**EVENT-DRIVEN MALICIOUS URL EXTRACTOR**

Ramanayaka A. M

(IT18021912)

B.Sc. (Hons) Degree in Information Technology Specializing in Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

**EVENT-DRIVEN MALICIOUS URL EXTRACTOR**

Ramanayaka A. M

(IT18021912)

Dissertation submitted in partial fulfillment of the requirements for the

Bachelor of

Science in Information Technology Specializing in Cyber Security

Department of Information Technology

Sri Lanka Institute of Information Technology

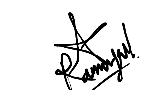
Sri Lanka

October 2021

# **Declaration**

“I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).”

Signature:  Date: 08/10/2021

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor: Date: 08/10/2021

# **Abstract**

People have become more used to using the internet for various purposes at the present and this usage will increase in the future with the rapid development of technology. With this development in technology there will always be advantages as well as disadvantages. Advantages are that the communication and work becomes much easier and faster for people then more and more people start to rely on these services, due to that comes the internet fraud and internet based security concerns arise which becomes a disadvantage. With that being said, the need for web security increases and more people has to be aware as to not get caught for any malicious activities which makes them vulnerable to security issues. Malicious URLs is one of the main concerns that has caught the attention of people because it can approach people in any form such as a phishing email that redirects them to a malicious website or a voice call that seems to be from a legitimate person which will ask for private and sensitive information such as credit card numbers or passwords or drive by downloads which installs malicious software on the devices. There are so many scenarios of which attackers use malicious URLs to target different people for different purposes but the end goal of the attackers will always be the same which is the personal and sensitive information of their victims. Categorization of malicious URL scenarios, URL feature classification, feature extraction, feature analysis and feature reduction are the main focus of my study and I will be studying more on how this can help with the speed and efficiency of the system for detecting phishing URLs. URL Feature reduction must be achieved with the approach to a better URL feature classification and URL feature extraction procedure that helps with better analysis of the extracted features therefore resulting in better feature reduction which delivers better efficiency and speed of the system. Feature classification and extraction will be done with reference to a lot of successful previous researches and then will be modified to achieve better accuracy which supports the model that we propose. Feature extraction, analysis and reduction will be done with python based coding and this will be achieved with tested datasets which are consisted of malicious URLs and legitimate URLs which would help with comparison of both of the datasets to extract the features and to analyse them.

*Keywords – internet fraud, web security, malicious urls*

# **Dedication**

My dissertation is dedicated to my family, my close friends and my supervisors who had been guiding me and helping me through this project. I am thankful for the encouragement I receive from family and for the assistance that I receive from my closest friends and the knowledge, help and guidance I receive from my supervisors and other lecturers who has been helpful throughout this research project. I would be dedicating this dissertation to all these people who had been here with me since the start of this project.

# **Acknowledgement**

This project would not have been a success if we did not receive a lot of guidance and help throughout the project from the very beginning of the proposal and up to now. I would like to kindly thank our project supervisor Mr Amila Nuwan Senarathne for providing us with the proper guidance, help, thoughts and ideas and for sharing his knowledge with us to achieve success throughout this project.

I would also like to show my gratitude to Sri Lanka Institute of Information Technology for giving us the opportunity to conduct a research project and gain valuable knowledge and experience.

**Table of Contents**

[**Declaration** III](#_Toc84580530)

[**Abstract** IV](#_Toc84580531)

[**Dedication** V](#_Toc84580532)

[**Acknowledgement** VI](#_Toc84580533)

[**Table of Contents** VII](#_Toc84580534)

[**List of Tables** IX](#_Toc84580535)

[**Table of Figures** X](#_Toc84580536)

[**1 INTRODUCTION** 1](#_Toc84580537)

[**1.1 Background & Literature Survey** 1](#_Toc84580538)

[**1.2 Research Gap** 9](#_Toc84580539)

[**1.3 Research Problem** 11](#_Toc84580540)

[**1.4 Research Objectives** 13](#_Toc84580541)

[**1.4.1 Main Objectives** 13](#_Toc84580542)

[**1.4.2 Specific Objectives** 14](#_Toc84580543)

[**2 METHODOLOGY** 15](#_Toc84580544)

[**2.1 Feature Classification** 16](#_Toc84580545)

[**2.1.1 Address Bar based features** 20](#_Toc84580546)

[**2.1.2 Domain based features** 21](#_Toc84580547)

[**2.1.3 HTML & JavaScript based features** 22](#_Toc84580548)

[**2.2 Feature Extraction** 24](#_Toc84580549)

[**2.3 Feature Analysis** 29](#_Toc84580550)

[**2.4 Feature Reduction** 31](#_Toc84580551)

[**Feature Reduction Techniques** 31](#_Toc84580552)

[**2.5 Scenario Categorization** 33](#_Toc84580553)

[**Deactivation scares** 33](#_Toc84580554)

[**Look-alike websites** 33](#_Toc84580555)

[**SEO Trojans** 33](#_Toc84580556)

[**Craigslist money scams** 33](#_Toc84580557)

[**Work mules** 34](#_Toc84580558)

[**SMS phishing** 34](#_Toc84580559)

[**Fake crisis notices** 34](#_Toc84580560)

[**Compromised Credit Card** 34](#_Toc84580561)

[**Social Media Request** 35](#_Toc84580562)

[**Fake Google Docs Login** 35](#_Toc84580563)

[**2.6 Commercialization** 36](#_Toc84580564)

[**2.7 Testing and Implementation** 37](#_Toc84580565)

[**3 RESULTS AND DISCUSSION** 39](#_Toc84580566)

[**3.1 Results** 39](#_Toc84580567)

[**3.1.1 Feature Extraction** 39](#_Toc84580568)

[**3.1.2 Feature Analysis** 43](#_Toc84580569)

[**3.2 Research Findings** 44](#_Toc84580570)

[**3.3 Discussion** 45](#_Toc84580571)

[**3.4 Summary of each Student’s contribution** 46](#_Toc84580572)

[**IT18071412 – S.W. Jonathan** 46](#_Toc84580573)

[**IT18021912 – Ramanayaka A. M** 46](#_Toc84580574)

[**IT18032666 – Wishvajith B.L.D.V** 47](#_Toc84580575)

[**IT18034400 – Renu Harshatha** 48](#_Toc84580576)

[**Conclusion** 49](#_Toc84580577)

[**References** 50](#_Toc84580578)

# **List of Tables**

[Table 1-First set of malicious URL features [11] 17](#_Toc84569307)

[Table 2-Host based features (2nd set) [11] 18](#_Toc84569308)

# **Table of Figures**

[Figure 1 - URL Components[2] 5](#_Toc84621217)

[Figure 2 - URL Feature Classification (3rd set) - 1 [10] 19](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621218)

[Figure 3 - URL Feature Classification (3rd set) - 2 [10] 19](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621219)

[Figure 4 - Feature Extraction - Loading the legitimate plus malicious URL set 39](#_Toc84621220)

[Figure 5 - Feature Extraction - Loading the phishing URL set 40](#_Toc84621221)

[Figure 6 - Feature Extraction - Collecting random URLs (Legitimate and malicious) 40](#_Toc84621222)

[Figure 7 - Feature Extraction - Collecting random URLs (Phishing) 41](#_Toc84621223)

[Figure 8 - Feature Extraction - Storing features of phishing URLs in a dataframe -1 41](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621224)

[Figure 9 - Feature Extraction - Storing features of phishing URLs in a dataframe -2 41](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621225)

[Figure 10 - Feature Extraction - Storing features of legitimate and malicious URLs in a dataframe -1 42](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621226)

[Figure 11 - Feature Extraction - Storing features of legitimate and malicious URLs in a dataframe -2 42](file:///C:\Users\Dell\Desktop\IT18021912-2021-085.docx#_Toc84621227)

[Figure 12 - Feature Extraction - Final concatenated dataset 42](#_Toc84621228)

[Figure 13 - Feature Analysis - Output after the analysis 43](#_Toc84621229)

# **1 INTRODUCTION**

## **1.1 Background & Literature Survey**

Internet is a boundless network that connects all the computers and smart devices in the world and people have become more used to it that they have arrived at a point where they cannot do any work without using the internet. Web services which are services provided through the internet are used in a wide variety of industries for many purposes. [1]

People have started to rely on these web services more than ever. Online communication to online money transfer and many more activities have now been established which makes it more useful for the people while also encouraging more and more malicious web activities to occur therefore more and more security concerns arise with the increasing usage of web services among people. The internet is not something that could be controlled and it is an open network which is used by billions of users where users are given an IP address as their ID which makes the users anonymous to others, therefore it is hard to control attackers and legitimate users in this network that connects the whole world.

There are so many vulnerabilities and security issues in this vast spread network which enables so many cyber-attacks and all these security issues and vulnerabilities cannot be tackled to reduce the risks because with the rapid development of the technology comes the development of better cyber-attacks and threats which poses security risks to the users of the internet. [2] These cyber-attacks also include the fake websites which are created by attackers to commit financial fraud and obtain sensitive information from users to commit data theft, identity theft, money theft, and these websites can trick users into downloading malware to the user’s computer which will then be installed for various malicious purposes. Existing security solutions is not enough at time to avoid these security breaches and attacks and attackers are smart enough to bypass some security measure with techniques like social engineering, phishing, drive-by-downloads and other hacking techniques to perform attacks. [3] Attacker use this as an advantage to target their victims. Malicious URLs, that are used for many attacks such as phishing, drive by downloads or spam, has become one of the main security concerns because it manipulates people in many ways and people get caught for them due to various reasons. In this method attackers work using psychological tricks to get their targets to do what they want and unfortunately lot of people get caught for these scams and phishing emails. Besides normal people, even the security experts get caught for these scams therefore these phishing sites, emails and URLs should be filtered by any means so we have chosen this area of study to build a system to detect malicious URLs.

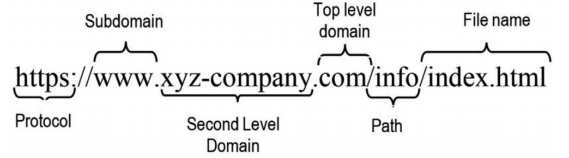
Phishing incidents increase every year and as statistics show, the number of phishing incidents were known to have increased by 15% from 2019 to 2020. [4] While phishing being a popular form of cyber-attack, phishing techniques and the skills have improved over the years so it is becoming harder to identify them.

Cyber criminals uses trending topics for their phishing emails as well such as in 2020 there were many phishing and scam emails related to and themed by the title of the COVID-19 pandemic. In a previous survey carried out before the covid pandemic, which was at the time of when the natural disaster of Hurricane Harvey happened, had said to show significant amount of cyber-attacks especially phishing based, scams and social engineering based attacks. A lot of respondents had stated that they received phishing emails with malicious links and some of them had already clicked on those provided links on the emails. It stated that more than 36% of the respondents had stated about receiving these phishing emails and 10% of the respondents had stated that they had clicked on the links that were given in the phishing emails. In 2020, we could see the COVID-19 pandemic affecting on peoples’ lives in various ways. People had to be at home, some people lost their jobs, some work from home, children had to start learning online through various video conferencing platforms and so did the workplaces follow the same ways of hosting meetings for their employees or any other business related meetings, and similarly online activity had been increasing with the effect of the pandemic and due to that we have been able to see that people are more prone to these cyber-attacks because they would always be using the internet for all these online activities so due to that attackers have the largest benefit on this pandemic as more people would be using internet for different online activities. Most of the attacks were known to be phishing emails or other fraudulent emails which contains malicious links or malicious attachments and sometimes these malicious links or other types of malicious attachments were also sent through social media platforms to the users. Attackers use various methods to approach the users due to this pandemic such as creating fake websites pretending to provide information regarding health and the pandemic, sending phishing emails with malicious links and attachments that are related to the pandemic and they would also create different applications which are health related which would trick users to download them and then finally download the needed malwares or other types of attachments to their computers. The users are unable to detect the malware or even understand whether a website is fake or not even because of the attackers using the pandemic as a disguise for their malicious activities which confuses the users and trick them into clicking the malicious links or download the malicious attachments or to provide sensitive information to them. [5]

These kind of trending scenarios are widely used by attackers to attract their targeted victims more easily. Especially in social engineering techniques, they use this as the disguise to trick people into believing that these phishing emails, or messages from social media or whatever fake website they visit is a legitimate one as people are more concerned on that topic for most of the time during that period of time. Scenario categorization is important so that we can know the trends and how different scenarios could be used for different trends and topics by the attackers. The scenarios need to be categorized with much more than what we can see with trending topics and need to be looked into more deeply to carry out the categorization much precisely and efficiently to speed up the system because scenario examples like COVID-19, fortune telling or war will not help with capturing malicious URLs. Finding what is common among even these examples will help to categorize the scenarios properly which will help us understand more about how we can also use it for feature reduction and where it would all link together.

The next area to be looked into is the malicious URLs which will again be studied on how to be detected using existing techniques and using URL feature based detection where URLs will be looked for special features that legitimate URLs contain and malicious URLs contain. The features of legitimate URLs and malicious URLs will both be checked in the beginning and they will be extracted separately and they will be checked to see what kind of features would be effectively drawing out the malicious URLs and what features could effectively separate the malicious URLs from a legitimate URL. Features such as having an IP address, having special characters, the age of the domain being too short and having too many dots are some of the features pointed out in previous researches that were done regarding this topic and they are features we can see in a malicious URL and we are not able to see these features in most legitimate URLs. [1] Attackers create fraudulent websites in which the interface and the overall layout is similar to popular websites to make them look authentic enough for attracting their targeted victims. The problem is that the attackers cannot get the exact same URL of the original website so they use different URLs. If a person is keen enough to see the difference in the URLs of the original website and the malicious website/webpage then they will most probably not click on these links but in real life people do not spend time to look into these details and they are not careful enough so if a phishing email arrives with a message that gives a sense of emergency such as to login to a website to confirm the user’s account to stop their account from being deleted then they will definitely click the link to perform that process and the user will easily give out their personal information such as a username and password or credit card numbers to these attackers. The users that uses the internet have very much less knowledge compared to the users who have basic IT knowledge but these social engineering techniques can make even the experts in the IT industry to fall for these phishing emails, scams, drive-by-downloads, etc. A lot of the users are unsure of how a legitimate URL appears, they have less knowledge about URLs, users are usually unaware of the redirection and hidden URLs, the users usually do not spend much of their time trying to figure out whether the URL is a malicious URL or whether it would be a legitimate URL and then accidently clicks on the URL which takes them to the malicious websites and due to the users having less knowledge, they are unable to understand the difference between a legitimate URL and a malicious URL. Therefore phishing attacks and other cyber-attacks which involves the use of malicious URLs have become more common and more people get caught in these attacks. [2]

Attackers use a lot of techniques on URLs and to understand how they use them and to understand what kind of changes they can do to disguise the malicious URL to look like a legitimate URL is important if we are concerned on providing enough security and providing knowledge for the model to find out the differences in a malicious URL and legitimate URL. Knowing URL components is one of the key factors of capturing malicious URLs. The components of the URL is shown in the figure below. [2]



**Figure 1 - URL Components** [2]

The protocol name used to access the web page is the first part of a URL, the organization name in the hosting server is represented by the subdomain and the second level domain name (SLD), and then we can find the top-level domain which shows the domains which could be found in the DNS’s root zone.

The domain name (host name) is the most unique and the most critical feature of the URL which is made up of the two components TLD (Top Level Domain) and SLD (Second Level Domain). The attacker is able to either find or buy the SLD for phishing but even without buying it the attacker can still use it to generate vast amount of URLs by adding path and file names to the SLD which will still make it look authentic for a normal person. The unique feature of the URL is the domain name therefore it has always been a great focus area of the cyber security companies to understand how domain names can be used by attackers to create fraudulent websites with domain names that would still look legitimate. In this way, the cyber security experts are able to find the IP addresses which are used by those domain names and then block that IP address so that users would be restricted from visiting other webpages that are created by that IP address which would protect users from becoming victims of attackers. The defence mechanisms that are implemented by experts in the cyber security field should take into consideration the use of different attack techniques on the URLs which would increase the maliciousness of the URL and therefore increasing the vulnerability of the users which leads the users into providing sensitive information willingly due to them having less knowledge about the malicious URLs. Attackers could use techniques like cybersquatting, random character usage or combined usage to enhance the effect of these malicious URLs therefore detection mechanisms should be implemented to become able to understand these attack techniques. [2]

According to many researches the most common way to detect malicious URLs is known as the blacklist method where it contains a database of URLs that were known to be malicious in the past and were added to the list after confirmation and this database is updated with new URLs whenever found. It is a method which is very easy to implement and is a very fast mechanism due to having a very simple query to be fulfilled and it is also known to have very low false positive rates but it was later known to have non-trivial false positive rates. This method is not very efficient in the present because new URLs are generated in large amounts and it is difficult to keep track of them. Next method is a heuristic approach which is a similar to blacklisting but this time it is not blacklisting URLs, instead it will blacklist signatures where they identify common attack types and categorize them and find the signature features, then it will search for these specific signatures throughout a webpage for any suspicious behaviour to produce a warning flag that shows that this web page is suspicious. Both these methods are however resource intensive and is hard to implement therefore another approach that is said to be more efficient than both these methods is the machine learning approach. In this approach the information of URLs will be analysed and compared with the original webpage URLs. The good features of the original webpage URLs will be extracted for training the system so that it will be easier to detect suspicious behaviours and features of phishing URLs when detected and being compared. [3]

The next area that I will be focusing on is the scenarios that the phishers/attackers use for phishing. It is true that they will be using scenarios like COVID-19 or any other health related scenario or else they will use a topic like giving out shopping deals. The topics vary time to time but for the categorization of scenarios it will not be helpful so what I did was find out what was more common for every scenario and I listed out a set of common scenarios that attackers use while masking them using trending topics so for every topic this scenario will be the same. While looking into few websites I found out that the categorization would go by; look-alike websites, deactivation scares, SEO Trojans, work mules, fake crisis notices, compromised credit card, and social media request. [6] [7]

Feature Reduction will be the next area that I will focus to improve the speed and efficiency of the system. When there is a large amount of features to analyse it is very difficult to keep a system efficient and on proper speed. It will also reduce the reliability of the system and will increase the use of computational power which is not recommended. We have to find methods to find patterns in a dataset that has a large amount of data which shows a large amount of features so that we can reduce the features to make the system work more efficiently. There are two types where one is feature selection and the other is dimensionality reduction. Feature reduction is where we keep the most relevant features and dimensionality reduction is where we try to find a small set of features that will be a combination of the basic features we find. [8]

Feature reduction is the main focus area that is followed by the other areas related to achieving the feature reduction effectively such as the feature classification, feature extraction and feature analysis which will finally lead to achieving feature reduction much effectively and this will benefit the model, that is built, by providing the speed and efficiency which is important for better and faster identification of the malicious URLs which is the main goal of the system that would be implemented.

Feature classification would be achieved in reference to many researches that were conducted previously by other researchers and will be checked for the features that are most effective by comparing and identifying a better list of features which would be much more effective and efficient. After the process of feature classification, then we are able to move on to feature extraction which would be done through python coding which would be implemented to extract those said features from the URLs. For the test case and to see how efficiently the coding would run, there will be two datasets that would be used to achieve this process successfully. UNB dataset which contains the legitimate URLs and the dataset from PhishTank which contains the malicious URLs will be used to extract the previously created list of features to understand how the coding would work with the datasets and then it will be used for the feature analysis later on which will further let us understand how effective the feature classification has been and further providing us the confirmation of how the feature analysis would work after the feature extraction. Finally feature reduction will be carried out after both the datasets has been checked for features and after a process of comparison and then we are able to obtain the dataset that would be created after feature reduction.

Feature reduction must be achieved with a lot of effort and due to having high volume and high speed of the data that we obtain, which is the URL data, it has become a great challenge to perform the feature classification and feature extraction which leads to better feature analysis and feature reduction. Having high dimensional URL features has also become a great challenge to face when training a model that is made for feature classification, extraction, analysis and reduction. Deep learning methods have said to be developed to produce better solutions for the feature classification and feature representation. [3]

Features will be classified according to different researches and based on different methods discussed in different researches which will then be used to provide a proper classification of URL features by combining all the related knowledge and by performing a better research on creating the most specific features for the training the model that is created. The features will then be extracted which are descriptive enough to be used in machine learning models so that they could be interpreted in a mathematical format to be used in these machine learning models. Ideally, the goal would be to get a good representation of the URL's various features based on some basic principles or heuristics. Some of these may include the string's statistical properties, the host's info, and the geo-location details of the host. [3]

## **1.2 Research Gap**

Scenario classification is not very much explained in many research papers and very rarely can it be seen on a research paper even regarding the malicious URL detection systems. This area of the research should be given more research focus due to having very less or no research papers which explains about the scenarios and scenarios classification or scenario categorization.

URL feature classification is another research topic that should be given more focus to and there are only a few research papers that focuses on listing out the features that can be used to distinguish between the legitimate URLs and the malicious URLs. The listed features are different from paper to paper and it is hard to find out common features and or a specific set of features that would fit for the model that we are trying to build.

There are many research papers that explains the suitable techniques to detect malicious URLs but there is not much detailed description on the classification and categorization of scenarios and how it would be detected when capturing them on phishing emails, scam emails or any malicious links that are sent to people. Relying on methods such as blacklisting will be efficient but it will not be able to detect future attacks sooner and sometimes a phishing webpage may not behave suspiciously in the beginning and may launch the attack later so with that problem we cannot be sure that the blacklist method would detect that URL and flag it.

Most researches explain about the keywords being used by different emails or URL features that would help to identify it as a malicious URL but there are not many research papers about scenario classification or categorization which would help with a machine learning approach or natural language processing analysis approach. Feature extraction and analysis is also a research topic which is rare to be found in the research papers. These topics are discussed but the techniques or procedures to be followed to properly carry out these processes are not very much explained.

Feature reduction techniques are not very much discussed in most of the research papers and needs to be looked up on. Feature extraction and feature classification is discussed to some extent but feature reduction should be looked into and improved in order to speed up the system and make it work more efficiently. Below table shows how we have looked into researches and identified the components covered in them and it shows that our proposed model will be able to cover all the components listed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Papers​** | **Scalability​** | **Accuracy ​** | **Classification of URLs​** | **High rate of detection/efficiency​** | **Ability to sort through keywords / Events​**  **​** | **Classification of Scenarios​** |
| Research A [9] | ​ | ​ | ​ | ​ | ​ | ​ |
| Research B [10] | ​ | ​ | ​ | ​ | ​ | ​ |
| Research C [11]​ | ​ | ​ | ​ | ​ | ​ | ​ |
| **Proposed Solution**​ | ​ | ​ | ​ | ​ | ​ | ​ |

## **1.3 Research Problem**

Scenario categorization, URL feature classification, URL feature extraction, URL feature analysis and URL feature reduction are research topics which are in actual fact the most challenging topics to search for in research papers because they are discussed really rarely especially about how to perform these tasks with proper, promising and working procedures. Many researches state that their models would effectively carry out the processes of URL feature reduction but only a very few has explained the whole process of how they would carry it out in their proposed model and therefore it is hard to understand how the process works and it is hard to have an idea about how to use any of that work in favour of the model that we will be building.

Classification of emails had been carried out in one study which they show that they have identified 15 features which are often used by the phishers when phishing and they explain all the features in detail. They have also explained that they achieved high accuracy with this method of classification and these features are accurate as well. A few features was noticed which might be of use for the system that we are building because we will not be analysing emails but the malicious URLs only so below are some of those features that were listed by the researchers [9]:

* URLs Containing IP Address [9]
* Disparities between “href ” Attribute and LINK Text [9]
* Number of Dots in the Domain Name [9]
* Number of Links Linked To Domain [9]

The above provided set of features would not provide the most accurate results even though the research states it that way because in the modern world the attackers have become more experts in using the features of a legitimate URL to actively fool even security experts at times and the features listed above are only a few that would help to distinguish between a legitimate URL and a malicious URL. Many more features have been listed in different researches such as in the research listed in the reference [10] and it gives us more insight to a set of more descriptive features that would help us see the differences in legitimate URLs and malicious URLs.

The feature extraction can be done by creating proper set of features which would be build up on feature classification, but in order to do that a lot of research must be carried out to obtain the best set of features as mentioned earlier. However feature extraction is another challenge in this research due to feature classification being difficult to achieve and due to less resources on understanding and creating the most specific set of features for carrying out the feature extraction in order to successfully carry out the feature analysis which will further help in bringing out the differences we can find in malicious URLs and the legitimate URLs.

Feature reduction is known mostly as dimensionality reduction which helps understand the features and list down the features of the URLs that would further be broken down to a list of the most specific URL features to ensure the efficiency and effectiveness of the model which is being implemented. Feature reduction or dimensionality reduction is used to reduce a dataset with high dimensionality or features as we call it, to lower and manageable dimensionality so that we can easily use that dataset to work on a model. Having less features or less dimensionality helps to ensure the efficiency of an algorithm, higher speed of a system and ensures less training time/computation time. Having a set of features which are small and having the needed basic characteristics of the input values in a system helps to ensure efficient and effective functionality of a system where we need to work with high amount of data, and therefore Feature Reduction is important but it is also a very rarely discussed topic among most of the research papers and due to that it is a challenge to properly implement the feature reduction strategy and the feature reduction process. [8]

## **1.4 Research Objectives**

### **1.4.1 Main Objectives**

The main focus and the main benefit that we expect by implementing the malicious URL detection system is to detect and block phishing sites and malicious URLs and to filter phishing and scam URLs because at the present day, attacks like phishing, scamming and drive by downloads has become very common and the techniques and tactic used by the attackers are evolving with the development in technology and the researches done so far has not found a proper solution that will completely get rid of this problem. Making sure that we can at least create an efficient, speedier and an accurate system than the systems created previously while using ideas taken from the previously created detection systems and the studies done so far. Identifying a suitable Ensemble Model using deep learning and machine learning algorithms. To build this system we will use the existing models and modify or go towards a hybrid approach for betterment of this system. The main reason for us to choose this area of study for the research is mainly because of the phishing, scamming and malvertising attacks that took place during this current period where the attackers targeted their victims using the pandemic situation which is COVID-19 that spread throughout the world. Then they used COVID-19 themed templates for their phishing sites and COVID-19 related phishing and scam emails were generated to attract more victims so due to this situation and due to people being at rest and mostly at home, it became easier to get people to fall for these attacks. People had more time to check on this kind of emails, malvertising related to the COVID-19 situation, and to spend time on social media and click on links that their friends might send them which could most probably be redirecting these people to a phishing website which appears to be legitimate. [5]

Main research objective on my focus for this project is the feature reduction and scenario categorization which helps the system to work and capture malicious URLs efficiently and effectively. Feature reduction will be done with the help of three more sub objectives which are feature classification, feature extraction and feature analysis which finally leads to feature reduction which is a big challenge but will be achieved with better research and more focus on these objectives.

### **1.4.2 Specific Objectives**

The main objectives are the scenario categorization and feature reduction work better towards the goal so individually a lot of areas will be looked into more closely so that we can create a better model using machine learning and deep learning algorithms. I will be looking into the following areas as I have been explaining earlier.

* URL Feature Classification will be done to understand how many features were found which will first help distinguish between the malicious URLs and the legitimate URLs and then those features will be narrowed down to see which features were thought to be the most common in malicious URLs and to see how effectively we can capture the malicious URLs.
* URL Feature extraction will be one of the sub objectives of feature reduction that will help us extract those specific features in the URLs so that we can extract the URLs with those specific features related to malicious URLs which then can be used to train the model for better detection of malicious URLs.
* URL Feature Analysis will be another objective which will be carried out after the feature extraction is done which will help to drop out the URLs which does not have the specific features or which seem to have NAN values.
* URL Feature Reduction can then be done to improve the efficiency of the system which will reduce the features that were extracted and analyzed by using the datasets which contains both legitimate and malicious URLs.
* Scenario Categorization will be another objective to fulfill into finding out the differences that it would cause for the features as well.

# **2 METHODOLOGY**

The methodology used for the Event Driven Malicious URL Extractor will be explained below where event driven would propose the use of proper scenario categorization and detecting malicious URLs would be done with the aid of feature reduction which would enhance the speed of the process that should be carried out to filter malicious URLs by properly distinguishing them between the legitimate URLs and the malicious URLs.

The term malicious URLs has become very common in the modern day society and it is a topic which is spoken a lot in the cyber security field and most security experts are still unable to find a solution that will last forever because of high velocity of the technology being developed and the attackers’ attack techniques would also develop day by day to the point where they could surpass these security mechanisms. The attackers are able to develop better attack tactics and mechanisms because they know how to find vulnerabilities that are present in even the security solutions. The attackers are able to generate a large number of malicious URLs which would help them when performing attacks and they are an important part of performing the most common attacks which most people fall for which are phishing attacks, drive by downloads, performing a redirection to bogus websites which are used to gain sensitive information from users by tricking them into thinking that these websites are legal, tricking users into installing malware or viruses or Trojans through drive by downloads, malvertising, cyber fraud and many more other attacks in which these malicious URLs are used for. However these attacks cause great losses to people which could even mean the people losing billions of dollars. [11]

## **2.1 Feature Classification**

URLs can be searched for properties which could be identified through a mechanism which uses lexical features to capture these properties and these properties are string based properties which can be found in most URLs. The main motivation for implementing URL-based features is to provide a better alternative to having a malicious URL that has a similar name. Also, since a URL can have numerous statistical properties, it is commonly used to extract them from a website. Lexical features are more focused on extracting features which are statistical properties and these statistical properties are length, number of sub-domains and many more which will be listed below. This is only a start to finding proper features for feature classification to identify and capture the malicious URLs and this is one feature set provided from the research referenced in [11]. Below is a categorization of the features according to the previously mentioned research:

* URL-related features such as the length of URL and the number of dots present in the URL [11]
* Domain based features such as the average length of the domain token or the number of those domain tokens [11]
* Directory related features such as the length of path token [11]
* File name features such as the length of the filename [11]
* Argument features such as the number of query parameters [11]

Apart from the five categories that were mentioned above, this research [11] had also presented a list of features that was used for their model and these provided features will be used to note down the list of features that helped in that particular research so that noting down it for our study would help us in narrowing it down to get the final list of features for proper feature classification.

The features that were listed are a set of important binary URL-based features which in that research were in actual fact a set of features proposed in a previous research and that researcher had stated that it is important to check for the number of redirections that could take place because most of these malicious URLs usually have a set of connected URLs, which the URL which is presented to the user, redirects them to and counting the number of these redirections is thought to be an important feature for the feature classification. A few other features are having an IP address or port number in the URLs. The following table lists these features which were presented in that research. [11]

Table 1-First set of malicious URL features [11]

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature Name** | **Type** |
| 1 | url\_len | Integer |
| 2 | url\_numbr\_hyphens\_dom | Integer |
| 3 | url\_numbr\_dom\_tokens | Integer |
| 4 | url\_path\_length | Integer |
| 5 | url\_filename\_length | Integer |
| 6 | url\_longest\_domain\_token\_len | Integer |
| 7 | url\_average\_domain\_token\_len | Integer |
| 8 | url\_longest\_path\_token\_len | Integer |
| 9 | url\_average\_path\_token\_len | Integer |
| 10 | url\_tld | String |
| 11 | url\_actual\_domain | String |
| 12 | url\_domain\_len | Integer |
| 13 | url\_actual\_hostname | String |
| 14 | url\_hostname\_len | Integer |
| 15 | url\_numbr\_dots | Integer |
| 16 | url\_numbr\_underscores | Integer |
| 17 | url\_numbr\_equals | Integer |
| 18 | url\_numbr\_slashes | Integer |
| 19 | query\_len | Integer |
| 20 | numbr\_query\_para | Integer |
| 21 | ip\_present | Binary |
| 22 | port\_present | Binary |
| 23 | absolute | Binary |
| 24 | subdomain\_present | Binary |
| 25 | url\_suspicious\_word | Binary |
| 26 | url\_suspicious\_tld | Binary |
| 27 | url\_count\_redirect | Integer |
| 28 | url\_domain\_contains\_word\_server\_client | Binary |
| 29 | url\_contain\_email | Binary |

In the same research there were some host based features which were listed by a few researched they had done and the following list are based on the domain of the URL. These features are specifically utilized to be geographical, expiration, updating date and many more features which will help in distinguishing between legitimate URLs and malicious URLs. Below are the 15 host-based features which were presented in the research [11]:

Table 2-Host based features (2nd set) [11]

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature Name** | **Type** |
| 1 | host\_age\_create\_mon | Integer |
| 2 | host\_age\_exp\_mon | Integer |
| 3 | host\_age\_update\_days | Integer |
| 4 | update\_date | Binary |
| 5 | create\_date | Binary |
| 6 | expiry\_date | Binary |
| 7 | zipcode | Binary |
| 8 | Status | String |
| 9 | org\_name | Binary |
| 10 | dnssec | String |
| 11 | city | Binary |
| 12 | state | Binary |
| 13 | country | Binary |
| 14 | whois\_server | Binary |
| 15 | referral\_url | Binary |

Another study proposed another set of features which are highly distinguishable and a set of features which has very less of a mutual connection and this was used in the feature classification task in that proposed study. There were 24 of these features that were used for that proposed model as given in the research at the reference [10]. These features, as discussed in the study are used to transform the chosen features into the feature vectors that they should be transformed in order to be used in their training model and we can see that specific set of features in the below two figures. [10]

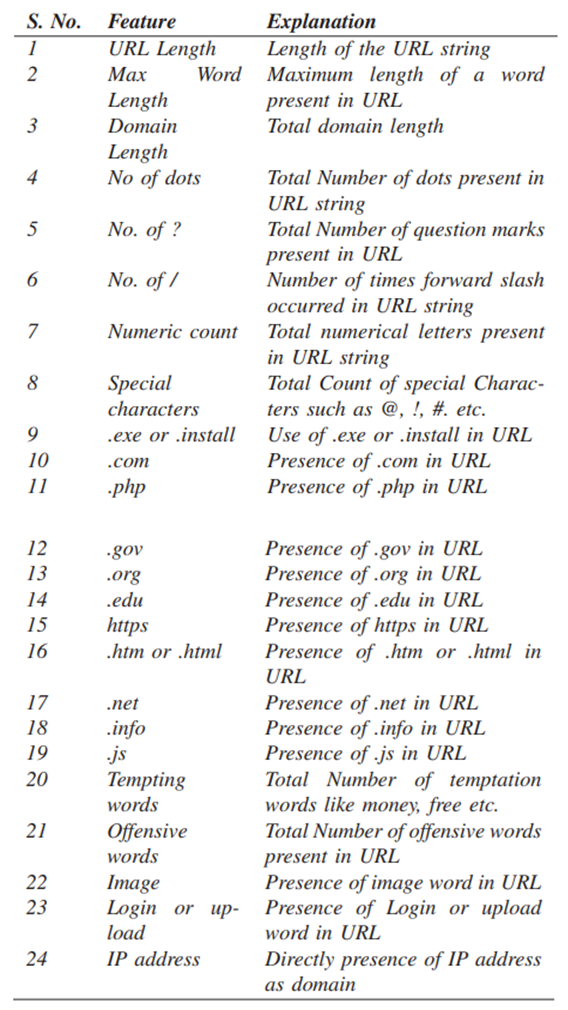


Figure 2 - URL Feature Classification (3rd set) - 1 [10]

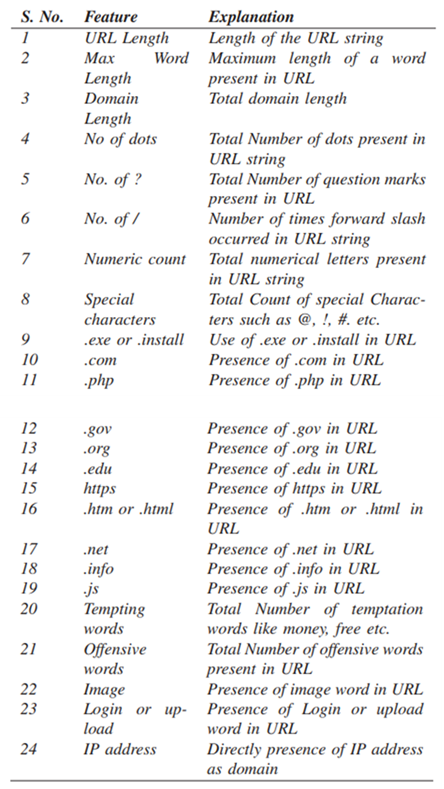


Figure 3 - URL Feature Classification (3rd set) - 2 [10]

The three sets of classified feature lists were extremely helpful in creating the final classification of the features. Feature classification needed for this model was possible because of the classified features that were provided from the researches that were performed in [10] [11]. However based on these classified feature lists the feature classification that was important for the modern solution that we propose was successful and the feature classification was done based on the information and with the aid of these researches. Below is the explanation for the feature classification that was created with the aid of the previous sets.

The features were categorized and classified as below after studying the feature sets that were found in the two researches mentioned previously. The categorization is as shown below:

1. Address Bar based features
2. Domain based features
3. HTML & JavaScript based features

### **2.1.1 Address Bar based features**

There are so many features that could be considered as the address bar based features but these specific features were chosen from the previous lists that were shown earlier.

* Domain of the URL – This is a feature is just the domain of the URL and the domain of the URL will be extracted in feature extraction but this is also a feature which does not have much significance in providing a good result and this feature could be dropped later after training the model and checking for the results.
* IP Address in URL – If there is an IP address present in place of a domain name, it could definitely mean that the URL is malicious and that the intention of using the URL is to steal personal information.
* “@” symbol in URL - The presence of the '@' symbol in the URL is checked. When the “@” symbol is used in a URL, the browser ignores anything before the “@” symbol, and the genuine address is commonly found after the “@” symbol.
* Length of the URL - Attackers can disguise the suspicious section of a URL in the address bar by using a lengthy URL. If the length of the URL is higher than or equal to 54 characters, the URL is categorized as malicious.
* Depth of the URL - Based on the '/', this feature estimates the number of subpages in the provided address.
* Redirection “//” in URL - The existence of "//" in the URL is checked and its presence in the URL route indicates that the user will be redirected to another website. The position of the "//" in the URL will be checked. If the URL begins with “HTTP,” the “//” should be placed in the sixth position. If the URL uses “HTTPS,” however, the “//” should occur in the seventh place and if it is found anywhere other than the said positions then the URL is considered as malicious.
* “http/https” in the Domain name - The existence of "http/https" in the domain section of the URL is checked. To deceive users, attackers may append the “HTTPS” token to the domain portion of a URL.
* Using URL shortening services such as “TinyURL” - URL shortening is a way of reducing the length of a URL while still directing to the desired webpage on the "World Wide Web." This is achieved by using a "HTTP Redirect" on a short domain name that connects to a webpage with a lengthy URL.
* Prefix or Suffix “-” in the Domain name - In legitimate URLs, the dash sign is used in very rare cases or not at all. Attackers frequently attach prefixes or suffixes to domain names, separated by (-), to give the impression that they are actually on a legitimate website.

### **2.1.2 Domain based features**

The following features were the features that we could obtain for this category to aid us in distinguishing between legitimate URLs and malicious URLs.

* DNS Record – In case of malicious URLs, the WHOIS database either does not recognize the stated identity or there are no records for the hostname. If the DNS record is empty or not discovered then this URL is said to be malicious.
* Website Traffic - This feature determines the popularity of a website by counting the number of visitors and the pages they view. However, because malicious websites are only active for a little time, the Alexa database (Alexa the Web Information Company., 1996) might not recognize them. In the worst-case scenario, genuine websites placed within the top 100,000. Furthermore, it is classed as "malicious" if the domain receives no traffic or is not recognized by the Alexa database.
* Age of Domain – This feature may be obtained from the WHOIS database. The majority of malicious websites only exist for a short time. For this initiative, a valid domain must be at least 12 months old. The term "age" refers to the difference in time between creation and expiration.
* End Period of Domain – This is also a feature that can be obtained from WHOIS database. The remaining domain time for this feature is determined by calculating the difference between the expiration time and the current time. For this project, we have set the legitimate domain's end time to be 6 months or less.

### **2.1.3 HTML & JavaScript based features**

Below are the features that could be found under this category which could be useful for this project.

* IFrame Redirection - iFrame is an HTML tag that allows you to insert another webpage inside the one you're now viewing. The “iframe” tag can be used by attackers to make the frame invisible, in other words, without frame borders. Attackers utilize the “frameBorder” property in this case, which causes the browser to generate a visual demarcation. It is considered malicious if the iframe is empty or if the response is not found.
* Status Bar Customization - Attackers can display a bogus URL in the status bar by using JavaScript. To get this feature, we'll need to look into the webpage source code, specifically the "onMouseOver" event, and see whether it affects the status bar in any way. If onmouseover is detected or if the response would be empty, then it is considered to be malicious.
* Disabling Right Click - Attackers block the right-click function with JavaScript, preventing visitors from seeing and saving the webpage source code. This feature is handled in the same way as "Hiding the Link using onMouseOver." We'll look for the event "event.button==2" in the webpage source code and see if the right click is disabled for this functionality. If the response is empty or onmouseover not detected, then it is considered malicious.
* Website Forwarding – The number of times a website has been redirected is one way to distinguish between a legitimate URL and malicious URL. Legitimate URLs, in most cases have been redirected for maximum only once, but malicious URLs have been redirected at least 4 times according to this feature.

This is the categorization of the features and the list of features which were found after researching through few proposed models and researches which had been effective after the implementation and therefore these highly discriminating and uncorrelated features will be used in the next part of the methodology which is feature extraction which will be done on both the datasets containing both the legitimate URLs and the malicious URLs.

## **2.2 Feature Extraction**

The feature classification which was done previously provided the classification of the features and the list of features to extract from the datasets that we will be using for this project which had been used in most of the studies and researches done previously. The datasets that would be used will be two datasets where one dataset will be containing legitimate URLs, spam, malware and defacement URLs and then the other dataset would contain phishing URLs. The dataset containing a collection of legitimate, spam, malware and defacement URLs will be the UNB dataset where UNB in actual fact stands for University of New Brunswick and this was created by a study done in that specific university therefore it gets the name from that. The dataset which contains the phishing URLs will be downloaded from PhishTank. The two datasets will be loaded into separate dataframes for feature extraction. Python based codes would be used for this whole task after feature classification is done. For the testing process however 10 random URLs from each dataset will be chosen and then feature extraction, feature analysis and reduction will be checked on them. However after feature extraction is done, the sorted out URLs which are malicious are once again written into two different “.csv” files as they are from two different datasets which has phishing URLs and a collection of other type of malicious URLs and legitimate URLs as well and then they are loaded into a dataframe once again for feature analysis. Feature analysis will later be discussed. Following will be the codes, procedures and explanations on how the dataset loading, feature checking and feature extraction is done.

Phishing URL dataset should be downloaded from PhishTank and then it can be loaded in to the program and then it will be loaded into a dataframe.

#Getting the phishing URLs

data0 = openRead ("url-data.csv")

columns of data0 = ['urls']

OUTPUT first 5 lines of data0

The legitimate URLs would also be taken in the same manner.

#Getting the files containing the legitimate and other malicious urls

data1 = openRead("Benign\_list\_big\_final.csv")

columns of data1 = ['URLs']

OUTPUT first 5 lines of data1

Feature extraction is then done after checking for the previously classified URL features. The URLs would be checked for each of these features one by one as it was previously mentioned. An “If …else” condition was used in order to capture those features from the URLs. If a URL is found with a malicious feature then the returning value for that malicious URL would be a binary value and the returning value will always be 1 for any malicious URL containing the features which indicates of the maliciousness of the URL. If it is considered a legitimate URL, then the returning value would always be a zero (0). The feature extraction function would then perform its process of feature extraction on the previously checked features on both the datasets of the malicious URLs and legitimate plus other malicious URLs.

This function is a function which would call all the other functions which were used to identify and extract all the features which were previously classified in feature classification. This function creates a list where all the features that were extracted would be stored. The function uses the extracted features of the URLs and then it appends it to the list that is created from the function.

The following text box contains the function which will create the extracted feature list from all the features extracted from the URLs through the other functions which are used for each feature. We can see the code calling each and every function which was used to extract each and every feature and then we can see the code using the append utility to append the features of the URLs to the list that would be created from this function. The coding is done in python language.

#Computing URL Features

#Extracting Features

function featureExtraction(pass in: url,label)

features = []

#Address bar based features (10)

Add to features list (call function getDomain(url))

Add to features list (call function havingIP(url))

Add to features list (call function haveAtSign(url))

Add to features list (call function getLength(url))

Add to features list (call function getDepth(url))

Add to features list (call function redirection(url))

Add to features list (call function httpDomain(url))

Add to features list (call function tinyURL(url))

Add to features list(call function prefixSuffix(url))

#Domain based features (4)

dns = 0

if

domain\_name = use whois to check registered owner(call function urlparse(url).netloc)

else

dns = 1

Add to features list (dns)

Add to features list(call function web\_traffic(url))

Add features list(1 if dns == 1 else call function domainAge(domain\_name))

Add to features list(1 if dns == 1 else call function domainEnd(domain\_name))

# HTML & Javascript based features (4)

if

response = get requests (url)

else

response = ""

Add to features list (iframe(response))

Add to features list(mouseOver(response))

Add to features list(rightClick(response))

Add to features list(forwarding(response))

Add to features list (label)

return features

There will be two more functions which would be used for the feature extraction in malicious URLs. This will create two separate lists according to the main function that will be used in these two functions therefore creating an extracted features list separately for both the legitimate URLs and the malicious URLs.

The features will be extracted from the legitimate URLs and the malicious URLs as shown below and then they will be stored in a list as it calls the feature extraction function. And then these lists will be converted to dataframes.

#Extracting features & storing them in a list

legi\_features = []

label = 0

for i in range from 0 to 10

url = legiurl['URLs'][i]

add to list of legi\_features(call function featureExtraction(url,label))

#converting the list to dataframe

feature\_names = ['Domain', 'Have\_IP', 'Have\_At', 'URL\_Length', 'URL\_Depth','Redirection', 'https\_Domain', 'TinyURL', 'Prefix/Suffix', 'DNS\_Record', 'Web\_Traffic', 'Domain\_Age', 'Domain\_End', 'iFrame', 'Mouse\_Over','Right\_Click', 'Web\_Forwards', 'Label']

legitimate = call python “pandas” package and create DataFrame(legi\_features, columns= feature\_names)

OUTPUT first 5 lines of legitimate

# Storing the extracted legitimate URLs features to csv file

Convert dataframe legitimate to csv with name 'legitimate.csv'

As shown above, two dataframes which are formed from the feature extraction that was done through legitimate URLs and from the feature extraction of the malicious URLs was shown. These two dataframes would then be concatenated into a single dataframe and then it will be exported as a “.csv” file for training purposes. This will be the final dataset which would be used for the training. And the number of URLs used for feature extraction through this coding would be changed to 5000 URLs which would provide us 10000 URLs in the final dataset. For the first test run, it will only use 10 random URLs from each dataset which would form a final dataset of 20 URLs with the extracted features.

#Extracting the features & storing them in a list

malicious\_features = []

label = 1

for i in range of 0 to 10

url = maliciousurl['urls'][i]

add to list of malicious\_features(call function featureExtraction(url,label))

#converting the list to dataframe

feature\_names = ['Domain', 'Have\_IP', 'Have\_At', 'URL\_Length', 'URL\_Depth','Redirection', 'https\_Domain', 'TinyURL', 'Prefix/Suffix', 'DNS\_Record', 'Web\_Traffic', 'Domain\_Age', 'Domain\_End', 'iFrame', 'Mouse\_Over','Right\_Click', 'Web\_Forwards', 'Label']

malicious = call python “pandas” package and create DataFrame (malicious\_features, columns= feature\_names)

OUTPUT first 5 lines of malicious

# Storing the extracted malicious URLs features to csv file

Convert dataframe malicious to csv with name 'malicious.csv'

## **2.3 Feature Analysis**

Feature analysis is the next task to be accomplished in the process to achieving the main goal which is feature reduction. Feature classification is completed, then the feature extraction was also done and the next to be done is the feature analysis. This is a part of the feature reduction which is a work in progress. The dataset that was exported as the finalized set of URLs and its features will be loaded back to the code which does the feature analysis. It will be loaded into a dataframe afterwards. Then it will check for the NaN values in the rows. If any rows are found with NaN values then those rows would be dropped. NaN stands for “Not a number”, and it is a numeric data type which is used to represent any value that is not defined or a value which cannot be presented properly. For example, 0/0 is not defined as a real number and is, therefore, represented by NaN.

file = 'finalurldata.csv'

df = openread (file)

# Removes all rows if they contain NaN values

Drop missing values in df (where axis='index', inplace=True)

As shown above it will remove rows with the NaN values after loading it into a dataframe.

Feature analysis is done to see how many features can be seen in these URLs and according to that the feature reduction could be done. However the coding was also made to check how many of each of the features were found in the 20 URLs that have been randomly chosen. Where 10 URLs were randomly chosen from the dataset containing a collection of legitimate, spam, defacing and malware URLs and another 10 URLs were randomly chosen from the dataset of the phishing URLs.

Below will be the coding used to check for how many of the features could be found on the set of 20 URLs which consists of both malicious URLs and the legitimate URLs as well.

haveIP = sum of df['Have\_IP']

print ('Have\_IP' with the value of variable haveIP)

haveAT = sum of df['Have\_At']

print ('Have\_At' with the value of variable haveAT)

urlLen = sum of df['URL\_Length']

print ('URL\_Length' with the value of variable urlLen)

urlDepth = sum of df['URL\_Depth']

print ('URL\_Depth' with the value of variable urlDepth)

redir = sum of df['Redirection']

print ('Redirection' with the value of variable redir)

httpsDomain = sum of df['https\_Domain']

print ('https\_Domain' with the value of variable httpsDomain)

tinyUrl = sum of df['TinyURL']

print ('TinyURL' with the value of variable tinyUrl)

presuf = df['Prefix/Suffix'].sum()

print ('Prefix/Suffix',presuf)

dnsrec = df['DNS\_Record'].sum()

print ('DNS\_Record',dnsrec)

webtraf = df['Web\_Traffic'].sum()

print ('Web\_Traffic',webtraf)

domAge = df['Domain\_Age'].sum()

print ('Domain\_Age',domAge)

domEnd = df['Domain\_End'].sum()

print ('Domain\_End',domEnd)

iframe = df['iFrame'].sum()

print ('iFrame',iframe)

mouseOver = df['Mouse\_Over'].sum()

print ('Mouse\_Over',mouseOver)

rclick = df['Right\_Click'].sum()

print ('Right\_Click',rclick)

webfor = df['Web\_Forwards'].sum()

print ('Web\_Forwards',webfor)

label = df['Label'].sum()

print ('Label',label)

## **2.4 Feature Reduction**

The translation of data from a high-dimensional space to a low-dimensional space so that the low-dimensional representation preserves some relevant features of the original data, ideally closer to its fundamental dimension, is known as dimensionality reduction or dimension reduction or as of this project, it is the feature reduction. Working with high-dimensional environments is inconvenient for a variety of reasons for instance, raw data is generally limited as a result of the problem with dimensionality, and interpreting the data is typically computationally difficult. Signal processing, voice recognition, and bioinformatics are all examples of areas that use dimensionality reduction to cope with huge numbers of observations and/or variables. Feature reduction can be done in many ways. There are a few python based methods that are recommended to use when the reduction needs to be done for a large amount of features where a feature selection should be done. These methods are Missing Value Ratio, Low Variance Filter, Random Forest, Backward Feature Extraction and Forward Feature Selection. [12][8]

### **Feature Reduction Techniques**

#### **Missing Value Ratio**

When there are missing values in a variable, if the threshold is more than 50% or a specific set percentage, then the variable should be dropped. This is how this approach will solve the problem. [10]

#### **Low Variance Filter**

When there are variables which has similar values as another variable can be identified and dropped accordingly. [10]

#### **Random Forest**

In random forest technique, we check for the most important features in the dataset. The importance of every feature is then checked to know which features to keep and use. The top most features will be kept. This is one of the most commonly used techniques. [10]

#### **Backward Feature Elimination**

This is done by first taking all the variables and then the model is trained, then the performance of the model is checked. Then one at a time these variables are dropped and the performance is checked each time. The variables that cause the slightest or no change at all is dropped. [10]

#### **Forward Feature Selection**

Each feature is checked with the model. If n number of features are there then the model is checked with the features separately n times. The feature that gives the best performance is used as the starting feature and then other features are added one at a time. This is done until there is no change in the performance. [10]

Both Backward Feature Elimination and Forward Feature Selection is time consuming and a lot of computational power is required to carry out these two techniques therefore it would be best to eliminate these techniques. We can look deeper into the other few techniques to see how compatible it would be with the ensemble model that will be built. [10]

These were the previously checked techniques and according to them, for this project, we have used random forest technique where we selected several features from different researches and then the top most features were extracted out of them to use for the training of the model in the end.

Now that the feature analysis is done, there is a way to find out which features are the features that could be found the most in malicious URLs. After checking for the commonness in the features we can keep or eliminate the features according to that.

## **2.5 Scenario Categorization**

According to a few websites that were found throughout the internet we were able to find out a few common scenarios of where malicious URLs would come in handy for the attackers and we can see them below.

### **Deactivation scares**

This is a lure that frequently works on people since nothing alarms individuals into responding speedier than a deactivation notice and you might have gotten one of these as well. Everyone almost every day gets an email pretending to come from an organization that we might – or might not – belong to or have used before. The email would claim that your account will be deactivated soon so that it will ask you to follow a link to take immediate action and will prompt you to enter your logon name and password and further, if the email pretends to be from a bank then it might ask you to update your credit card then the stolen credit card information is used to commit further crimes. These could be easily spotted but nowadays they look very realistic and they would even attach real links to the company that they are claiming to be from and also will include warnings about scammers and reassure the individuals by including notices that say that this email was scanned and cleaned by an antivirus software. It’s easy to ignore these phishes if you don’t have an account with the companies they claim to represent but if you do have then it is better to check by going to the legitimate site by manually typing its URL. [7]

### **Look-alike websites**

The links that phishing emails contain redirects the victims to websites that seem very legitimate and it has been difficult to spot the difference between a fake site and a real site. The fake websites has become spot on copies of legitimate websites that are used to trick users. The URL has a part of the real URL of the real website as well so that it looks real. If you looked closely at the link you can see that it points to a different domain but this is not the case when people click on the links because the website looks exactly like the real one. [7]

### **SEO Trojans**

Another common example is SEO Poisoning which attracts victims by showing up at the top of the search results as an advertisement or as a solution to what you have searched. If the victim clicks on this link then the legitimate looking website will lure the victim to download a software saying that it will fix the problem and victim is drawn to trust that website and will end up downloading that software which has malicious code in it or it is most probably malware. [7]

### **Craigslist money scams**

Another target of attackers is Craigslist. People tend to show up at them more and are willing to share their personal details, exchange money and or click on links. It happens mostly when people go there to sell. A buyer would appear immediately and will offer to pay the full price and buy it and that buyer would say that they would pay more if the seller is willing to use the buyer’s trusted intermediary to handle shipping and other payments. They will first hand over a large check and will ask you to remove your portion and forward the rest to the intermediary and then later the bank would return the check because it is bogus and now the seller will be accused for sending fraudulent funds. [7]

### **Work mules**

Job hunters have become another target of these attackers where the attackers would offer jobs on sites and social media and lure the victims to work for them. In the beginning they will give legitimate looking tasks which are not really legitimate and pay them well and the victims definitely fall for it thinking they have finally got a good job that pays them well. Victims do not understand that they are being conned. The attackers will ask the victims to withdraw the funds that the attacker deposited on the victim’s bank account and transfer it to somewhere else while keeping some amount for themselves. Sometimes they will ask the victim to convert the money to Bitcoin first and some are asked to do more work to make it seem like real work. [7]

### **SMS phishing**

Cell phones used to be safe from spam and phishing but not anymore. Nowadays we get spam voice calls or text messages which are known as smishing and vishing. Smishing may not sound realistic but a lot of people fall for it. If the victim clicks on the link then the victim will be prompted to install a Trojan software or will be asked to call a number and most commonly these messages will be about a compromised credit card and will ask you to give your details like credit card number and then the victim is scammed. Vishing happens in the same manner but it is directly a voice call scamming the victim. [7]

### **Fake crisis notices**

Phishing scammers usually use rely on deception and creating a sense of urgency to achieve success. Crises such as the coronavirus pandemic amplify the sense of urgency and provide new opportunities for deception. People are on edge. They want information and look to their employers, the government, and other relevant authorities for direction. An email that appears to be from one of these entities and promises new information or instructs recipients to complete a task quickly will likely receive less scrutiny than prior to the crisis. [7]

### **Compromised Credit Card**

In this method the attacker will use some known personal details of the victim to trick him and get further information. For example the attacker might know that the victim recently made a purchase from a popular vendor, and the attacker will pretend to be a personnel from the customer support of that organization and send an email to the victim with the notice that the victim’s credit card is compromised and will ask to take immediate action to protect the victim’s account. [6]

### **Social Media Request**

People usually accept friend requests from a person that has mutual friends or is followed by one of their followers. If the attacker sends a friend request to the victim which the attacker has the same Facebook friends as the victim then the victim tends to think that the person is someone whom the victim might know. Then the attacker will send a link or some kind of attachment which will make the victim download or click on it and the victim ends up downloading and eventually installing malware to their PC and if this was downloaded to a workplace computer then it will potentially spread through the company network as well. [6]

### **Fake Google Docs Login**

Cyber criminals also send phishing emails that would contain a link to a fake Google Docs login page which tricks the victim to login to the attacker’s fake website. For example the phishing email might read “We’ve updated our login credential policy, please confirm your account by logging into Google Docs.” The attacker will also have created a fake Google email address to trick the victim into believing it. [6]

## **2.6 Commercialization**

One part of the targeted user base for this system is small-medium enterprises (SME), which has higher usability and a lower cost. The system can be commercialized on the note of providing basic security.

Two versions of this system can be implemented.

* A free version that SMEs and Researchers can use to collect URL lists based on events with limited export capability.
* A paid version that will provide seamless export capability in addition to the basic endpoint protector using the browser plugin.

# 

|  |  |
| --- | --- |
| **Free Version** | Rate Limit on event-based malicious URL list and restricted export capabilities. |
| **Paid Version** | Browser plugin to protect users from malicious pages with basic reporting to the administrator  No limit on export and event-based malicious URLs |

## **2.7 Testing and Implementation**

Feature reduction is done under 4 steps which is feature classification, feature extraction and feature analysis. In the methodology I explained the coding which is used for feature extraction and feature analysis. The processes followed in feature classification, feature extraction and feature analysis helps in having a proper understanding about the features and how they can be found on the URLs so that we can distinguish between malicious URLs and legitimate URLs. This will help with dropping the features which shows the lowest effect or result on the process thus improving and achieving feature reduction.

Feature classification is already done and there is not testing needed because the classified features are the features used for the feature extraction and analysis. The only part to cover is the feature extraction and feature analysis steps in this process and for testing the code and for testing and extracting the feature, datasets which contains legitimate URLs and malicious URLs should be downloaded so there will be two separate datasets used for this process.

Datasets for the testing purposes would be downloaded from two different sites. PhishTank contains the dataset that has the malicious URLs in it. The other dataset is the dataset which is actually the UNB dataset that contains a collection of spam, legitimate, malware and defacement URLs which will be used for the testing purposes. As it was explained before these two datasets will be loaded in to the program and then they will be loaded separately into two dataframes which will then be used for the feature extraction purposes. There are several functions which are written to extract each and every feature from these URLs.

The coding is done in python language and Jupyter Notebook was used to write and run the coding. Jupyter Notebook can be used through a platform that provides various other benefits and coding uses such as the Anaconda Navigator. The coding was written from loading datasets in to a dataframe, to the little functions which were done to extract the features and then exporting the next created data lists, in separate lines which is one feature of using Jupyter Notebook and then it can be used to run each line of coding and see if it runs properly or if it shows any errors. With this benefit we can easily spot any errors on the coding and work on them separately rather than running a whole code and not knowing what went wrong or why it went wrong.

It even lets us download and install python utilities which are missing or which will be needed to update which is an advantage of using this platform and it is a recommended platform to be used when coding in python language which is also easier to use and the interface is user friendly as well.

# **3 RESULTS AND DISCUSSION**

## **3.1 Results**

Results for both the testing processes done on feature extraction and feature analysis will be shown here which ultimately leads to feature reduction. The previously shown coding will be done line by line to extract the features and then to analyze them.

### **3.1.1 Feature Extraction**

The datasets will first be loaded into a dataframe as shown below. One dataset contains a collection of both legitimate and malicious URLs and the other dataset contains phishing URLs.

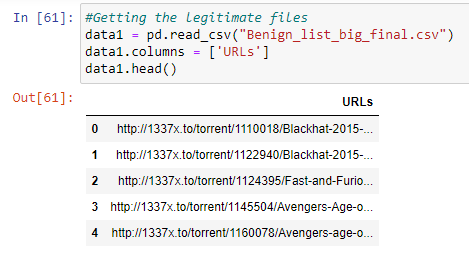


Figure 4 - Feature Extraction - Loading the legitimate plus malicious URL set

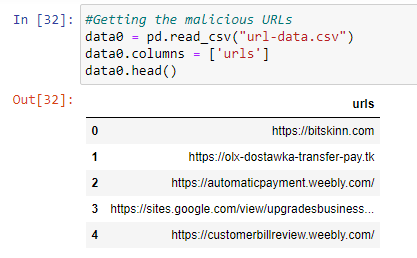


Figure 5 - Feature Extraction - Loading the phishing URL set

Random set of URLs will be collected from each dataset. 10 URLs will be collected from both the datasets which will be a total of 20 URLs and then the feature extraction will be done on these URLs separately on the two sets of URLs.

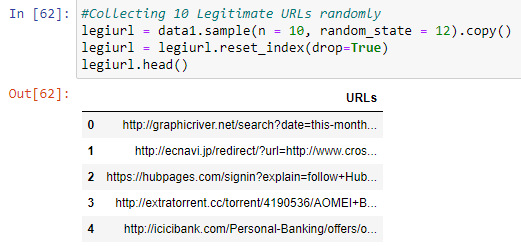


Figure 6 - Feature Extraction - Collecting random URLs (Legitimate and malicious)

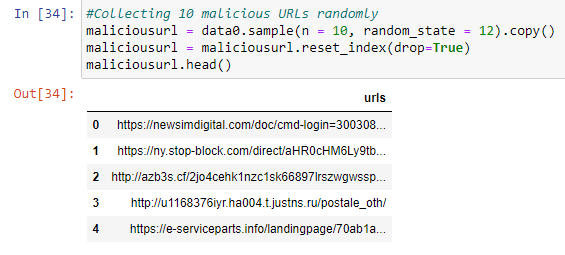
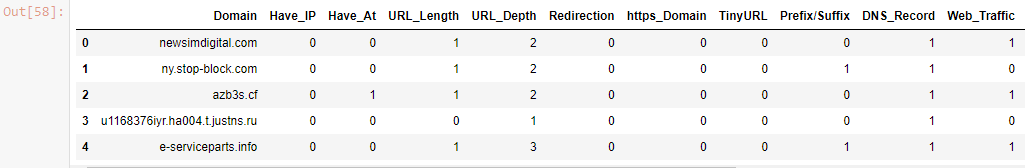


Figure 7 - Feature Extraction - Collecting random URLs (Phishing)

The features will be extracted and then they will be stored in a different dataframe again as shown below. Both the screenshots were taken as two screenshots because the table was too long to capture.



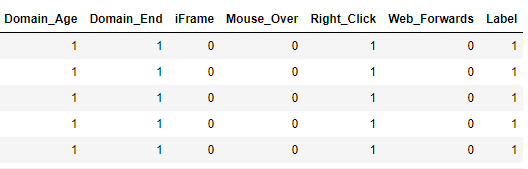


Figure 8 - Feature Extraction - Storing features of phishing URLs in a dataframe -1

Figure 9 - Feature Extraction - Storing features of phishing URLs in a dataframe -2

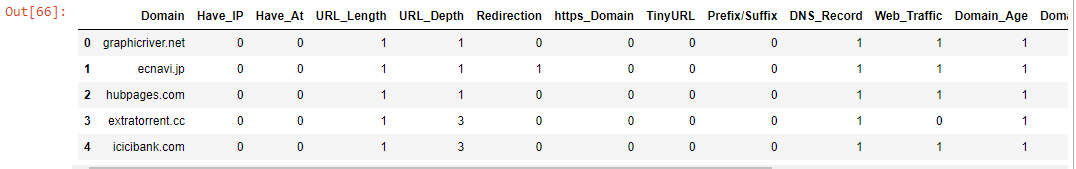
****

Figure 10 - Feature Extraction - Storing features of legitimate and malicious URLs in a dataframe -1

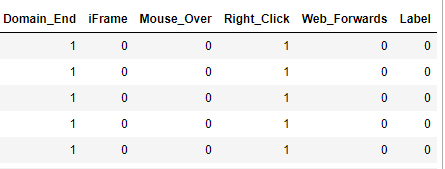
****

Figure 11 - Feature Extraction - Storing features of legitimate and malicious URLs in a dataframe -2

The two dataframes would be concatenated into one dataframe after this and then they will be exported as a “.csv” file for using as the final dataset to be used for feature analysis.

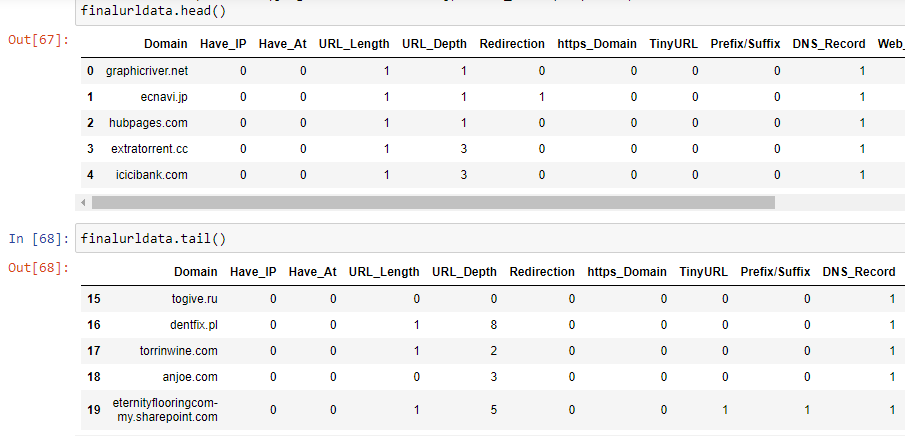


Figure 12 - Feature Extraction - Final concatenated dataset

### **3.1.2 Feature Analysis**

Feature analysis is done with the aid of the final dataset which was created from feature extraction. The concatenated set of all malicious URLs with their feature extraction is available in that dataset and then will be used to analyze the features in the coding which was created for the feature analysis.

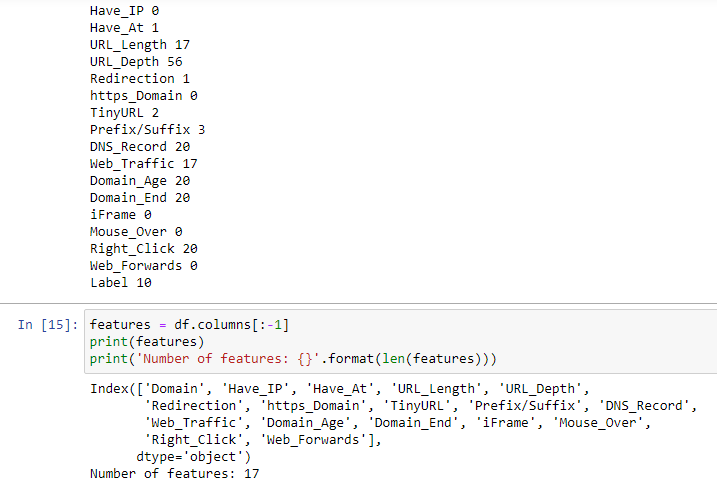


Figure 13 - Feature Analysis - Output after the analysis

This is the final output of the feature analysis process and this helps us to understand which features were most common among the malicious URLs and how many of them could be found finally after the deletion of row which contained NaN values. As we can see above we are able to see how many URLs out of the 20 selected random URLs contained which features. If there is any feature which has a zero (0) value of no effect on the dataset at all then they will be removed thus achieving feature reduction where we used random forest method to achieve it.

## **3.2 Research Findings**

The final findings of the project as we are able to see are that the URLs which are malicious URLs have specific features that actually help us distinguish between the malicious URLs and the legitimate URLs easily.

First a few studies which explained about proper feature classification helped us achieve the categorization of features and the classification of features of the malicious URLs which we cannot find or can rarely be found in legitimate URLs. These features were classified after a thorough study of 3 sets of classified URL features which were found in previous studies and researches done by other researchers who has much more experience and capability in understanding and finding proper ways to classify the best and the most specific features which helps to distinguish between legitimate URLs and malicious URLs.

Feature extraction helped in extracting the classified features from the URL datasets and it helped us identify the URLs which contained the exact features and they were malicious URLs.

Feature extraction aided in extracting the features that were classified earlier and then they were extracted using functions to extract each of the features and as expected it provided the output with the required return value which was a 1 for malicious URLs and returning 0 (zero) for legitimate URLs. It was also checked for NaN values and the rows were deleted if they had any NaN values. Then with this output, feature analysis was able to sum up and give a count of each feature which showed the commonness of the features.

By checking for the commonness in features we could perform feature reduction where features would be removed if it had no effect and a sum of zero on the feature extraction and analysis. This is how feature reduction was able to be achieved effectively.

## **3.3 Discussion**

The whole purpose of performing this research and working on this project is to implement a better model and a system which would capture malicious URLs effectively and efficiently. One challenge when making this model was the challenge of working with a huge amount of URLs which are being produced each day and attackers are also capable of producing too many URLs for malicious purposes therefore to work with those URLs efficiently we needed to find a set of features which would help us distinguish between malicious URLs and legitimate URLs.

While doing this project I was able to get an insight into how URLs work, the parts of a URL, the specific features of a URL which would help us distinguish between a malicious and a legitimate URL and to understand what kind of information about a URL can we extract from it.

This kind of information and knowledge which was obtained from various studies and researches were able to help us face the challenges in doing this project because studies and researches which discussed about feature reduction was very less but with the limited resources and knowledge we could come up with a proper feature reduction strategy and process thus feature reduction was successfully achieved.

## **3.4 Summary of each Student’s contribution**

### **IT18071412 – S.W. Jonathan**

Main responsibility was to build the Ensemble model by creating and testing multiple diverse modeling algorithms to predict malicious URLs with improved accuracy without resulting in decrease in performance of the system. The following algorithms were tested by the student to find the ideal algorithm to be used in the system, which was concluded as the Bidirectional LSTM + CNN.

* Recurrent Neural Networks (RNNs) + Convolutional Neural Networks (CNNs)
* Artificial Neural Network (ANN) + K-Nearest Neighbors (KNN) + Decision Tree (C4.5) + Random Forest Classifier (RFC)
* Long-Short Term Memory [LSTM] + Convolutional Neural Networks (CNNs)
* Monotone Multi-Layer Perceptron Neural Network, Multi-Layer Perceptron, Neural Networks

In order to test these algorithms, datasets were acquired as a sub objective from public open sources; PhishTank, OpenPhish, Common Crawler and Alexa.

Another sub objective includes building the website using the Python Flask Framework with following UI features,

* Provide data visualization to viewers
* Export event-based malicious URL data lists in .CSV Format

### **IT18021912 – Ramanayaka A. M**

As the main objective, feature reduction of the collected URLs from the ensemble model was performed by the student. In order to achieve this, there are three main tasks to be done which is feature classification, feature extraction, feature analysis and then the feature reduction will be performed.

Feature classification was done through a thorough study of a few sets of features which were classified in previous researches and studies and then a final classification was done to obtain the most effective features with feature classification.

Next, feature extraction was done on the URLs which were classified in feature classification where the URLs were checked for each and every feature and then were extracted from the URLs for the feature analysis.

In feature analysis, the output dataset from the feature extraction process will be checked for NaN values and the rows will be deleted accordingly. It will also provide an output of how common the features had been. Feature reduction is done using random forest method so the top most features were used for the training and testing and with that feature reduction is achieved.

### **IT18032666 – Wishvajith B.L.D.V**

As the main objective student’s main responsibility was to build a strategy to populate keywords from the user inputted keyword/query via the web interface of the system and compare those populated keywords with the identified malicious URLs from the ensemble model to find a potential match. In order to achieve this objective, two models were created with the following capabilities,

1. Keyword populator model

* Extracting real time tweets through Twitter API relating to the user input keyword
* Deriving keywords from the tweets
* Exporting of populated keywords as a .CSV file

1. Comparison model.

* Tokenization and stemming of the captured URLs
* Matching of the populated keywords with URL stems
* Export matching sets as a .CSV file

And tested with a sample dataset containing 549,346 URLs to test the accuracy of the models. Testing was also carried out to find the best strategy for the matching of the keywords with URLs by the student and concluded that the use of levenshtein distance to calculate the similarities was efficient.

### **IT18034400 – Renu Harshatha**

Student’s main objective was to install and integrate the Nomad with the developed malicious URL detection system to scale the overall system.

As a sub objective creation of the browser endpoint extension was carried out, with the objectives of,

* Monitoring capability to track user behavior ( URLs visited)
* Capturing and storing the URLs in JSON format using JavaScript
* Automated blacklisting of detected malicious URLs upon visiting

# **Conclusion**

Despite the fact that many previous research have been done on recognizing and blocking malicious URLs using various algorithms and techniques, they fall short of capturing the characteristics of malicious URLs early in the attacking cycle, along with overall efficiency. In this article, we presented an ensemble model as well as a browser plugin to detect fraudulent domain names including event-related keywords in order to address these difficulties.

Feature reduction was an effective way to produce an effective and efficient system which is able to process a large amount of URLs and the feature classification helped us in identifying the most specific features present in a malicious URL which lets us distinguish between legitimate and malicious URLs. Feature extraction and feature analysis also played a role in bringing out the best effect of feature reduction and the feature reduction which was done according to random forest method produced great results at the testing process.

Utilizing filled keywords from the user-inputted event keyword, our model was able to effectively detect and create a list of malicious domain names with an average accuracy of 98.1 percent. To do so, we trained and tested our model using 549,346 URLs gathered and processed via Kaggle. Unlike prior studies on the topic, our method looks to be a feasible option for decreasing computing time and overhead. This is due to its ability to detect malicious URLs with high accuracy using the ensemble model and map them with event-related keywords in real time, early in the attack lifecycle. Future research will focus on experimenting with other algorithms and technologies to improve the models' time efficiency.

# **References**

|  |  |
| --- | --- |
| [1] | Y. G. F. Z. Y. Y. W. W. a. T. Z. Ping Yi, “Web Phishing Detection Using a Deep Learning Framework,” *Hindawi,* vol. 2018, p. 9, 2018. |
| [2] | E. B. O. D. B. D. Ozgur Koray Sahingoz, “Machine learning based phishing detection from URLs,” *Elsevier,* p. 14, 2018. |
| [3] | C. L. S. C. H. H. Doyen Sahoo, “Malicious URL Detection using Machine Learning: A Survey,” 2019. |
| [4] | D. Warburton, “F5,” F5 Labs - Application Threat Intelligence, 11 November 2020. [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/2020-phishing-and-fraud-report#:~:text=The%20number%20of%20phishing%20incidents,at%20the%20access%20control%20layer.. |
| [5] | J. I. &. R. Islam, “Detecting Malicious Urls of COVID-19 Pandemic,” p. 17, 2020. |
| [6] | “19 Examples of Common Phishing Emails…And How to Avoid Them,” 2021 Terranova Worldwide Corporation, [Online]. Available: https://terranovasecurity.com/top-examples-of-phishing-emails/. |
| [7] | R. A. Grimes, “14 real-world phishing examples — and how to recognize them,” CSO Online, 9 April 2020. [Online]. Available: https://www.csoonline.com/article/3235520/15-real-world-phishing-examples-and-how-to-recognize-them.html#. |
| [8] | P. Sharma, “The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes),” Analytics Vidhya, 27 August 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/. |
| [9] | A. O. A. Andronicus A. Akinyelu, “Classification of Phishing Email Using Random Forest Machine,” *Hindawi,* vol. 2014, p. 7, 2014. |
| [10] | B. S. Yogendra Kumar, “A lightweight machine learning based security,” in *International Conference on Communication Systems & Networks (COMSNETS)*, 2021. |
| [11] | S. Garg, “Feature-Rich Models and Feature Reduction for Malicious URLs Classification and Prediction,” *Creative Components,* no. 179, p. 58, 2019. |
| [12] | “Dimensionality reduction,” Wikipedia, 17 February 2021. [Online]. Available: https://en.wikipedia.org/wiki/Dimensionality\_reduction. |