**PhishTales**

**CIS 3296 Section 704**

**Spring 2023**

**Team Members:**

Thomas Rau

Ali Shahid

Anthony Roman

Djakaridja Ouedraogo

**Repository URL:**

<https://github.com/cis3296s23/applebaum-projects-phishtales>

**Project Board:**

<https://temple-cis-projects-in-cs.atlassian.net/jira/software/c/projects/PHT/boards/44>

Table of Contents

[**Project Proposal 3**](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.isx3z0dv8cr9)

[Project Abstract 3](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.2mft1lem1f9t)

[Conceptual Design 3](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.fce0euit4oyd)

[Proof of Concept 3](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.prvoay6xhggh)

[Background 4](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.gjfihxjfko6i)

[Required Resources 4](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.w2b81y3ec34i)

[**Project Design 5**](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.stuj01oaba9m)

[Vision 5](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.vn9a9xg59ulk)

[Personas 5](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.qzp7t44pai2k)

[Persona 1: 15-30 User Age Group 5](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.nasobqvmg7bc)

[Persona 2: 30-50 User Age Group 5](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.kwfvylq04u7i)

[Persona 3: 50-65 User Age Group 5](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.v1csk2wrv32o)

[Persona 4: 65-90 User Age Group 6](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.6fr82r1ftw6s)

[Class Diagram 6](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.z9soljfebcgq)

[**Project Progress 7**](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.fybvgrl4owlb)

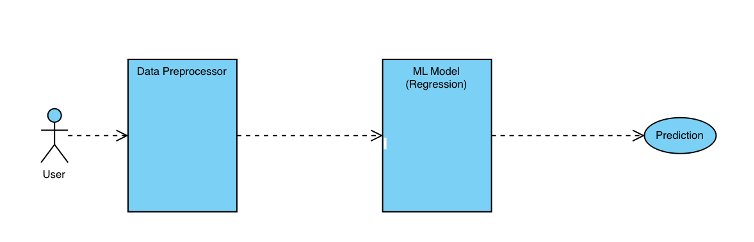
[Week 2 7](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.i75l88bd1uie)

[Week 3 8](https://docs.google.com/document/d/1Df5ctSu7En-RfoHQMPK1E3usQUIcaf1zIXyS-8lEvCg/edit#heading=h.s17g8kbclcr4)

Project Proposal

Project Abstract

This project is designed to be a program and/or integrated web extension that inputs a URL and outputs a probability that the inputted URL is a phishing scam. From a user point of view, all that occurs is a simple URL input, or in the case of a browser extension, a click of a button. The output is the reliability score of the website in question. The project uses machine learning to predict if a website is a phishing scam or not.



Conceptual Design

This project will mainly use Python, as it has many packages and libraries used for data science and machine learning. Models such as K—nearest neighbor, decision trees, naïve bayes, and support vector machines will be used for classification. XGBoost, polyfit, and linregress will be used for regression, with regplot from seaborn will be used to visualize data. These models can be obtained from NumPy, SciPy, pandas, matplotlib, and scikit-learn packages freely available using pip. Python3 will be used for the actual data preprocessing, model training, model testing, an model evaluation. To implement a web extension, HTML, JavaScript, and CSS will be used. This will create a simple interface that detects the current web URL, passes it through the ML model, and outputs a classification and regression value indicating the reliability of the website in question. HTML and CSS will be used to build and style the extension, while JavaScript will simply deal with user interaction.

Proof of Concept

<https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques/>

This project on GitHub employs similar techniques to build a ML model to determine if a website is likely a phishing scam. Similar to what I would like to employ, it uses known phishing URLs and known legitimate URLs to train a ML model (in this case, it employs multiple models and compares their accuracy) and ultimately predict (classify) a URL's reliability. The features extracted from the URLs are similar to what I would like to employ, such as features of the actual URL, features of the domain, and HTML/javascript features. I would like to create my own project rather than further developing this one, because it is trained off of data that is 2+ years old, it also may have out-of-date learning models which could be optimized with state of the art techniques, and it lacks the ability to perform tasks live while in a browser. This project also uses classification, a subset of supervised learning, while I would like to employ regression techniques.

Background

This tool will check URLs for domain squatting-- which is when a common website is registered under a different domain (ex: registering apple.io vs apple.com), URL hijacking-- which takes common website domain typos and registers them to make a website appear to be legitimate (ex: goggle.com vs google.com), also checking other details in the URL such as its length, the number of subdomains in the URL, and the Top-Level Domain (TLD). The project will also check the domain name and its IP address to see if it is blacklisted in any commonly known phishing databases (ex: https:openphish.com/phishing\_database.html). Page-based features will be checked to determine how reliable the website seems, websites (ex: PageRank and AWS) can be used for reference data. Finally, content-based features can parse through the code used to develop the website and detect the reliability of said website. All these features combined will use a decision-tree ML algorithm to create a score that assesses the likelihood that a given URL is a phishing scam or not. While other projects exist that use similar ideologies, this project will employ updated models and will use newly updated data to train these models. With all of the different features used to create a predicted output, this project has potential to be more accurate than similar projects.

Required Resources

Resources required for this project can be obtained with a simple internet connection and the ability to access open-source python libraries. Hardware requirements are quite simple as the model will be built on a well-equipped machine and further testing will not require much computing power. Software requirements are the ability to run python files, chrome browser, and either MacOS, Windows, or Linux.

Project Design

Vision

For people browsing the internet who want to be safe when clicking a link, PhishTales is a website or browser extension that helps determine if a website is phishing.

Unlike Phishing Website Detection by Machine Learning Techniques, our product uses newer website data and machine learning techniques.

Personas

Persona 1: 15-30 User Age Group

Emily is a tech-savvy individual who spends most of her time online. She enjoys browsing social media, streaming content, and playing online games. She is knowledgeable about the latest tech trends and is always up-to-date with the latest gadgets. She wants to protect herself from online threats such as phishing attacks. She is aware of the potential dangers of phishing scams and doesn't want to fall prey to them. She wants to find a reliable phishing detection product that can help her stay safe online.

Emily is a busy college student and doesn't have a lot of spare time to research and evaluate different phishing detection products. She is also concerned about the cost of such products and wants to find an affordable solution. She finds Phishtales.

Persona 2: 30-50 User Age Group

Sarah is a 40 year old, busy stay-at-home mom with two young children. She spends a lot of time managing her household, running errands, and taking care of her kids. She loves social media and buying things online.

She uses her phone, laptop, and tablet for basic web browsing and computing tasks. She does not know much about cyber security practices. Sarah is concerned about the safety of her personal information online and wants to find a reliable phishing detection product that can help protect her and her family from cyber attacks. She wants to ensure that her kids are safe when using the internet.

Persona 3: 50-65 User Age Group

John, a 55-year-old Cyber Security Analyst based in New York City, relies on PhishTales to enhance his ability to detect and mitigate potential security threats. With a background in Computer Science and a focus on Cyber Security, John is passionate about protecting his organization from cyberattacks, such as phishing and malware. PhishTales, an innovative phishing URL detection extension, seamlessly integrates into John's browser and automatically scans links in emails, web pages, and other digital sources. By flagging potentially unsafe URLs, PhishTales empowers John to promptly investigate and neutralize threats, ensuring a secure digital environment for his organization.

Persona 4: 65-90 User Age Group

Eleanor is an elderly 80 year old woman who lives alone and spends most of her time at home. She enjoys reading books and watching television. She has limited experience using the internet and with 21st century technology.

Eleanor wants to stay connected with her family and friends through email and social media. She also wants to be able to manage her bank account securely. She wants to find a reliable phishing detection product that can help her protect herself from online scams and fraud. She does not know what phishing is, and is the most vulnerable to phishing attacks.

Class Diagram

Diagram

Description automatically generated

Figure 1. – This is the UML Class diagram for PhishTales. The beginning of the diagram shows how the machine learning model is built from the dataset. The data is split into a training and testing set, then pre-processed, followed by feature extraction and training/evaluation. This requires the use of a number of python packages. After the model is complete, it is exported using pickle to become a .pkl file. This pickle file is implemented into a python driver program which predicts new outcomes based on a URL input. The driver program also takes input from a user and checks its validity, extracts the features from the URL, and outputs the prediction from the mode. From here, this program is implemented into a flask web application which allows a user to use the model on the web to receive a result.

Project Progress

Week 2

**Sprint Goal:** In this sprint, the goal is to create a website for the users to enter websites. The start of the ML model will also be developed.

**Backlog Features**

* **Website for entering URLs**

|  |  |  |
| --- | --- | --- |
| **Tasks in Sprint** | **Task Status at end of Sprint** | **Assigned To** |
| **Product Vision** | Completed | Thomas Rau |
| **Modify Project Proposal** | Completed | All |
| **Persona Powerpoint** | Completed | Ali Shahid |
| **Start of ML Model** |  | Anthony Roman |
| **Project Report Document - Week 2 Scrum Report** |  |  |

Week 3

**Sprint Goal:** In this sprint, the ML model will be connected to the website. The accuracy of the ML model will also be fine-tuned.

**Backlog Features**

·   User-visible features worked on during this spring

|  |  |  |
| --- | --- | --- |
| **Tasks in Sprint** | **Task Status at end of Sprint** | **Assigned To** |
|  | Not Started / Partially completed / Completed |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Remove all text in yellow before submitting.