International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 1 Jan. 2023, pp: 808-813 www.ijaem.net ISSN: 2395-5252

Malicious Activity Detection Over Internet Via Machine Learning

Mamady kante[1], Daniel zinyongo[2], Hassad Hassan [3], Ali Jabir[4]
Department of Computer Science & Engineering
School of Engineering & Technology
Sharda University
Greater Noida, Uttar Pradesh, India

Date of Submission: 12-01-2023 Date of Acceptance: 25-01-2023

ABSTRACT: We live in a generation whereby everybody now has access to a computer, weather it is a desktop or a microcomputer such as a cell phone or tablet. In our present, the growth of technology has also presented an environment where everything is now done digitally such as file storage via the cloud, surfing the web and also communication. However, with this easy accessibility combining into a global village, it has brought problems regarding one's security on their device, and even with the vast malware detection software and security measures taken to avoid this, it is quite hard to detect even the newest of malware that is being developed every single day. The answer to this would be using machine learning to fight off malware and anti-malware.

I. INTRODUCTION

Cybersecurity threats including ransomware, phishing, malware injection, etc. have significantly increased recently on numerous websites throughout the world. These attacks have cost a variety of financial institutions, e-commerce businesses, and individuals a tremendous amount of money.

As new attack types emerge daily, it is extremely difficult for cybersecurity experts to contain a cybersecurity attack in this kind of circumstance.

1.1 Problem Statement

Computer and electronic security have become a very crucial problem ever since the easy accessibility of owning a device. Nearly 600,000 computer viruses are developed every day and most of these antiviruses and antimalware software cannot catch up. The way this software is developed are meant to capture malware that's already been built, and cannot catch the malware until it has been updated to search for it. There are many ways to unleash malware and steal people's data such as sneaking the malware into downloads (Trojans), phishing and pharming and several other tactics. The way the antimalware and antivirus software are built are meant to capture and detect malware that already exists, which leaves a loophole for the malware that's being developed which the software does not know of and makes it a huge danger to people who do not know about malware or the antivirus itself.

1.2 Project Overview

A solution to the current problem would be to use machine learning. Artificial intelligence has grown into a very powerful technology over the past years, hence applying machine learning to the detection of malicious software and various malicious activities would prove effective as a security measure. Machine learning seems to be a suitable solution because its adaptive, like the human brain, learning as it goes from the start from detecting the easiest of malware to the advanced malware that's not easily trackable. The tradition al way of tracking malware is not as advanced as the code is not adaptive and is fixed on the known malware that exists, and focuses on specific malware only known, leaving room for malware that is potentially dangerous to infect devices.

1.3 Expected Outcome

The aim is to produce an adaptive, advanced antimalware software that can save time for cybersecurity initiates and professionals to fight the war on malware spreading. The antimalware is meant to be advanced enough to do all the work, while having time spent on maintenance so little, since it is supposed to learn about its main goal; to find malware and malicious activity and eliminate it. The second aim is for compatibility; the ability for this program to work on several devices stretching from personal computers to mobile devices.

1.4 Hardware & Software Specifications
The hardware needed is not meant to be demanding, for everyone holds a device which fits in a specific budget. The current requirements needed are:

- Windows 8.1 and above
- Dual Core CPU, 1GHz and above
- 5GB internal memory and above
- 2GB Ram and above

II. LITTERATURE SURVEY

2.1 Existing Work

There have been several experiments and reviews about malware detection, and many of them have been successful. Malware analysis is required in order to create an effective approach to malware detection. The goal of malware analysis is to understand how the specific code of malware works so that defenses can be built to combat these malwares and protect the system and the network. The static analysis method examines programmed or executable binaries without running them. During static analysis, the programmed is broken down using various reverse engineering techniques and tools in order to rebuild the original source code. This procedure is mostly carried out by hand.

2.2 Proposed System

Machine Learning has the advantage of analyzing the data that falls in the current knowledge it has been programmed with. With this knowledge, the AI can reason with the unseen properties as time goes by. The mathematical analysis is called a model. Machine learning has the ability to create a broad variety of options it takes to acquire a solution rather than sticking to a single model.

Advantages of Machine learning

- There is very little, if any, human interaction required.
- Machine learning algorithms improve on their own. The accuracy and efficiency increases, resulting in better decisions.
- Machine learning algorithms are also very good at dealing with multidimensional and multivariable data. Even in dynamic or uncertain environments.
- Machine learning is also used in a variety of applications

2.3 Feasibility Study

In the last five years, significant advances have been made in vision, speech recognition and generation, natural language processing (understanding and generation), image and video generation, multi-agent systems, planning, decision-making, and the integration of vision and motor control for robotics. Although artificial intelligence (AI) cannot identify and correct every possible malware or cyberthreat issue, it may be a successful and strong weapon against even the most complex malware when it models both a program's ill and positive goals.

This has mainly been a success on the android platform. The dependence on smartphone devices is now higher than ever before. Since the 90s, malware has posed a serious risk to computer systems and networks. However, malware complexity has increased since the beginning of its detection. The evolution of computer malware functionality and behavior can be divided into five distinct generations [2], and with the multiple improvements to the Android framework and the ongoing evolution of Android malware, detecting malware over time in an effective and scalable manner is difficult [3].

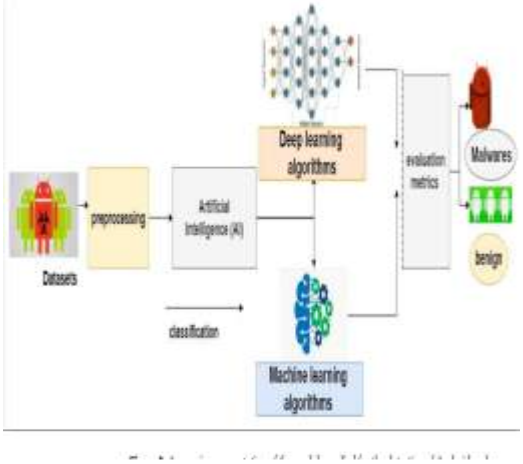
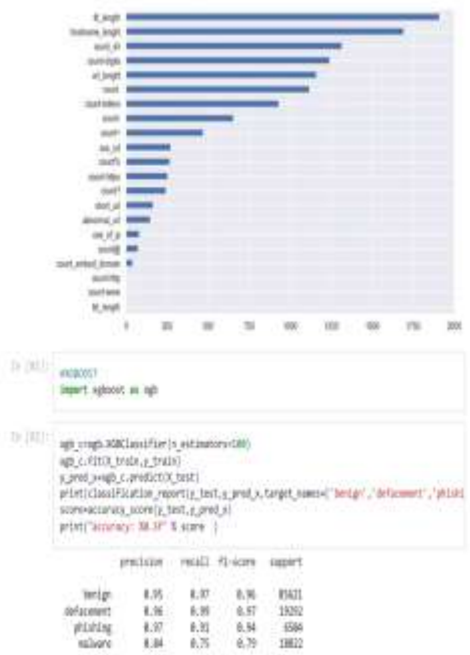


Fig1 malware detection in Android system

The primary defense against cyberattacks for users is software from anti-malware corporates. However, as the anti-malware sector becomes more impactful at identifying and preventing malicious activities, more sophisticated malware samples may appear in the wild. As a result, the arms race between malware defenders and malware authors rages on [5]. Fig 2 shows the metrics used to research the sites used to visit and how many users access safe and unsafe URLs.



```
#XGBOOST
import xgboost as xgb
```

```
xgb_c=xgb.XGBClassifier(n_estimators=100)
xgb_c.fit(X_train,y_train)
y_pred_x=xgb_c.predict(X_test)
print(classification_report(y_test,y_pred_x,target_names=['benign','defacement','phishi
score=accuracy_score(y_test,y_pred_x)
print("accuracy: %0.3f" % score )
```

```
              precision    recall  f1-score   support

      benign       0.95      0.97      0.96     85621
  defacement       0.96      0.99      0.97     19292
    phishing       0.97      0.91      0.94      6504
     malware       0.84      0.75      0.79     18822
```

Fig 2. Metrics representing users who access safe and unsafe

III. METHODOLOGY

The research is evaluated on the quality, reliability, and accuracy of the methodology on which it was conducted and the analysis of the information provided. This section examines how the data was collected for the research.

We will generate lexical numeric characteristics from the given URL because machine learning techniques only accept numeric input. As a result, rather than real raw URLs, the numeric lexical properties will be used as input to machine learning algorithms. If you are unfamiliar with lexical characteristics, you may look for an explanation on Stack Overflow. As a result, in this case study, we will employ the three well-known machine learning ensemble classifiers Random Forest, Light GBM, and XGBoost. Later, to discover which characteristics are critical for predicting harmful URLs, we will evaluate their performance and generate an average feature importance plot.

A. source

Our data was imported from kaggle.Com which is a data science-based community website.

B. Decription of the data

We employed a harmful URL dataset of 6,51,191 urls in our case study, comprising 4,28,103 benign or safe urls, 96,457 defacement urls, 94,111 phishing urls, and 32,520 malware urls. Our data set has 6,51,191 entries, each with two columns: URL, which contains the raw urls, and type, which is the target variable.

Following that, we looked at the numerous categories of urls in our dataset, such as benign, malicious, phishing, and defacement urls.

•Safe URLs: These are URLs that are safe to visit. Here are a few examples of safe URLs:
-facebook.com
- google.co.in
-myspace.com

• Malware URLs: When a victim visits these urls, malware is injected into their system. Here are some harmful urls examples:
• protoplast.co.nz
• microencapsulation.readmyweather.com

• Defacement URLs Defacement URLs are typically generated by hackers in order to get access to a web server and replace the hosted website with one of their own, utilizing techniques such as code injection, cross-site scripting, and so on. Government websites, banking websites, and business websites are common targets for URL spoofing.

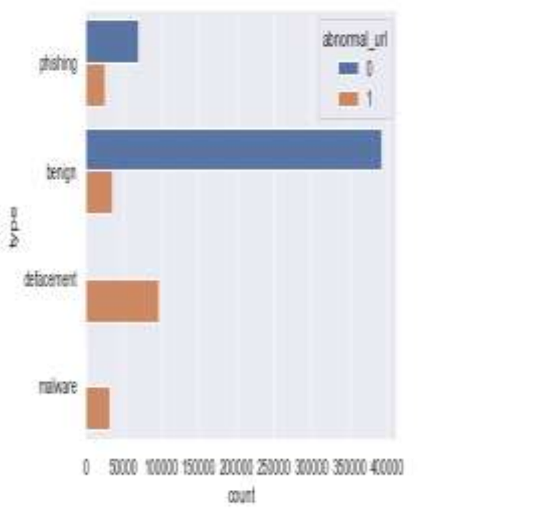
• Phishing URLs: Hackers use phishing URLs to obtain sensitive personal or financial information such as login credentials, credit card numbers, internet banking information, and so on.

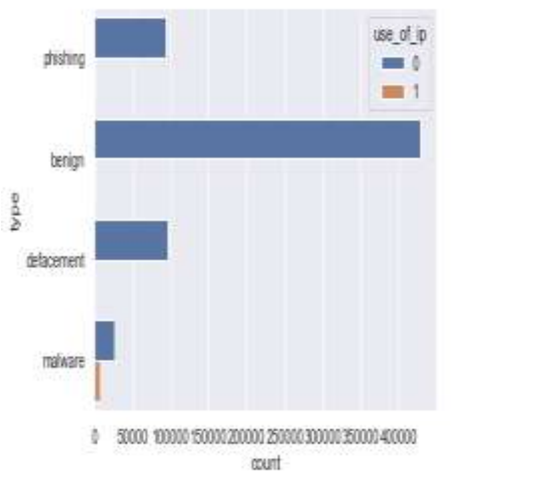
To do data pre-processing and exploratory analysis, we imported all of the essential Python libraries for this project: numpy, pandas, matplotlib, seaborn, sklearn, metrics, and so on..

1. Data preparation and pre-processing

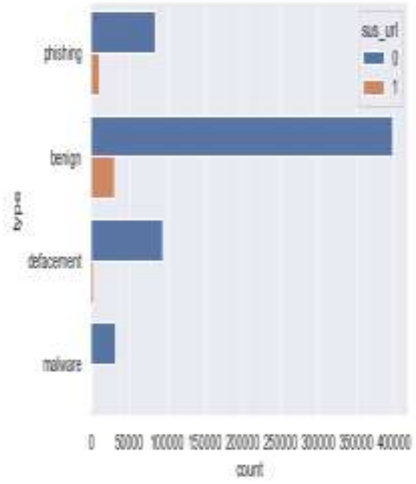
- To do the pre-processing and exploratory analysis of the data we imported all the necessary python libraries which will be used in this project: numpy, pandas, matplotlib, seaborn, sklearn, metrics, etc.
- Exploratory Data Analysis (EDA)

Data visualization or EDA is the process of analyzing data in the form of graphs or maps, which makes it much easier to understand trends or patterns in the data. In this step, we checked the distribution of different features for all four classes of URLs.

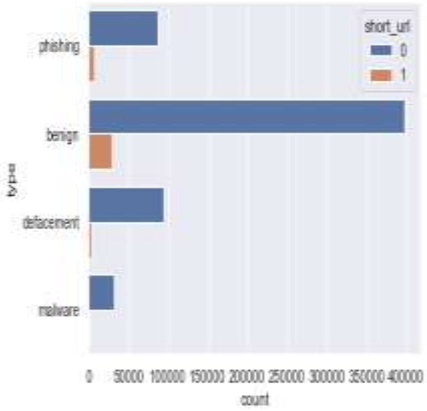




As we can observe from the above distribution of use_of_ip feature, only malware urls have IP addresses. In the case of abnormal_url, defacement urls have higher distribution.



From the distribution of suspicious URLs, it is clear that benign URLs have the highest distribution while phishing URLs have the second highest distribution. As suspicious URLs consist of transaction and payment-related keywords and generally genuine banking or payment-related URLs consist of such keywords that's why benign URLs have the highest distribution.



As per the short_url distribution, we can observe that benign urls have the highest short urls as we know that generally, we use URL shortening services for easily sharing long-length urls.

2. Preprocessing

Before using our data to train models for machine learning, it is necessary for us to clean our data. We encoded the target variable to the numerical form by using LabelEncoder because machine learning algorithms only understand numeric target variables. After we will use the train_test_split function to split our data into two groups which are the trainset and test set.

Outcome produced

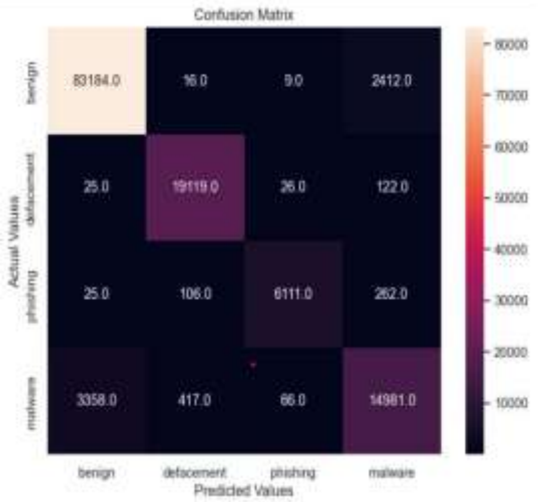


Fig 2:Confusion matrix for random forest

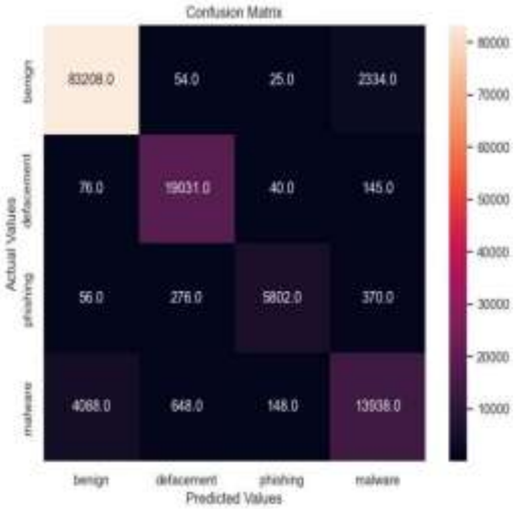


Fig3 :Confusion matrix for Light GMB classifier

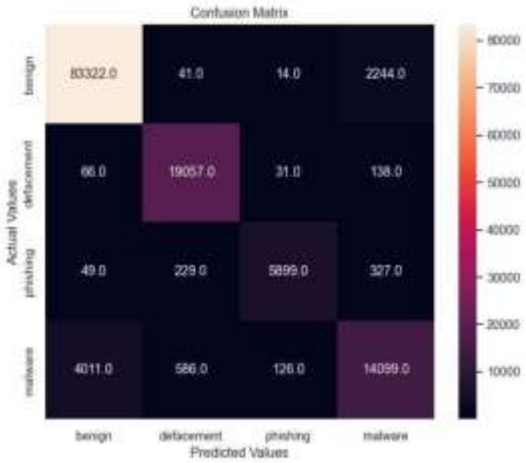


Fig4: Confusion matrix for XGBOOST

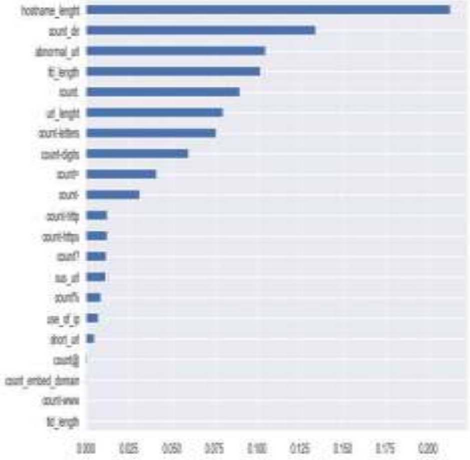


Fig5: Feature importance of random forest

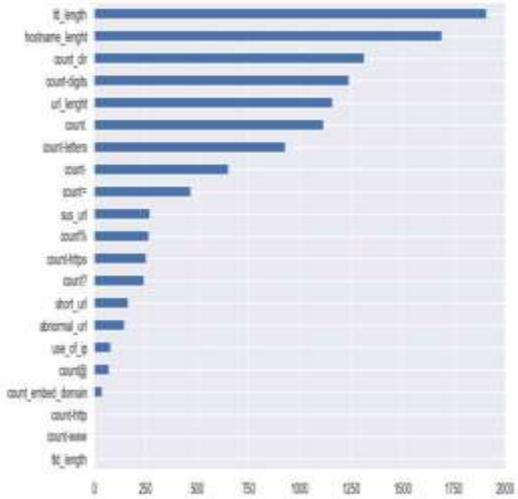


Fig6: Feature importance of Light GBM Classifier

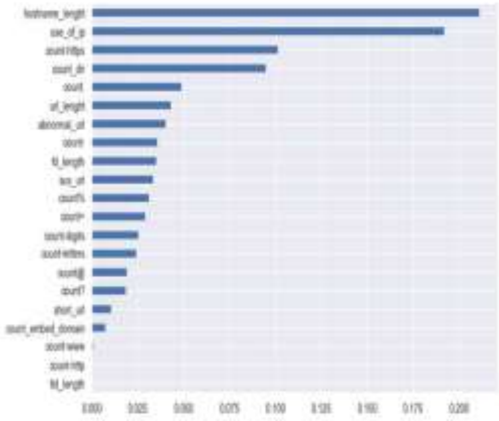
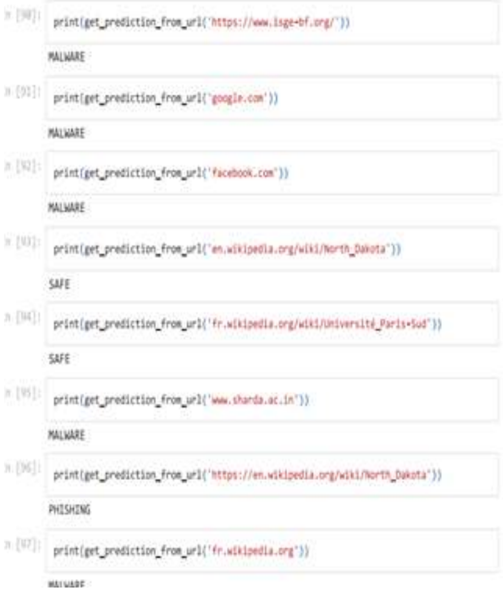


Fig7: Feature importance of Xgboost



```
In [90]: print(get_prediction_from_url('https://www.isge-bf.org/'))

MALWARE

In [91]: print(get_prediction_from_url('google.com'))

MALWARE

In [92]: print(get_prediction_from_url('facebook.com'))

MALWARE

In [93]: print(get_prediction_from_url('en.wikipedia.org/wiki/North_Dakota'))

SAFE

In [94]: print(get_prediction_from_url('fr.wikipedia.org/wiki/Université_Paris-Sud'))

SAFE

In [95]: print(get_prediction_from_url('www.sharda.ac.in'))

MALWARE

In [96]: print(get_prediction_from_url('https://en.wikipedia.org/wiki/North_Dakota'))

PHISHING

In [97]: print(get_prediction_from_url('fr.wikipedia.org'))

MALWARE
```

IV. CONCLUSION AND FUTURE SCOPE

We demonstrated how to use machine learning to detect dangerous URLs. We retrieved 22 lexical features from raw URLs and trained the XG Boost, Light GBM, and Random Forest machine learning models on these. Furthermore, we assessed the efficacy of the three machine learning models and observed that Random Forest outperformed the others, reaching the highest accuracy of 94.3%.

By charting the feature relevance of Random forest, we determined that the top 5 characteristics for recognizing malicious URLs are hostname length, count dir, count-www, fd length, and URL length. Random Forest, the prediction technique for classifying any raw URL using our stored model, has finally been developed.

For the time being, the program works in the back-end environment. In the future it is hoped that we will have a front-end development where users have options to select and filter out the URL's which they feel safe which the program might have missed. The other improvement being worked on is the complete accuracy of automatic detection rather than having to update the database of URLs since the program only searches for types of malwares on the websites but cannot discern which website is safe or not. The process of selecting websites is mainly done manually.

REFERENCES

[1] Omar N. Elayan∗ , Ahmad M. MustafaAndroid Malware Detection Using Deep Learning

[2] Sadia Noreen† , Shafaq Murtaza† , M. Zubair Shafiq‡ , Muddassar Farooq†Evolvable Malware

[3] YANFANG YE, TAO LI, DONALD ADJEROH, S. SITHARAMA IYENGAR,A Survey on Malware Detection Using Data Mining Techniques

[4] Jeffrey C Kimmell∗, Mahmoud Abdelsalam†, and Maanak Gupta Analyzing Machine Learning Approaches for Online Malware Detection in Cloud

[5] John Demme Matthew Maycock Jared Schmitz Adrian Tang Adam Waksman Simha Sethumadhavan Salvatore Stolfo

[6] Dragos¸ Gavrilut¸1,2 , Mihai Cimpoes¸u1,2 , Dan Anton1,2 , Liviu Ciortuz Malware Detection Using Machine Learning

[7] Kaspersky Labs Malware Detection Using Machine Learning

[8] Kateryna Chumachenko MACHINE LEARNING METHODS FOR MALWARE DETECTION AND CLASSIFICATION

[9] Nejad, M. B., Nejad, E. B., & Karami, A. (2012). Using Data mining Techniques to increase efficiency of Customer Relationship management process. Research Journal of Applied Sciences, Engineering and Technology, 4(23), 5010-5015.

[10] Tiago, M. T. P. M. B., & Verı́ssimo, J. M. C. (2014). Digital marketing and social media: Why bother?. Business horizons, 57(6), 703-708.