VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering **(SCOPE)**

BCSE309L: Cryptography and Network Security

WebShield Pro: Chrome Extension Harnessing Machine Learning for Dynamic Phishing Detection

F2+TF2 Slot, Winter Semester 2023-2024

by

Atharva Marathe - 21BCE1558

Akshay Thakur- 21BCE1512

Anirudh Soma - 21BCE5537

Ayush Tripathi- 21BCE1148

Submitted to

Dr. Sobitha Ahila S

School of Computer Science and Engineering (SCOPE)

Vellore Institute of Technology,

Chennai Campus-600127

Abstract: This project is dedicated to harnessing the power of machine learning to create a robust system for detecting phishing and harmful online links. Phishing, a deceptive tactic that masquerades as legitimate websites, presents a significant cybersecurity threat by duping users into revealing sensitive information. To combat this menace, traditional methods such as blacklists, heuristic analysis, and certain hashing techniques are commonly employed. However, to enhance detection accuracy and adaptability, we are integrating machine learning into our solution. Our approach involves exploring various classification algorithms, including Logistic Regression, Support Vector Machines (SVM), Decision Trees, and Neural Networks. These algorithms will be trained and evaluated using a dataset of phishing URLs sourced from the UCI Machine Learning repository. By leveraging machine learning, our aim is to develop a versatile and accurate detection system capable of effectively identifying phishing attempts and malicious links in real-time. The ultimate objective is to integrate the developed solution into a browser plugin, potentially as a Chrome extension. This plugin will provide users with enhanced protection against phishing attacks while seamlessly integrating into their browsing experience. Through this endeavour, we seek to contribute to the ongoing efforts in cybersecurity by empowering users with proactive measures to safeguard their online activities against evolving threats.

Keywords: Phishing Detection(PD), Chrome Extension(CE), Random Forest(RF), Support Vector Machine(SVM), Neural Networks(NN)

Introduction: Phishing, a prevalent cyber threat, continues to jeopardize the security and privacy of online users across various domains, including financial services, e-commerce, and social media. It involves malicious actors using deceptive tactics to trick individuals into divulging sensitive information such as usernames, passwords, and financial details. Despite efforts to combat phishing through conventional methods, such as blacklisting known phishing URLs and educating users about common phishing techniques, the dynamic and evolving nature of these attacks poses significant challenges to effective detection and prevention. In response to the escalating threat landscape, our project aims to develop a robust and adaptive phishing detection system leveraging a combination of advanced techniques. We recognize that traditional approaches, while valuable, may not suffice in addressing the intricate tactics employed by phishers. Hence, our project focuses on integrating four key methods: cryptography, heuristic rules, URL analysis, and similarity-based approaches, to enhance phishing detection capabilities. Cryptography plays a crucial role in verifying the authenticity and integrity of web content, thereby safeguarding against phishing attempts that involve tampering with communication channels or manipulating digital certificates. By employing cryptographic techniques such as secure hashing algorithms, our system can verify the legitimacy of websites and detect any unauthorized alterations to web content. Heuristic rules provide an effective means of identifying phishing attempts based on common characteristics and behaviors associated with malicious websites. By defining rules that flag suspicious activities, such as unusual URL structures, unexpected redirects, or deceptive content, our system can proactively alert users to potential phishing threats. URL analysis involves scrutinizing the structure and components of URLs to identify patterns indicative of phishing attempts. By analyzing elements such as domain names, subdomains, and query parameters, our system can assess the credibility of URLs and distinguish between legitimate and malicious websites. Similarity-based approaches leverage machine learning algorithms to compare web content and detect similarities with known phishing pages. By training models on a dataset of labelled examples, our system can learn to recognize common patterns and characteristics of phishing websites, enabling it to identify previously unseen phishing attempts with high accuracy. By integrating sophisticated algorithms and methodologies into a comprehensive software application, we aim to empower end-users with robust protection against phishing attempts in real-time. Here, we present a novel approach to phishing detection that combines the strengths of cryptography, heuristic rules, URL analysis, and similarity-based techniques. Building upon existing research and methodologies, our approach prioritizes adaptability and effectiveness in identifying and mitigating phishing threats. Through the integration of machine learning algorithms, we endeavour to augment the detection capabilities of our system, enabling it to extract new patterns and adapt to evolving attack vectors.

Literature Review: Phishing attacks represent a pervasive and persistent cybersecurity threat, exploiting human vulnerabilities to deceive individuals and organizations into divulging sensitive information. Fan (2021)[1] conducted a study focused on detecting and classifying phishing websites using machine learning techniques. Leveraging original data from PhishTank, the research employed Support Vector Machines (SVM) and Bayesian methods to accurately classify phishing websites. By exploring the impact of different feature types and quantities on classification model performance, Fan highlighted the effectiveness of ensemble learning and feature selection in improving detection accuracy. The study also introduced a depth detection system for phishing web pages, incorporating characteristics such as blacklists, visual similarity, and text classification. Similarly, Ripa et.al (2021)[2] further explored the landscape of phishing attacks, highlighting the critical role of machine learning models in detection and prevention. Their study encompassed the detection of phishing URLs, emails, and websites, employing classifiers such as XGBoost, Naive Bayes, and Random Forest. The research yielded high accuracy rates in identifying phishing vectors, underscoring the effectiveness of machine learning in combating phishing threats across diverse attack surfaces. In addition to traditional features, lexical and host-based features were integrated into the detection process. Lexical features focused on the structure and composition of URLs, capturing characteristics such as URL length, presence of security-sensitive words, and directory attributes. Host-based features delved into domain-related attributes, including domain length, token counts, and the presence of suspicious top-level domains. By incorporating both lexical and host-based features, the detection system could analyze a comprehensive range of attributes to identify potential phishing URLs. Furthermore, they emphasized the importance of leveraging online URL reputation services for categorizing URLs based on their reputation and trustworthiness. These services utilize vast databases of known malicious URLs, along with real-time analysis techniques, to assess the reputation of URLs and determine their likelihood of being associated with phishing activities. By integrating information from these reputation services into the detection process, the system could augment its capabilities and enhance its accuracy in identifying phishing threats. In a study conducted by Kumar et al. (2020)[3], they emphasized the importance of understanding the lexical structure of URLs, highlighting how attackers manipulate URLs for phishing purposes. The study delved into data pre-processing techniques for feature selection and extraction, categorizing features into URL lexical structure-based, domain name-related, and page-based features. URL-based features extracted by them encompassed various attributes such as URL length, presence of security-sensitive words, and characteristics of the URL path and file. Domain-based features included domain length, token counts, and the presence of suspicious top-level domains, while page-related features comprised aspects such as the age of the domain and zip code of the domain holder's address. By leveraging these features and employing machine learning algorithms, the study demonstrated the efficacy of data-driven approaches in detecting phishing websites. The comprehensive feature set facilitated nuanced analysis, enabling the detection system to discern subtle indicators of phishing activities. Zabihimayvan et.al (2019) [4] utilized Fuzzy Rough Set (FRS) theory to select effective features for phishing website detection. Their approach, applied to three classification methods, improved detection accuracy. By comparing against other methods, they found FRS outperformed in feature selection. The study identified universal features contributing to robust detection, even without domain-based features, achieving a high F-measure of approximately 93%. Sharma et.al (2023) [5] proposed a real-time phishing attack detection method utilizing advanced machine learning techniques. Initially, a dataset comprising labelled phishing and non-phishing instances is collected and pre-processed. Feature engineering is then applied to extract relevant characteristics from the dataset, incorporating URL analysis, content analysis, and behavior analysis. Subsequently, three machine learning algorithms—Random Forest, Support Vector Machines (SVM), and a Deep Neural Network (DNN)—are selected for training on the extracted features. The resulting models are combined using an ensemble method to improve performance. For real-time detection, the system continuously monitors incoming network traffic, extracts real-time features, and employs the ensemble model to predict phishing attempts. Based on the prediction, appropriate alerts are generated, ensuring swift and accurate detection of phishing attacks in real-time.

Proposed Solution: We propose that machine learning be used to overcome the limitations of traditional phishing detection methods. Because large volumes of data on phishing attack patterns are readily available, the problem of phishing detection is a perfect target for machine learning solutions. The main idea is to utilise machine learning algorithms on a dataset of phishing pages to create a model that can be used to determine if a given web page is a phishing page or a legitimate webpage in real time. We plan to turn the learned model into a software application that can be simply deployed to end users to counteract phishing attacks. For this, we choose to create a Chrome extension using JavaScript to design a machine learning algorithm from scratch. We will be able to quickly publish the learnt model on the Chrome Web Store, where anyone can download and utilise our phishing detection solution. When selecting a machine learning method for our product, we must consider three criteria in order to complete this project successfully. First, the trained model's accuracy should be good, as a product utilised by end customers in the actual world should not produce incorrect results. Second, the algorithm being used should be able to produce classifications in real time, which means it should have a very short execution time and utilise very few computer resources. Third, while selecting a machine learning algorithm for the problem of phishing detection, false positives and false negatives must be taken into account. This is due to the fact that a user should not be induced to assume that a phishing website is authentic. As a result, when choosing a phishing detection classifier, we should consider these three restrictions.

The training dataset for our project comes from the UCI Machine Learning repository's "Phishing Websites Data Set." There are 11,055 items in the database, with 6157 phishing scams and 4898 valid ones. Each instance contains 30 traits that are commonly linked with phishing or suspicious web pages, such as the inclusion of an IP address in the URL domain or the existence of JavaScript code that modifies the information in the web browser address bar. A rule is assigned to each feature. If the rule is met, we consider it to be a sign of phishing, unless it is otherwise valid. Only discrete values were included in the dataset after it was standardised. Each instance's feature will have a value of '1' if the rule associated with that feature is satisfied, '0' if it is partially satisfied, and '-1' if it is unsatisfied.

The features represented by the training dataset can be classified into four categories,

A) Address bar-based features
   - Using IP address
   - Long URL to hide suspicious part
   - Use of URL shortening services
   - Use of "@" symbol
   - Redirection with "//"
   - Adding prefix or suffix separated by "-" to the domain
   - Sub domains and multi sub domains
   - HTTPS
   - Domain Registration Length
   - Favicon
   - Using Non-Standard Port
   - The Existence of "HTTPS" Token in the Domain Part of the URL

B) Abnormal based features
   - RequestURL
   - URL portion of anchor tag
   - Links in <meta>, <script> and <link> tags
   - Server Form Handler (SFH)
   - Submitting Information to Email
   - Abnormal URL

C) HTML and JavaScript based features
   - Status bar customization
   - Disabling right click option
   - Using pop-up window
   - IFrameRedirection

D) Domain based Features
   - Age of Domain
   - DNS Record
   - Website Traffic
   - PageRank
   - Google Index
   - Number of Links Pointing to Page
   - Statistical-Reports Based Feature

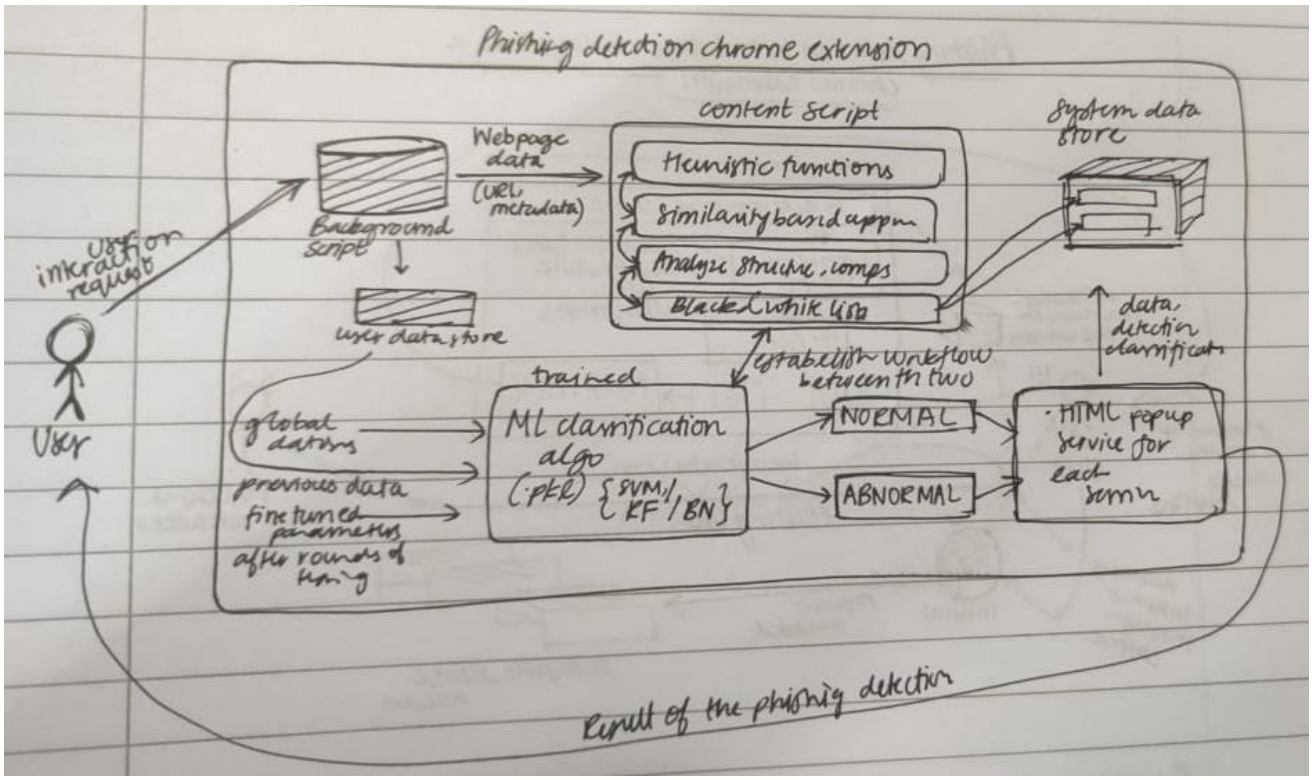
In this methodology, the utilization of black-and-white lists, heuristic rules, URL link analysis, and similarity-based approaches also serve as integral components of the phishing detection system. These techniques complement each other to enhance the system's accuracy and effectiveness in identifying potential threats.

Firstly, blacklists and whitelists play a crucial role in the initial screening process. By maintaining a database of known phishing URLs, domains, and IP addresses, the system can quickly flag suspicious web pages that match entries in the blacklist. Conversely, whitelists provide assurance to users by confirming the legitimacy of trusted

websites. However, to ensure relevance and efficacy, these lists require continuous updating to keep pace with evolving phishing tactics and the dynamic nature of the web. Secondly, heuristic rules contribute to the detection process by establishing criteria based on common characteristics of phishing websites. These rules, such as analyzing URL length, HTTPS usage, redirects, and domain-SSL certificate pairs, help identify suspicious patterns indicative of phishing attempts. By evaluating webpage content and behavior against these rules, the system can generate alerts when multiple heuristic violations are detected. Additionally, URL link analysis is employed to scrutinize the structure and components of URLs for potential phishing indicators. This involves examining domain names, subdomains, path parameters, and special characters to identify deviations from typical patterns associated with legitimate websites. By flagging suspicious links embedded in emails, messages, or webpages, the system can pre-emptively warn users about potential phishing threats. Furthermore, similarity-based approaches are utilized to compare accessed webpages against known legitimate websites or templates. Phishing websites often mimic genuine sites but may exhibit subtle differences in content or structure. By calculating similarity scores between accessed webpages and reference sets of legitimate sites, the system can detect deviations and alert users to potential phishing attempts. These techniques in the proposed methodology provide a comprehensive and robust approach to phishing detection.

Functional Architecture: The functional architecture depicted in the diagram outlines the components and workflow of the phishing detection system implemented as a Chrome extension and content script. Here's a breakdown of the components and their functions:

1. User Interaction:
    - This component represents the users' engagement with the system through their web browser.
    - Users initiate actions such as browsing websites, clicking on links, and interacting with web content.
2. Content Script:
    - The content script is a JavaScript program injected into web pages by the browser extension.
    - It operates within the browser environment and has access to the Document Object Model (DOM) of the web page.
    - The content script's primary role is to monitor and analyze user activities on web pages in real-time.
    - It intercepts web requests made by the browser and extracts relevant data for phishing detection.
3. Phishing Detection:
    - This refers to the process of identifying potential phishing attempts based on the analysis of web requests and page content.
    - This component receives data from the content script and applies various detection techniques to assess the likelihood of phishing.
    - Techniques may include URL analysis, content analysis (e.g., inspecting HTML markup and JavaScript code), and behavior analysis (e.g., tracking user interactions).
    - Advanced algorithms and heuristics are employed to detect patterns indicative of phishing activities, such as suspicious URLs, malicious scripts, or deceptive page elements.
4. Detection Module:
    - This module serves as the core component responsible for analyzing web request data and detecting phishing attempts.
    - It receives input from the content script and processes the data using a combination of rule-based heuristics and machine learning algorithms.
    - The module maintains a database of known phishing indicators and regularly update its detection mechanisms to adapt to evolving threats.



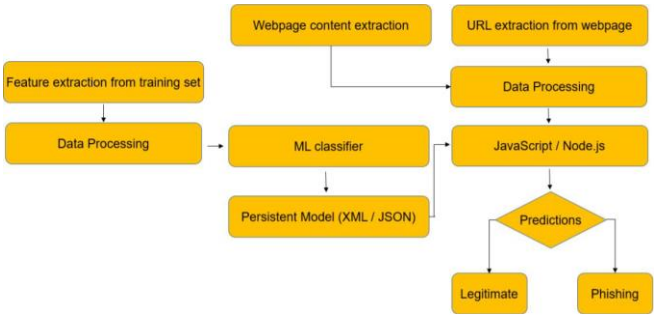
5. Feature Extraction:
    - This is the preprocessing step where relevant characteristics are extracted from web requests and page content.
    - Features may encompass various attributes, including URL structure (e.g., length, domain, path), content attributes (e.g., keywords, links), and user behavior patterns (e.g., click frequency, navigation sequence).
    - Extracted features serve as input to machine learning algorithms for classification and prediction.

6. Machine Learning Algorithm:
    - This plays a crucial role in analyzing extracted features and making predictions about the likelihood of phishing.
    - Supervised learning techniques are employed to train models on labelled datasets containing examples of phishing and legitimate web traffic.
    - Algorithms such as logistic decision trees, support vector machines (SVM), and neural networks are used to classify web requests as either phishing or legitimate.

7. Real-Time Phishing Detection:
    - Real-time phishing detection ensures that potential threats are identified promptly as users navigate web pages.
    - The system continuously monitors user activity and incoming web requests, allowing for immediate detection and mitigation of phishing attempts.
    - Real-time detection mechanisms enable the system to provide timely alerts and warnings to users, preventing them from inadvertently disclosing sensitive information to malicious actors.

8. User Alerts:
    - When a potential phishing attempt is detected, the system generates alerts to notify users and prompt appropriate actions.
    - Alerts are displayed as warning messages, pop-up notifications, or visual cues within the browser interface.
    - Users may be advised to exercise caution, avoid interacting with suspicious content, or navigate to safer web pages to mitigate the risk of falling victim to phishing scams.

Technical Approach: The proposed method aims to develop a sophisticated browser extension for phishing detection, leveraging advanced machine learning techniques. Specifically, the implementation relies on Support Vector Machines (SVM) due to their flexibility in handling margin and reduced computing complexity. This allows for the creation of an SVM-trained persistent model capable of accurately identifying potentially dangerous websites for classification purposes. Given its widespread use, the extension is primarily designed to support the Chrome browser. Additionally, extensions are web-independent, consolidating multiple files into a single downloadable package for users, simplifying installation as a one-time action. The architecture of the browser extension involves training the model with available data using an SVM discriminative classifier. Subsequently, the trained persistent model is integrated into the extension, enabling it to predict the authenticity of websites accessed by users. This prediction process occurs in real-time, generating alerts to indicate the legitimacy of the browsed URL upon every page load. To achieve this, a combination of Python-based training and JavaScript-based testing modules is employed. The training component, developed in Python, capitalizes on its ability to effectively handle complex numeric computing tasks. Conversely, the testing module focuses on site content and feature extraction, minimizing heavy computation operations to mitigate client-end computation performance latency.



The Chrome extension adheres to Google's guidelines and comprises three essential files: manifest.json, content.js, and background.js. The manifest.json file serves as a repository for all metadata pertaining to the extension, listing all associated files and resources. Once installed, the content.js file is loaded on every page within the Chrome browser. However, it operates as an unprivileged module, possessing direct access solely to DOM components. To interact with external APIs and manipulate the browser's user experience, content.js relies on supporting files, with background.js serving as a supplementary script facilitating these interactions, commonly referred to as message forwarding.

Within the content.js script, multiple functions have been implemented for web content and URL feature extraction, crucial for identifying phishing portals. These functions include:

- isIPInURL(): Identifies the presence of an IP address in the URL.
- isLongURL(): Validates if the length of the URL exceeds 75 characters.
- isTinyURL(): Identifies URLs smaller than 20 characters.
- isAlphaNumericURL(): Checks for alphanumeric '@' symbols in the URL.
- isRedirectingURL(): Verifies if '//' appears more than once within the URL.
- isHyphenURL(): Checks for the presence of '-' adjacent to the domain name in the URL.
- isMultiDomainURL(): Ensures that the domain name is restricted to top-level domain, country-code, and second-level domain.
- isFaviconDomainUnidentical(): Verifies if links on the given web page are loaded from other domains.
- isIllegalHttpsURL(): Identifies the presence of multiple 'https' instances in the URL string.
- isImgFromDifferentDomain(): Validates if images on the given web page are loaded from other domains.
- isAnchorFromDifferentDomain(): Detects if links on the given web page are loaded from other domains.
- isScLnkFromDifferentDomain(): Identifies if scripts on the given web page are loaded from other domains.

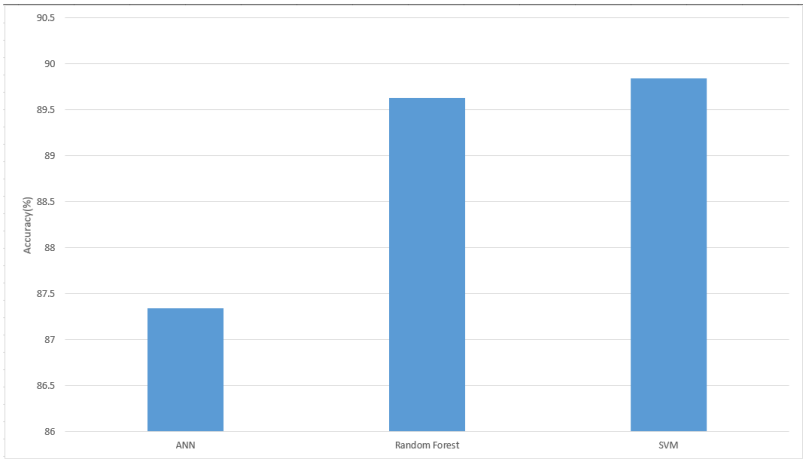
- isFormActionInvalid(): Detects invalid or blank form submissions.
- isMailToAvailable(): Checks for anchor tags incorporating 'mailto'.
- isStatusBarTampered(): Validates if the status bar display is manipulated on mouseover.
- isIframePresent(): Identifies sites that exhibit iframes in the DOM.

These functions collectively provide the extension with the capability to thoroughly analyze webpage content and URL characteristics, enabling effective detection of potential phishing attempts.

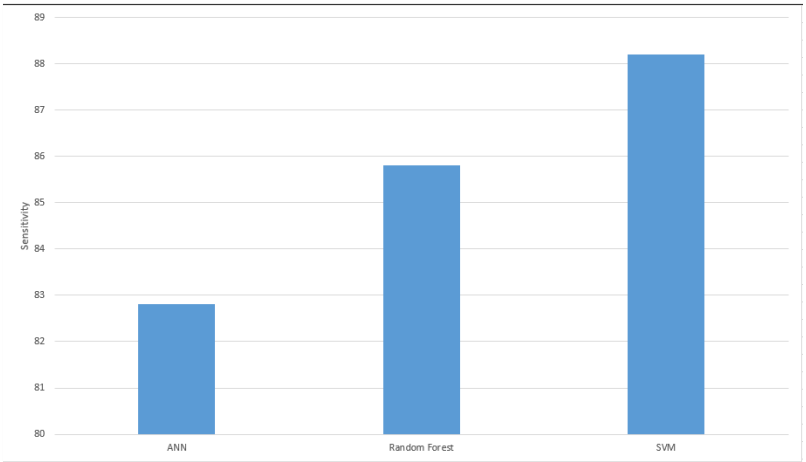
Results: On the phishing dataset, our project rigorously evaluates the performance of all the classifiers presented. Through extensive testing on 3317 test samples, we employ a diverse set of performance criteria to assess the effectiveness of each algorithm. The results of these evaluations are meticulously documented and presented through graphical representations, allowing for clear and insightful interpretation of the findings.

The confusion matrices provide valuable insights into the performance of three machine learning classifiers—Random Forest, Artificial Neural Network (ANN), and Support Vector Machine (SVM)—in distinguishing between phishing and legitimate URLs. In the case of Random Forest, out of a total of 3317 test samples, it correctly predicted 1249 phishing URLs and 1680 legitimate URLs, yielding a true positive rate of 37.7% for phishing URLs and a true negative rate of 50.6% for legitimate URLs. The false negative rate, indicating legitimate URLs incorrectly classified as phishing, stands at 5.5%, while the false positive rate, denoting phishing URLs classified as legitimate, is 4.9%. Artificial Neural Networks correctly identified 1205 phishing URLs and 1692 legitimate URLs, resulting in a true positive rate of 36.3% for phishing URLs and a true negative rate of 51.0% for legitimate URLs. The false negative rate is 5.1%, and the false positive rate is 7.5%. Lastly, the SVM confusion matrix reveals that SVM accurately classified 1293 phishing URLs and 1731 legitimate URLs, achieving a true positive rate of 39.0% for phishing URLs and a true negative rate of 52.2% for legitimate URLs. The false negative rate is 3.9%, while the false positive rate is 6.2%.

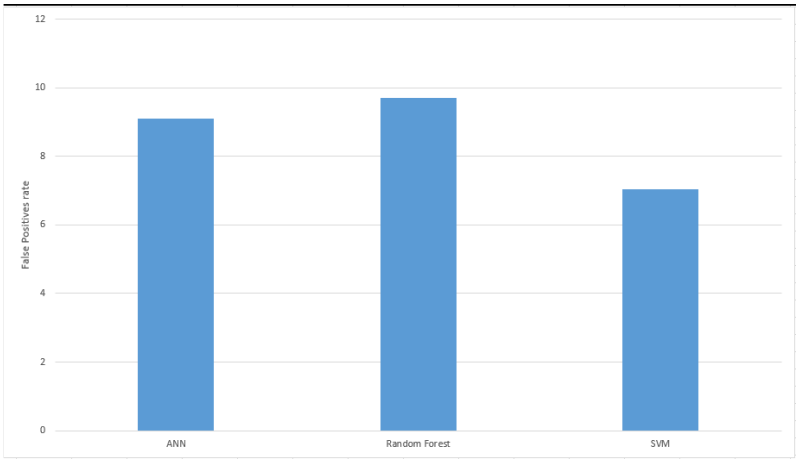
Algorithms	Accuracy(%)	Specificity(%)	Sensitivity(%)
Artificial Neural Networks	87.34	91	83
Random Forests	89.61	90	86
SVM	89.93	93	89



Sesnitivity of classifiers:

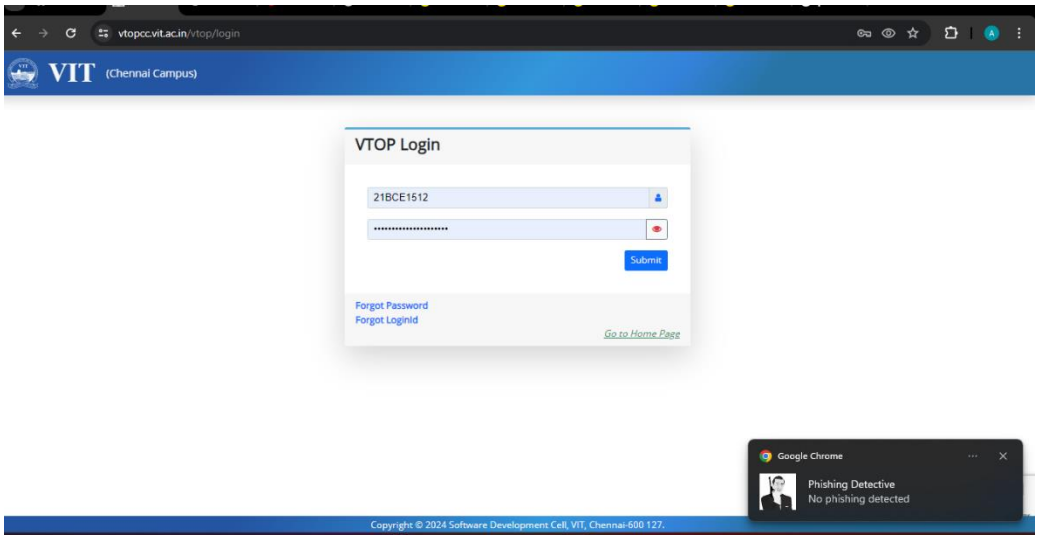


Rate of false positives:

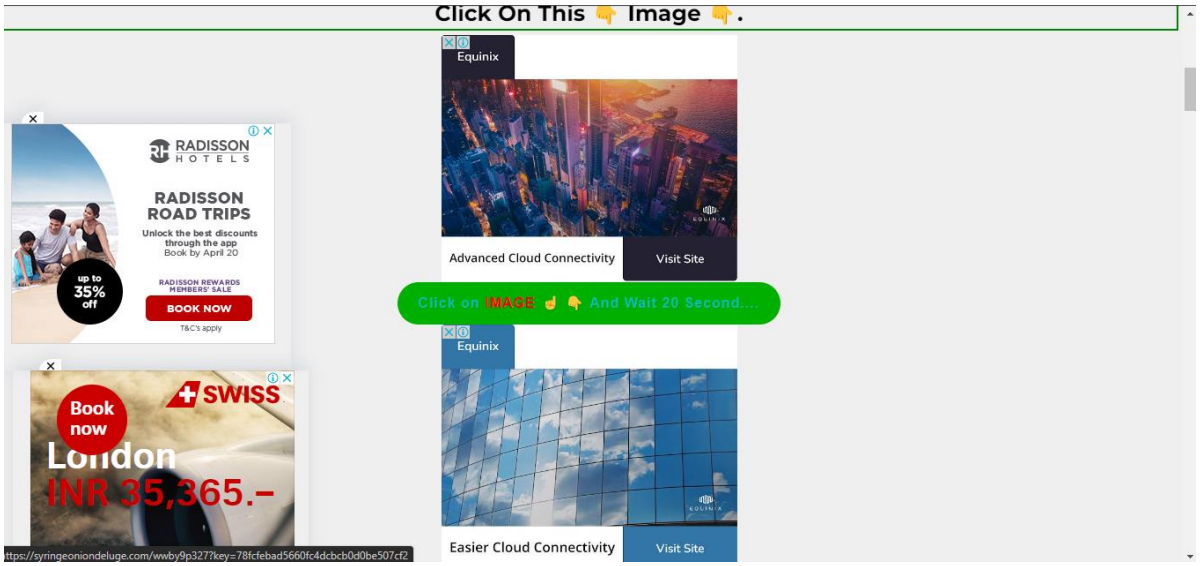


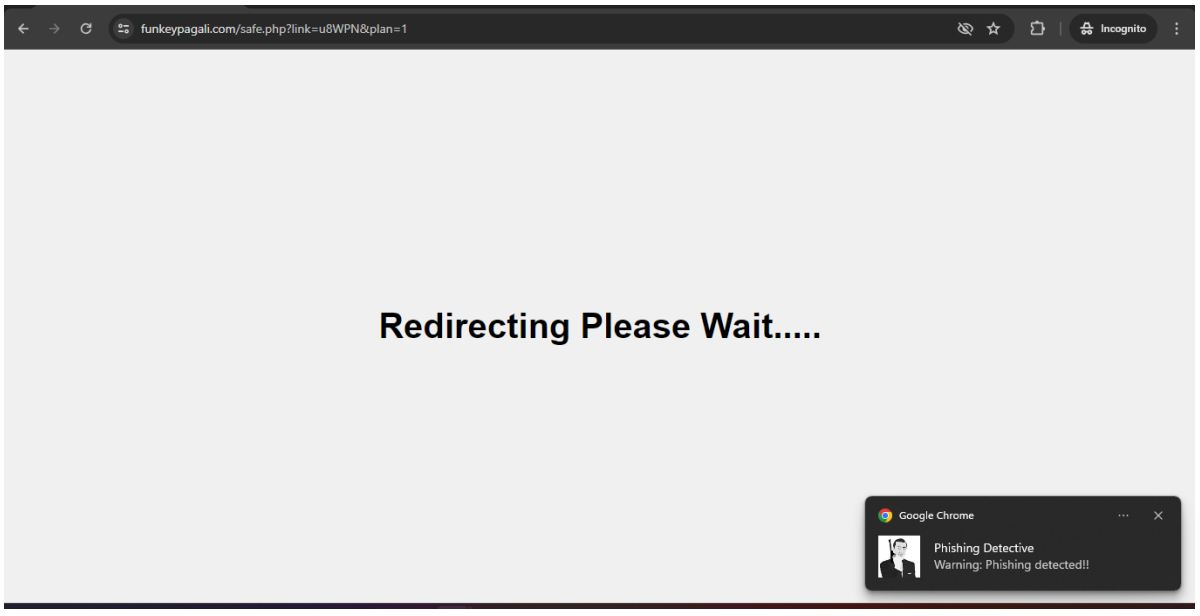
Demonstration on web pages:

Legitimate website:



Phishing Website:





Conclusion: The study underscores the pressing nature of the phishing threat and emphasizes the necessity for robust detection methods to safeguard internet users. Through rigorous testing of machine learning techniques on the Phishing Websites Dataset, we have identified promising avenues for effective detection. The development of a Chrome plugin based on the selected algorithm represents a tangible step towards empowering end users with accessible tools to identify phishing websites. Looking ahead, our focus shifts towards further refining the phishing detection system. Future improvements will involve transforming the system into a scalable web service with online learning capabilities, enabling it to dynamically adapt to evolving phishing attack patterns and continuously enhance model accuracy. Furthermore, incorporating real-time data feeds and threat intelligence sources into the detection system can bolster its capability to identify emerging phishing threats promptly. By continuously monitoring and analyzing new data, the system can stay ahead of evolving attack techniques and provide users with timely warnings about potential risks. By prioritizing ongoing innovation and adaptation, we aim to stay at the forefront of combating phishing threats and promoting online safety and security for all users**.**

References:

1. *Detecting and Classifying Phishing Websites by Machine Learning*. (2021, July 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9712124

2. *The Emergence Threat of Phishing Attack and The Detection Techniques Using Machine Learning Models*. (2021, July 8). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9528 2043

3. *Phishing Website Classification and Detection Using Machine Learning*. (2020, January 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9104161

4. *Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection. (2019, June 1).* IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8858884

5. *Real-Time Phishing Attack Detection through Advanced Machine Learning Techniques. (2023, December 8)*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10456013