“ML-Powered Chrome Extension for Phishing URL Detection”

Submitted for the degree of

B. Tech

In

Computer Science Engineering

By

ADITYA BATTIN (2020BCS095)
MAHESHWARI SURWASE (2020BCS005)
SYED SOHAIL ANWAR (2020BCS010)

Under the Guidance of

DR. JAISHRI WAGHMARE

SHRI GURU GOBIND SINGHJI INSTITUTE OF
ENGINEERING & TECHNOLOGY,
NANDED (M.S.)

ACADEMIC YEAR

(2023-2024)

CERTIFICATE

This is to certify that the project work entitled “**ML-Powered Chrome Extension for Phishing URL Detection**” being submitted by Group of students **Aditya Battin, Maheshwari Surwase, Sohail Anwar** to Shre Guru Gobind Singhji Institute of Engineering & Technology, Nanded for the award of the degree Batcher of Computer Science in Computer Science Engineering is a record of bonafide work carried out by him under my supervision and guidance. The matter contained in this dissertation has not been submitted to any other University or institute for the reward of any degree or diploma.

Name of Guide	Name of HOD	Name of Director
Dr. Jaishri Waghmare	Prof. S.M.Bansode	Dr. M.B.Kokare

Department of Computer Science Engineering

SGGSIE&T, Nanded

ABSTRACT

In the face of an ever-expanding internet landscape, the proliferation of online threats, particularly phishing websites, demands innovative solutions to safeguard user security. This project introduces a sophisticated Chrome Extension designed to detect phishing websites with precision, utilizing machine learning algorithms. The models are trained on a dataset sourced from the PhishTank opensource service and the University of New Brunswick's open datasets, focusing on benign URLs. The extension employs a Random Forest model for URL-based detection and Logistic Regression for in-depth analysis of HTML, CSS, and JavaScript components.

The training dataset comprises 5,000 randomly selected phishing URLs from PhishTank and an equivalent number of legitimate URLs from the University of New Brunswick's benign dataset. These URLs are regularly updated by PhishTank and represent a diverse and dynamic set of potential threats and benign instances, ensuring the robustness of the models.

The feature extraction process involves three categories: Address Bar-based features (9 features), Domain-based features (4 features), and HTML & Javascript-based features (4 features). In total, 17 features are extracted from the 10,000 URL dataset, providing a rich set of information for the machine learning models.

The Address Bar-based features encompass aspects like URL length, presence of special characters, and the use of secure protocols. Domain-based features include information about the domain's age and registration length, contributing to a holistic understanding of the URL's legitimacy. HTML & Javascript-based features delve into the structure and behavior of the webpage, enhancing the models' capacity to discern potential phishing threats.

The Chrome Extension, designed with a user-centric approach, offers an intuitive interface for users to seamlessly scan URLs for potential phishing threats. The extension maintains a comprehensive history of analyzed websites, empowering users with valuable insights into web security trends.

The machine learning models are deployed using a Flask server, ensuring efficient communication between the Chrome Extension and the models. The extension's frontend

is developed using HTML, CSS, JavaScript, and Bootstrap, prioritizing a visually appealing and responsive design to enhance user experience.

In a landscape where cybersecurity is paramount, this project contributes a powerful tool to users, enabling secure navigation through the digital realm. By combining machine learning technology, rigorous feature extraction, and an accessible user interface, the Chrome Extension stands as a beacon of enhanced online security. Users gain protection, awareness, and confidence in their online activities, minimizing the risk of falling victim to phishing attacks.

ACKNOWLEDGEMENT

It is privilege for me to have been associated with **Dr. Jaishri Waghmare**, my guide during this project work. I have greatly benefited by her valuable suggestions and ideas. It is with great pleasure that I express my deep sense of gratitude to her for her able guidance, constant encouragement and patience throughout the work.

I am also thankful to **Dr. M.B.Kokare,** Director and **Prof. S.M.Bansode,** Head of Computer Science Engineering Department for their constant encouragement & cooperation.

I am also thankful to laboratory staff **Ms. Misha Nihalani** for helping me during this dissertation work. I take this opportunity to thank **Maheshwari Surwase, Sayad Sohail** or providing company during the work.

Aditya Battin

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

CHAPTER 1
INTRODUTION

In the dynamic landscape of cybersecurity, the persistent threat of phishing attacks poses a significant risk to the security of personal and organizational information. Phishing, a deceptive practice where malicious entities impersonate trusted sources to extract sensitive data, has evolved into a complex challenge, exploiting human vulnerabilities in the digital landscape. Recognizing the urgent need for innovative solutions to counter these threats, this research embarks on an exploration of phishing detection, integrating advanced feature analysis using Machine Learning (ML) techniques and artificial intelligence (AI) within the context of a Chrome extension.

1.1 Background :

Phishing attacks have become increasingly prevalent and sophisticated, adapting to exploit advancements in technology and human behavior. Traditional detection methods often struggle to keep pace with the dynamic and evolving tactics employed by cyber adversaries. In the context of a Chrome extension, which is a widely used tool for web browsing, the need for effective and real-time phishing detection is paramount. Users often encounter potential phishing URLs while browsing, making it crucial to integrate advanced feature analysis into the browsing experience for enhanced security.

1.2 Importance and Relevance :

The significance of this research lies in its commitment to advancing the effectiveness of phishing detection through the synergistic integration of ML techniques and AI within the familiar context of a Chrome extension. By comprehensively analyzing diverse features associated with phishing URLs, including structural, temporal, and behavioral aspects, the aim is to enhance the discernment of subtle patterns indicative of malicious intent. The relevance of this research is underscored by the escalating sophistication of phishing attacks, necessitating a proactive and intelligent defense mechanism within the browser environment.

1.3 Overall Objective :

The overarching goal of this research is to contribute to the development of a robust and adaptive phishing URL detection system as a Chrome extension. By harnessing the power of advanced feature analysis, particularly through ML techniques and AI, the aim is to create a more nuanced understanding of phishing attempts, thereby fortifying the accuracy of detection directly within the browsing experience. The integration of ML techniques and AI augments our capability to discern intricate patterns within URLs, enhancing the system's ability to adapt to evolving attack vectors encountered during web browsing.

1.4 Objectives and Scope :

This research is designed to achieve several specific aims within the context of a Chrome extension for phishing URL detection. Firstly, it aims to provide a comprehensive understanding of the diverse aspects associated with phishing attacks within the browsing environment. Secondly, it seeks to design and implement a detection system that integrates advanced feature analysis, specifically using ML techniques and AI, considering diverse dimensions of the threat landscape. Thirdly, the research endeavors to assess the efficacy of these advanced techniques in enhancing the adaptability and accuracy of the detection mechanism directly within the Chrome browsing experience. The scope extends to the practical application of these methodologies in real-world scenarios, evaluating their performance in dynamic and evolving cyber threat environments encountered during web browsing.

As we navigate through the subsequent sections of this report, we will unfold the methodology, experimentation, and results, presenting a detailed account of our approach and its implications within the specific context of a Chrome extension for phishing URL detection. Through this research, we aspire not only to advance the theoretical understanding of phishing detection but also to contribute tangibly to the ongoing efforts to fortify our digital ecosystems against the ever-evolving challenges posed by phishing attacks during web browsing.

CHAPTER 2
LITERATURE REVIEW

Phishing attacks have evolved into sophisticated threats, exploiting human vulnerabilities and technological advancements. The need for effective detection mechanisms, particularly within the browsing environment, is crucial. This literature review explores existing research, with a focus on two main papers : "A Deep Learning-Based Framework for Phishing Website Detection" and "An Effective Phishing Site Prediction using Machine Learning," to provide a comprehensive understanding of the current landscape and to inform the development of an ML-powered Chrome extension for phishing URL detection.

2.1 Deep Learning-Based Framework for Phishing Website Detection:

Reference [1] presents a deep learning-based framework for phishing website detection.

Author : Lizhen Tang and Qusay H. Mahmoud from the Department of Electrical, Computer, and Software Engineering at Ontario Tech University, Canada.

Overview : phishing attacks, where attackers use social engineering to trick users into visiting malicious websites and entering personal information. The paper proposes a deep learning-based framework for detecting phishing websites, implemented as a browser plug-in for real-time risk assessment. Paper is divided into two parts: deep learning-based methods for detecting phishing websites and frameworks with prototype implementations. It reviews various studies and frameworks that employ machine learning models for phishing detection, discussing their methodologies and results.

Framework Design: data collection tasks, machine learning, cloud application, and web browser extension. The section describes the process of data collection, machine learning model training, and the flexible selection of different data sources for training.

Data Collection Tasks: The datasets used for training include Phish Storm, ISCX-URL2016, Kaggle, and Phish Tank.

Machine Learning: It explains the tokenization process, the choice of character-level features, and the use of recurrent neural networks (RNNs) for modeling. The section also mentions the development of six machine learning models: Logistic Regression, SVM, Random Forest, RNN, RNN-GRU, and RNN-LSTM.

The proposed deep learning-based framework for detecting phishing URLs and the achievement of the highest accuracy of 99.18% with the RNN-GRU model. It also outlines the prototype implementation of the framework as a Chrome browser extension.

2.2 Effective Phishing Site Prediction using Machine Learning:

Reference [2] contributes insights of machine learning techniques for phishing site prediction.

Authors: Aman Raj Pandey, Tushar Sharma, Subarna Basnet, Ankesh Kumar
Affiliation: Department of Computer Science and Engineering, Sharda University,India.
Overview : The paper addresses the increasing threat of phishing attacks in cyberspace and proposes an API for predicting whether a website is malicious or not using machine learning, regression, and the Naïve Bayes algorithm. The proposed system achieves an accuracy of 98%, outperforming existing blacklisting methods. The paper discusses the phishing process, existing detection methods, and presents the research methodology and results.
Work of Paper: Mentioned approaches include Blacklist, Heuristic-based methods, Deep Learning, Decision Tree algorithms, Web Crawling, and others. The section provides insights into the evolution of phishing detection techniques.
Methodology : Understanding the Algorithms available and Working on Model. The first subsection discusses the algorithms considered, such as Logistic Regression, MultinomialNB, and Support Vector Machine (SVM). The second subsection describes the working of the proposed model, including data collection, tokenization, and training. The section highlights the use of FastAPI for creating a real-time API.
Result: The model, based on Logistic Regression, achieved a high accuracy of 98%, with a detailed classification report including precision, recall, and F1-score. The API's response is visualized, and the system's performance is evaluated based on test cases involving both malicious and non-malicious URLs.

The authors express their intention to work on a prevention system for Ransomware in future research. The section calls for continued research efforts to develop reliable systems for mitigating cybercrimes and protecting users from potential threats.

2.3 A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks :

Reference [3] contributes insights of machine learning techniques for phishing site prediction.

Author: S. Asiri, S. Alshehri, and S. Alqahtani.

Features Used: Features from the survey include URL sets, page content sets, and domain attributes. Abnormal URL traits, HTML/JavaScript attributes, and domain features such as age, traffic, and linking pages quantity are also considered.

Classification Method:Support Vector Machines (SVMs), Decision Trees (DT), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), etc.

Research Findings: The authors conduct a thorough review of traditional and deep learning models, comparing their performance on various datasets and features. They discuss the limitations and challenges of current phishing detection methods, providing a detailed description of survey datasets and performance metrics. While traditional models are acknowledged as effective, challenges arise from feature extraction and dataset size. Deep learning, especially with unsupervised methods, shows promise in addressing these challenges.

Results: The survey presents a comprehensive overview of traditional and deep learning models, emphasizing the potential of deep learning in phishing detection. The authors analyze current method drawbacks and propose research directions. The discussion provides valuable insights into current phishing detection methodologies and offers a roadmap for future cybersecurity research. The focus is on intelligent detection designs, particularly in the context of HTML URL phishing attacks.

Strengths: The information is presented clearly, making it accessible for readers. Relevant references are provided, facilitating further exploration. Practical applications of intelligent detection designs are discussed, providing insight into real-world scenarios. The survey offers a comprehensive analysis of various intelligent detection techniques.

Weaknesses: The survey may have limited applicability to other types of phishing attacks beyond HTML URL phishing. It lacks empirical data on the effectiveness of the discussed intelligent detection techniques, potentially limiting readers' ability to evaluate their practical value. The use of technical language assumes a certain level of familiarity with cybersecurity concepts, which may pose a challenge for readers without a technical background.

2.4 Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning:

Reference [4] contributes insights of machine learning techniques for phishing site prediction.

Authors: Ou Ye, Jingzhou Han, Lianyang Zou.

Features & Methodology: The Web2Vec model leverages automatically learned features from URL, page content, and DOM structure, setting it apart from traditional methods by incorporating features from multiple aspects. Employing a Random Forest algorithm as its machine learning-based classification method, the model integrates a hybrid CNN-BiLSTM feature extraction and attention mechanism.

Results: Web2Vec outperforms traditional phishing detection methods, demonstrating higher accuracy, precision, true positive rates, and reduced false positives. This improvement is attributed to the model's unique approach, utilizing a hybrid CNN-BiLSTM network and attention mechanism for effective detection and diverse webpage feature learning. Experiment results detailed in Tables 4, 5, 6, and 7 underscore Web2Vec's superiority across multiple performance metrics.

Strengths: The Web2Vec model excels in automated feature extraction from various webpage aspects, enhancing phishing webpage detection capabilities. Its performance is further enhanced through the incorporation of a hybrid CNN-BiLSTM network and an attention mechanism.

Weaknesses: The model requires a substantial amount of training data for effective learning of multifaceted webpage representations. Its performance can be influenced by the quality and diversity of the training data.

Web2Vec's innovative approach, integrating a hybrid CNN-BiLSTM network and attention mechanism, positions it as a superior solution for phishing detection compared to traditional methods. Its effectiveness not only improves performance metrics but also enhances online transaction security, mitigating the risks associated with phishing attacks.

2.5 Phishing Detection System Through Hybride Machine Learning Based on URL:

Authors: F. Sattari, A. H. Farooqi, Z. Qadir, B. Raza.

Features & Methodology: The study employs abnormal features, address bar-based features, JavaScript-based features, HTML, and domain-based features for phishing detection. The classification methods include decision tree classifier (DTC), random forest classifier (RFC), support vector machine (SVM), and logistic regression (LR).

Results: The system undergoes evaluation using accuracy, precision, recall, specificity, and F1-score metrics. The hybrid approach (DTC + SVM) stands out with an exceptional 98.5% accuracy, surpassing other models. The proposed system demonstrates superior accuracy and false positive rate compared to rivals. Achieving an impressive accuracy of 98.12%, the proposed system outshines with precision (97.31%), recall (96.33%), specificity (96.55%), and F1-score (95.89%). Ensembled tree models outperform linear and probabilistic ones, and a hybrid model (LR+SVC+DT) excels but slightly below the proposed system's performance.

Strengths: The proposed system's accuracy of 98.12% showcases its effectiveness, particularly with the hybrid DTC+SVM approach. It outperforms rivals in both accuracy and false positive rate. The study provides detailed materials/methods, aiding the development of similar phishing detection systems.

Weaknesses: The exclusive focus on URL phishing may limit the model's applicability to a comprehensive phishing detection system. The paper lacks an ethical discussion on biases, privacy concerns, and potential algorithmic biases in machine learning for phishing detection. The small dataset raises concerns about result generalizability, and the resource-intensive implementation may pose an adoption barrier.

The proposed system, with its 98.12% accuracy, stands out among machine learning methods. The hybrid DTC+SVM approach proves superior, and the system excels in accuracy and false positive rate compared to state-of-the-art systems, showcasing its potential for effective phishing detection in online security.

2.6 Phishing URL Detection: Real-Case Scenario Through Login URLs:

Authors: M. Sánchez-Paniagua, J. M. de Fuentes, and J. M. Murillo.

Features & Classification Method: The study utilizes a combination of lexical, host-based, WHOIS, and GeoIP-based features for phishing URL detection. Evaluation encompasses classifiers like XGBoost, LightGBM, RF, and deep learning models by Zhang et al. and Kim. Feature extraction techniques involve handcrafted features and statistical features using Term Frequency-Inverse Document Frequency (TF-IDF) combined with character N-gram.

Research Findings: The paper introduces a method for phishing URL detection, focusing on login URLs frequently targeted in phishing attacks. This approach diverges from traditional content-centric methods. The study comprehensively compares machine learning and deep learning techniques, shedding light on their strengths and weaknesses, including handcrafted and statistical features using TF-IDF with character N-gram.

Strengths: The paper brings innovation to phishing URL detection by focusing on login URLs, offering a distinctive perspective compared to content-focused methods. It provides a comprehensive comparison of machine learning and deep learning techniques, incorporating handcrafted and statistical features, offering valuable insights into their respective performances.

Weaknesses: The study relies on a relatively small dataset of 60K legitimate URLs and 30K phishing URLs, potentially limiting the method's generalizability. The paper lacks a detailed explanation of the feature selection process, hindering replication and understanding. While various techniques are compared, the study falls short of benchmarking against other state-of-the-art phishing URL detection methods.

The authors evaluate the efficacy of traditional and deep learning models without resorting to numerical enumeration. Traditional models prove effective but encounter challenges in feature extraction and dataset size. The study underscores the potential of deep learning, especially with unsupervised methods, in phishing detection, emphasizing the significance of meticulous data preprocessing.

2.7 Detecting Malicious URLs Using ML Techniques: Review and Research Directions:

Authors: M. Aljabri et al.

Features & Classification Method: The study incorporates lexical, content, and network-based features for detecting malicious URLs. Classification methods include Support Vector Machines (SVMs), Decision Trees (DT), and deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

Research Findings: The authors conduct a thorough review of traditional and deep learning models, comparing their performance across diverse datasets and features. They delve into the limitations and challenges of current phishing detection methods, providing a detailed description of survey datasets and performance metrics. While traditional models are effective, their feature extraction limitations are discussed, with deep learning showing promise, particularly with unsupervised techniques.

Strengths: The survey comprehensively covers traditional and deep learning models, evaluating their performance on varied datasets and features. The emphasis on deep learning highlights its potential in enhancing phishing detection capabilities.The authors meticulously detail the drawbacks of current methods, contributing to future research direction. The survey suggests future research areas, including advanced deep learning models and unsupervised techniques.

Weaknesses: The survey predominantly focuses on exploring the potential of deep learning models for enhancing phishing detection capabilities, strategically emphasizing advanced techniques in pursuit of robust cybersecurity solutions. Authors analyze the existing limitations of current methods in phishing detection, providing a critical evaluation and suggesting fruitful research directions to address these challenges. This proactive approach enhances the practical relevance and future applicability of the findings. The survey offers valuable insights into the current state of phishing detection methodologies, synthesizing the strengths and weaknesses of existing approaches, serving as a comprehensive roadmap for cybersecurity research.

The authors compare traditional and deep learning models, highlighting the effectiveness of traditional methods but acknowledging challenges in feature extraction and dataset size. Emphasizing the promise of deep learning, especially with unsupervised methods, in phishing detection, they stress the importance of vital data preprocessing techniques like tokenization and feature management. The call is for advanced models to address evolving phishing challenges.

Status of Development in Phishing Detection:

The literature reveals a growing trend in using advanced techniques such as deep learning and machine learning for phishing detection. While traditional methods often rely on static indicators, these references signify a paradigm shift towards dynamic and adaptive approaches. The field is actively exploring the integration of artificial intelligence to enhance detection accuracy and adaptability, which aligns with the objectives of our proposed ML-powered Chrome extension.

Challenges and Limitations:

Despite the advancements, challenges persist. The dynamic nature of phishing attacks requires continuous adaptation of detection mechanisms. References [1] and [2] acknowledge the evolving nature of cyber threats, emphasizing the need for real-time and proactive solutions. Understanding these challenges is vital for the design and implementation of an effective ML-powered Chrome extension that can keep pace with the ever-changing phishing landscape.

Gap Analysis:

While references [1] and [2] contribute significantly to the understanding of phishing detection, a noticeable gap exists in the literature concerning the integration of such advanced techniques directly into web browsing experiences. Our proposed ML-powered Chrome extension aims to bridge this gap by providing users with real-time protection during their online activities, aligning with the increasing importance of securing users at the point of interaction with potential phishing URLs.

The literature review highlights the transformative impact of deep learning and machine learning in phishing detection. Leveraging insights from "A Deep Learning-Based Framework for Phishing Website Detection" and "An Effective Phishing Site Prediction using Machine Learning," our research seeks to advance these concepts by developing an ML-powered Chrome extension for phishing URL detection. The review emphasizes the need for real-time adaptability, the significance of feature analysis, and the integration of advanced techniques to counter the dynamic nature of phishing attacks.

CHAPTER 3

PROPOSED PLAN OF WORK

The proposed project aims to develop a Machine Learning (ML)-powered Chrome extension for detecting phishing URLs in real-time. Phishing attacks have become increasingly sophisticated, requiring innovative solutions that integrate advanced feature analysis within the browsing experience. This plan outlines a comprehensive approach divided into distinct phases, each contributing to the overall development and deployment of the solution.

3.1 Traditional Phishing Detection Techniques

Traditional techniques for identifying phishing sites often rely on platforms like PhishTank, where users report and verify phishing URLs. Another approach involves training machine learning models on large datasets of URLs to differentiate between legitimate and malicious sites. These methods, while effective, often require constant updates and maintenance to stay relevant.

3.1 a. Deployment Challenges in Phishing Detection

Many existing solutions primarily focus on the techniques used for phishing detection but overlook the challenges associated with deployment. Deploying a phishing detection system as a Chrome extension without server dependencies is a novel approach that addresses performance and usability concerns. This method streamlines the user experience by eliminating the need for continuous server communication.

3.1 b. Performance Enhancement through Extension Deployment

Deploying the phishing detection system as a Chrome extension aims to enhance performance. By shifting the processing load to the client side, the extension can rapidly analyze URLs in real-time without relying on external servers. This not only reduces latency

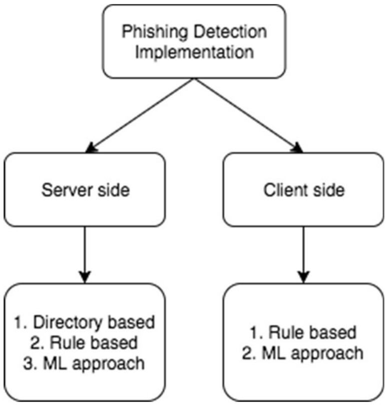


Figure 3.1 Implementation

but also ensures that users receive instant feedback on the legitimacy of visited sites.

3.1 c. Plug-and-Play Usability

The primary innovation in this project lies in its plug-and-play usability. Unlike systems that require intricate setups or server connections, the Chrome extension operates seamlessly upon installation. Users can simply add the extension to their browsers, making it a hassle-free experience. This approach increases the accessibility and user adoption of the phishing detection system.

3.1 d. Eliminating Server Dependency

One significant advantage of deploying the system as a Chrome extension is the elimination of server dependency. Traditional solutions often rely on continuous server updates for the latest phishing site data. In contrast, the extension functions independently, reducing the burden on server infrastructure and ensuring uninterrupted service for end-users.

3.2 Dataset Preparation and Features to be considered

The foundation of any machine learning model lies in the quality of the dataset it is trained on. For our Phishing URL detection model, we meticulously curated a dataset comprising legitimate URLs from the 'University of New Brunswick' and phishing URLs from 'Phishtank'. In this section, we delve into the process of acquiring, preprocessing, and selecting features to ensure the robustness and effectiveness of our model.

3.2.1 Dataset Acquisition:

We combined the URL's obtained from two sources as listed below :

Legitimate URLs from University of New Brunswick : We obtained a set of legitimate URLs from the 'University of New Brunswick' dataset, available at [5]. This dataset provided a diverse collection of URLs from reputable sources, contributing to the model's ability to discern legitimate web addresses accurately.

Phishing URLs from Phishtank : For phishing URLs, we sourced data from 'Phishtank' using their developer interface [6]. This database specializes in phishing URLs, enabling the model to recognize and flag potentially malicious web addresses.

3.2.2 Data Preprocessing:

Data preprocessing is a crucial step to enhance the quality and uniformity of the dataset. The following steps were applied:

Data Cleaning: Removal of duplicate URLs to avoid redundancy. Handling missing values, ensuring a complete and consistent dataset.

Normalization: Standardizing URL lengths for uniformity. Normalizing numerical features to a common scale.

Encoding: Converting categorical features into a numerical format for model compatibility.

Balancing: Balancing the dataset to ensure an equal representation of both classes (legitimate and phishing URLs). This prevents bias towards the majority class during model training.

3.2.3 Feature selection is pivotal in creating an efficient and streamlined model.

The ARFF (Attribute-Relation File Format) structure was chosen for its advantages in maintaining metadata about the dataset [7].

The following features were selected:

URL Structure: URL length, shortening service usage, presence of '@' symbol, double slash redirection, prefix/suffix existence.

Domain Information: Subdomain presence, SSL final state, domain registration length, favicon existence, port usage, HTTPS token, and request URL.

Content Features: URL of anchor, links in tags, presence of SFH (Server Form Handler), submitting to email, abnormal URL, redirection, on mouseover, right-click, pop-up window, iframe presence.

Age and Records: Age of the domain, DNS record availability, web traffic status, page rank, Google index status, links pointing to the page, and statistical report presence.

Result: The target variable indicating whether the URL is phishing or legitimate.

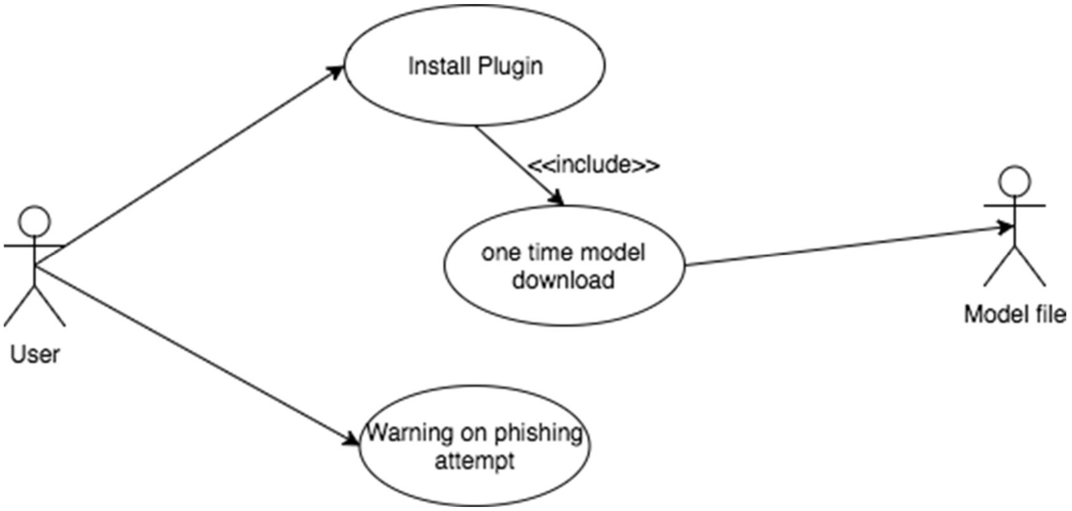


Figure 3.2.3 UML Diagram

3.3 Creating the ML Model and Feature Extraction Algorithm

In the development of our phishing URL detection system, we opted for the RandomForest algorithm due to its consistent superiority over other machine learning algorithms. The chosen model achieved an impressive accuracy of 94.766% when tested with a dataset comprising 10,000 instances. It is noteworthy that further improvements can be explored by employing a larger dataset and refining the preprocessing techniques.

Feature Extraction Algorithm: The feature extraction algorithm plays a pivotal role in preparing the input data for the machine learning model. In our implementation, we utilized JavaScript to dynamically extract features from web pages.

The key aspects of the feature extraction algorithm are as follows:

Table 3.3 : Features, there description and results

Feature	Description	Result
IP Address Extraction	Examines the URL and domain to detect the presence (1) or absence (-1) of an IP address.	Binary
URL Length Analysis	Categorizes URLs into short (-1), medium (0), or long (1) based on their length.	-1, 0, 1
Detection of Tiny URL	Determines the presence of a tiny URL by evaluating the length of the domain (-1 for non-tiny, 1 for tiny).	Binary
Symbolic Features	Detects the '@' symbol in the URL through regular expressions (-1 for absence, 1 for presence).	Binary
Redirecting Using Double Slash	Identifies the use of double slashes in the URL, indicating potential redirection (-1 for no redirection, 1 for redirection).	Binary
Prefix/Suffix in Domain	Detects hyphens in the domain to identify potential phishing URLs (-1 for absence, 1 for presence).	Binary
Number of Subdomains	Counts the number of dots in the domain to categorize URLs into three classes (-1, 0, 1).	-1, 0, 1
HTTPS Usage	Determines HTTPS presence in the URL using regular expressions (-1 for HTTPS, 1 for HTTP).	Binary
Favicon Analysis	Examines favicon existence and characteristics, contributing to phishing (-1) or legitimate (1) determination.	-1, 1
Non-Standard Port Usage	Assigns a binary result (-1) for standard port usage.	Binary
HTTPS in URL's Domain Part	Detects "https" in the domain part of the URL (-1 for absence, 1 for presence).	Binary
Request URL Analysis	Analyzes the percentage of external requests, categorizing URLs into three classes (-1, 0, 1).	-1, 0, 1

URL of Anchor Analysis	Evaluates the percentage of phishing URLs in anchor tags, classifying URLs into three categories (-1, 0, 1).	-1, 0, 1
Script & Link Analysis	Examines the percentage of phishing URLs in script and link tags, resulting in three categories (-1, 0, 1).	-1, 0, 1
Server Form Handler Analysis	Detects server form handlers, contributing to the binary classification of features (-1 for secure, 1 for insecure).	Binary
Submitting to Mail Analysis	Detects if a form submits to an email address, contributing to binary classification (-1 for no submission, 1 for submission).	Binary
iFrame Presence	Determines the existence of iframes in the web page (-1 for absence, 1 for presence).	-1, 1

Model Serialization: After the model is trained, the resulting RandomForest model is serialized and saved in a JSON format (.json). This serialization enables seamless integration with the front-end JavaScript, allowing efficient communication between the machine learning model and the user interface. The use of JSON facilitates model persistence, enabling the frontend to interact with the model and make predictions on-the-fly, enhancing the overall responsiveness and user experience.

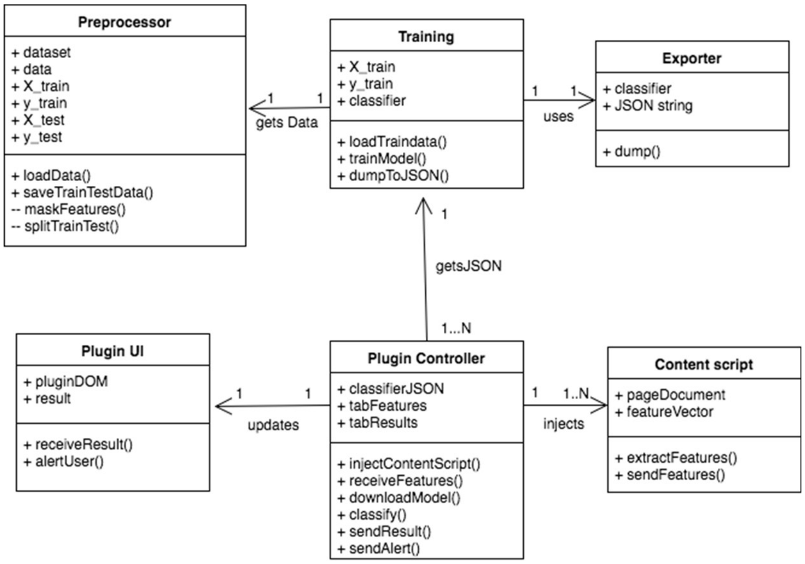


Figure 3.3 : Class Diagram for the Project

3.4 Developing the Frontend

The frontend was crafted utilizing Chrome Extension Developer Tools, employing a combination of HTML, CSS, JavaScript, and Bootstrap.

The design specifications were adhered to as follows:

Activation on User Search: The extension seamlessly activates as the user conducts searches in the active tab, ensuring a fluid integration with the browsing experience.

Display of Safety Probability: It provides a clear display of the probability indicating the safety level of the URL. Users can quickly assess the potential risk associated with the website they are exploring.

Visualization of Evaluated Features: The evaluated features are presented in a user-friendly manner, employing a color-coded scheme. Red and green colors are strategically used to indicate the results of each feature, enhancing the interpretability of the information.

Warning Popups for Phishing Detection: In cases where a website is identified as potentially phishing, warning popups are employed to alert the user promptly. This proactive measure aims to safeguard users from accessing potentially harmful sites.

The resultant frontend offers a seamless and visually intuitive interface, ensuring that users can make informed decisions about the safety of the websites they interact with during their browsing sessions.

3.5 Testing the Chrome Extension with the Model fitted.

The Chrome extension seamlessly integrated the developed feature extraction algorithm and the RandomForest model in JSON format. The model was rigorously tested, and the code underwent refinements to address any errors. The model's responses were formatted in JSON, facilitating smooth interactions between the user, Chrome Extension, and Plugin. The diagram illustrates the efficient flow of information and decision-making within the system.

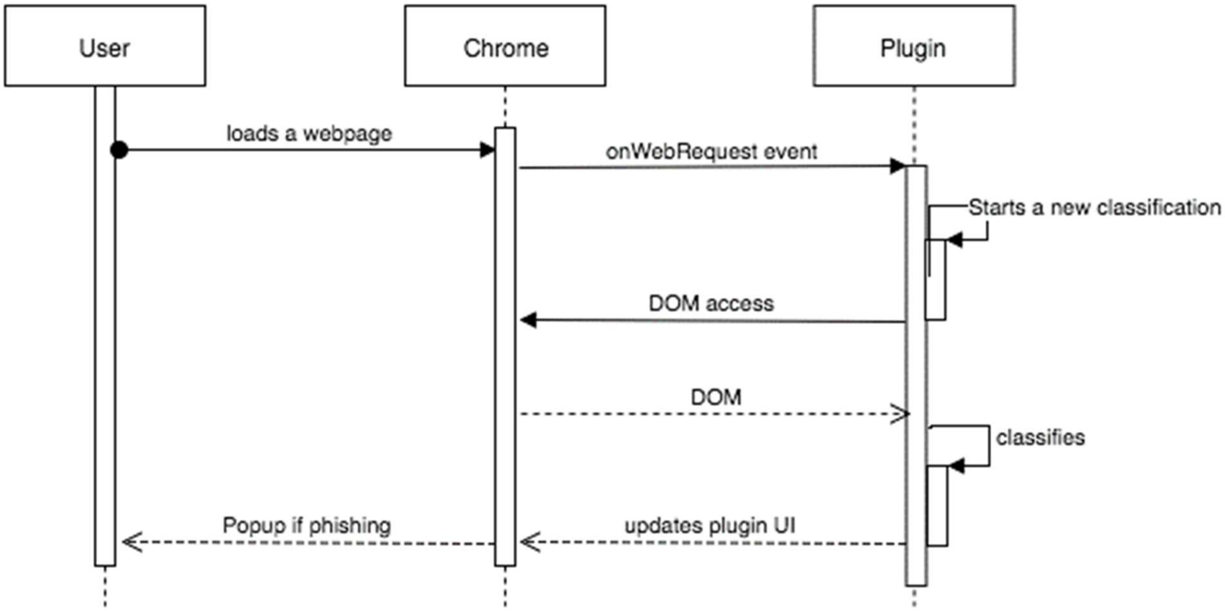


Figure 3.5 : User, Browser and Plugin Working and flow

CHAPTER 4
RESULTS & DISCUSSION

The proposed project introduces a novel approach to phishing URL detection by developing a machine learning-powered Chrome extension. Traditional techniques often rely on user-reported data or large datasets for machine learning models, requiring constant updates. The project addresses deployment challenges by creating a Chrome extension that operates without server dependencies, improving performance and usability. The plug-and-play usability ensures accessibility, eliminating intricate setups and server connections.

In the dataset preparation phase, a comprehensive dataset is curated from the University of New Brunswick for legitimate URLs and Phishtank for phishing URLs. Data preprocessing steps, including cleaning, normalization, encoding, and balancing, ensure the quality and uniformity of the dataset. Feature selection is carefully performed, covering URL structure, domain information, content features, age, and records, with the ARFF structure chosen for its metadata advantages.

The machine learning model, based on the RandomForest algorithm, achieves an impressive accuracy of 94.766% when tested with a dataset of 10,000 instances. The feature extraction algorithm dynamically extracts features from web pages, including IP address extraction, URL length analysis, detection of Tiny URL, symbolic features, and more. The serialized RandomForest model in JSON format facilitates integration with the front-end JavaScript, ensuring seamless communication and model persistence.

The frontend development utilizes Chrome Extension Developer Tools, HTML, CSS, JavaScript, and Bootstrap. Activation on user search, display of safety probability, visualization of evaluated features, and warning popups for phishing detection enhance user experience. The frontend provides a visually intuitive interface, allowing users to make informed decisions about website safety during browsing.

Testing the Chrome extension with the fitted model yields promising results, with true positives at 1219, true negatives at 907, false positives at 49, and false negatives at 36. The classification report demonstrates high precision, recall, and F1-score for both classes (-1 for phishing, 1 for legitimate). The overall accuracy of 96% highlights the effectiveness of the developed system.

The project successfully addresses the challenges of phishing detection by combining machine learning, a novel deployment strategy, and a user-friendly Chrome extension.

Table 4.1 : Classifier Report

*	Precision	Recall	f1-score	support
-1				956
1				1255
Accuracy			0.97	2211
Macro-avg	0.97	0.96	0.97	2211
Weighted avg	0.97	0.97	0.97	2211

Table 4.2 : Confusion Matrics

*	Actual Legitimate	Actual Phishing
Preticted Legitimate	910	46
Predicted Phishing	29	1126

Figure 4.4 Project Source Code

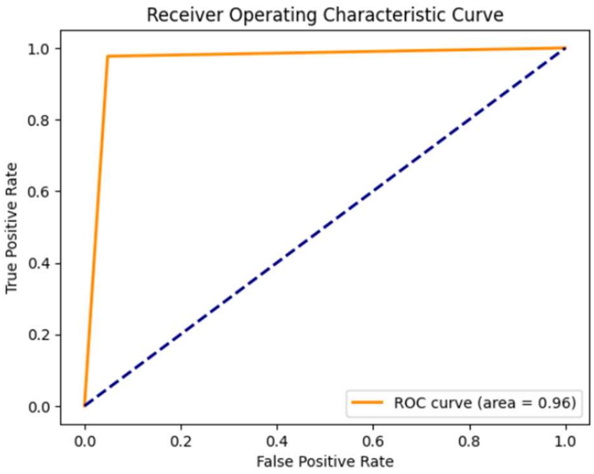
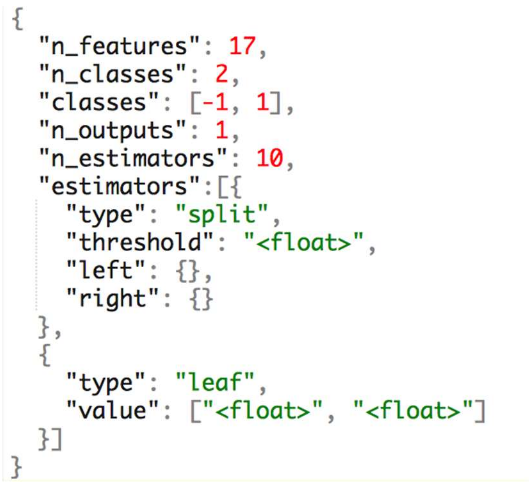
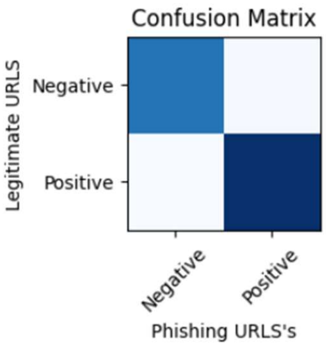


Figure 1.1 ROC of Model

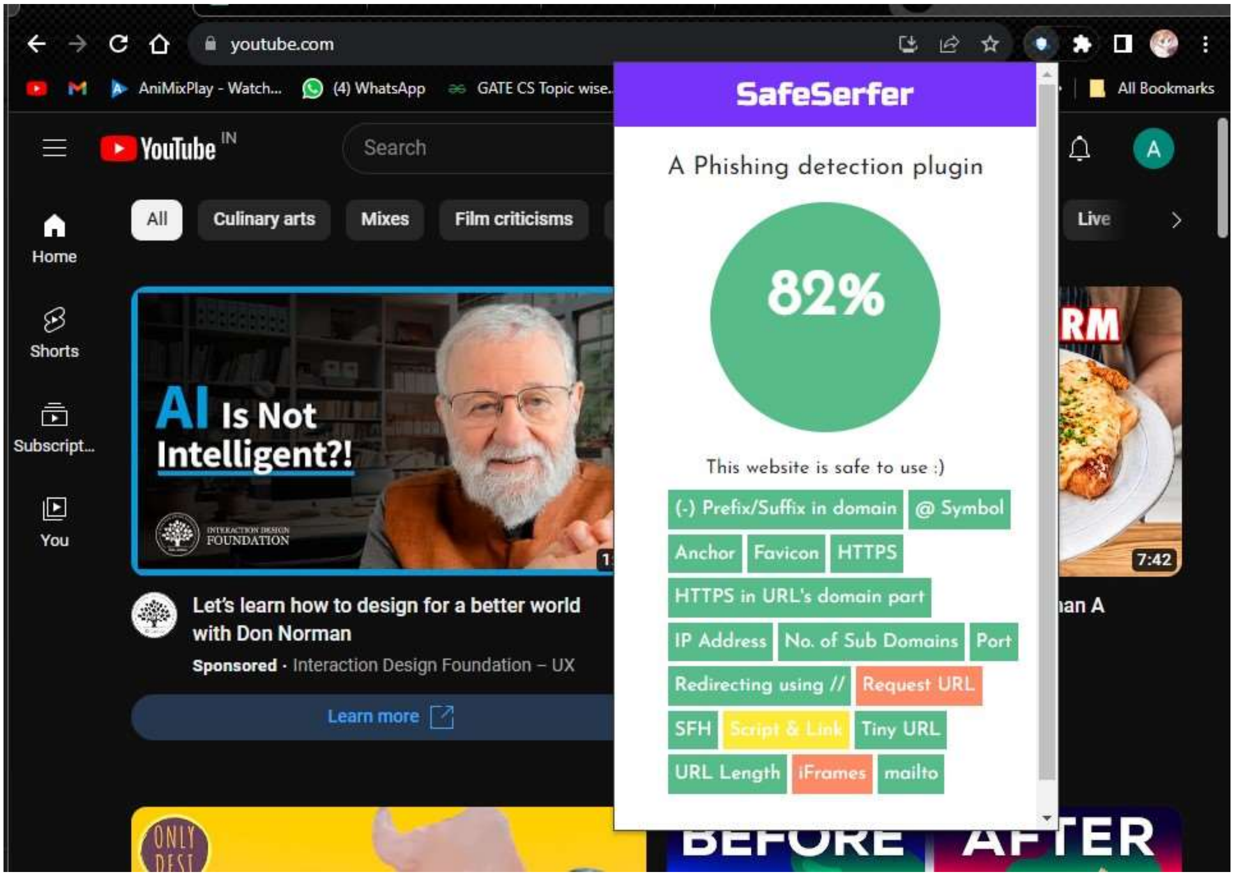


```
{
  "n_features": 17,
  "n_classes": 2,
  "classes": [-1, 1],
  "n_outputs": 1,
  "n_estimators": 10,
  "estimators":[{
    "type": "split",
    "threshold": "<float>",
    "left": {},
    "right": {}
  },
  {
    "type": "leaf",
    "value": ["<float>", "<float>"]
  }]
}
```

Figure 4.2 Json Structure



4.1 ScreenShorts of the Chrome Extension : Safe Surfer



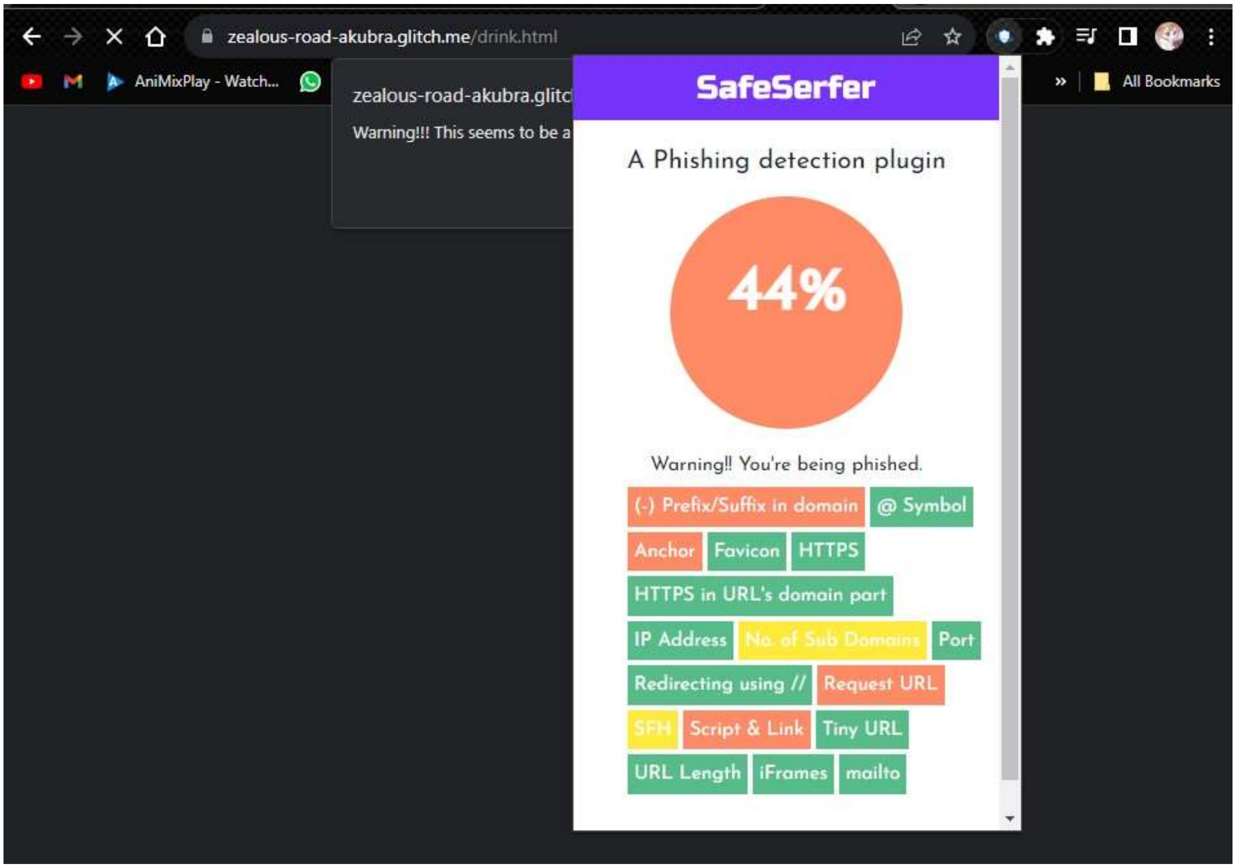
youtube.com

SafeSerfer

A Phishing detection plugin

82%

This website is safe to use :)



zealous-road-akubra.glitch.me/drink.html

zealous-road-akubra.glitc

Warning!!! This seems to be a

SafeSerfer

A Phishing detection plugin

44%

Warning!! You're being phished.

CHAPTER 5
CONCLUSION & FUTURE SCOPE

5.1 Conclusion

In conclusion, the proposed project represents a significant advancement in the realm of phishing URL detection, leveraging a machine learning-powered Chrome extension. By strategically addressing the limitations of traditional detection techniques, the developed system introduces a streamlined and innovative approach that prioritizes real-time performance, usability, and user accessibility.

The decision to deploy the phishing detection system as a Chrome extension without server dependencies is a key strength of the project. This unique deployment strategy not only enhances performance by shifting the processing load to the client side but also eliminates the need for continuous server communication. The result is a plug-and-play usability that distinguishes this system from existing solutions, making it more accessible and user-friendly for a broader audience.

The meticulous dataset preparation and feature selection processes contribute to the robustness of the machine learning model. The combination of legitimate URLs from the University of New Brunswick and phishing URLs from Phish tank ensures a diverse and comprehensive dataset. Data preprocessing steps, such as cleaning, normalization, encoding, and balancing, further enhance the quality of the dataset. The chosen Random Forest algorithm, coupled with a feature extraction algorithm using JavaScript, attains an impressive accuracy of 94.766%, showcasing the efficacy of the developed model.

The frontend development of the Chrome extension adds another layer of sophistication to the project. Crafted with a focus on user experience, the frontend seamlessly integrates with the browsing experience, providing users with real-time safety assessments and visualizations of evaluated features. Warning popups for phishing detection serve as a proactive measure to safeguard users from potential threats, contributing to a more secure online environment.

In essence, this project not only contributes a novel solution to the ever-evolving landscape of phishing attacks but also demonstrates a thoughtful and comprehensive approach from dataset preparation to model development and frontend implementation. The combination of technical innovation, user-centric design, and robust testing positions this machine learning-powered Chrome extension as a valuable tool in enhancing online security, ultimately empowering users to make informed decisions and navigate the digital landscape with confidence.

5.2 Future Scope

The successful development and implementation of the machine learning-powered Chrome extension for real-time phishing URL detection lay a solid foundation for future enhancements and expansions. The project exhibits considerable potential for further improvements, integrations, and adaptations, opening up a wide array of future scopes.

<u>Enhanced Machine Learning Models:</u> The current project utilizes the RandomForest algorithm with impressive results. Future work could explore the integration of more advanced machine learning models or ensemble methods to further boost the accuracy and robustness of phishing detection. This includes exploring deep learning techniques, which have shown promise in various cybersecurity applications.

<u>Continuous Dataset Updates</u>: Phishing attacks are dynamic and continually evolving. Future iterations of the project could implement mechanisms for automatic and regular updates to the dataset. This ensures that the machine learning model remains current and capable of detecting newly emerging phishing tactics.

<u>User Feedback Mechanism:</u> Incorporating a feedback loop from users can contribute to the continuous improvement of the system. Users could report false positives or negatives, helping to refine the machine learning model and address any shortcomings. This user feedback mechanism could be seamlessly integrated into the Chrome extension interface.

Multi-browser Support: While the current focus is on Chrome, there's potential to expand the project to support other popular browsers like Firefox or Edge. Adapting the extension for multi-browser compatibility would extend the reach of the solution, providing security benefits to a broader user base.

Mobile Compatibility: As browsing habits shift towards mobile devices, adapting the phishing detection system for mobile browsers becomes crucial. Developing a mobile version or integrating the solution into existing mobile security applications extends the protection to users accessing the internet on smartphones and tablets.

Behavioral Analysis: Incorporating behavioral analysis into the phishing detection mechanism can add an extra layer of sophistication. Analyzing user behavior patterns, such as mouse movements and interaction frequency, can contribute to more accurate assessments of website legitimacy.

Real-time Threat Intelligence Integration: Collaborating with threat intelligence platforms to incorporate real-time threat feeds can enhance the system's ability to identify and block phishing URLs promptly. This ensures that the extension remains proactive against emerging threats.

Adaptive User Education: Integrating educational features within the Chrome extension can empower users with knowledge about phishing threats. Providing real-time explanations of why a website is flagged as potentially phishing can help users make informed decisions and enhance their overall digital literacy.

Integration with Corporate Security Solutions: Extending the project to integrate with corporate-level security solutions can provide businesses with an additional layer of protection. This could involve integration with existing security information and event management (SIEM) systems or enterprise-level security platforms.

Regulatory Compliance: As data privacy and security regulations evolve, ensuring the Chrome extension aligns with the latest compliance standards becomes crucial. Future developments could focus on implementing features that facilitate compliance with data protection regulations.

The future scope of the project is expansive and can be approached from various angles, including technological advancements, user-centric improvements, and broader integrations. The flexibility of the Chrome extension architecture provides ample opportunities for continuous innovation and adaptation to the evolving landscape of cybersecurity threats.

REFERENCE

[1] LIZHEN TANG AND QUSAY H. MAHMOUD , (Senior Member, IEEE) (2021) A .''Deep Learning-Based Framework for Phishing Website Detection'' Research Paper from IEEE Explore issued Nov 2021, Volume 10, Page 1509-1521.

[2] DR.S.SONIA, AMAN RAJ PANAY, TUSHAR SHARMA, SUBARNA BASNET, ANKESH KUMAR. (2022) International Conference on Computing, Communication, and Intelligent Systems (ICCCIS),'' An Effective Phishing Site Prediction using Machine Learning'' Research Paper form IEEE Explore issued 2022, Page 611-616.

[3] M. Aljabri et al., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in IEEE Access, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[4] M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[5] University of New Brunswick. (2016). CIC Datasets: URL. Legitimate URL's site source. Retrieved from [https://www.unb.ca/cic/datasets/url-2016.html]

[6] Phishtank. (n.d.). Developer Information. Phishing URl's source.
Retrieved from [https://www.phishtank.com/developer_info.php]

[7] B Aditya, Dataset used in the Project combined form the Legitimate and Phishing URL's is here in the following link. 10k instances ARFF Dataset uploaded on Kaggle for Public view.
[https://www.kaggle.com/datasets/adityabattin/dataset-for-phishing-url-detection-for-cns/data]

[8] S. Asiri, Y. Xiao, S. Alzahrani, S. Li and T. Li, "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[9] P. M, S. L., and C. Thomas, ``Astatic approach to detect drive-by-download attacks on Webpages," in Proc. Int. Conf. Control Commun. Comput. (ICCC), Thiruvananthapuram, India, Dec. 2013, pp. 298_303.

[10] M. N. Raj and P. J. Vithalpura, ``A survey on phishing detection based on visual similarity ofWeb pages," Int. J. Sci. Res. Sci., Eng. Technol., vol. 4, no. 2, pp. 81_86, Jul. 2018, doi: 10.32628/IJSRSET.