Computer Engineering Department

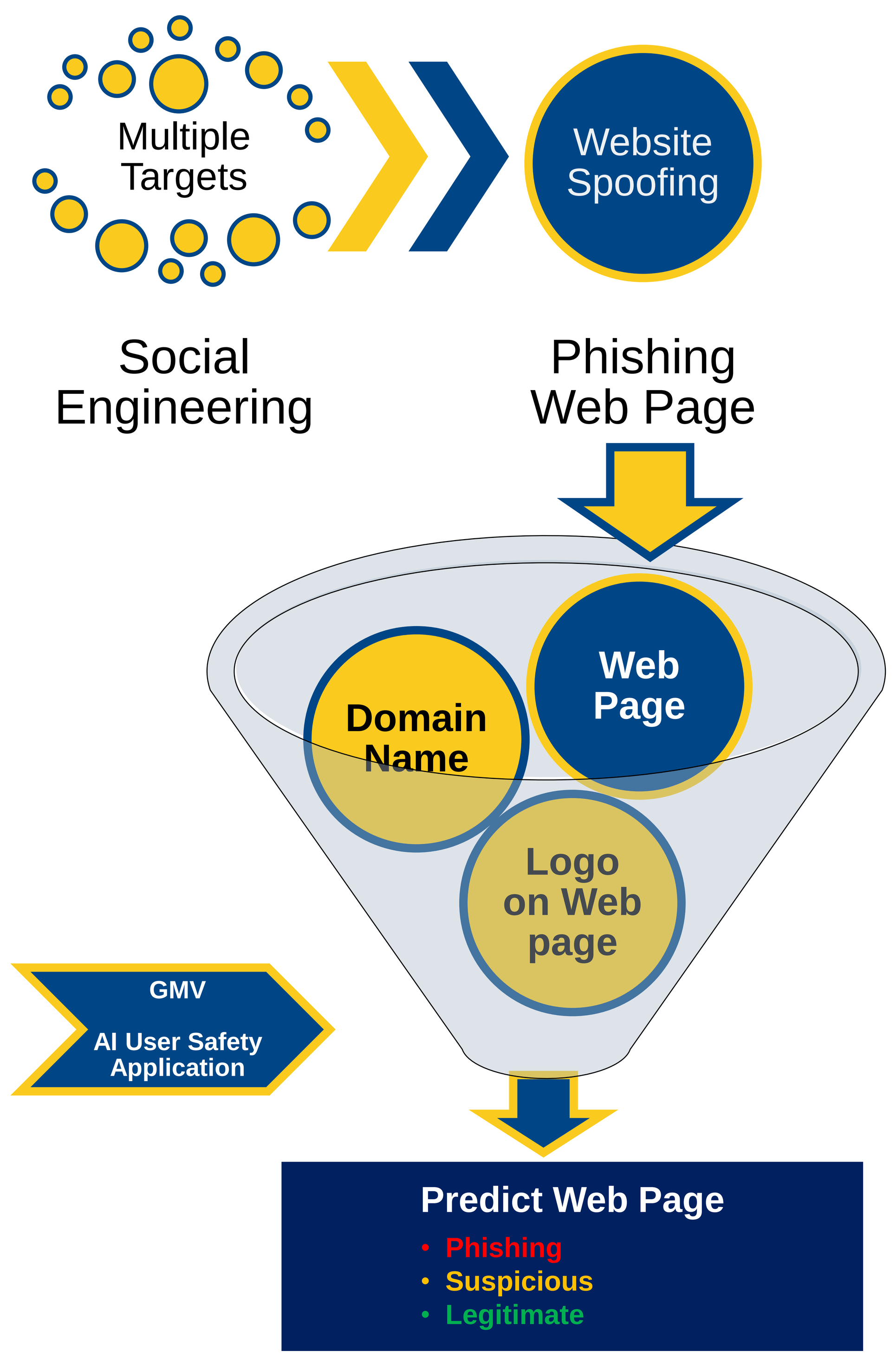
AI User Safety Application

Project Advisor: Vijay Eranti

Abolghasemi, Mirsaeid (MS Software Engineering)
Bhaseen, Varun (MS Computer Engineering)
Timokhina, Gulnara (MS Software Engineering)

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.



Here in this project, we use an ensemble model of Char-CNN and YOLOv3 Object Detection for phishing detection and aim to create an AI agent which users can use to detect the current web page status.

Following are the key objectives that are achieved from the AI User Safety Application:

- Combining object detection techniques YOLOv3 with NLP (Natural Language Processing) using Char-CNN.
- Deploying an ensemble model on a web browser application framework optimized for performance and speed without compromising on CPU/GPU usage and memory footprint at runtime.

Methodology

Model 1: YOLOv3 based Logo Detection Model

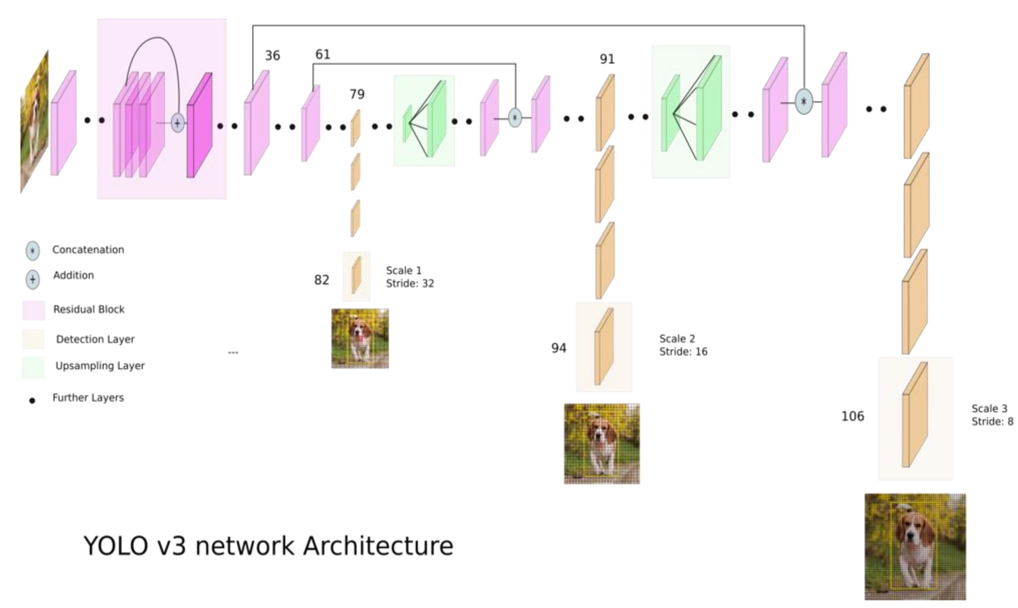


Figure 35: YOLO v3 Model Architecture. Source: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

The YOLO model in this project detects one or more logos in a webpage as an object. The web pages are provided as an image that is base64 encoded string which is decoded at the backend into an image for input. If the logo provided is not there in the trained class, then the model returns an empty string. If the logo provided is in existence, then the model returns the Class Name and Confidence Score, along with the image with bounding boxes.

Model 2: Char-CNN based URL Validation Model

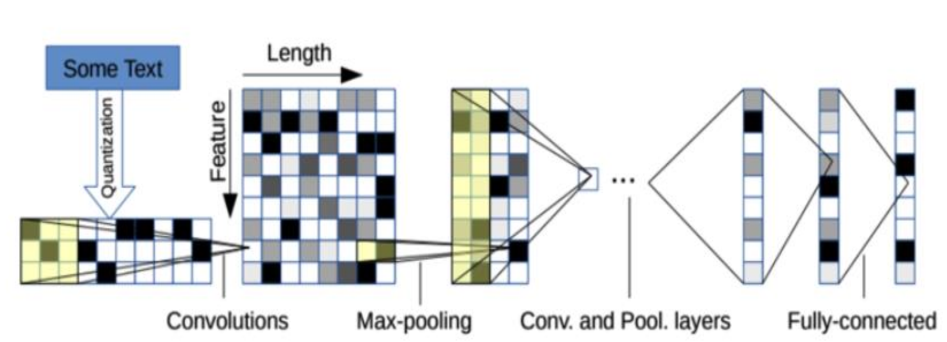


Figure 39: Char-CNN Model Architecture diagram: Source: Character level CNN with Keras. In this notebook, we will build a... | by Xu LIANG | Towards Data Science

The Char-CNN model in this project is a character recognition model which is used to recognize different characters of the URL individually. The backbone of the model is a CNN (Convolutional Neural Network). The model is a classification model which takes the strings as input and carries out a scan to identify potential anomalies and phishing patterns and then determine the output as a binary classification (1 – Phishing, 0 – Legitimate) with Sigmoid activation giving output between 0.0 and 1.0. The output can be interpreted as the probability of the "Phishing" class that is encoded as 1. Assuming the threshold is 0.5 if the output is greater than or equal to 0.5 the predicted class is 1 (Phishing), and if the output is less than 0.5 the predicted class is 0 (Legitimate).

Ensemble Model for Phishing Detection

A unified output was achieved with the ensemble of both models using logical operators to give a single output of **Phishing**, **Suspicious** and **Legitimate**.

Analysis and Results

Individual Deep Learning Model Performance

The key metrics used for each model can be seen from below:

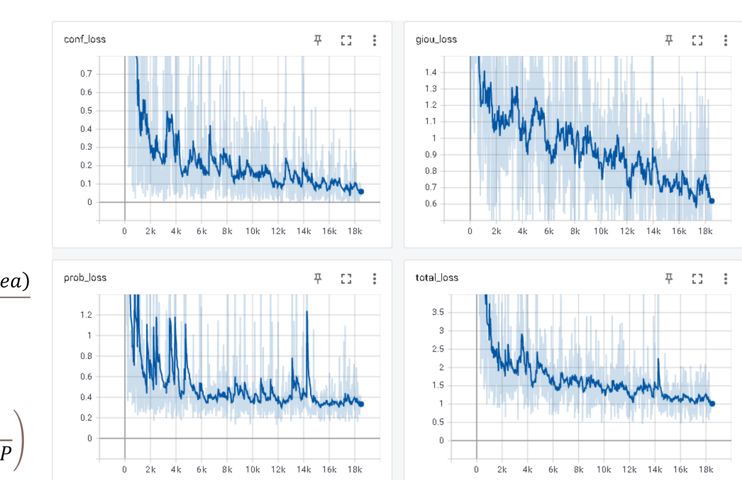
YOLO

- GIoU

$$GIoU = iou - 1.0 \times \frac{(enclosed\ area - union\ area)}{enclosed\ area}$$

- mAP

$$mAP = \mu\left(\sum \frac{TP}{TP + FP}\right)$$



Char-CNN

- Accuracy

$$Accuracy = \frac{\sum TP + TN}{\sum TP + TN + FP + FN}$$

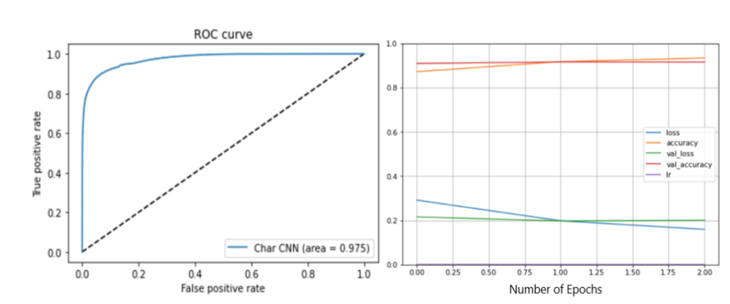
- Precision

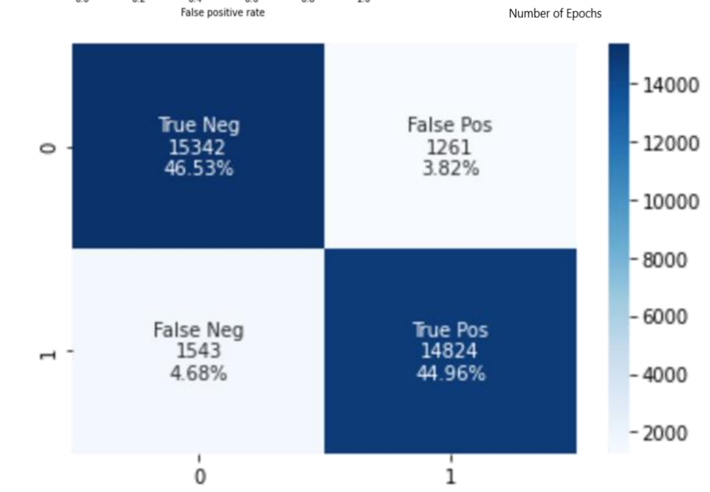
$$Precision = \frac{\sum TP}{\sum TP + FP}$$

- Recall

$$Recall = \frac{\sum TP}{\sum TP + FN}$$

- The accuracy of model as observed from truth table is 91.2%



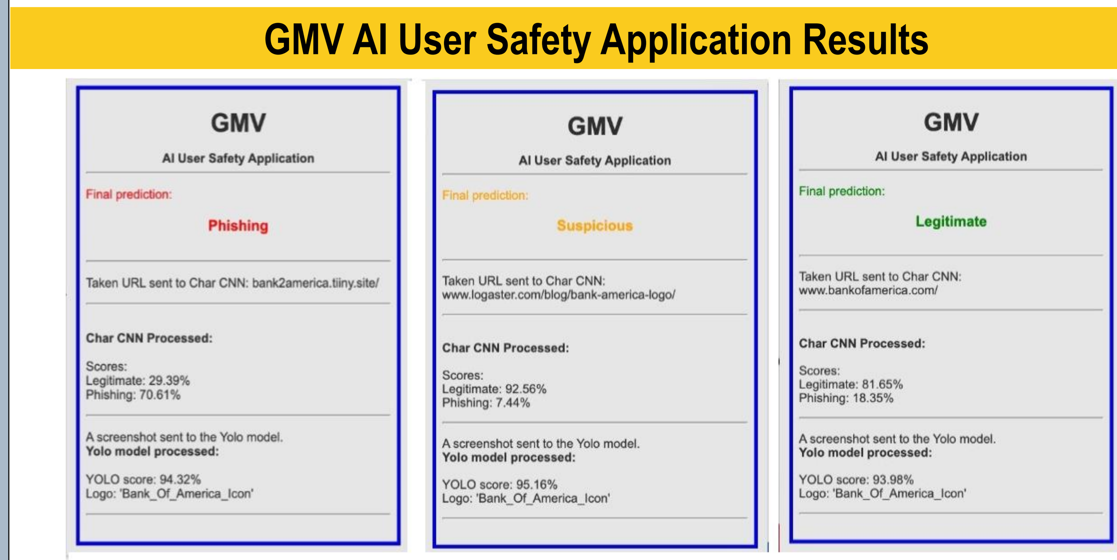


	precision	recall	f1-score	support
0	0.909	0.924	0.916	16603
1	0.922	0.906	0.914	16367
accuracy			0.915	32970
macro avg	0.915	0.915	0.915	32970
weighted avg	0.915	0.915	0.915	32970

The YOLOv3 and Char-CNN model is trained and evaluated over a custom dataset.

- The overall mAP score for the YOLO model is 90 (out of 100) and can accurately predict the objects as a logo.
- The ROC curve for the Char-CNN model is 97.5% which means the model is not overfitting

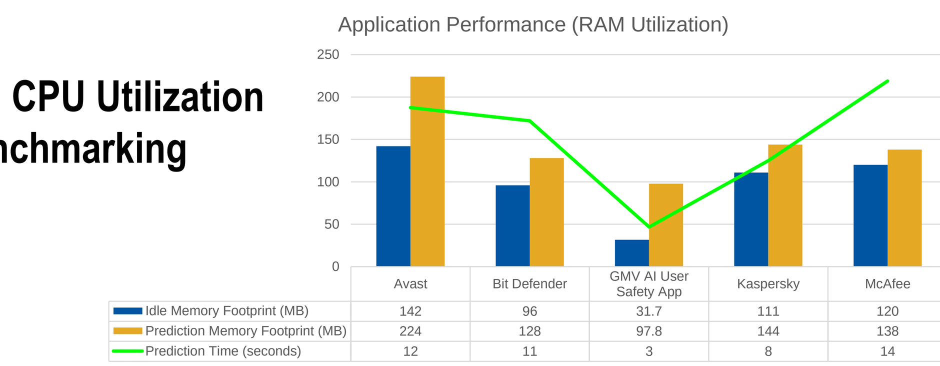
GMV AI User Safety Application Results



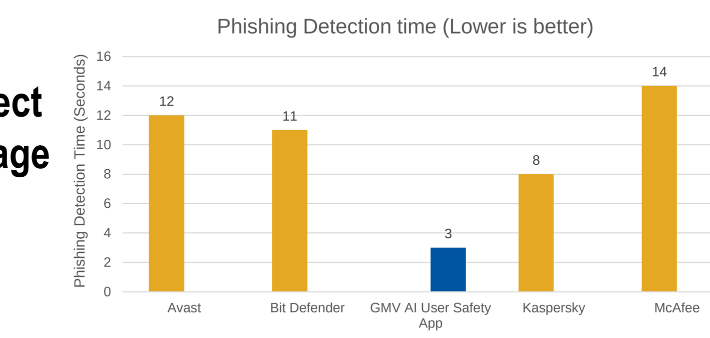
AI User Safety Application Benchmarking

One of the most **primary objectives** of the application is to ensure that a faster, efficient, and accurate phishing detection can be achieved without compromising on the user's resources following are the three criteria around which the entire application is benchmarked:

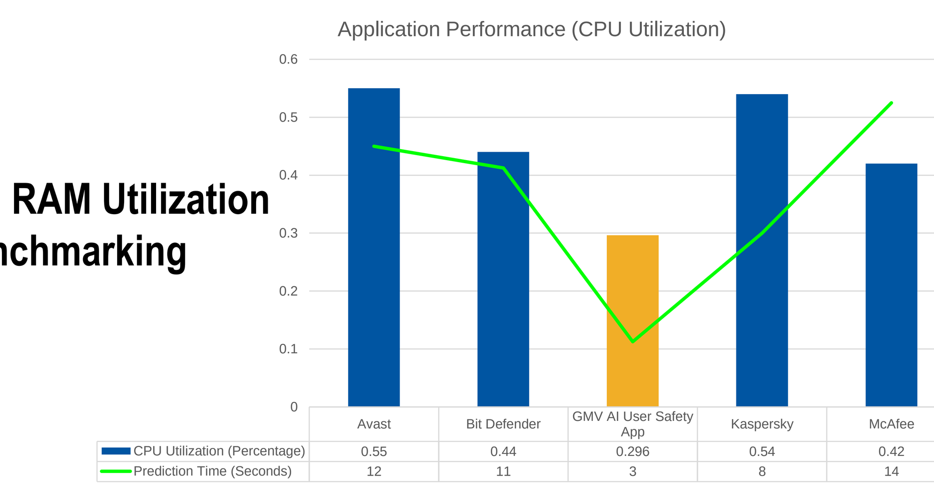
1. CPU Utilization Benchmarking



2. Time to Detect Phishing web page



3. RAM Utilization Benchmarking



Summary/Conclusions

The key objective of the AI User Safety Application was to demonstrate an application that can detect phishing in a faster, efficient, and accurate manner on a browser on a real-time basis. The approach that was taken by us to validate a URL string and validate the logo of the page, has been successful.

The overall idle memory footprint of the application is at least 60% less than that of commercial solutions and a maximum of 22% increase in greater accuracy in identifying phishing websites.

Acknowledgements

We would like to extend our sincere thanks to Professor Vijay Eranti who guided us towards a solution that could help us achieve our key objectives.

We would also like to extend our gratitude towards our family members who supported us throughout the process, our classmates, and dear friends who will forever be part of all the great moments we had in our brief time.

Lastly, we would like to thank SJSU staff, faculty members, infrastructure which enabled us to realize and learn as we moved along.

Key References

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
2. Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sep. 2014.
3. M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model Cards for Model Reporting," *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.