User Data Protection using AI Cybersecurity

A Project Report
Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Software Engineering

By
Mirsaeid Abolghasemi
Varun Bhaseen
Gulnara Timokhina

May 2021

APPROVED

Vijay Eranti, Project Advisor

Dan Harkey, Director, MS Software Engineering

Rod Fatoohi, Department Chair

ABSTRACT

User Data Protection using AI Cybersecurity

By

Mirsaeid Abolghasemi, Varun Bhaseen, Gulnara Timokhina

A user's privacy protection is of prime importance nowadays. The majority of user privacy violations happen through phishing attacks. As per the reports published by the "Anti-phishing working group for Q4 2020", it has been observed that the number of phishing sites detected was 637,302 of which 84% now use Secure Sockets Layer (SSL) protection. A phishing attack can be detected from paid or freeware anti-virus software, corporate email phishing detection, and user intelligence or awareness.

The primary goal in a phishing attack is to exploit human weaknesses. The challenge with current phishing detection is the lack of availability of a reliable state-of-art tool which could compensate for these weaknesses. The majority of phishing detection technology is based on a classical approach where the agent (detector) relies on information on which it has been trained. The agent does not factor into a real-time artificial intelligence-based detection which can detect the most recent evolved techniques on which phishing attacks are based.

In this project, we will be proposing an artificial intelligence-based real-time detector that can protect user's privacy details by constantly scanning a web page for malicious scripts, phishing contents, domain authenticity, and logo identifiers. Our approach will be using not only natural language processing but also computer vision to detect the authenticity of the web page and the use of logos or images on that web page. The outcome will be to run a plugin on a browser that can consume minimal resources and can give a quick scan notification about the safety of the web page

Acknowledgments

The authors are deeply indebted to Professor Vijay Eranti for his invaluable comments and assistance in the preparation of this study.

Table of Contents

List of Figures

Chapter 1. Project Overview

1.1. Introduction

Phishing is a security vulnerability that aims to trick unsuspecting users by mixing social engineering and website spoofing techniques into stealing their sensitive details (e.g., password, bank, or financial details). A typical phishing attack's lifecycle begins with the receipt of a fake e-mail, SMS, or instant message from scammers trying to make users think and believe that it comes from a legitimate source. The messages typically use persuasive claims and a link pointing to a fake web page that mimics the legitimate web page of the target brand. If the user enters their credentials, the life cycle of the attack will conclude by submitting confidential information to phishers which can be misused for online fraud or the misuse of private data.

Phishing is one of the oldest and well-known security vulnerability exploitation technique which relies more on human weaknesses rather than a technological weakness as stated by Bozkir et al. [1] and because of the human factor involved it becomes difficult to eliminate it. Although there are many ways in which a phishing website can be detected the traditional approach has been to rely on databases which maintain a list of these websites and can prevent them from opening at users browser but as mentioned by Abdelnabi et al. [5] any zero-day detections are not possible in these traditional approaches. The update of the database takes a considerable amount of time and usually, a certain number of users have already been affected.

Hence to compensate for human weakness Huang et al [2] in their paper have suggested a more artificial intelligence or deep learning-based technique in which the model can more readily be used to identify the malicious URL for the web page. Based on validation the output of the model will classify whether the web page is phishing or not. Whereas Le et al. [3] suggest a more focused and object detection-based approach where they have enhanced certain features and validate the URL based on a certain character and word frequencies.

Here in this paper, we try to combine various techniques for phishing detection and aim to create an AI agent which can assist users by flagging messages ensuring that users are aware of the

current web page status. The AI agent will scan the entire web page to determine if the web page is safe or is intended as phishing.

1.2. Proposed Areas of Study and Academic Contribution

The project scope is bounded on assumption that attackers will try to modify or morph slight changes to existing web pages and use the same to deceit an unsuspecting user into revealing details. The slight changes can be minute to human awareness like changes to Logo, or web page URL. The project hence is focusing on three major aspects with first being validating the authenticity of a logo followed by validating the URL address and finally trying to predict if the web page has a malicious (phishing) intent.

The project is combining all three checking or validation criteria into a single ensemble model. The models are based on deep learning techniques and will be ensembled based on a multitask multi classification techniques. The model will eventually be saved and used in a web browser extension as a knowledge model or AI agent model which will compensate and enable users to be safe from phishing attacks.

Following are the proposed areas of study for this project from academic perspective:

- Use pipelines to create a production level application model
- Use ensemble technique to balance the multitask multi class output
- Use ML operations techniques to create a commercial-ready to market application
- Combining object detection techniques with NLP (Natural Language Processing) techniques.
- Deploying an ensemble model on a web browser application framework optimized for performance and speed without compromising on CPU/GPU usage at runtime.

1.3. Current State of the Art

Phishing as stated earlier and with reports from APWG (Anti Phishing Work Group) is a multi-million-dollar industry and hence there are many traditional and AI based techniques that are

available to use commercially. Here we have listed out few techniques which in similar context to our project and we have inspired few techniques from each of them.

One of the novel techniques we came across was in article “LogoSENSE” [4]. In this paper, the author highlights what are the different underlying features that are present in the dataset and how to carry out feature extraction. Also, it explains techniques like HOG (Histogram of Oriented Gradients) and what role it plays in detecting the phishing website by comparing logos and images. This paper uses machine learning techniques rather than deep learning techniques, but the approach to enhance the features in the dataset is useful and can be reused. The authors of this research paper have created their dataset (LogoSENSE) on which they carry out feature engineering and data preprocessing to identify whether a given page is phishing or not by simply analyzing the logo in the page.

The second paper that we have identified is “Phishing URL Detection via CNN and Attention-Based Hierarchical RNN” [1]. The authors have used CNN and RNN to extract URL texts or characters or words and use them as features to authenticate and verify a website. The authors worked on a dataset that takes a screenshot of a web page and then uses neural networks to extract features and classify the webpages.

The third technique which we identified is URLNet [2] The authors of this paper focus on using lexical features in a URL to identify whether the website is phishing. This is achieved by using a CNN model that can focus on character level and word level embeddings. The authors have built their dataset and carried out feature extraction using position-sensitive bigrams and trigrams. The conclusion for this paper was to build a state-of-the-art model which can detect phishing website by analyzing the URL. We will be using the feature extraction technique and try to combine it with other URL validation techniques listed earlier for the detection of the phishing website

Then there is Malicious Web Content Detection Using Machine Learning [3]. This paper addresses the imbalanced dataset and what approach should be taken for sampling. The paper also addresses

the different dependencies that are there for the creation of applications for data science (Machine learning and Deep learning models).

VisualPhishNet highlights the zero-day phishing website detection approach. What it essentially highlights is that if a phishing website is created today and it is not a part of any phishing website database then how good the model is to detect this fresh new website as a Phishing website [5].

Chapter 2. Project Architecture

2.1. Browser Extension Application Architecture

The figure below provides a high-level architecture for the application. In our application concept, we use two deep learning models based on YOLO and CNN-RNN multi-task multi-label classification. To get better phishing prediction, these algorithms will be applied to the features extracted from web pages like logo, texts, and address bar.

The deep learning models will be used in our application to get these extracted features and then it will classify the webpages to show that a webpage is phishing, suspicious, or legitimate. The application will be packaged as a web browser extension and/or a plugin

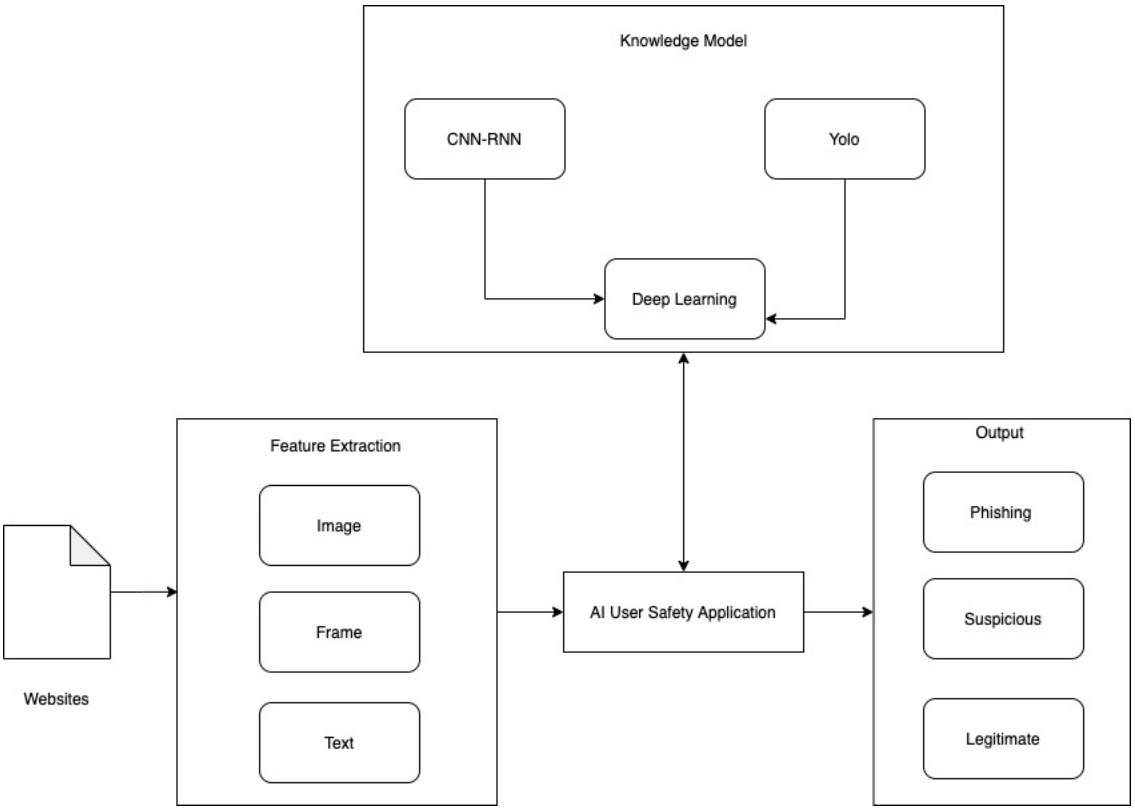

Figure 1: High level application Architecture

2.2. Model Architecture

It is useful to understand the underlying sub architecture of the deep learning based artificial intelligence agent. As seen in figure-2 below we have two different techniques that we will ensemble using a multi-task multi-gate model.

The first technique is YOLO (You Only Look Once), that is a object detection technique which is used validate a logo present in a web page. YOLO scans a web page, locates the logo as an object. Then it calculates a confidence score. We have set a threshold value (currently 65%) for the confidence score to qualify a logo as a legitimate logo.

The second technique is Char-CNN-RNN model (Character Convolutional Neural Network and Recurrent Neural Network). First the URL of a web page is extracted and then Char-CNN-RNN is used to analyze the URL. The URL of a web page is usually morphed by phishers to subdue an unsuspected user and trick them into revealing their information. The model will essentially check to ensure that the URL is a valid URL or if it is phishing URL and will subsequently raise the flag.

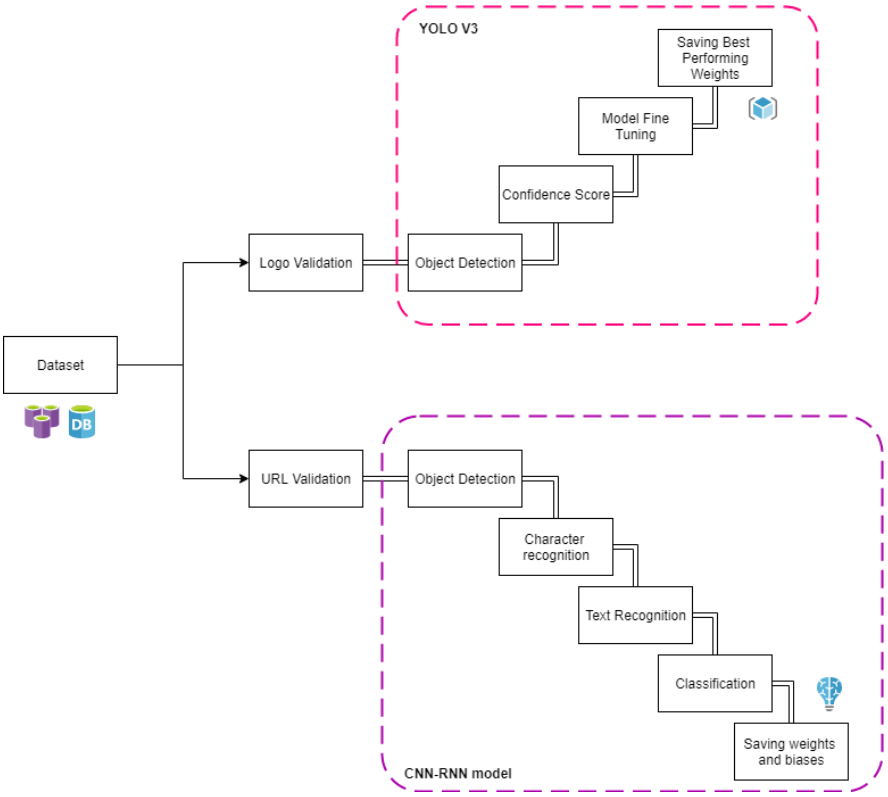

Figure 2: Deep Learning Model Architecture

Chapter 3. Technology Descriptions

3.1. Client-End Plugin

3.1.1. Package Structure

<Add about folder structure, Manifest and Project Structure>

3.1.2. Plugin Styling

<Add about CSS-HTML>

3.1.3. Plugin Libraries

<Add about Tensorflow JS>

<Add about Chrome JS>

3.2. Middleware Hooks

3.2.1. HTTP POST/REST Chrome transmission

3.2.2. HTTP POST Cloud Server

3.2.3. Data Encryption

3.3. Back-end Server

3.3.1. Cloud Service Provider

3.3.2. Web Server Host (Docker-Nginx-UWSGI)

<Add about docker image, docker file>

<Add about Nginx and UWSGI>

3.3.3. Flask Application

<Add about Flask-TensorFlow Server>

Chapter 4. Project Design

4.1. Plugin Application Design

4.1.1. Wireframe Design

4.1.2. UML Class Diagram

4.1.3. UML Sequence Diagram

4.2. AI Cybersecurity Model Design

4.3.1. ML-DL model design

<Add machine learning/deep learning model design>

4.4. Predictor UML Class diagram

Chapter 5. Project Implementation

5.1. Plugin Application Implementation

5.1.1. Key Objectives

5.1.2. Market Innovation

5.1.3. Design Principles and Guidelines

5.2. AI Cybersecurity Model Implementation

5.2.1. Key Objectives

<Add machine learning/deep learning model design>

5.2.2. Ensemble vs Stand-alone vs Cascade modelling

<Add why cascading model was used and what were the experiments done with ensemble and stand-alone models>

Chapter 6. Testing and Verification

6.1. AI Model Validation and Testing

6.1.1. Model Validation Process

6.1.2. Model Testing Process

6.2. Application Testing

6.2.1. Test Plan

6.2.2. Unit Tests

6.2.3. Security Testing

<Include encryption testing as well along with OWASP testing>

Chapter 7. Performance and Benchmarks

Describe any performance and benchmarking criteria you used for your project. In addition, describe any benchmarking results you observed in your project.

Chapter 8. Deployment, Operations, Maintenance

8.1. Standalone Chrome Package Deployment

8.1.1. Package Updates and Releases

8.1.2. Auto-update vs Manual

<Add why manual update was chosen>

8.2. Client-Server Package Deployment

8.2.1. Advantage over standalone deployment

<Zero downtime and efforts for customers>

8.2.2. Ease of AI Model update and deployment

<Summary of current model used and how model can be iteratively improved and updated with newer versions and improved datasets>

Chapter 9. Summary, Conclusions, and Recommendations

9.1. Summary

9.2. Conclusions

9.3. Recommendations for Further Research

Glossary

References

1. Huang, Yongjie, Yang, Qiping, Qin, Jinghui, & Wen, Wushao. (2019). **Phishing URL Detection via CNN and Attention-Based Hierarchical RNN**. *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*

2. Hung Le, Quang Pham, Doyen Sahoo, Steven C.H. Hoi. 2018. **URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection**. *In Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17), 13 pages.*

3. Desai, A., Jatakia, J., Naik, R., & Raul, N. (2017, May**). Malicious web content detection using machine learning**. *In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 1432-1436). IEEE.*

4. Bozkir, Ahmet Selman, & Aydos, Murat. (2020). **LogoSENSE: A companion HOG based logo detection scheme for phishing web page and E-mail brand recognition.** *Computers & Security, 95, 101855.*

5. Abdelnabi, Sahar, Krombholz, Katharina, & Fritz, Mario. (2019). **VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity**. https://arxiv.org/abs/1909.00300

6. Redmon, Joseph, Divvala, Santosh, Girshick, Ross, & Farhadi, Ali. (2016). **You Only Look Once: Unified, Real-Time Object Detection**. *2016, 779-788.*

Appendices

Appendix A.