Surefire: A Framework to Detect Phishing Websites

Team02 - Daniel Wang, Gaurav Pawaskar, Rohit Pitke(rpitke3@gatech.edu), Surabhi Chembra

Summary

New phishing techniques evolve rapidly in response to new phishing detection methods. In contrast, existing phishing detection technologies can quickly become outdated because they are static, failing to adjust parameters or improve their success rate over time.

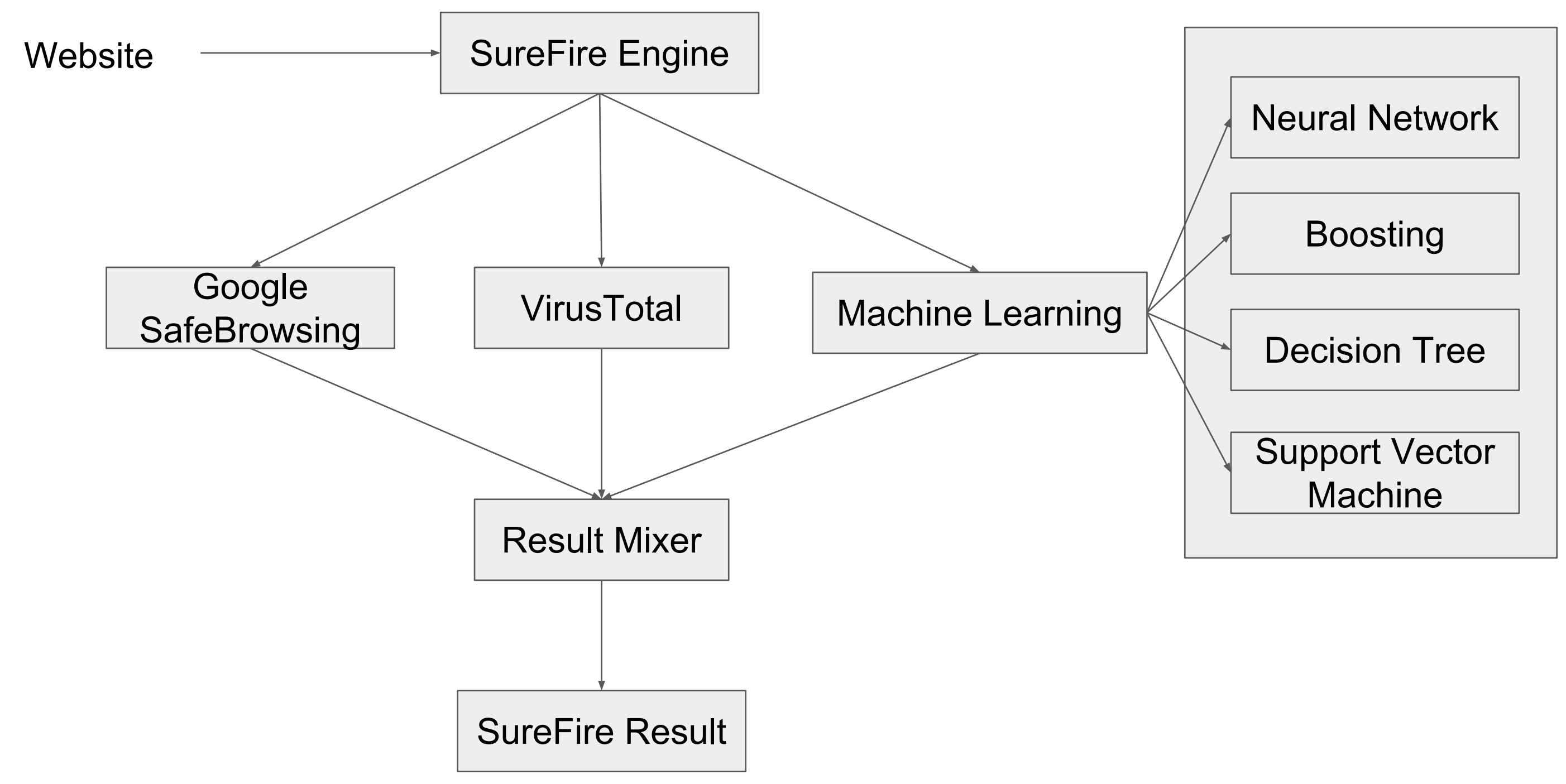
Phishing detection methods like Machine Learning, Google Safebrowsing or traditional signature based methods like VirusTotal are inadequate when used in isolation. We ran numerous experiments and found evidence of the same conclusion. When used in combination, these methods have a much higher chance to improve detection and provide safety to user's internet experience. These methods in conjunction found to be cancelling weaknesses of individual methods and provide reliable and accurate information.

Intuition

Based on our analysis, phishing detection methods in isolation have accuracy of around 80%. Individual methods like machine learning, crowdsourcing and signature based detection suffer due to inherent bias and variance present in features of world wide web. Combination of these methods cancel out these weaknesses and able to identify phishing websites with accuracy upto 98%.

Phishing website type	Detection by ML models	Detection by crowdsourcing like Google	Detection by signature based methods like Virustotal
Subdomain takeover websites(e.g An attacker claims the domain pointed by blog.example.com and host phishing webpage)	Low Accuracy (Website is similar as trustworthy site, classifiers often fail)	High Accuracy (Human intervention, DNS labelling feedbacks help)	Low Accuracy (Website is similar as trustworthy site, same signatures as good site)
Phishing websites on convoluted domains (e.g linked-in.com/example)	High accuracy due to fine grained features extraction	Low accuracy (heavily biased to high traffic sites)	Moderate accuracy as signatures may or may not be similar to spooky website
Phishing website hosting on popular domains like Google site, AWS S3, Azure	Low-Moderate accuracy, depends on classifiers	Moderate accuracy based on DNS labelling, user traffic	High accuracy as signatures like SSL certificate, image tags can match with spooky signatures

SureFire's Approach



Dataset

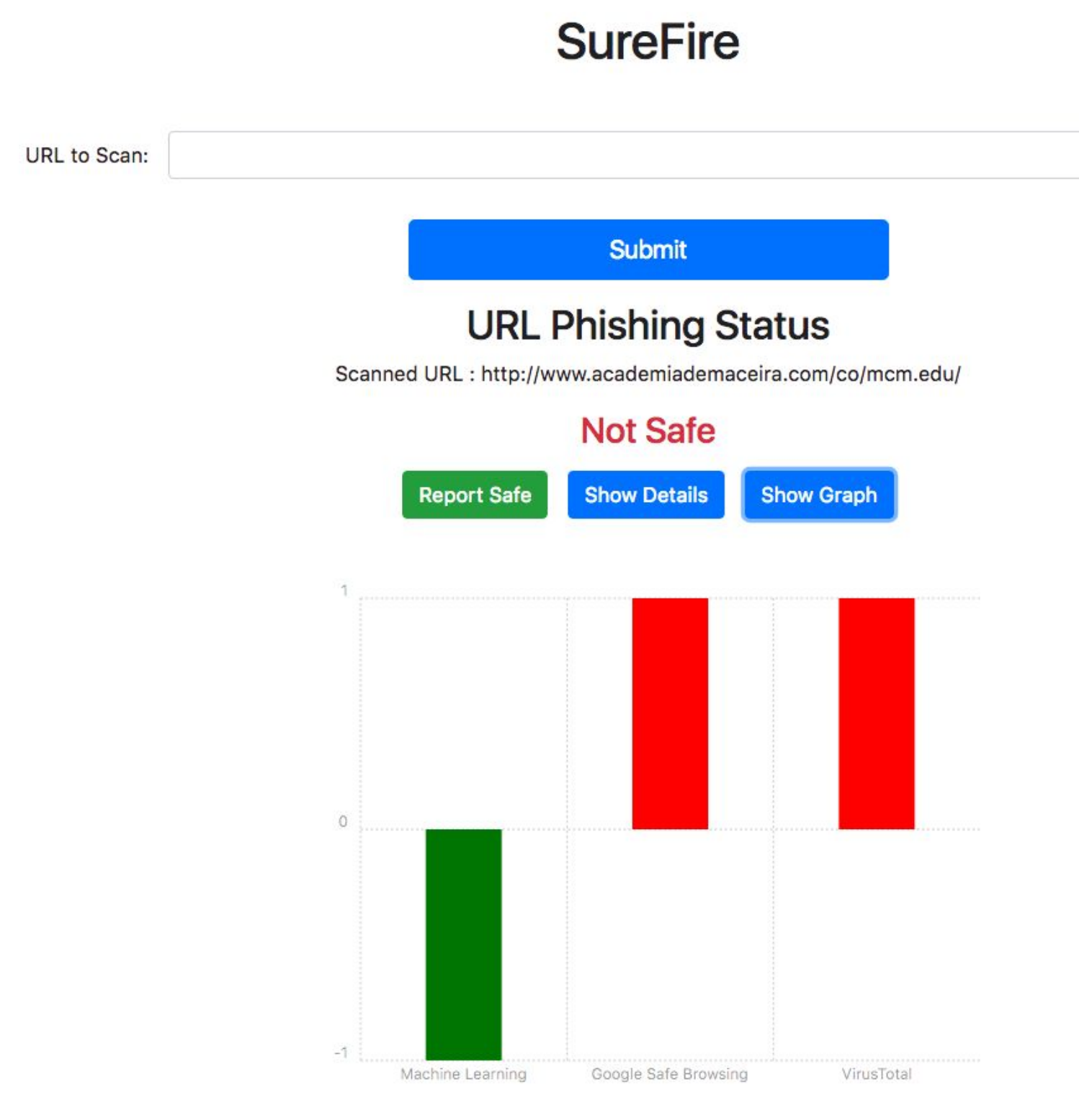
We used two data sources for downloading urls of websites. Combination of sites from phishtank.com and list of urls provided by Alexa of most popular websites is used. This helped us to get 100k+ records for processing.

Features extraction for all urls obtained was performed by custom python scripts. We used python libraries like requests, urllib, whois for extracting features.

For some features we used Alexa, Google Safe Browsing and VirusTotal APIs.

Total Number of records	100035
Phishing sites	27049
Clean sites	75637
Features extracted	8

SureFire Visualization

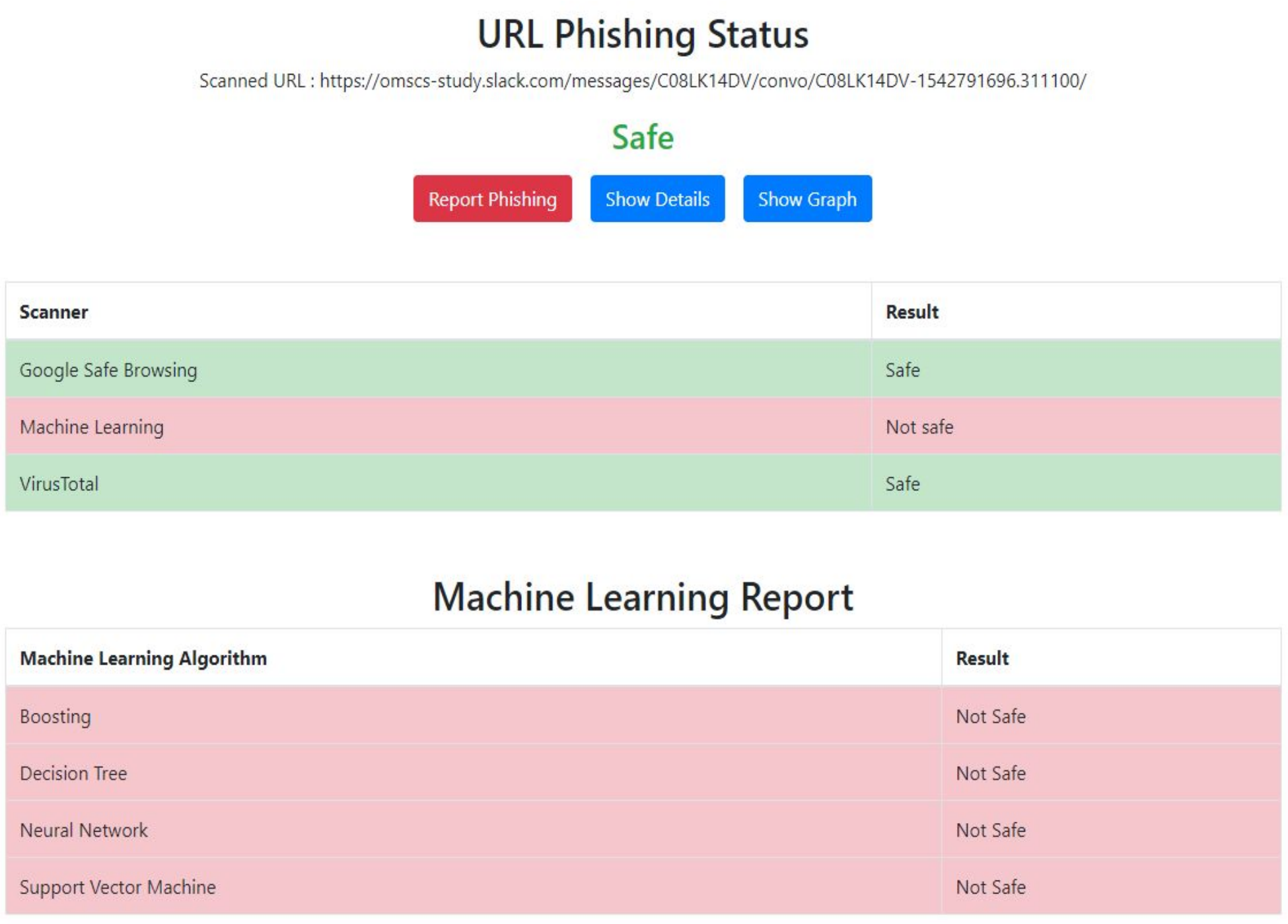

SureFire

URL to Scan:

URL Phishing Status

Scanned URL : http://www.academiademaceira.com/co/mcm.edu/

Not Safe


URL Phishing Status

Scanned URL : https://omscs-study.slack.com/messages/C08LK14DV/convo/C08LK14DV-1542791696.311100/

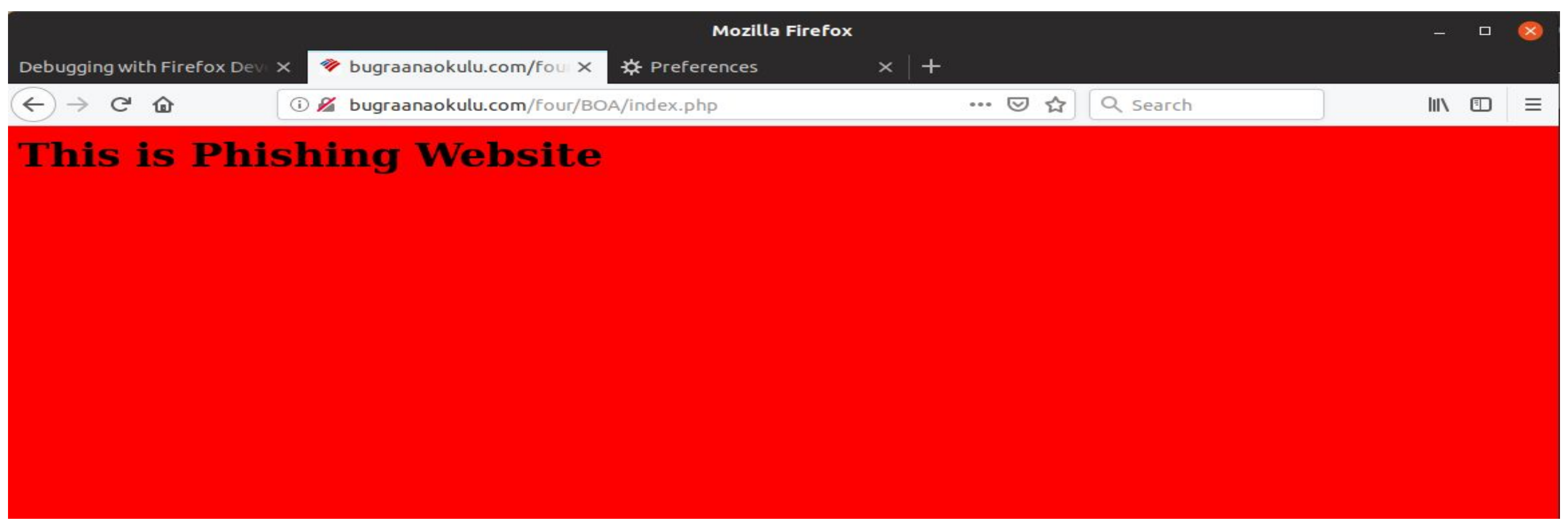
Safe

Scanner	Result
Google Safe Browsing	Safe
Machine Learning	Not safe
VirusTotal	Safe

Machine Learning Report

Machine Learning Algorithm	Result
Boosting	Not Safe
Decision Tree	Not Safe
Neural Network	Not Safe
Support Vector Machine	Not Safe

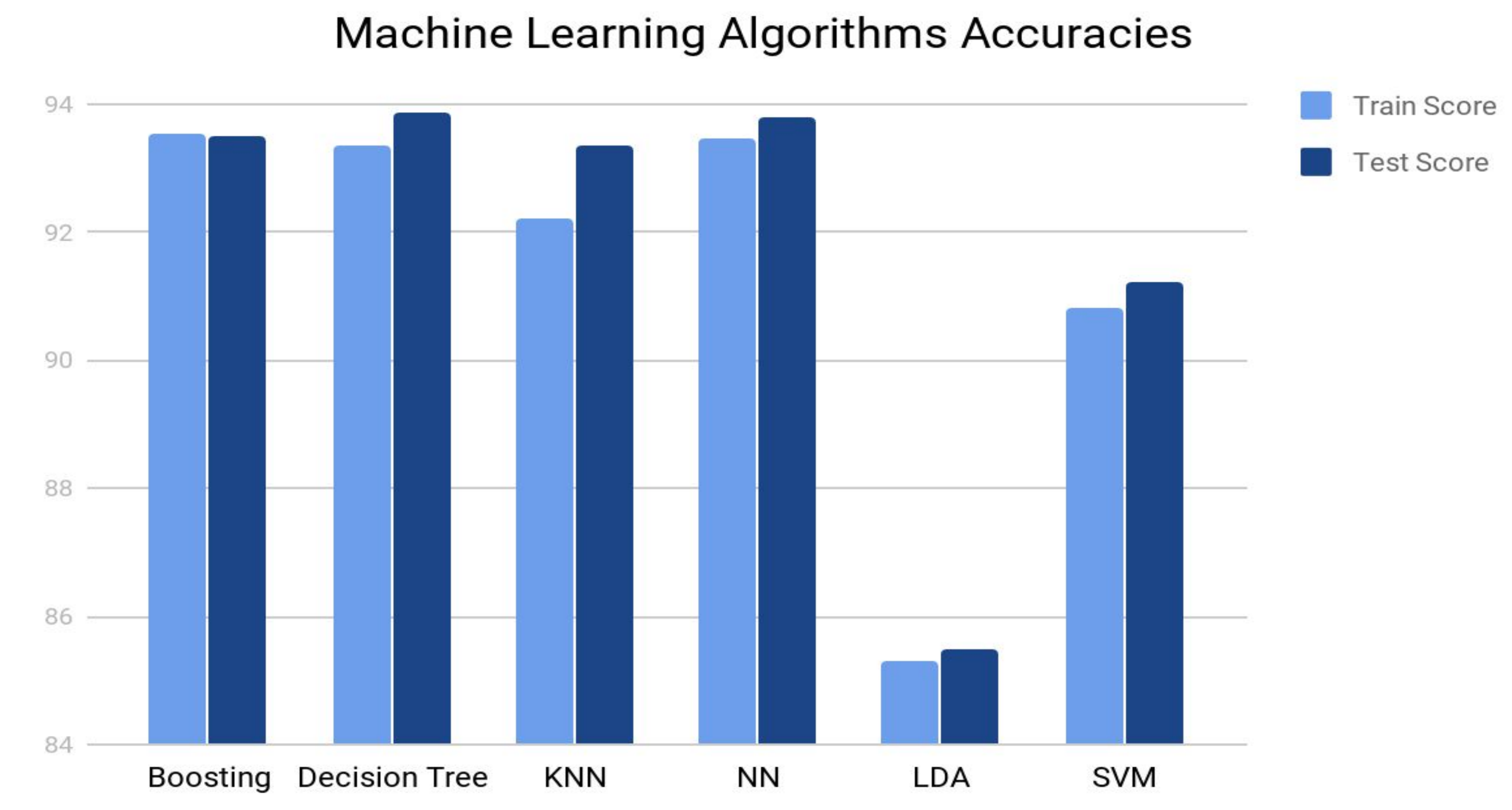
Browser Plugin

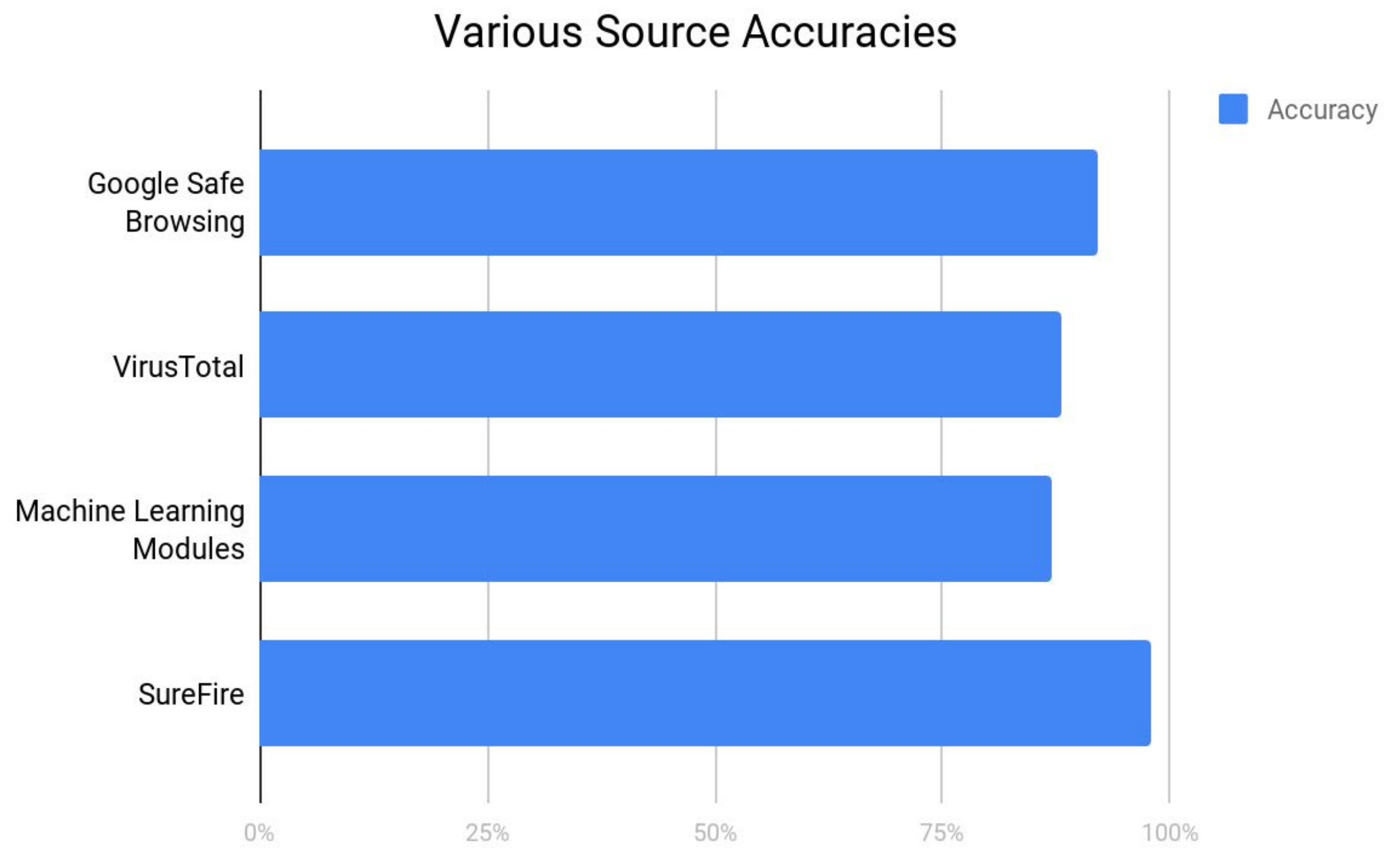

Mozilla Firefox
Debugging with Firefox Dev
bugraanaokulu.com/fou
Preferences
bugraanaokulu.com/four/BOA/index.php
Search

This is Phishing Website

Experiments & Observations

Experiment	Rationale	Evaluations/Result
Determine the most important features for phishing websites	Phishing websites have 30 features that are considered for model selection. Scraping one million websites and then extracting these many features is mammoth job and some features may not be worthy of extraction. We ran feature reduction and extracted important features that have higher influence on overall phishing, without affecting accuracy.	We reduced size of features from 30 to 8 using feature reduction. We then validated our results to see if we maintain same accuracy post features reduction. Before features reduction, our accuracy was 90.50%. Post features reduction, we were able to achieve 92% accuracy and reduced time significantly for extracting relevant features(only 8 now) of other data samples.
Determine the most effective machine learning model	For supervised learning, we have many choices like decision tree, neural network, linear regression, KNN etc. We found that, after feature extraction, some algorithms like decision trees were giving less accuracy than neural network, perhaps due to bias in the dataset. We also experimented with K-nearest neighbor algorithm. The results of various machine learning algorithms for an example instance is shown in Bar chart.	Among algorithms LDA shows relatively less accuracy and KNN, being an instance-based method, needs to store the huge dataset and takes time to traverse through it for each query. So, LDA and KNN were not used for the final product. The Machine-learning (ML) module consists of the rest 4 algorithms in below figure where their cross-validation train accuracies and test accuracies are shown. The confidence values (average probability) with which the ML module determines if the site is phishing and not, are computed and based on which one is greater, the site is classified as phishing and not-phishing respectively.
How to mix results from various methods?	Every phishing detection method generates different results in worst case. We need to provide single unified score to users by abstracting out these inherent variations.	We found that every method is good for detecting some class of websites while suffer from some other class as mentioned above. We ran accuracy detection on our training dataset. Google's APIs were producing 90%+ accuracy on training dataset while machine learning and VirusTotal were just shy of 90%. We decided to apply higher weightage to Google's result and equal to VirusTotal and Machine learning. We will re-train as we get more user feedback and modify weightage accordingly.
How to return data quickly to waiting client?	Surefire is API based system and we must provide final result quickly to client. We saw TCP timeouts when we were not running methods in parallel.	We threaded our system and started running all 3 methods in parallel. Threads then return data to result mixer for final score calculation. We were able to get final score quickly to waiting clients.





Conclusion

Combination of results from various phishing detection techniques help in improving accuracy to greater extent. We showed from our experiment and product that we can offer high-performance detection even for modern phishing websites. Our product also shows that, in isolation, even the state-of-the-art machine learning models can be bypassed and hence have to be augmented with traditional models especially in Information security space.