AI-enabled Phishing Links Detection and Alert System

A Mini-Project Report
Submitted For
Partial Fulfillment of the Requirements of the Degree of
Bachelor of Engineering
In

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
(Semester V)

By

Punit Gavali 9712

Janet Nelson 9717

Shelson Chettiar 9697

Akshat Sarraf 9742

Under the guidance

of

Dr. Jagruti Save

Fr. Conceicao Rodrigues College of Engineering
Bandra (w), Mumbai:400050
University of Mumbai
Dec 2023

CERTIFICATE

This is to certify that the mini-project entitled **“ AI-enabled Phishing Links Detection and Alert System”** is a bonafide work of “ Punit Gavali - 9712, Janet Nelson - 9717 , Akshat Sarraf - 9742 , Sheldon Chettiar - 9697” submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Artificial Intelligence and Data Science** (Semester- V).

(Name and Sign)

Guide/ Supervisor

(Name and sign)	(Name and sign)
Head of Department	Principal

Approval Sheet

Mini Project Report Approval for B.E. (Semester-V)

This mini-project report entitled **AI-enabled Phishing Links Detection and Alert System** submitted by Punit Gavali - 9712, Janet Nelson - 9717 , Akshat Sarraf - 9742 , Sheldon Chettiar - 9697 is approved for the degree of Bachelor of Engineering in **Artificial Intelligence and Data Science** (Semester-V).

Examiners 1.————————————

2.————————————

Date:

Place

:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: Signature

Punit Gavali 9712

Janet Nelson 9717

Sheldon Chettiar 9697

Akshat Sarraf 9742

ABSTRACT

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords, and credit card details (and money), often for malicious reason. It is typically carried out by email spoofing or instant messaging, and it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site, the only difference being the URL of the website in concern.

Communications purporting to be from social web sites, auction sites, banks, online payment processors are often used to lure victims. Phishing emails may contain links to websites that distribute malware. Detecting phishing websites often include lookup in a directory of malicious sites. Since most of the phishing websites are short lived, the directory cannot always keep track of all, including new phishing websites.

So the problem of detecting phishing websites can be solved in a better way by machine learning techniques. Based on a comparison of different ML techniques, the random forest classifier seems to perform better.As Scikit -learn , 2 random forest , plugin Only way for an end user to benefit from this is to implement detection in a browser plugin. So that the user can be warned in real time as he browses a phishing site. However, browser extensions have restrictions such as they can be written only in javascript

Table of Content

Chapter 1
INTRODUCTION

1.1 System Introduction

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, passwords, and credit card details (and money), often for malicious reason. It is typically carried out by email spoofing or instant messaging, and it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site, the only difference being the URL of the website in concern. Communications purporting to be from social web sites, auction sites, banks, online payment processors are often used to lure victims. Phishing emails may contain links to websites that distribute malware. Detecting phishing websites often include lookup in a directory of malicious sites. Since most of the phishing websites are short lived, the directory cannot always keep track of all, including new phishing websites. So the problem of detecting phishing websites can be solved in a better way by machine learning techniques.

Based on a comparison of different ML techniques, the random forest classifier seems to perform better. Only way for an end user to benefit from this is to implement detection in a browser plugin. So that the user can be warned in real time as he browses a phishing site. However, browser extensions have restrictions such as they can be written only in javascript and they have limited access to page URLs and resources.

1.2 Objective

Creating a browser plugin to warn users of phishing websites, without relying on external services, demands an approach that relies on local, heuristics-based detection. The plugin should analyze website URLs and SSL certificates locally, cross-referencing them with a predefined database of known phishing sites, and employ heuristic methods to identify suspicious behavior in real-time. This way, users can receive instant warnings before potentially sharing sensitive information on phishing sites, all while safeguarding their browsing data from external exposure.

1.3 Scope

According to wikipedia, In 2022, 76% of organisations experienced phishing attacks. Nearly half of information security professionals surveyed said that the rate of attacks increased from 2016. In the first half of 2021 businesses and residents of Qatar were hit with more than 93,570 phishing events in a three-month span. With increasing number of internet users, there is a prominent need for security solutions again attacks such as phishing. Hence this plugin would be a good contribution for the chrome users

1.4 Requirements of the System(Features of the System)

FUNCTIONAL REQUIREMENTS

The plugin warns the user when he/she visits a phishing website. The plugin should adhere to the following requirements:

• The plugin should be fast enough to prevent the user from submitting any sensitive information to the phishing website.

• The plugin should not use any external web service or API which can leak user's browsing pattern.

• The plugin should be able to detect newly created phishing websites.

• The plugin should have a mechanism of updating itself to emerging

NON FUNCTIONAL REQUIREMENTS

User Interface There must be a simple and easy to use user interface where the user should be able to quickly identify the phishing website. The input should be automatically taken from the webpage in the current tab and the output should be clearly identifiable. Further the user should be interrupted on the event of phishing.

1.5 Software and Hardware Requirements:-

Hardware

2nd generation Core i5 (2 GHz+), 3rd/4th-generation Core i5 processor, or equivalent operating system you're using (e.g., Windows 10, macOS Catalina, etc.). A computer with at least 4GB of RAM is generally recommended.

Software

- Python for training the model
- Chrome browser

Performance

The plugin should be always available and should make fast detection with low false negatives.

1.6 Advantages and Applications

Advantages:

Privacy Protection: By not relying on external services, the plugin ensures that the user's browsing data remains private and is not shared with external servers, reducing the risk of data leaks or privacy breaches.

Instant Detection: The plugin provides real-time warnings, allowing users to be alerted to potential phishing sites immediately, which is crucial for preventing

Offline Functionality: Users can benefit from phishing protection even when they are offline or in situations with limited internet connectivity, as the plugin operates locally.

Customization: Users can often customize the plugin's settings and define the level of strictness for detecting phishing sites, enhancing the user experience.

Low Latency: Because the detection process occurs locally, there is minimal latency in providing warnings, ensuring a seamless browsing experience.

Applications:

Phishing Prevention: The primary application is to protect users from falling victim to phishing attacks, helping them avoid disclosing sensitive information such as login credentials, credit card details, and personal information to malicious websites.

Online Banking Security: Banks and financial institutions can use this plugin to enhance the security of online banking and financial transactions, ensuring that their customers are not inadvertently directed to phishing sites.

Email Security: Integrated with email clients, the plugin can prevent users from clicking on phishing links embedded in emails, reducing the risk of email-based attacks.

eCommerce Safety: E-commerce platforms can use the plugin to protect customers from fake shopping websites, thereby enhancing trust and online shopping security.

Educational Tools: Educational institutions and organizations can encourage the installation of the plugin among their members to provide additional security awareness and protection from phishing scams.

Chapter 2

LITERATURE REVIEW

2.1 Existing Systems

AI-enabled phishing link detection and alert systems have witnessed significant developments in recent years. Established companies like Microsoft and Google have integrated AI-powered technologies such as Microsoft Defender SmartScreen and Google Safe Browsing into their web browsers, providing real-time warnings to users when potentially malicious websites, including phishing sites, are detected.
Additionally, community-driven platforms like PhishTank contribute to the collective effort by maintaining up-to-date blacklists of known phishing websites. Security solution providers, such as Cofense, Symantec, and Barracuda, have harnessed the power of AI and machine learning to develop anti-phishing tools that offer robust protection against evolving threats. Startups in the cybersecurity space are also actively innovating in this domain, leveraging advanced machine learning models to analyze URLs, email content, and other indicators of phishing attacks.

These systems collectively contribute to the growing arsenal of AI-driven defenses against phishing threats, but it's essential to stay updated on the latest developments to ensure effective protection.

2.2 Research Gap

The research gap in the field of AI-enabled phishing link detection and alert systems lies in the need for further advancements to address several key challenges and limitations:

Real-Time Detection and Alerts: Many existing systems are effective but may not provide real-time protection. Research should focus on reducing the time between the identification of a phishing attempt and the alerting of the user to minimize potential risks.

Enhanced Accuracy: While AI models have improved detection accuracy, there is room for enhancement. Researchers can explore methods to reduce false positives and negatives, making the systems more reliable.

Adaptive Learning: Phishing techniques constantly evolve, requiring adaptive systems. Research can aim to create systems that can quickly adapt to new phishing tactics, thus reducing the window of vulnerability.

User-Friendly Interfaces: Usability and user experience are crucial. Future research should address the design of intuitive and user-friendly interfaces that effectively convey alerts and educate users on potential threats.

Cross-Platform Protection: Phishing threats extend beyond web browsers and email. There is a need for systems that provide comprehensive protection across various platforms and communication channels, including mobile apps and social media.

Privacy-Preserving Solutions: Research should explore methods for protecting user privacy while still effectively identifying and alerting users to phishing threats. This is especially important in light of increasing privacy concerns and regulations.

Multilingual and Regional Considerations: Expanding the scope to detect

2.3 Proposed System

The proposed AI-enabled phishing link detection and alert system aims to revolutionize online security by offering real-time protection against evolving phishing threats. Leveraging cutting-edge machine learning models, it will swiftly analyze URLs, email content, and user behavior to detect potential phishing attempts. This adaptive system will continuously learn and update its algorithms to counter new tactics. With a user-friendly interface, cross-platform compatibility, and a focus on privacy, it will provide comprehensive protection while prioritizing the user experience.

Multilingual support, behavioral analysis, and interoperability with existing solutions will further enhance its capabilities. This system's deployment will be accompanied by thorough impact measurement, ensuring it effectively reduces successful phishing attacks and enhances user security awareness.

2.4 Social relevance and Market Review

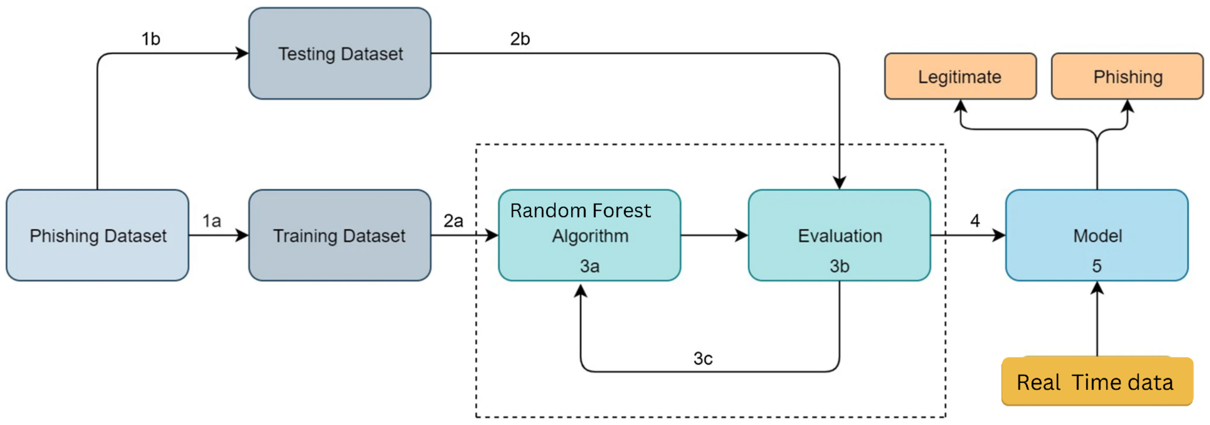
The proposed AI-enabled phishing link detection and alert system is of paramount social relevance as it directly combats the pervasive and ever-evolving threat of phishing attacks. Phishing poses a significant risk to individuals, businesses, and society at large, resulting in financial losses, data breaches, and compromised identities. This system not only safeguards users from falling victim to such attacks but also educates them about the risks of phishing, contributing to a more informed and vigilant online community. Furthermore, it addresses the growing concerns about data privacy by prioritizing the protection of user information. In the broader context, its implementation has the potential to reduce cybercrime and its associated costs, making it a crucial asset in the battle against digital threats.

In terms of market review, the system aligns with the increasing demand for advanced cybersecurity solutions, especially in the face of ever-sophisticated phishing attacks. It enters a competitive landscape where established cybersecurity companies, innovative startups, and tech giants are actively involved in the development of AI-based security solutions. The market's growth is driven by the need for comprehensive defense mechanisms, encompassing individual consumers, enterprises, and government entities. Furthermore, the system responds to the growing regulatory environment, where data protection regulations and compliance requirements are shaping the market's trajectory.

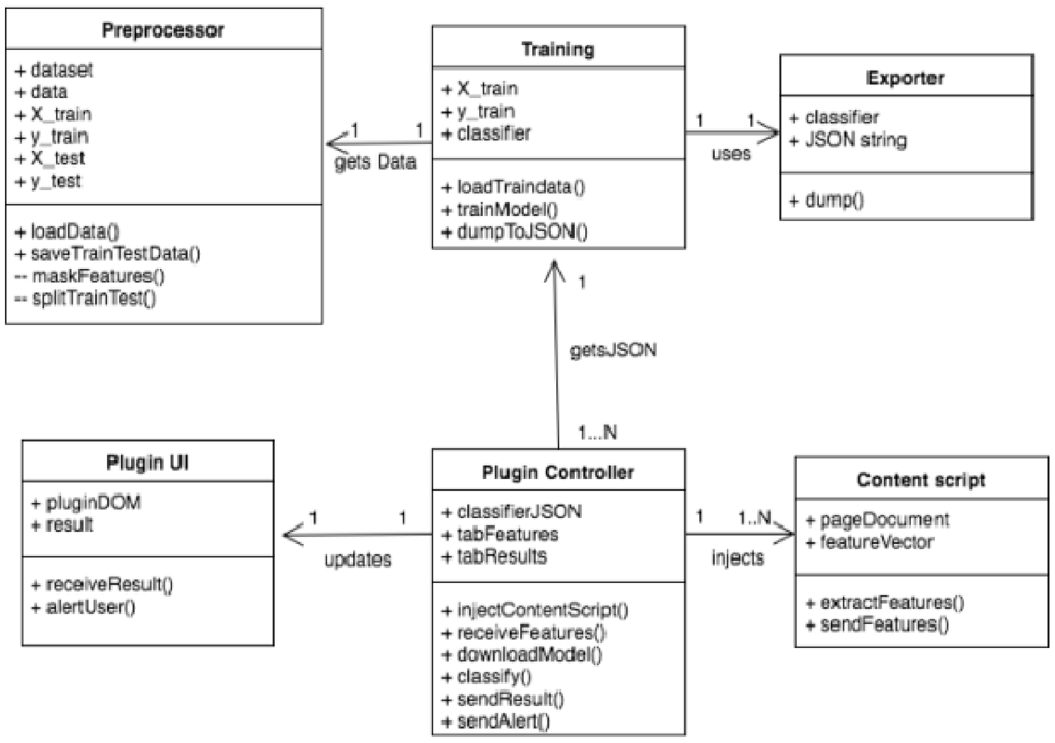
Chapter 3

System Design

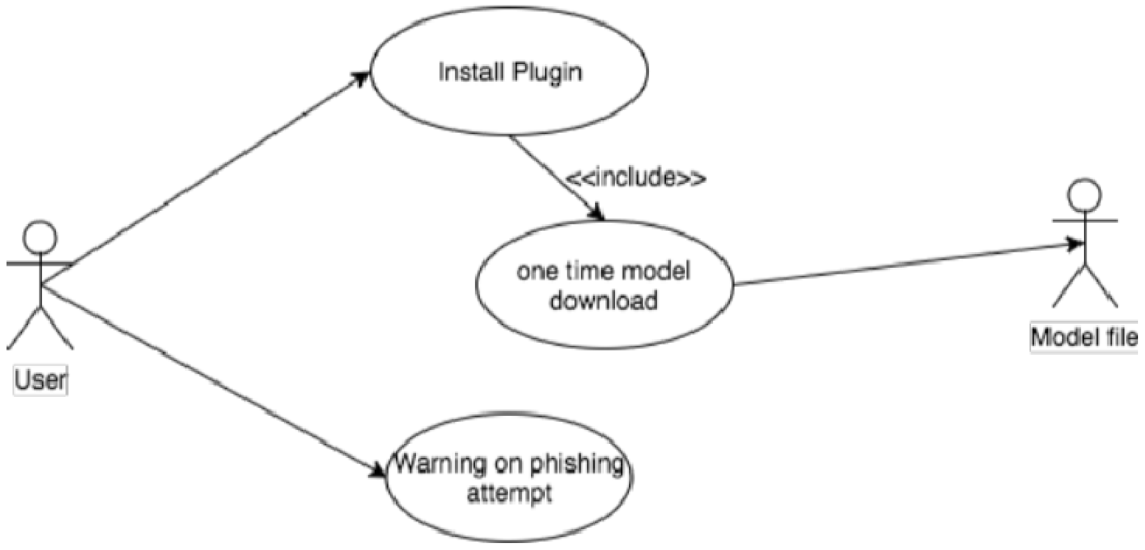
3.1 Block Diagram



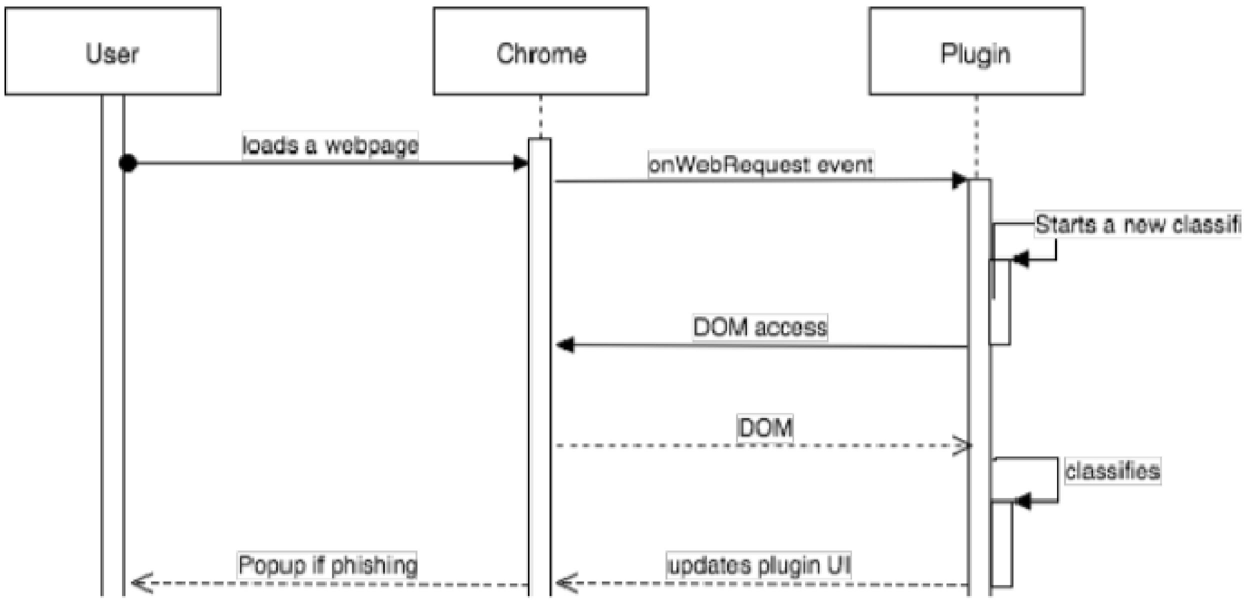
3.2 Class Diagram



3.3 Use Case Diagram



3.4 Sequence Diagram

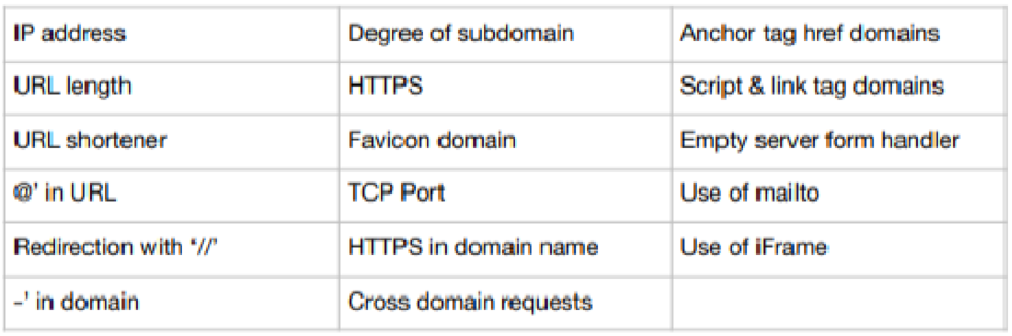


Chapter 4
METHODOLOGIES

4.1 Module Design

Dataset Description :- The dataset was downloaded from UCL Repository :- https://archive.ics.uci.edu/dataset/327/phishing+websites . As it contains of 30 features from which we selected 17 of those such as IP address, URL length , HTTPS , URL shortner , Favicon Domain , @ in URL , TCP port , Redirection with '//' , HTTPS in domain name , '-' in domain , Cross domain request , Anchor tag href domain , script & link tag domain , script & link tag domains , Empty server from handler , Use of mailto, Use of iFrame .

Preprocessing :- The dataset is downloaded from UCI repository and loaded into a numpy array. The dataset consists of 30 features, which needs to be reduced so that they can be extracted on the browser. Each feature is experimented on the browser so that it will be feasible to extract it without using any external web service or third party. Based on the experiments, 17 features have been chosen out of 30 without much loss in the accuracy on the test data. As we selected 17 features of those:-



IP address	Degree of subdomain	Anchor tag href domains
URL length	HTTPS	Script & link tag domains
URL shortener	Favicon domain	Empty server form handler
@' in URL	TCP Port	Use of mailto
Redirection with '//'	HTTPS in domain name	Use of iFrame
-' in domain	Cross domain requests	

Table 4.1.1

Then the dataset is split into training and testing set with 30% for testing. Both the training and testing data are saved to disk.

Training:-

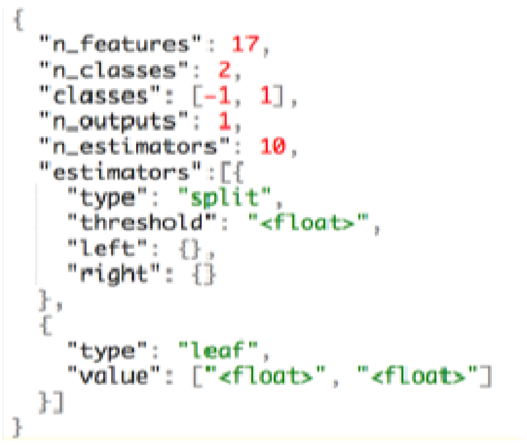
The training data from the preprocessing module is loaded from the disk. A random forest classifier is trained on the data using scikitlearn library. Random Forest is an ensemble learning technique and ensemble of 10 decision tree estimators is used. Each decision tree follows CART algorithm and tries to reduce the gini impurity.

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

The cross validation score is also calculated on the training data. The F1 score is calculated on the testing data. Then the trained model is exported to JSON using the next module.

Exporting Model

Every machine learning algorithm learns its parameter values during the training phase. In Random Forest, each decision tree is an independent learner and each decision tree learns node threshold values and the leaf nodes learn class probabilities. Thus a format needs to be devised to represent the Random Forest in JSON.

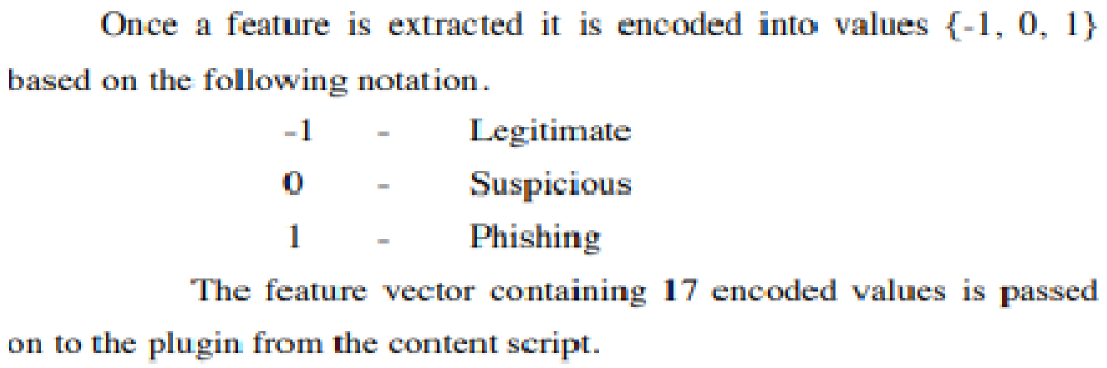


```
{
  "n_features": 17,
  "n_classes": 2,
  "classes": [-1, 1],
  "n_outputs": 1,
  "n_estimators": 10,
  "estimators":[{
    "type": "split",
    "threshold": "<float>",
    "left": {},
    "right": {}
  },
  {
    "type": "leaf",
    "value": ["<float>", "<float>"]
  }]
}
```

4.1.2 Random Forest JSON structure

Plugin Feature Extraction :-

The above mentioned 17 features needs to be extracted and encoded for each webpage in realtime while the page is being loaded. A content script is used so that it can access the DOM of the webpage. The content script is automatically injected into each page while it loads. The content script is responsible to collect the features and then send them to the plugin. The main objective of this work is not to use any external web service and the features needs to be independent of network latency and the extraction should be rapid. All these are made sure while developing techniques for extraction of features.



Once a feature is extracted it is encoded into values {-1, 0, 1} based on the following notation.

-1	-	Legitimate
0	-	Suspicious
1	-	Phishing

The feature vector containing 17 encoded values is passed on to the plugin from the content script.

Classification:-

The feature vector obtained from the content script is ran through the Random Forest for classification. The Random Forest parameters JSON is downloaded and cached in disk. The script tries to load the JSON from disk and incase of cache miss, the JSON is downloaded again. A javascript library has been developed to mimic the Random Forest behaviour using the JSON by comparing feature vector against the threshold of the nodes. The output of the binary classification is based on the leaf node values and the user is warned if the webpage is classified as phishing.

4.2 Exporting Algorithm

The algorithm used to export Random Forest model as JSON is as follows.

TREE_TO_JSON(NODE):

```
1. tree_json ← {}
2. if (node has threshold) then
3. tree_json[“type”] ← “split”
4. tree_json[“threshold”] ← node.threshold
5. tree_json[“left”] ← TREE_TO_JSON(node.left)
6. tree_json[“right”] ← TREE_TO_JSON(node.right)
7. else
8. tree_json[“type”] ← “leaf”
9. tree_json[“values”] ← node.values
10. return tree_json
```

RANDOM_FOREST_TO_JSON(RF):

```
1. forest_json ← {}
2. forest_json['n_features'] ← rf.n_features_
3. forest_json['n_classes'] ← rf.n_classes_
4. forest_json['classes'] ← rf.classes_
5. forest_json['n_outputs'] ← rf.n_outputs_
6. forest_json['n_estimators'] ← rf.n_estimators
7. forest_json['estimators'] ← []
8. e ← rf.estimators
9. for (i ← 0 to rf.n_estimators)
10. forest_json[‘estimators’][i] ← TREE_TO_JSON(e[i])
11. return forest_json
```

Chapter 5

IMPLEMENTATION

5.1 manifest.json

```
{
  "name": "Zphisher",
  "version": "1.0.01",
  "description": "A phishing detector plugin",
  "permissions": ["activeTab","declarativeContent", "storage",
"webNavigation"],
  "background": {
    "service_worker": "js/bg-loader.js"
  },
  "action": {
    "default_popup": "plugin_ui.html"
  },
  "content_scripts":[
    {
      "matches": ["http://*/*","https://*/*"],
      "js": ["js/features.js"]
    }
  ],
  "manifest_version": 3
}
```

5.2 plugin_ui.html

```
<!DOCTYPE>
<html>
   <head>
     <meta charset="utf-8">
     <title>Zphisher</title>
     <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.c
ss"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmY
m81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
```

```
      <link
href="https://fonts.googleapis.com/css?family=Josefin+Sans|Russo+One"
rel="stylesheet">
      <link href="plugin_ui.css" type="text/css" rel="stylesheet">
    </head>
    <body>
      <div id="plugin_name">
        <h1>Zphisher</h1>
      </div>
      <h2 style="font-size: 20px">A Phishing detection plugin</h2>
      <div class="rounded-circle" id="res-circle">
        <h1 id="site_score">93%</h1>
      </div>
      <h2 id="site_msg">This website is safe to use :)</h2>
      <ul id="features">
      </ul>
      <a href="/test.html">View model test results</a>
      <!-- <script src="features.js" type="text/javascript"></script> -->
      <script src="js/jquery.js" type="text/javascript"></script>
      <script src="js/plugin_ui.js" type="text/javascript"></script>
    </body>
  </html>
```

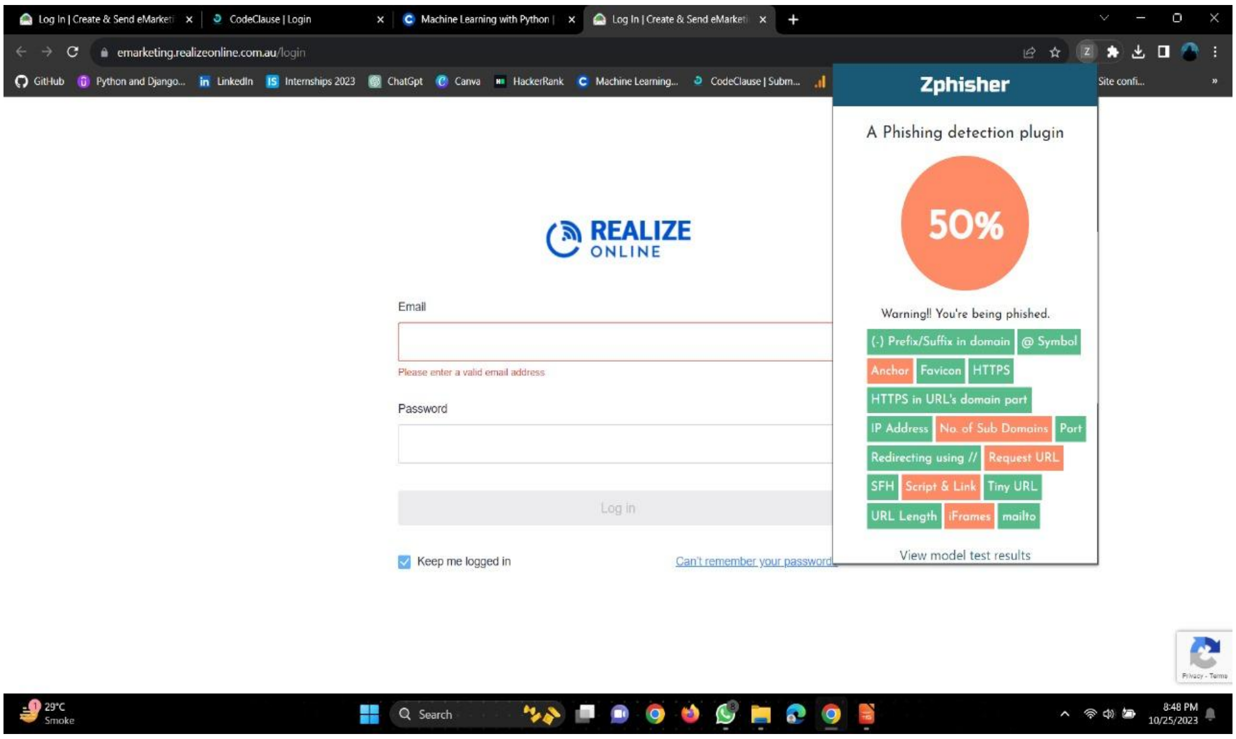
5.3 test.js

```
function test_model() {
$.getJSON("https://raw.githubusercontent.com/picopalette/phishing-detectio
n-plugin/master/static/classifier.json", function(clfdata) {
    var rf = random_forest(clfdata);
$.getJSON("https://raw.githubusercontent.com/picopalette/phishing-detectio
n-plugin/master/static/testdata.json", function(testdata) {
      var X = testdata['X_test'];
      var y = testdata['y_test'];
      for(var x in X) {
        for(var i in x) {
          x[i] = parseInt(x[i]);
        }
      }
      var pred = rf.predict(X);
      var TP = 0, TN = 0, FP = 0, FN = 0;
      for(var i in pred) {
        if(pred[i][0] == true && y[i] == "1") {
          TP++;
        } else if(pred[i][0] == false && y[i] == "1") {
          FN++;
        } else if(pred[i][0] == false && y[i] == "-1") {
          TN++;
        } else if(pred[i][0] == true && y[i] == "-1") {
          FP++;
        }
      }
      var precision = TP/(TP+FP);
      var recall = TP/(TP+FN);
      var f1 = 2 * precision * recall / (precision + recall);
      $('#precision').text(precision);
      $('#recall').text(recall);
      $('#accuracy').text(f1);
    });
  });
}
test_model();
```

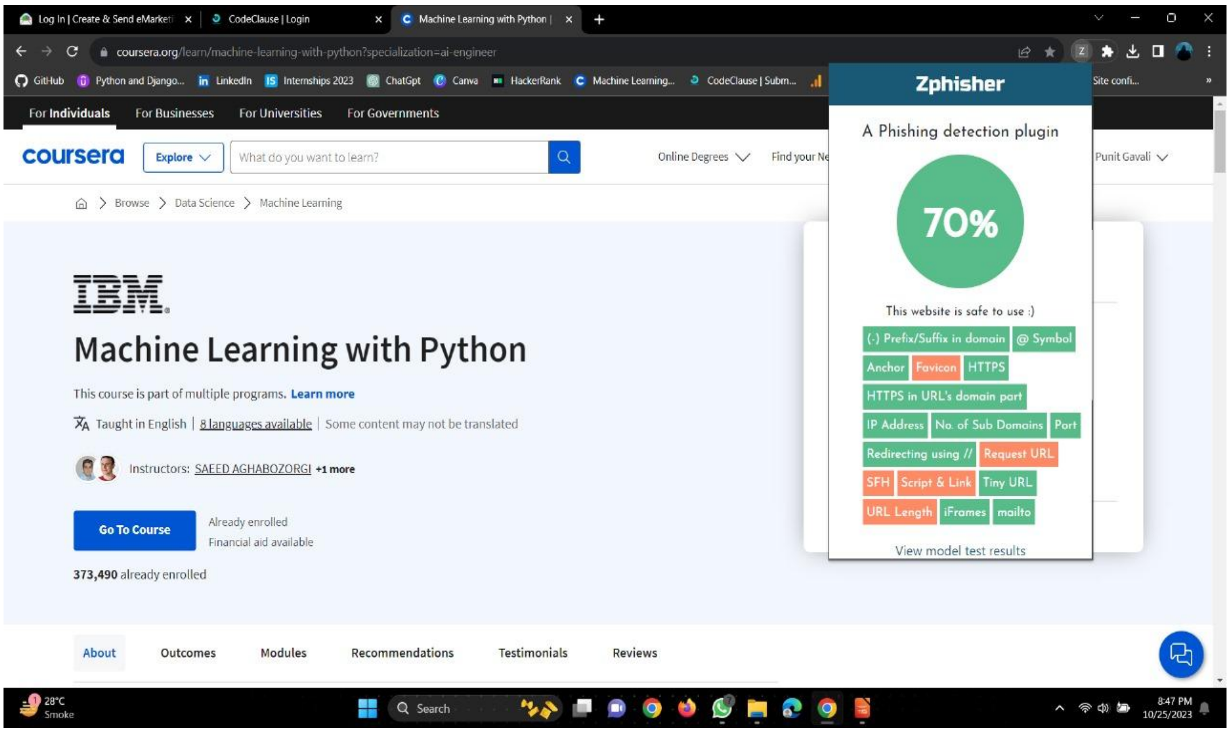
5.4 plugin_ui.js

```
var colors = {
   "-1":"#58bc8a",
   "0":"#ffeb3c",
   "1":"#ff8b66"
};
var featureList = document.getElementById("features");

chrome.tabs.query({ currentWindow: true, active: true }, function(tabs){
   chrome.storage.local.get(['results', 'legitimatePercents', 'isPhish'],
function(items) {
        var result = items.results[tabs[0].id];
        var isPhish = items.isPhish[tabs[0].id];
        var legitimatePercent = items.legitimatePercents[tabs[0].id];

        for(var key in result){
            var newFeature = document.createElement("li");
            //console.log(key);
            newFeature.textContent = key;
            //newFeature.className = "rounded";
            newFeature.style.backgroundColor=colors[result[key]];
            featureList.appendChild(newFeature);
        }

        $("#site_score").text(parseInt(legitimatePercent)+"%");
        if(isPhish) {
            $("#res-circle").css("background", "#ff8b66");
            $("#site_msg").text("Warning!! You're being phished.");
            $("#site_score").text(parseInt(legitimatePercent)-20+"%");
        }
    });

});
```

OUTPUT:-


Log In | Create & Send eMarketi
CodeClause | Login
Machine Learning with Python |
Log In | Create & Send eMarketi
emarketing.realizeonline.com.au/login
REALIZE ONLINE
Email
Please enter a valid email address
Password
Log in
Keep me logged in
Can't remember your password
Zphisher
A Phishing detection plugin
50%
Warning!! You're being phished.
(-) Prefix/Suffix in domain
@ Symbol
Anchor
Favicon
HTTPS
HTTPS in URL's domain part
IP Address
No. of Sub Domains
Port
Redirecting using //
Request URL
SFH
Script & Link
Tiny URL
URL Length
iFrames
mailto
View model test results

5.5


coursera.org/learn/machine-learning-with-python?specialization=ai-engineer
For Individuals
For Businesses
For Universities
For Governments
coursera
Explore
What do you want to learn?
Online Degrees
Browse
Data Science
Machine Learning
IBM
Machine Learning with Python
This course is part of multiple programs. Learn more
Taught in English | 8 languages available | Some content may not be translated
Instructors: SAEED AGHABOZORGI +1 more
Go To Course
Already enrolled
Financial aid available
373,490 already enrolled
About
Outcomes
Modules
Recommendations
Testimonials
Reviews
Zphisher
A Phishing detection plugin
70%
This website is safe to use :)
(-) Prefix/Suffix in domain
@ Symbol
Anchor
Favicon
HTTPS
HTTPS in URL's domain part
IP Address
No. of Sub Domains
Port
Redirecting using //
Request URL
SFH
Script & Link
Tiny URL
URL Length
iFrames
mailto
View model test results

5.6

Chapter 6

CONCLUSION AND FUTURE WORK

This is a phishing website detection system that focuses on client side implementation with rapid detection so that the users will be warned before getting phished. The main implementation is porting of Random Forest classifier to javascript. Similar works often use webpage features that are not feasible to extract on the client side and this results in the detection being dependent on the network. On the other side, this system uses only features that are possible to extract on the client side and thus it is able to provide rapid detection and better privacy. Although using lesser features results in mild drop in accuracy, it increases the usability of the system. This work has identified a subset of webpage feature that can be implemented on the client side without much effect in accuracy.

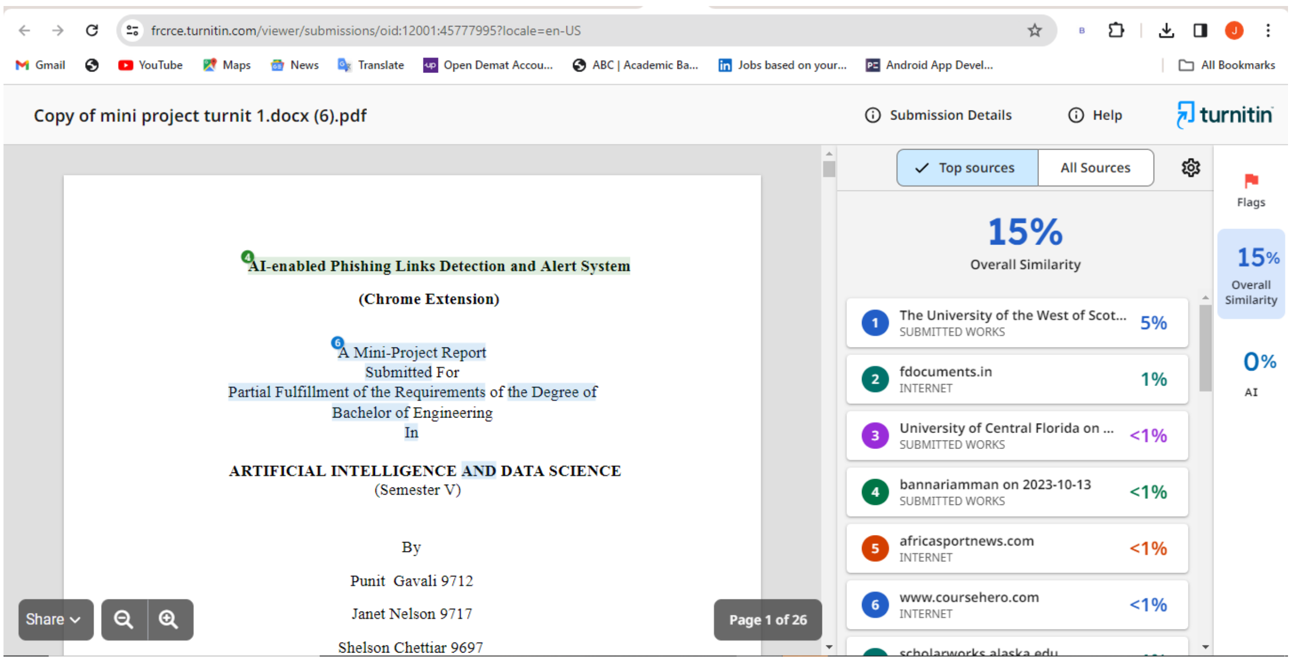
- 94 % Accuracy for random forest classifier.
- Results : Calculated live from the data
  Precision - 0.82176022510471
  Recall   - 0.9631665750412315
  F1 Score - 0.8868640850417616

It acknowledges the evolving regulatory landscape by prioritizing data privacy and security. As the system is developed and deployed, it is positioned to enhance online security, protect user privacy, and reduce the impact of phishing attacks in a digital age marked by evolving threats.

Future Work :

The classifier is currently trained on 17 features which can be increased provided that, they don't make the detection slower or result in loss of privacy. The extension can made to cache results of frequently visited sites and hence reducing computation. But this may result in pharming attack being undetected. A solution needs to be devised for caching of results without losing the ability to detect pharming. The classification in javascript can be done using WorkerThreads which may result in better classification time.

Plagiarism Report



frcre.turnitin.com/viewer/submissions/oid:12001:45777995?locale=en-US

Copy of mini project turnit 1.docx (6).pdf

Submission Details | Help | turnitin

AI-enabled Phishing Links Detection and Alert System

(Chrome Extension)

A Mini-Project Report
Submitted For
Partial Fulfillment of the Requirements of the Degree of
Bachelor of Engineering
In

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
(Semester V)

By

Punit Gavali 9712
Janet Nelson 9717
Shelson Chettiar 9697

Page 1 of 26

Top sources | All Sources

15% Overall Similarity

	Source	Type	Similarity
1	The University of the West of Scot...	SUBMITTED WORKS	5%
2	fdocuments.in	INTERNET	1%
3	University of Central Florida on ...	SUBMITTED WORKS	<1%
4	bannariamman on 2023-10-13	SUBMITTED WORKS	<1%
5	africasportnews.com	INTERNET	<1%
6	www.coursehero.com	INTERNET	<1%

15% Overall Similarity

0% AI

Reference

[1] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, “Intelligent phishing website detection using random forest classifier,” 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Nov. 2020.

[2] “UCI Machine Learning Repository: Phishing Websites Data Set,” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/ phishing websites.

[3] J.-H. Li and S.-D. Wang, “PhishBox: An Approach for Phishing Validation and Detection,” 2019 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2019.

[4] A. A. Ahmed and N. A. Abdullah, “Real time detection of phishing websites,” 2021 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021.

[5] R. Aravindhan, R. Shanmugalakshmi, K. Ramya, and S. C., “Certain investigation on web application security: Phishing detection and phishing target discovery,” 2022 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), 2022

[6] S. B. K.p, “Phishing Detection in Websites Using Neural Networks and Firefly,” International Journal Of Engineering And Computer Science, 2016.
[7] “An Efficient Approaches For Website Phishing Detection Using Supervised Machine Learning Technique,” International Journal of Advance Engineering and Research Development, vol. 2, no. 05, 2019.

[8] S. Gupta and A. Singhal, “Phishing URL detection by using artificial neural network with PSO,” 2019 2nd International Conference on Telecommunication and Networks (TEL-NET), 2019.

[9] Ammar ALmomani, G. B. B, Tat-Chee Wan, Altyeb Altaher, and Selvakumar Manickam, “Phishing Dynamic Evolving Neural Fuzzy Framework for ...,” Jan-2022. [Online]. Available: https://arxiv.org/pdf/1302.0629