injection, and others. In our project, we concentrated on developing a web extension that was intended exclusively for identifying phishing websites. However, it's critical to recognise that the threat landscape is always changing and that new attack methods are consistently being developed. While the initial focus of our study is phishing detection, we are aware of the potential to broaden its applicability to include the detection of other kinds of attacks. We can improve the browser extension to offer a more thorough defence against a wider spectrum of harmful actions by utilising the power of machine learning and regularly upgrading our model with new attack samples.

One of the most frequent and damaging types of cyberattacks in the modern digital environment is the phishing attack. They immediately go for consumers' trust and take advantage of their weaknesses to get access to private data. You can combat phishing assaults, a serious hazard that impacts people, companies, and organisations in a variety of industries.

While other attack types also present serious risks, focusing on phishing assaults enables you to go in-depth and obtain a thorough grasp of the attack methodology, detection techniques, and mitigation strategies. It offers a chance to make a significant difference in the battle against phishing and to create a specialised solution that tackles the peculiar difficulties presented by this specific kind of assault.We will create a browser plugin as a Chrome extension based on the chosen model. The plug-in will examine URLs found

while browsing, warning users of potential phishing dangers and empowering them to decide for themselves whether a website is legitimate.

The importance of our initiative rests in the creation of a cutting-edge, machine learning-driven strategy to counter phishing attempts. We seek to improve internet user security, safeguard against money loss, and lessen the dangers associated with falling prey to phishing scams by leveraging the power of machine learning.

Although the current iteration of our study concentrates on phishing detection, we are aware of a wider range of online dangers. We can keep improving our browser extension and ultimately help create a better and more secure online environment for all users by recognising the potential for expansion and embracing developments in machine learning and cybersecurity.

To build an effective phishing detection model, we employed a machine learning approach. We collected a comprehensive dataset of known phishing URLs and legitimate websites, which served as the foundation for training our model. Relevant features were extracted from website URLs to capture distinguishing characteristics between legitimate and phishing websites. These features encompassed factors such as domain length, presence of suspicious keywords, subdomain count and the use of HTTPS. We utilized a machine learning algorithm, Random Forest, to train the model on the labeled dataset. The algorithm learned the patterns and characteristics associated with phishing websites, enabling it to classify URLs accurately.

## Problem Description

Phishing attacks pose a significant threat to individuals and organizations, exploiting human vulnerability to trick users into revealing sensitive information. Traditional methods of detecting phishing websites, such as maintaining static blacklists or relying on manual inspection, are insufficient to keep pace with the ever-evolving tactics employed by attackers. Therefore, there is a pressing need for a proactive and automated solution to identify and alert users about potential phishing websites in real-time.

The existing browser-based security measures often fall short in effectively detecting sophisticated phishing attacks. Users may unknowingly visit malicious websites, putting their personal information, financial data, or login credentials at risk. This highlights the need for a more robust and accurate phishing detection mechanism that can seamlessly integrate with popular web browsers, such as Chrome, to provide users with a safer online experience.

The project focuses on creating a machine learning-powered Chrome plugin for phishing detection in order to solve this issue. By utilizing sophisticated algorithms and feature extraction techniques, the addon intends to analyze website URLs in real-time and assess their propensity to be connected to phishing attempts. We aim to improve the capability to recognize and alert consumers about potential phishing threats by utilizing machine learning models built on a diverse dataset of known phishing websites and legal URLs.

The ultimate objective is to offer customers a trusted and approachable tool that can successfully differentiate between authentic websites and phishing attempts. Users will gain from real-time notifications by incorporating this Chrome extension into their surfing experience, lowering their risk of falling for phishing scams, and strengthening their online security.

.

# Literature Review

## Chapter 2

* 1. **Literature Survey**

The authors Vipin Das, Vijaya Pathak, Sattvik Sharma “Network Intrusion Detection System Based Machine Learning Algorithms”, IEEE-2019 [1], they proposed a process that involves capturing network packets, preprocessing the data using Rough Set Theory (RST) to reduce dimensions and using Support Vector Machine (SVM) for learning and testing. This approach effectively reduces data density and improves accuracy. The author's research addresses the challenge of network intrusion detection by utilizing RST to identify relevant features and reducing computational complexity. SVM, employed as the classification model, effectively distinguishes between normal network behavior and intrusions, resulting in improved detection performance. Comparative analysis with Principal Component Analysis demonstrates the superiority of the RST and SVM approach. It minimizes false positives, optimizes resource allocation and enhances overall accuracy, making it a robust solution for network intrusion detection.This study paves the way for further research, encouraging exploration of alternative machine learning algorithms and techniques for network security applications.

The authors Biswanath Mukherjee, L. Todd Heberlein and Karl N. Levitt “Network Intrusion Detection”, IEEE 2018 [2], they proposed Intrusion Detection as a novel approach to enhance security in existing computer systems and data networks. Their research emphasizes the significance of Intrusion Detection Systems (IDS), which are designed to detect potential attacks through host-audit-trail analysis and network traffic analysis. The primary objective of these systems is to identify and respond to attacks promptly, even in real-time scenarios. The authors successfully developed several prototypes of Intrusion Detection Systems, demonstrating the practical implementation of their concept. The promising outcomes obtained from these prototypes highlight the potential effectiveness of IDS in ensuring network security. Furthermore, the research paper provides an insightful categorization of different types of Intrusion Detection Systems, along with a discussion of their respective advantages and disadvantages. This comprehensive overview allows readers to gain a better understanding of the various approaches and their applicability in diverse network environments.

The authors Abdul Razaque , Mohamed Ben Haj Frej, Dauren Sabyrov “Detection of Phishing Websites using Machine Learning”, IEEE-2019 [3], they address the significant issue of phishing attacks. Phishing exploits email communication to distribute malicious links or attachments, posing risks such as financial loss and identity theft. The study focuses on developing a Google Chrome web browser extension for effective phishing prevention. Implemented using JavaScript, the extension combines Blacklisting and semantic analysis techniques. It analyzes website content, including text, links, images and other data. By comparing website URLs against a blacklist database, the extension blocks access to known phishing sites, adding an extra layer of security. Furthermore, semantic analysis examines website text for suspicious patterns and deceptive language, providing real-time protection. Through rigorous testing and comparison with existing approaches, our solution demonstrates high accuracy in detecting and blocking phishing attempts, significantly mitigating the problem. This innovative extension offers robust protection and safeguards users while browsing the internet.

The authors Anish Halimaa A and Dr. K. Sundarakantham "Machine Learning Based Intrusion Detection System", IEEE-2020 [4], they emphasize the importance of intrusion detection and intrusion prevention in the current technological landscape. With our increasing reliance on networks and information systems for everyday activities, the need for effective intrusion detection and prevention mechanisms becomes crucial. The paper highlights that numerous approaches have been developed and applied in intrusion detection systems. Among these approaches, machine learning techniques play a significant role. Machine learning algorithms provide the ability to analyze and learn from large amounts of data, enabling the detection of anomalous activities and potential intrusions. The authors specifically focus on two machine learning algorithms, Support Vector Machine (SVM) and Naïve Bayes, in their analysis. These algorithms are applied to a dataset consisting of 19,000 instances, allowing for a thorough evaluation of their performance in intrusion detection. The research findings indicate that SVM outperforms Naïve Bayes in terms of intrusion detection accuracy. This highlights the effectiveness of SVM as a machine learning algorithm for identifying and classifying intrusions in the given dataset.

The authors Amani Alswailem, Dr.Aram Alsedrani, Norah Alrumayh “Detecting Phishing Websites Using Machine Learning”, IEEE-2019 [5], they tackled the important problem of phishing websites, which steal sensitive data like usernames and passwords by preying on human weaknesses rather than software flaws. We presented an intelligent algorithm that can successfully identify phishing websites to address this issue. Our technology works as an add-on for web browsers, adding an extra degree of security by instantly alerting users if a potential website is found. We adopted the Random Forest technique using machine learning, specifically supervised learning, to take advantage of its superior performance in classification problems. Our research's major goal was to create a high-performance classifier for locating phishing websites. To do this, we carefully looked at the characteristics that are frequently found on phishing websites. They trained the classifier to obtain the best accuracy by carefully choosing and combining the most useful features. Our technology proved exceptional performance after a thorough examination, obtaining a remarkable accuracy rate of 98.8%..

The authors Junaid Rashid, Toqeer Mahmood, Muhammad Wasif Nisar “Phishing Detection Using Machine Learning Technique”, IEEE-2020 [6], addressed the pervasive threat posed by phishing attempts in today's internet-dependent culture by proposing an effective machine learning-based technique for phishing detection. The suggested technique provides outstanding performance by combining the Support Vector Machine (SVM) classifier with novel functionality. With an astonishing 95.66% accuracy rate, experimental results show that the suggested technique may successfully discern between authentic and phishing websites. Furthermore, the method merely makes use of 22.5% of the novel capabilities, underscoring its usefulness and efficiency. Benchmarking the suggested method's performance against industry-standard phishing datasets from the "University of California Irvine (UCI)" repository regularly confirms its superior performance. The technique establishes itself as a popular solution for phishing detection based on machine learning due to its high accuracy and favourable comparison results. In conclusion, this study offers an important contribution to the machine learning-based fight against phishing. The proposed method's effectiveness in correctly identifying phishing websites is demonstrated when it is combined with the SVM classifier. The method has a lot of potential to safeguard users against the negative effects of phishing assaults in a variety of real-world applications by boosting internet security.

## Comparative Analysis of the Related Work

The table 2.1 discusses the comparative analysis of the current systems in light of the suggested proposal.

**Table 2.1 Comparative Analysis Of The Related Work**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Author(s)** | **Algorithms/Techniques** | **Performance Measures** |
| **1.** | Vipin Das, Vijaya Pathak, Sattvik Sharma | Rough Set Theory (RST) and Support Vector Machine (SVM) | Accuracy Compatibility |
| **2.** | Biswanath Mukherjee, L. Todd Heberlein and Karl  N. Levitt | Based on host-audit-trail and network traffic analysis | Accuracy |
| **3.** | Abdul Razaque , Mohamed Ben Haj Frej, Dauren Sabyrov | Rigorous testing and comparison with existing approaches | Accuracy |
| **4.** | Anish Halimaa A and Dr.  K. Sundarakantham | Support Vector Machine (SVM) and Naïve Bayes | Accuracy |
| **5.** | Amani Alswailem, Dr.Aram Alsedrani, Norah Alrumayh | Random Forest technique | Accuracy |
| **6.** | Junaid Rashid, Toqeer Mahmood and Muhammad Wasif Nisar | SVM classifier | Accuracy |

## Summary

The research papers address the significant issue of phishing attacks and propose machine learning-based techniques for their detection and prevention. Phishing attacks exploit email communication to distribute malicious links or attachments, posing risks such as financial loss and identity theft. The researchers present innovative solutions to enhance internet security and protect users. The first paper introduces a Google Chrome browser extension that combines blacklisting and semantic analysis techniques. It analyzes website content and compares URLs against a blacklist database, effectively blocking access to known phishing sites. The second paper presents an intelligent system implemented as a browser extension, utilizing supervised learning and the Random Forest technique to detect phishing websites. Thorough examination of features associated with phishing enables the system to achieve a high accuracy rate. The third paper proposes an efficient technique integrating the Support Vector Machine (SVM) classifier, accurately distinguishing phishing websites from legitimate ones. The technique demonstrates remarkable performance and efficiency. Collectively, these research papers contribute valuable insights and methodologies for combating phishing attacks through machine learning. They offer effective means of detecting and preventing phishing attempts, thereby enhancing internet security and safeguarding users from the detrimental consequences of phishing attacks.

# Problem Formulation

## Chapter 3

* 1. **Problem Statement**

This project addresses the issue of the need for a real-time, efficient phishing detection method to improve online security. The goal is to create a machine learning-based Chrome plugin that quickly and accurately detects suspected phishing websites by analysing website URLs. The extension aims to give users prompt warnings or alerts when they come across dubious or misleading websites, reducing the risk of falling for phishing scams and protecting their personal and sensitive information. It does this by utilising machine learning algorithms and feature extraction techniques.

## Objectives of the Present Study

The objectives of the proposed project are as follows:

* + 1. Enhance user security by detecting and preventing phishing attacks in real-time.
    2. Educate users about phishing techniques and indicators to promote awareness and prevention.
    3. Improve user confidence in online transactions by safeguarding sensitive information.
    4. Provide a seamless and user-friendly browsing experience while protecting against phishing threats.
    5. Continuously adapt to evolving phishing techniques to maintain high detection accuracy.

## Summary

The goal of the project is to provide a clever method of proactively identifying and warning consumers about dangerous phishing websites. The plugin gives users prompt cautions or notifications when they come across suspicious websites by using machine learning algorithms and real-time monitoring of website URLs. The project improves online security, gives consumers the information they need to make wise choices and helps in the ongoing battle against phishing scams.

# Requirements and Methodology

## Chapter 4

* 1. **Hardware Requirements**

The hardware requirements for the proposed project are depicted in Table 4.1.

**Table 4.1: Hardware Requirements**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Hardware/Equipment** | **Specification** |
| 1. | Processor | Intel Core i5 or above |
| 2. | RAM | 4GB or above |

## Software Requirements

The software requirements for the proposed project are depicted in Table 4.2.

**Table 4.2: Software requirements**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Software** | **Specification** |
| 1. | Browser | Chrome |
| 2. | Python | Python 3 and above |
| 3. | JavaScript | ES2015 |

## Methodology Used

The proposed project is implemented using the following steps: Step 1 : Problem Definition

* + - Clearly define the problem of detecting phishing attacks in real-time within the Chrome browser.

Step 2 : Dataset Collection

* + - Gather a reliable and diverse dataset of labeled phishing and legitimate websites for training the machine learning model.

Step 3 : Dataset Preprocessing

* + - Clean and preprocess the dataset by removing duplicates, handling missing values and normalizing the data.

Step 4 : Feature Extraction

* + - Extract relevant features from URLs or web pages that can help differentiate between phishing and legitimate websites.

Step 5 : Model Selection

* + - Choose a suitable machine learning algorithm, such as logistic regression or random forest, for phishing detection.

Step 6 : Model Training

* + - Train the selected model using the preprocessed dataset, with the extracted features as input and the phishing/legitimate labels as the target.

Step 7 : Model Evaluation

* + - Evaluate the performance of the trained model using appropriate metrics like accuracy, precision, recall and F1-score.

Step 8 : Chrome Extension Development

* + - Create a Chrome extension using JavaScript and the Chrome Extension API to interact with web pages and detect phishing attacks.

Step 9 : Integration with Machine Learning Model

* + - Integrate the trained machine learning model into the Chrome extension to make predictions based on the extracted features.

Step 10 : Real-time Phishing Detection

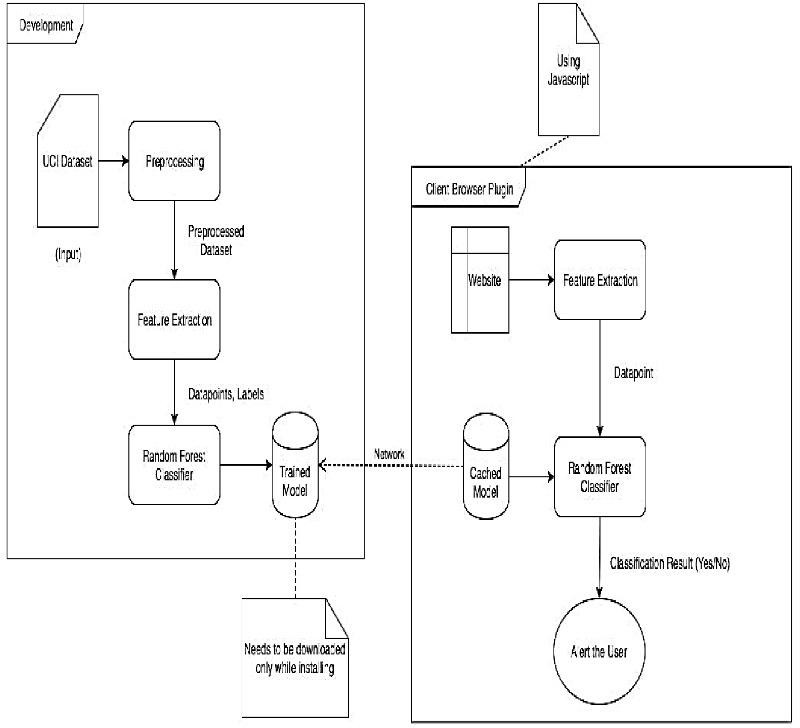
* + - Implement mechanisms in the extension to analyze URLs or web page content and communicate with the machine learning model for real-time phishing detection.

**Chapter 5**

# System Design

## Architecture of the Proposed System

Figure 5.1 shows the architecture of the proposed system.



**Figure 5.1: Architecture Of The Proposed System**

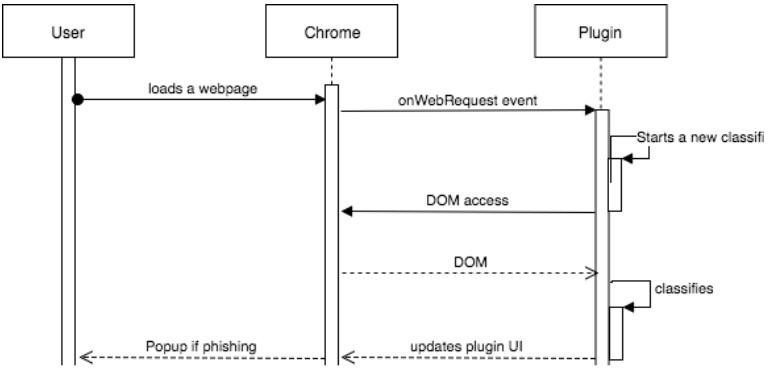
At the core of the architecture is the Chrome extension interface, which serves as the user-facing component. It interacts with the user, displaying alerts, notifications and visual indicators to convey the phishing detection results. When a user visits a web page, the URL analysis component captures the URL and sends it for analysis. The URL analysis module examines the URL for suspicious patterns and indicators commonly associated with phishing attacks. The feature extraction component plays a crucial role in extracting relevant features from the URL or web page content. These features can include domain characteristics, subdomain information, URL structure, SSL certificate

details, redirects, or other attributes that aid in distinguishing legitimate websites from phishing websites.

The extracted features are then fed into a trained machine learning model specifically designed for phishing detection. The model analyzes the features and applies learned patterns to classify the URL as either phishing or legitimate. Based on the output of the machine learning model, the phishing detection decision is made. If the URL is classified as a potential phishing attack, the Chrome extension interface generates an appropriate notification or alert to inform the user about the potential threat. To keep up with evolving phishing techniques, the system architecture includes mechanisms for model updating and maintenance. This can involve periodic model retraining using new datasets, incorporating user feedback, or leveraging external threat intelligence sources to enhance the model's accuracy and effectiveness.

## System Flowchart

A system flowchart is a way of depicting how data flows in a system and how decisions are made to control events. Figure 5.2 depicts the system flowchart.



**Figure 5.2: System Flowchart**

# Implementation

## Chapter 6

**6.1 Pseudocode**

// Function to perform phishing detection function detectPhishing(url) {

// Extract relevant features from the URL or web page content features = extractFeatures(url);

// Load the trained machine learning model model = loadModel();

// Make a prediction using the model prediction = model.predict(features);

// Return the prediction (1 for phishing, 0 for legitimate) return prediction;

}

// Function to extract features from the URL function extractFeatures(url) {

// Perform feature extraction by analyzing the URL or web page content features = analyzeURL(url);

// Return the extracted features return features;

}

// Function to load the trained machine learning model function loadModel() {

// Load the pre-trained machine learning model model = loadModelFromFile();

// Return the loaded model return model;

}

// Function to analyze the URL and extract features function analyzeURL(url) {

// Perform analysis on the URL and extract relevant features features = analyzeURLFeatures(url);

// Return the extracted features return features;

}

// Main function function main() {

// Get the current URL from the browser currentURL = getCurrentURL();

// Perform phishing detection on the current URL result = detectPhishing(currentURL);

// Display the phishing detection result to the user showResult(result);

}

**Code of Random Forest algorithm** :

let decision\_tree = function(root) { var predictOne = function(x) { var node = root; while(node['type'] == 'split') {

var threshold = node['threshold'].split(' <= '); if(x[threshold[0]] <= threshold[1]) { //Left node = node['left'];

} else { //Right

node = node['right'];

}

}

return node['value'][0];

}

var predict = function(X) { var pred = [];

for(let x in X) { pred.push(this.predictOne(X[x]));

}

return pred;

}

return {

'predict': predict, 'predictOne': predictOne

}

}

let random\_forest = function(clf) { var predict = function(X) {

var pred = [];

for(let e in clf['estimators']) {

let tree = decision\_tree(clf['estimators'][e]); pred.push(tree.predict(X));

}

pred = pred[0].map((col, i) => pred.map(row => row[i])); var results = [];

for(var p in pred) {

let positive=0, negative=0; for(let i in pred[p]) { positive += pred[p][i][1]; negative += pred[p][i][0];

}

results.push([positive>=negative, Math.max(positive, negative)]);

}

return results;

}

return {

'predict': predict

}

}

**Chapter 7**

# System Testing, Results and Discussion

## System Testing

**Table 7.1: Unit Test Cases**

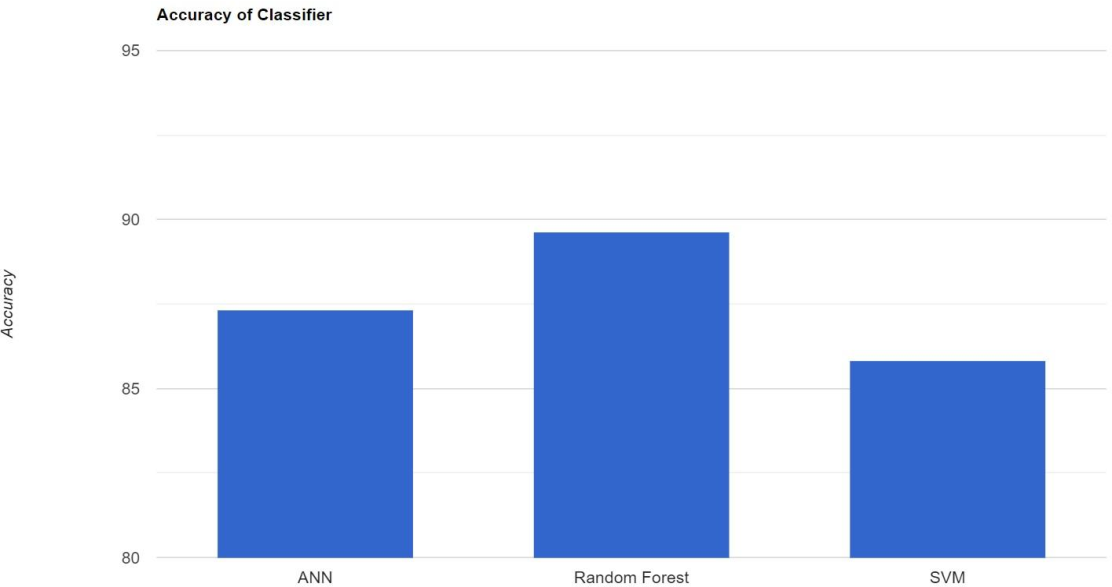
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case number** | **Input** | **Stage** | **Expected behavior** | **Observed behavior** | **Status P=Pass F=Fail** |
| 1 | A phishing website trying to steal user credentials | Testing | Shows popup saying website is malicious | As expected | P |
| 2 | A legitimate website URL | Testing | Shows popup saying website is not malicious | As expected | P |
| 3 | A URL that redirects to a phishing website | Testing | Shows popup saying website is malicious | As expected | P |

## Result Analysis

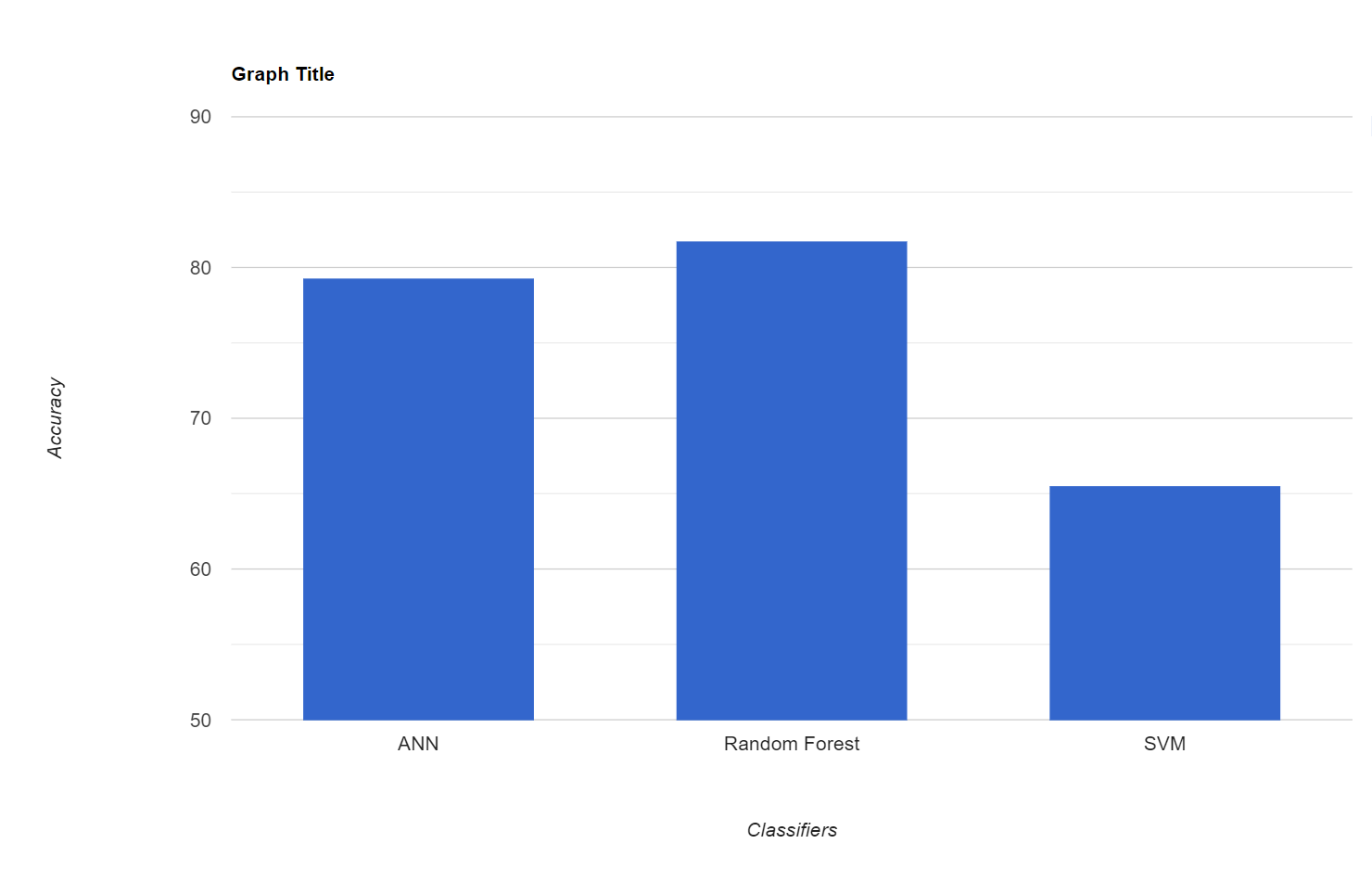
**Table 7.2: Analysis Of The 3 Algorithms**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Training Size** | **Testing Size** | **Accuracy (%)** | | |
| **Artificial Neural Network** | **Random Forest** | **SVM** |
| 1 | 80% | 20% | 87.34 | 89.63 | 85.84 |
| 2 | 70% | 30% | 79.27 | 81.77 | 65.54 |

Figure 7.1 shows the bar graph for the accuracy of the three algorithms where the train set size was 80% and the test set size was 20%.

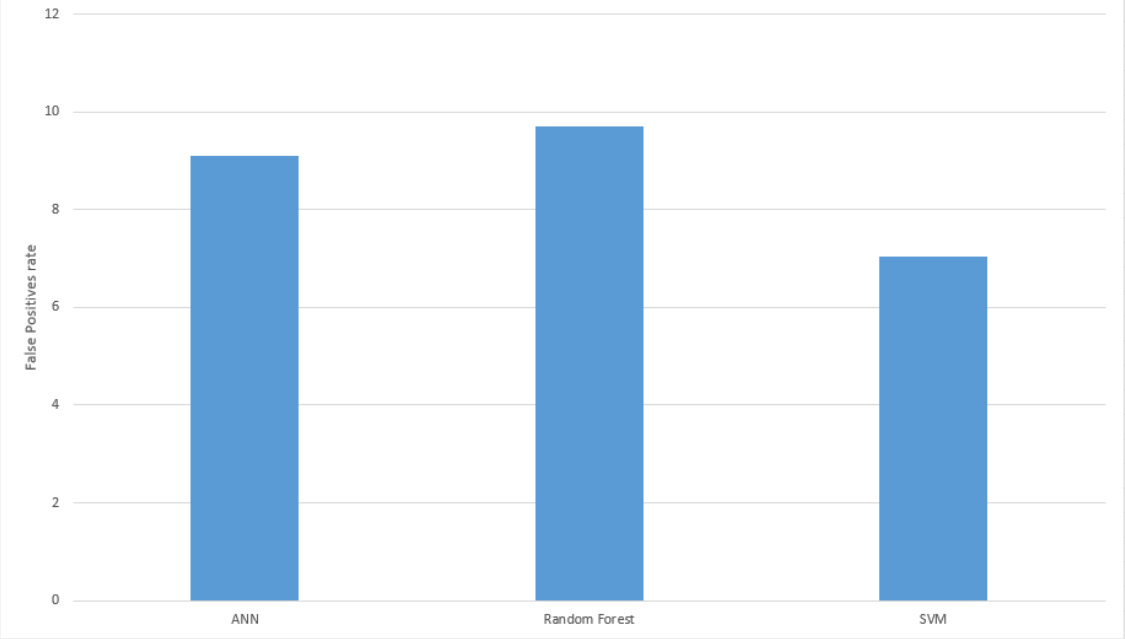


**Figure 7.1: Graph Analysis Of The Test Case 1**

Figure 7.2 shows the bar graph for the accuracy of the three algorithms where the train set size was 70% and the test set size was 30%**.**

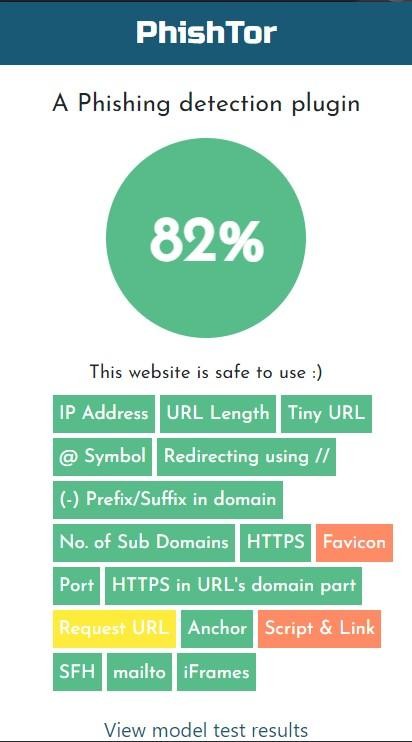
**Figure 7.2: Graph Analysis Of The Test Case 2**

Figure 7.3 shows the bar graph for the false positive rate of classifiers



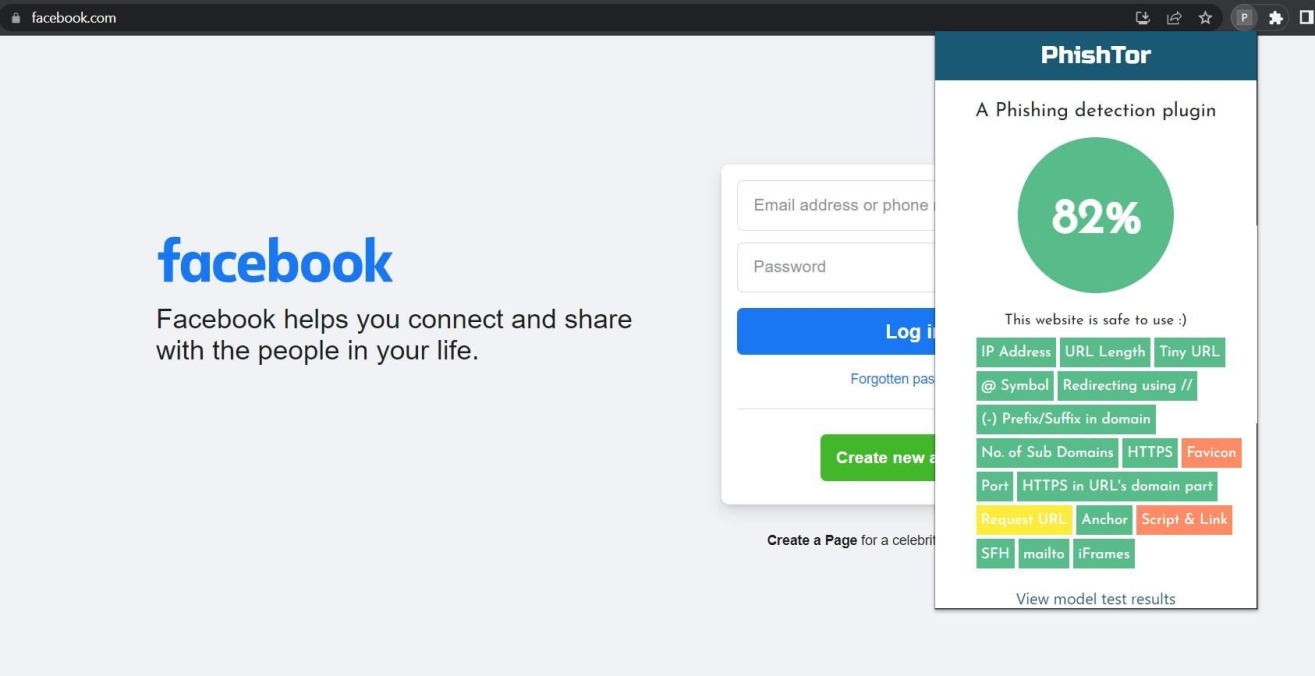
**Figure 7.3: False Positive Rate Of Classifiers**

Figure 7.4 is the UI Design of the proposed project



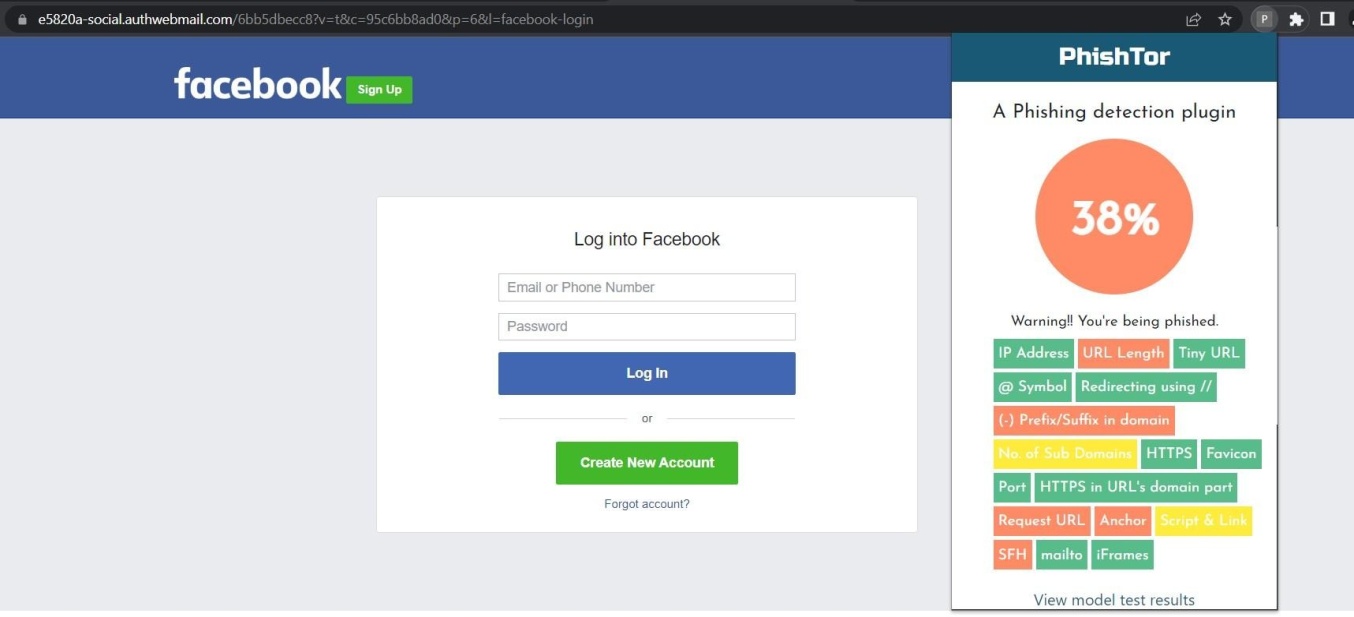
**Figure 7.4: UI Design**

Figure 7.5 is the output of the proposed project. Green circle indicates legitimate site and light red indicates phishing.



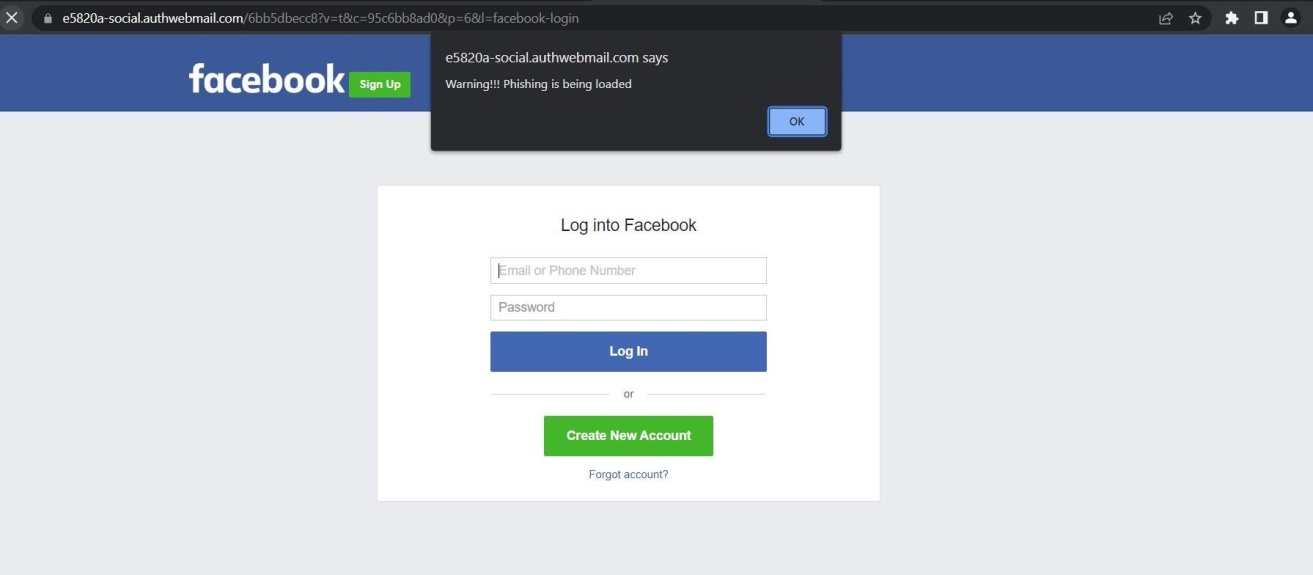
**Figure 7.5: Output For Legitimate Site**

Figure 7.6 is output for phishing websites



**Figure 7.6: Output For Phishing Websites**

Figure 7.7 is pop-up message for phishing websites



**Figure 7.7: Pop-Up Message**

## Summary

The execution of the proposed project includes critical steps for real-time phishing detection within the Chrome browser. A wide range of sites, both phishing and legitimate, are gathered and carefully preprocessed to guarantee high-quality data. Relevant characteristics, such as domain attributes, URL structure, and SSL certificate information, are collected from URLs or web content and are used to distinguish between phishing and legitimate websites. The model is trained using the preprocessed dataset using an appropriate machine learning approach, such as logistic regression or random forest. Accuracy, precision, recall, and F1-score are performance indicators that evaluate the model and direct improvement. This Chrome extension provides real-time phishing detection by interacting with users through pop-ups, notifications, or visual signals. It is created using JavaScript and the Chrome Extension API. The plugin incorporates the trained model, which makes predictions about potential dangers based on the characteristics and URL of the webpage. The add-on notifies the user if phishing is found. Accuracy is improved via methods for updating models, adding fresh data, and utilizing user feedback. User experience, flexibility, privacy, and cooperation with security communities are given top priority during deployment. Success is ensured by user-friendly interfaces, secure code, and knowledge exchange. By taking these actions, the team hopes to improve Chrome browser security by providing a dependable, real-time phishing defense mechanism.

**Chapter 8**

# Conclusion and Scope for Future Work

## Conclusion

In conclusion, the study successfully used feature extraction and machine learning methods to address real-time phishing detection within the Chrome browser. The plugin improves internet security by protecting users against phishing assaults. Key features gathered from URLs and web content were used to train the algorithm on a varied dataset of labeled legitimate and phishing websites. Performance assessment made sure that phishing and legal websites could be distinguished with accuracy. The Chrome extension integrates seamlessly, foreseeing hazards based on URLs visited and sending alarms, empowering users to actively protect their data. For complete security, future prospects include extending to detect malware, ransomware, and social engineering assaults. Advanced algorithms, behavioral analysis, and user input are all used to improve the machine-learning model. Security is improved through educating users about phishing, and the extension's defenses against complex phishing are strengthened by collaboration with security communities and integration with current solutions. By addressing a variety of threats, strengthening detection capabilities, and equipping users with knowledge about online safety, the project lays a solid foundation for Chrome's real-time phishing detection, positioning it to provide an even more secure surfing experience.

## Scope for Future Work

The project's future work will focus on improving feature extraction methods for better phishing attack detection from URLs or site content. Studying modern methods such as image analysis and natural language processing is part of this. Deep neural networks and other sophisticated machine-learning techniques can also increase accuracy. For evaluating dynamic web pages, real-time analysis can be developed, and user comments can help to improve the model. Coverage will be improved by including mobile platforms and working with cybersecurity professionals. A thorough defence against emerging phishing attempts will also be reliant on the integration of privacy controls and user education.