Asian Journal of Social Science and Management Technology
ISSN: 2313-7410
Volume 4, Issue 2, March-April, 2022
Available at www.ajssmt.com

Novel Approach to Secure Websites with Machine Learning Classifiers

Chetanpal Singh[1], Ishtiaque-Al-Mahmood[2], Mohimen-Al-Tahsin[3]

[1,] *Department of Information System Holmes Institute, Australia*
[2,] *School of Business and Tourism Southern Cross University, Australia*
[3,] *Department of Computer Science and Engineering Brac University, Bangladesh*

Abstract: The Internet has become essential and most for regular life. This ascent in the boundless utilization of innovation brought a rise in various problems. There have been so many loops and data breaches in web content that it's become straightforward for criminals to manipulate those security gaps and inject viruses or other unwanted and harmful contents. Almost every one of us faces the problem of getting trapped in malicious sites or clicking on some popped up advertisements, which ended up into absolutely an indecent web page that contain potential threats. As a result, most of our private data is being compromised or leaked, creating so many problems in our corporate and personal lives. So, it's become a top priority for security analysts and consults to ensure web security as it has become an integral part of every sphere. Keeping these facts in our mind, we have come up with a smart protection concept for websites that we regularly browse in everyday life. We aim to build a model that will prevent us from accessing various malicious sites, unwanted link redirection, and pornographic image contents that pop up quite frequently in the time of browsing webpages. To make our system detect these problems, we will implement machine learning and image processing algorithms and provide the users with a filtered, smooth, and safe user experience.

Index Terms: Malicious Site, Machine learning, Link redirection, Image processing

1. INTRODUCTION

The advancement in communication technologies and global networking has made routine activities like e-commerce, social networking, and electronic banking very easy. Still tactical, but at the same time, it has made all the data related to these activities available in cyberspace. Due to the uncontrolled and open internet architecture, computer systems and networks are always prone to cyber-attacks, a severe issue. Though some security risks can be prevented by using the networks with utmost care and taking the help of professionals, one cannot entirely escape from the threat of phishing scams. For performing a phishing attack, the attackers acquire personal details of the users so that they can manipulate the experienced user. Such phishing attacks that target end-users result in significant loss of personal or confidential information of the user, which sometimes also involves monetary loss. The Morris Worm is broadly considered the first Internet virus; however, its author did no longer make it with the motive of causing damage.

In 1988, Robert Morris, a graduate pupil at Cornell University, determined to measure the Internet. To test this, he created software that might implant itself in UNIX computer systems as it travelled around the usage of networking commands. Robert Morris has become the first person to be convicted of Computer Fraud and Abuse because his computer virus had caused lots of greenbacks in lost productiveness. He, without difficulty, admitted his quick sightedness, pronouncing he ought to have tested the bug 39s replication technique before sending it into the wild. But the massive use of technology brought an enormous surge in cybercrime (Yost and Jaiswal, 2017). This number of cyber-attacks is increasing daily at an exponential rate. According to the Symantec Internet Security Threat report 2019, the use of destructive malware by different groups grew by about 25% in 2018 (Symantec, 2019). Almost every one of us faces the problem of getting trapped in phishing sites or clicking on some popped up advertisements, which ended up in indecent web pages that we did not desire. As a result, most of our very private data is being compromised or leaked, creating so many problems in our corporate and personal lives. According to a Gallup study, the attacks are causing panic among users; according to a Gallup study, more than 70% of Americans are worried about losing personal or financial data by getting hacked (Gallup, 2021). According to the report, the business analyst claimed that a company spends approximately 2.4 million USD because of malicious attacks (Accenture, 2018). So, security analysts and consults have become the top priority to ensure security as cyber security has become an integral part of every sphere.

There have been so many loops and data breaches in web content; as a result, it's become straightforward for the criminals to break that security breach of web contents and inject viruses and other unwanted and harmful content. Sometimes they use nude pictures and animated videos to attract the user. When this unwanted content pops up to the web browser, the user is attracted by this pornographic content and presses this from their attraction. According to the reports from Symantec, there has been a connection between these pornographic images and malicious content (Symantec, 2010).

Not only that. It is also often seen that when we press a link, it sometimes changes the destination and takes us to another link. These kinds of link redirection are commonly used to redirect malicious websites that contain potential threats (OWASP, 2014).

Moreover, the random popping up of websites in our browser may indicate many things on our computer. We also see many online popping up advertisements when we open our browser or on any website. These online ads come in banners, text, pop-ups, images, or transitional formats. Advertisers began implementing more excellent intrusive techniques to attract users' attention (Krammer and Taylor, 2008). They lively motion pictures or pictures generate sounds or cover the main content to force user attention. Furthermore, these online ads have become a goal of malicious entities wishing to cause harm or achieve profit (Post and Sekharan, 2015).

To reduce distraction, malware infections, and unwanted link directories, smart protection systems can be a great option, though some use preinstalled pop-up blockers. Using a smart protection system, the pages that user's access are secure and free from internet threats, such as malware and phishing scams. It is also free from unwanted link redirection, designed to trick users into providing personal records.

There have been so many loops and data breaches in web content. As a result, it's become effortless for criminals to break that security breach of web contents and inject viruses and other unwanted and harmful content. Sometimes they use nude pictures and animated videos to attract the user. When this unwanted content pops up in the web browser, the user is attracted by this pornographic content and press this from their attraction. But these pornographic can also additionally cause pages with malicious threats. There is an interconnected relationship between pornography and the dissemination of malware. The random popping up of websites in our browser may indicate many things on our computer. We also see many online popping up advertisements when we open our browser or on any website. Not only that, but we also face many more problems while browsing any website. Sometimes, when we click on a website, it enters another website without entering that website. That's mean when anyone clicks on that main URL, they will be taken to another page instead. Keeping these facts in our mind, we have come up with a smart protection concept for websites that we regularly browse in everyday life. Using smart safety, we ensure that the pages that user's access are secure and free from internet threats, such as malware phishing scams. It is also free from unwanted link redirection, designed to trick users into providing personal records. So, we aim to build a model that will prevent us from various malicious data and unwanted and indecent content that pop up quite regularly when browsing or surfing webpages. Our supervisor helps us learn about the advancement of machine learning algorithms. He also suggests that we use some techniques that play an essential role to reduce link redirection of a website. We rely on web surfing or browsing all day for any random daily basic needs. So, any problem or uneasiness regarding web surfing can seriously impact productivity and peace of mind as they become intolerable. Some significant issues, such as malicious data malware, hamper our privacy by compromising our private data without our will.

Again, we see many inappropriate Ads and indecent pictures between the web content, which is very uncomfortable. So, we think nudity detection is constantly a significant issue of web indexes, person to person communication sites and other web channels. Yet, this issue is becoming more genuine these days since there is increasingly more measure of information. The significant issue is that individuals have perceived the calculation behind the sifting of nude pictures, which depends on text-based separating of picture labels and subtitles (Gavrilut et al., 2009). Moreover, these inappropriate things cover the web content as well. Then we also face link redirection to an undesired web destination almost every time we surf from one web address to another.

Phishing Attack has been a crucial research subject for researchers in recent years. The term phishing attack means fishing of the victims. It is an attractive technique used by attackers or phishers to lure users by opening fraudulent websites similar to legal sites that compel users to provide their personal or financial information. Such websites have the exact design and a graphical user interface similar to legitimate websites, but their Uniform Resource Locators (URLs) are different from the original webpage. Experienced and professional users can easily detect these fraudulent websites by recognizing the fake URLs. However, most of the time, due to lack of time or simply ignorance, the users do not carefully check the address of websites forwarded through an email message, other web pages, or social networking platforms. These web pages entered by the user are so similar to the original websites that the user provides his/her sensitive details to it without any doubt. When the unsuspecting users click on these fraudulent websites, the phisher or attacker gets access to their sensitive information like personal information, financial details, passwords or usernames, etc. Research has shown that the users fall into the trap of phishers mainly due to five reasons.

- The users have to lack detailed knowledge about the legal URLs.
- Due to hidden URLs or redirection, the users do not see the complete address of the webpage.
- The users do not know which web pages are trusted ones.
- The users enter into these webpages accidentally
- Or they are not able to differentiate the legal and fraudulent web pages

In cyber-attacks, even experienced users can also be targeted by the attackers with new techniques of attacking the networks. The experienced users can also believe the malicious website as the legal one and provide his sensitive information. Such attacks can cause significant harm to the end-users; hence, there is a need for additional support systems or technology to protect the computer networks and users from such attacks. These protection systems can either deploy some auxiliary programs for the users or increase their awareness of this aspect. In such a situation, the software-based phishing detection systems are trusted and implemented as Decision Support Systems for the users. Most widely used techniques in this category include Natural Language Processing, Machine Learning, Image Processing of the webpage, etc.

Machine Learning is one such technique in supervised learning which is implemented for developing predictive models. Machine Learning can manage the issue of phishing attacks by transferring it into classification. For training and evaluating the model, the labelled historical data of websites are used. This technique can detect phishing activities by integrating models into web browsers. It has been noticed that when new techniques are proposed to counter phishing attacks, the attackers also come up with some new forms of attacks after identifying the loopholes in the proposed solutions. Hence, it is recommended to deploy hybrid models to ensure network security instead of implementing a single technique. Also, an automated phishing attack detection model can be more effective if the features of websites are integrated with the input dataset derived from a large number of websites. These are the most common problems everyone faces, but there is hardly any software to deal with these issues. So, our aim is to solve these issues and make the web surfing e more experience more interactive and user friendly. The problems we discussed earlier needed different algorithms from different sections like we; if we have to deal with unwanted image detection, we have to implement an image processing method. Then for detecting malicious content and ensuring web security, we will use a machine learning algorithm and various web security protocols. For that, we have read so many research papers related to our project. Keeping all these issues in mind, we came up with a possible solution of introducing a model containing machine learning and image processing algorithms that can handle these above problems. Here is a brief idea of our plan to solve the above problems:

As we have just started working on our project, we are trying to figure out the possible algorithms to run our software. But we have made our goals very clear about what we are going to resolve. First of all, we will secure our web browsing from malware and malicious data by implementing a machine learning algorithm. We will provide an initial dataset on the existing cyber threats, and based on that, we will train our machine. Then for the new threats, our system will learn the datasets and act according to it.

Secondly, our system will keep track of the URL links of a requesting website. As all the information of a website is kept in the HTML file of that web page, we will trace the links and scan for all the unnecessary directories.

Furthermore, we will implement an image processing algorithm to detect inappropriate and indecent pictures or ads among all web content. We will implement a convolution neural network algorithm to scan for these unwanted contents, and after detecting those, our system will remove them or hide them from the web page. This paper focuses on protecting websites from such cyber-attacks by using different machine learning algorithms and image processing techniques. All in all, we still in our early period of our project so gradually we will try to integrate all of these features for our system and then implement them in our software.

2. Literature Review

There are three wings in our thesis. The wings are malicious site detection, scanning invalid link directory and nudity detection. No work had been done before, which has the combination of all three.

Malicious content has genuinely been a danger to individuals and relationships since the mid-1970s when the Creeper virus first appeared. Starting now and into the foreseeable future, the world has been persevering through a surge from a colossal number of different malicious contents varieties, all with the arrangement of causing the most unsettling influence and damage as could be expected under the circumstances. Malicious contents and their variations may change a great deal from content marks. They share some conduct highlights at a more elevated level which are more exact in uncovering the genuine plan of malicious contents (Rokkathapa and Kanrar, 2019). It is anything but difficult to identify the known malicious program in a framework, yet the issue emerges when the malicious contents are obscure. Since obscure malicious contents can't be recognised by utilising accessible known malicious contents marks. Mark based discovery procedures neglect to detect unknown and zero-day assaults. An epic methodology is needed to speak to malicious contents, including successfully detecting obfuscated, anonymous, and mutated malicious contents (Liu et al., 2019).

Sometimes we face problems like we are in a website and entering another website with a different domain. These websites or links having other domains are called the invalid link directory. In our thesis, we detect those false or invalid links and warn the user about that invalid link. The identification of malevolent site page's strategies incorporates a black-list and white-list approach. In any case, the black list and white list advancement are vain if a particular URL isn't in the list (Samantray et al., 2019).

Nudity detection methodologies assume a significant function in arrangements zeroing in on controlling admittance to improper substance. These systems apply channel or comparable methods to identify nakedness in computerised pictures *(*Fleck *et al., 1996)*. To achieve our goal, we need to introduce an image classifier algorithm that can classify an image and label it as safe or not safe. There have been many works published on pornographic image detection over time. Earlier, people used to measure nudity based on a human structure model (Forsyth and Fleck, 1996; Forsyth and Fleck, 1999; Duan et al., 2002)

In most cases, we have seen they have used simple backgrounds and completely naked people, making it very simple to detect and classify them; however, in reality, the images and backgrounds are very complex. So, those primitive methods don't work efficiently as per the expectations. Some different chips away at explicit picture identification have been distributed where they picked skin tone to recognise bare pictures against ordinary pictures (Jeong et al., 2004; Lee et al., 2007).

Additionally, there are some other progressed approaches on pixel-based technique plays out a looking through Region of Interest (ROI), using skin location, at that point performs highlights extraction in the ROI, for example, colour moment, histogram (Rowley, 2006; Daeef et al., 2016). Yet, it likewise can't give the expected outcome. We chose to utilise CNN for picture order. The convolutional neural organisation (CNN) as one of the techniques in profound learning has indicated prevalence in grouping these pictures.

There is some extensive literature on malicious website detection using machine learning algorithms pornographic images detection using image processing algorithms. The relevant papers which most influenced our work will be mentioned here. Through many approaches, nudity can be detected.

One of the pioneering works is done by Gavrilut et al. (2009). This paper presented a machine learning system for malware detection planning to get as barely any fake positives as possible by utilising a simple and essential multi-stage combination (cascade) of various adaptations of the perceptron algorithm. They were extremely near the objective, even though they have a non-zero fake positive rate.

In another research paper (Rokkathapa and Kanrar, 2019), he proposed a method if a massive amount of malicious files has been context to identify the best classifier and maximise QoE for malware detection sources. In extracted with the help of the cross-validation method in machine learning, one can classify malware samples, and those samples can be predicted about the maliciousness present in the sample. Malware classification using Behavioral specification is modelled under supervised learning. In this research paper, around 3000 datasets were experimented with, among which 1400 were malicious programs.

This type of work is done by Daeef et al. (2019). This paper creates a recognition framework with a comprehensive security scope utilising URL includes just relying upon users straightforwardly managing URLs to surf the web and gives a decent way to deal with malicious URLs as demonstrated by past studies. This paper proposes a framework called spam filtering, which can be coordinated into such a process to expand the detection performance in real-time. The proposed framework's simulation results demonstrated phishing URLs recognition exactness with 93% and gave the online process of a particular URL in an average season of 0.12 seconds.

A similar kind of work is also done by Xuan et al. (2020). They utilised AI algorithms to characterise URLs dependent on the features and conducts of URLs. The features are extricated from static and dynamic behaviours of URLs and are new to the literature. Those recently proposed features are the principle commitment of the research. AI algorithms are an aspect of the entire malicious URL discovery framework. Two administered machine learning algorithms are utilised, Support vector machine (SVM) and Random Forest (RF).

There is interesting work done by Bala et al. (2010). This paper examines and executes a novel technique to channel questionable indecent pictures shown on websites. This issue remains a fascinating issue to be tended to regarding the present situation. The algorithm works with the human body area utilising shape, shading and picture focused pixel scanning analysis. The output of the pixel checking approach is broken down, and an enhanced integrated filter is intended to improve the exhibition. The idea of the pixel filtering approach is tested and shown in real-time, utilising a web-based interface.

Adult content on a website can be detected by a pixel-based approach presented by Garcia et al. (2018). For this, all the multimedia files are being processed, segmented and filtered to analyse skin-coloured pixels by processing in YCbCr space and then classifying it as skin or non-skin pixels. They developed an application grounded from a pixel-based approach and a skin tone detection filter to detect images and videos with a large skin colour count. It is considered pornographic, classifying images and video frames containing nudity. With pixel-wise kin detection with image processing for this pornography filtering, the precision of 90.33% and accuracy of 80.23% were obtained.

Santos et al. (2012) proposed a nudity detection algorithm that offers a system for nudity discovery in images, separated into two modules: (1) filter skin identification; and (2) image zoning. The target of the image zoning module is to partition pictures into separate parts dependent on the presumption that a nude image presents a higher measure of skin pixels in its focal locale. These features must be the most regular data utilised to manage nudity location in pictures. At last, nudity recognition is performed utilising SVM (Support Vector Machines).

Pan et al. (2019) has discussed web attacks detection. End to end system attack has been addressed in the research paper. The author has focused on stacked diagnostic and autoencoders in attack detection. Upon reading the paper, one will learn how the author has developed an improved approach surrounding the benchmark framework that focuses on machine learning approaches and relies on label training data sets. The research has discussed the particular approach's crucial aspects and the benchmark. The approach has no relation to feature extraction engineering (Pan et al., 2019). The technique is helpful for conventional detection methods. Design software artefacts have been utilised to carry on this research, and it has been observed that with the help of the software artefacts, the research was much clearer.

The anti-phishing approach was proposed by Jain and Gupta (2018), where machine learning has been used by extracting 19 features. The 19 features have been used on the client-side so that authors can distinguish between phishing websites and legitimate ones. To carry out the research, a total of 2141 phishing pages have been used from PhishTank and open fish, and the number of legitimate pages counts to 1918 from Alexa popular website. There were also some online payment gateways and top banking websites in their approach. According to statistics, there was a 99.39% actual positive rate with the help of machine learning.

The research contribution by Xiang et al. (2018) has been divided into two phases. The first phase saw two kinds of attributes to carry out phishing. The features are genuine and interaction attributes that the authors used in the first phase. A feature-rich machine learning framework followed this, and the research achieved a 92% true positive rate and only a .4% false-positive rate with the attributes.

The most common practice of phishing detection is through URL; however, some researchers focus on detecting phishing emails with the help of the data stored in the email packets. The neural network approach was adopted by Smadi et al. (2018) to detect phishing attacks. There are 50 features in the proposed system, classified into four categories: mail headers, content URL, HTML content and the main text. The system has primarily focused on the emails, but the URL extracted features have made the system similar to the model.

Apart from machine learning approaches, Rao and Pais (2018) has also adopted an imaging approach in some research known to be a hybrid method. However, the image checking approach has a difficulty that is it requires an initial image database and proper knowledge about the History of the webpage. However, the approach that was proposed does not have any such dependencies. Hyperlink based features, third-party-based features and URL obfuscation features are the three categories of these features.

Challenges to detect cyber Attacks

It is challenging for researchers to develop proper techniques and detect phishing that is made on systems. One of the most popular techniques to detect phishing is the machine learning technique. Phishing attacks have increased over the recent past. Kumar and Faizan (2018) has come up with a proposal to evaluate a phishing detection approach in the paper. The URL has linguistic features that the researchers use to figure out patterns for spam and phishing detection. To avoid this situation, it is essential to retain the classified information. Gradual, abrupt, incremental and reappearance are the four types of features where the situation frequently occurs (Qin and Wen, 2018). Moreover, these features throw new challenges to machine learning techniques, and it is difficult to monitor the concept of drift and update or classifier using new data.

The research paper has reported some of the significant challenges in detecting phishing. According to Sountharrajan et al. (2020), the computer has a lot of difficulty in identifying the common dialects. Moreover, the authenticity and security instrument depend on the client-side's inclination that comes from the program and user’s side. However, the practice of phishing attacks is widespread nowadays, and attackers choose different ways to attack a system.

The lack of training data for prediction techniques is one of the most challenging tasks in the machine learning model in detection. Phishing detection classifiers are provided with data from the different repositories (Alloghani et al., 2020). UCI machine learning repository is one of the repositories. For the primary predictors, various features have been utilised. Moreover, the email technique variants also pose challenges to the researchers. There can be a study about the email technique variance to judge the causes of ML variation and detect the website that has been attacked due to phasing.

According to Basit et al. (2020), it is challenging to detect fishing with conventional email techniques; conventional email techniques have a low percentage of accurate detection. The traditional methods are appropriate for small data sets. The study is focused on heuristic techniques to explore high false-positive rates. To detect malicious emails and block them without users' intervention, the research paper has proposed a new solution. The authors implemented the approach of presenting a framework or a plug so that the website seeks login details. If the website asks for login details, this will create an alarm for the website owner, and the owner would get to know that the website is experiencing malicious activity.

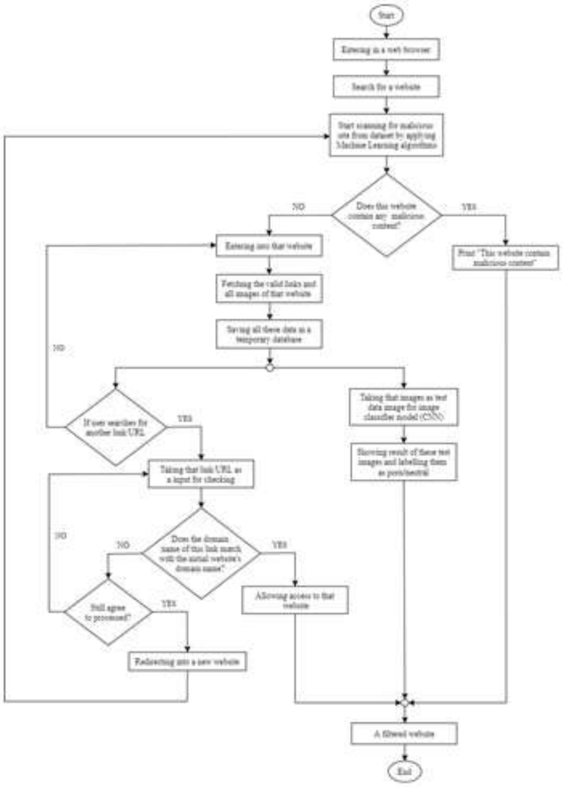
Certain machine learning-based approaches have encountered phishing detection challenges in the real environment. The machine learning-based techniques are prone to adverbial attacks. According to Alotaibi and Alotaibi (2020), the email techniques are not enough to defend the website from attack. Another challenge with email-based approaches is the vast data from the URLs. Thirdly there can be some reasons to halt the benign URL. The benign URL can be expired or not required and can come under phishing attacks. Thus, it will become difficult for the users to figure out the attack with a benign URL when an attacker uses it.

3. Proposed Approach

To build up our system model, we first had to set our work plan properly. Without proper planning, our system would not work the way we expect it to do. To ensure a better web surfing experience, we had to note the several parameters that needed to be solved to provide a good user experience. We have noted down three significant problems from all of those and decided to act according to it. Those are malicious site detection, link redirection, and pornographic image detection.

Our system ensures the overall security of websites with the help of machine learning and image processing algorithms. Usually, whenever we request a website, we often see lots of inappropriate and indecent content on the web pages. There is hardly any integrated system that simultaneously deals with inappropriate link redirection, malicious site detection, indecent image detection. Our approach deals with these problems in real-time and generates user-friendly and filtered web content. We will implement image processing algorithms to detect unwanted pictures. We also keep track of the URL links, which will stop issues like an invalid, undesired link redirection into trap sites or fishing sites. Our main motto here is to find out a minimal solution by which we can provide the users with a safe, smooth, and good browsing experience. Figure 3.1 shows the workflow diagram of the complete system. The system will follow the above-mentioned steps to fulfill our requirement.

According to the above figure, when a user search for a specific link/URL of a website, our system will first check whether that website is a malicious site or not. Our system will scan this using machine learning algorithm like Logistic algorithm, K-Nearest Neighbors (K-NN), Random forest, XG-Boost and Extra Tree Classifier. If our system finds that link contains any malicious content, our system will show a warning and will not allow the browser to access that malicious website. But if that site is safe from malicious content then our system will allow the user to enter into that website and our system will fetch all the valid links and images relate to that website in background. These data will be stored in a temporary database. In this stage, our system will perform two tasks simultaneously. First of all, our image classifier module will take the images that are fetched from that website using web scrapping. These fetched images will be use as a test image data for our CNN image classifier and provide a probable result of those images being porn and neutral. On the other hand, if the user asking to browse for another link then our link redirection model will check whether the desired input link and the destination link matches or not. This checking will occur based on the valid links that are fetched previously using web scrapping in a temporary database. The checker will work based on the comparison between the domain address of the parent website and the domain ad- dress of the link that the browser is redirecting for. Here, if this checker matches then the user can access to that link but if it doesn't then the user will get a warning and also get an option where he either can access into that or not. If the user decides to access into that unmatched link directory then for that new website our system will check from the beginning like starting with scanning for malicious website to nudity detection and ended up providing the user a user-friendly and smooth web surfing experience all together.



Start
Entering in a web browser
Search for a website
Start scanning for malicious site from dataset by applying Machine Learning algorithms
Does this website contain any malicious content?
NO
YES
Entering into that website
Print "This website contain malicious content"
Fetching the valid links and all images of that website
Saving all these data in a temporary database
NO
If user searches for another link URL
YES
Taking that images as test data image for image classifier model (CNN)
Showing result of these test images and labelling them as porn/neutral
Taking that link URL as a input for checking
NO
Does the domain name of this link match with the actual website's domain name?
NO
YES
Still agree to processed?
YES
Allowing access to that website
Redirecting into a new website
A filtered website
End

Figure 3.1: Work plan of the research

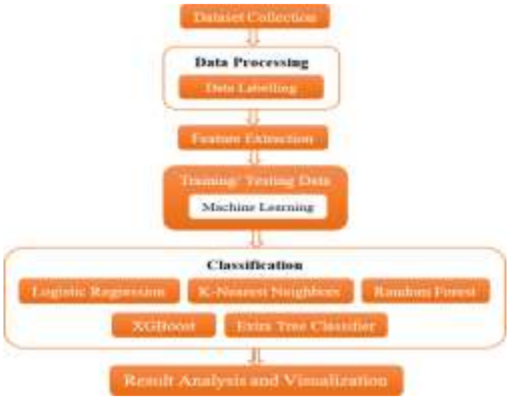


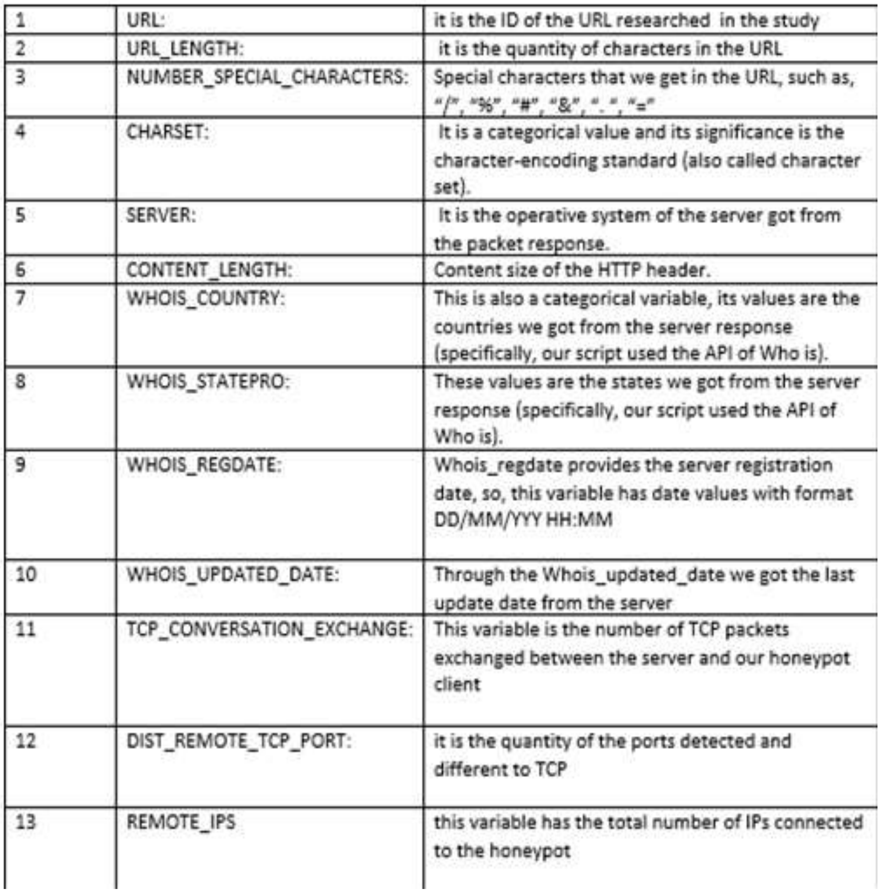
Figure 3.2: workflow of malicious site detection

Here we discuss about the data processing method and feature extraction for our research. The work process of our methodology is offered above to recognize dark malicious content by machine learning.

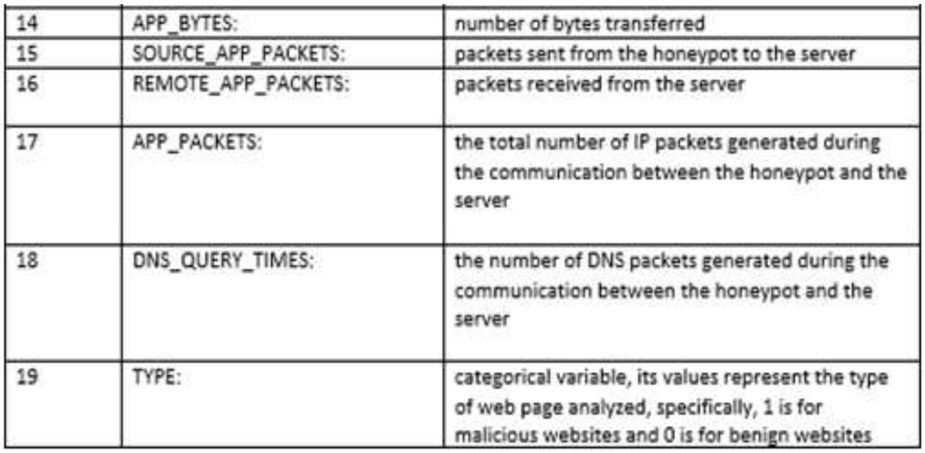
4. Data Description and Processing

In order to detect malware, our proposed model is discussed in this section We collected our dataset from kaggle competition. It is enormous dataset with an immense number of highlights and properties which are import. Besides, there are around 37000 examples in the dataset.

The data we are using in the dataset are spoken to in columns where every column depict an alternate standard.



1	URL:	it is the ID of the URL researched in the study
2	URL_LENGTH:	it is the quantity of characters in the URL
3	NUMBER_SPECIAL_CHARACTERS:	Special characters that we get in the URL; such as, "/", "%", "#", "&", ". ", "="
4	CHARSET:	It is a categorical value and its significance is the character-encoding standard (also called character set).
5	SERVER:	It is the operative system of the server got from the packet response.
6	CONTENT_LENGTH:	Content size of the HTTP header.
7	WHOIS_COUNTRY:	This is also a categorical variable, its values are the countries we got from the server response (specifically, our script used the API of Who is).
8	WHOIS_STATEPRO:	These values are the states we got from the server response (specifically, our script used the API of Who is).
9	WHOIS_REGDATE:	Whois_regdate provides the server registration date, so, this variable has date values with format DD/MM/YYY HH:MM
10	WHOIS_UPDATED_DATE:	Through the Whois_updated_date we got the last update date from the server
11	TCP_CONVERSATION_EXCHANGE:	This variable is the number of TCP packets exchanged between the server and our honeypot client
12	DIST_REMOTE_TCP_PORT:	it is the quantity of the ports detected and different to TCP
13	REMOTE_IPS	this variable has the total number of IPs connected to the honeypot



14	APP_BYTES:	number of bytes transferred
15	SOURCE_APP_PACKETS:	packets sent from the honeypot to the server
16	REMOTE_APP_PACKETS:	packets received from the server
17	APP_PACKETS:	the total number of IP packets generated during the communication between the honeypot and the server
18	DNS_QUERY_TIMES:	the number of DNS packets generated during the communication between the honeypot and the server
19	TYPE:	categorical variable, its values represent the type of web page analyzed, specifically, 1 is for malicious websites and 0 is for benign websites

Figure 3.3: Dataset Description

In data processing, data is mined in such way that it changes raw data to comprehensible arrangement. Real world data that might have some incompleteness or error is proven to be solved by data processing. We tried to figure out the “nan” values and replace them with zeroes. We also took only the numbers as input for our algorithms.

There were principally 20 inputs however, while extraction we made sense of 2 of them were pointless and we worked with the rest 18. Featured extraction is a cycle that is basically done to get the most important sources of info and work with them without unnecessary problems These recoveries from chopping down the time span and furthermore the entanglements that might have emerged from the not all that significant data.

Training/Testing of Machine Learning Classifier

After completing feature selection method, the subsequent stage is testing and training of Machine Learning classifiers. We divided our dataset into training and testing sets. From that point forward, we test and train the dataset in some classification algorithm Preparing information is utilized to fit and tune our model. Here, test information is spoken to as inconspicuous information to assess the models (Southarrajan et al., 2020). For our approach, we have used five classification algorithms.

Here are the five classifiers:
ˆ Logistic Regression
ˆ K-Nearest Neighbor (KNN)
ˆ Random Forest
ˆ XGBoost
ˆ Extra Tree Classifier

We have used these five classification algorithms to detect malicious sites.

After getting the outcome, they were analyzed and visualized. After the classified result, we got from the algorithms, the outputs were analyzed and after that visualized using graphical representations. For the analyzing, we used some metrics like Accuracy, TPR, and FPR. TPR and FPR were calculated from the confusion metrics. We used “Heat Map”, “Pie Plot”, and “ROC” curves for our final visual representation.

The scanning link directory is another wing of our system. Here, we have used the method of web scraping using python. Web scraping is the process to collect and parsing data from any website. But there are some websites that do not give permission or access to parse their data with the web scraping tools.
Various programming languages can be used for web scraping. Here, we have used python language because we can use python for various kinds of web scraping tools. The web scraping tools are called web scrapers. There are many types of web scrapers. For example, ParseHub, BeautifulSoup, Scrapy, Octoperse etc.

ˆ ParseHub: PerseHub is a web scraping tool which does not need any coding. It is also called a data mining tool which is very simple to use. If a user provides any link in PerseHub, it will automatically extract all the data of that website and exports the data in JSON or EXEL format.

ˆ BeautifulSoup: BeautifulSoup is a Python library for getting data out of HTML, XML, and other markup languages. There are a few site pages that show information pertinent to exploration, for example, date or address data, yet that don't give any method of downloading the information straightforwardly. Beautiful Soup encourages to pull specific substance from a page, eliminate the HTML markup, and spare the data. It is an instrument for web scraping that causes to tidy up and parse the reports that have pulled down from the web.

Scrapy: Scrapy is a web scraping library for Python developers hoping to construct adaptable web crawlers.

Octoparse: Octoparse is a phenomenal device for individuals who need to separate information from websites without coding, while as yet having com- mand over the full cycle with their simple to utilize UI.

In our system, we are used BeautifulSoup because It is very simple and easy to learn and import. Another reason for using BeautifulSoup is, it has great extensive documentation which encourages us to get familiar with the things rapidly and it has great network backing to make sense of the issues that emerge while we are working with this library.

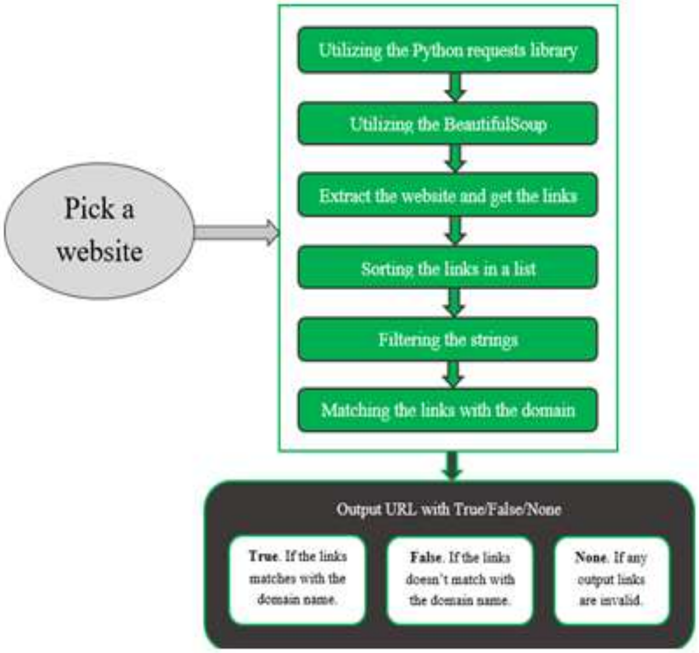

Pick a website
Utilizing the Python requests library
Utilizing the BeautifulSoup
Extract the website and get the links
Sorting the links in a list
Filtering the strings
Matching the links with the domain
Output URL with True/False/None
True. If the links matches with the domain name.
False. If the links doesn't match with the domain name.
None. If any output links are invalid.

Figure 3.4: Workflow of Link Redirection Methodology

Working Methodology of Scanning Link Directory

Here we discuss how this part of our model works. When someone ssearchesfor a website, from then on our system will start working. When our system starts scraping the web, it sends a request to the server that’s hosting the page we specified using the Python request library. Basically, our code downloads that page’s source code, just as a browser would. The BeautifulSoup library extracts the website and get all the links that are included in that particular website. Then all the links will be sorted in a list. After checking the domain names of these links, our system will show an output, indicating whether it has entered any link other than the one user wanted to enter.

1) Steps to Scan Invalid Link Directory

For our system we used Python code to scan invalid link directories. We used a few Python libraries to make our system effective and efficient. Steps we have followed to scan invalid link directory for any websites is given below: The request library: The primary thing we’ll have to do to scrap a web page is to download the page. We can download pages utilizing the Python requests library. The request library will make a GET request to a web server, which will download the HTML substance of a given page for us.

```
import requests
page = requests.get(inputt)
```

Import Beautiful Soup: We can utilize the Beautiful Soup library to parse this document, and extract the content from the p tag. We initially need to import the library and make an occurrence of the Beautiful Soup class to parse our document.

```
from bs4 import BeautifulSoup
bSoup = BeautifulSoup(page.content, ’html.parser’)
```

Now the BeautifulSoup will extract the website and get all the links that are inincluded in that particular website. For example, we are taking www.netflix.com as the input.

```
inputt = ’https://www.netflix.com’
```

After parsing we get all the links that are linked to netflix.com. After that this all links will be stored in a list.

```
links list = bSoup.find all(’a’)
```

Now it will filter the string. String means the link that is being used as input.

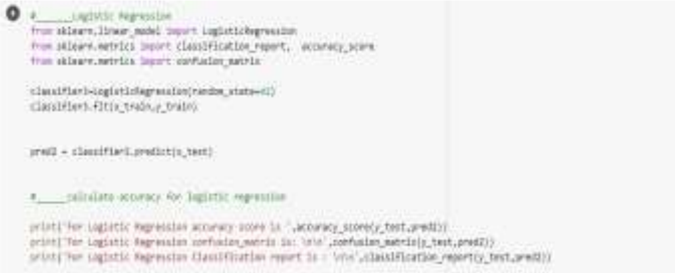
```
domain link = research(’.*?(\.com/\.net/\.gov/\.org/)’, domain link).group(0)domain link
= resub(’ˆhttps?://(m\.)?’,”, domain link).strip()
domain link = resub(’ˆwww\.(m\.)?’,”, domain link).strip()
domain link = resub (’\.com/?$\.net/?$\.gov/?$\.org/?$’, ”, domain link).strip()
```

After filtering, it will match the links with the domain name. If it matches, it will return True. If it does not match it will return False and give a warning that it is entering another website and for the invalid links, it will return None.

Here, True means If the user wants to enter that link, there will be no problem to enter. That means the user can enter where he/she wanted to enter as a destination link. But False shows when URL domains are redirected to a different domain and None used for invalid links. In our system, false or invalid links will give a warning to the user about unexpected link redirection.

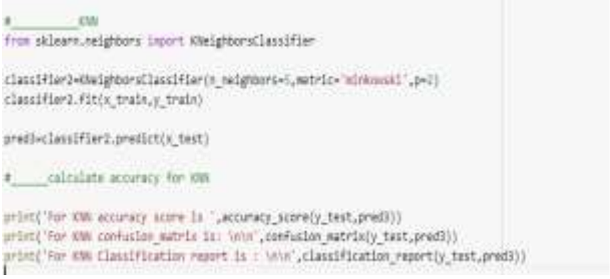
5. RESEARCH METHODOLOGY

Here the algorithm uses an S shape curve to detect true or false from the value. from sklearn.linear model and metrics. We imported logistic regression, classification report, confusion matrix. For the data set, it produces an S shape curve then it tells us the website is malicious or not.



```
#_____Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,  accuracy_score
from sklearn.metrics import confusion_matrix

classifier1=LogisticRegression(random_state=0)
classifier1.fit(x_train,y_train)


pred2 = classifier1.predict(x_test)


#_____calculate accuracy for logistic regression

print('For Logistic Regression accuracy score is ',accuracy_score(y_test,pred2))
print('For Logistic Regression confusion_matrix is: \n\n',confusion_matrix(y_test,pred2))
print('For Logistic Regression Classification report is : \n\n',classification_report(y_test,pred2))
```

Figure 4.1: Logistic Regression



```
#_________KNN
from sklearn.neighbors import KNeighborsClassifier

classifier2=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
classifier2.fit(x_train,y_train)

pred3=classifier2.predict(x_test)

#_____calculate accuracy for KNN

print('For KNN accuracy score is ',accuracy_score(y_test,pred3))
print('For KNN confusion_matrix is: \n\n',confusion_matrix(y_test,pred3))
print('For KNN Classification report is : \n\n',classification_report(y_test,pred3))
```

Figure 4.2: K-Nearest Neighbor

Malicious Site Detection Methodology

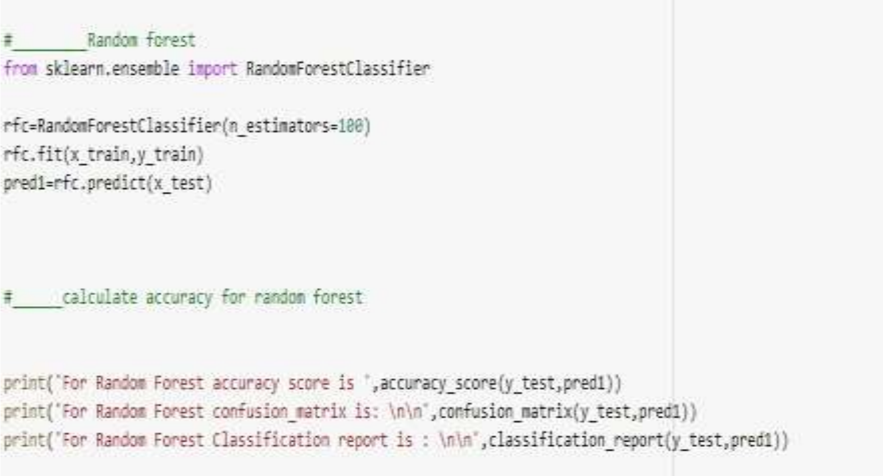
K-Nearest Neighbor

neighbours

In this section, we have used KNN where classifiers determine the region and the neighneighboursting with a data set with known categories. Then clustering that data and a new data with unknown category comes we do not know this category then we put the data here and it finds the close neigfindshbors.

Random Forest

We know random forest uses trees. Here Random Forest makes trees from our data and for the most votes we get our required data and the level of accuracy is good though not good new data. We grow multiple trees as opposed to a single tree in the court model to classify a new object based on attributes each tree gives a classification. Tree votes for that class the forests choose the classification having the most votes over all the other trees in the forests and in the case of regression take the average of the outputs by different trees.



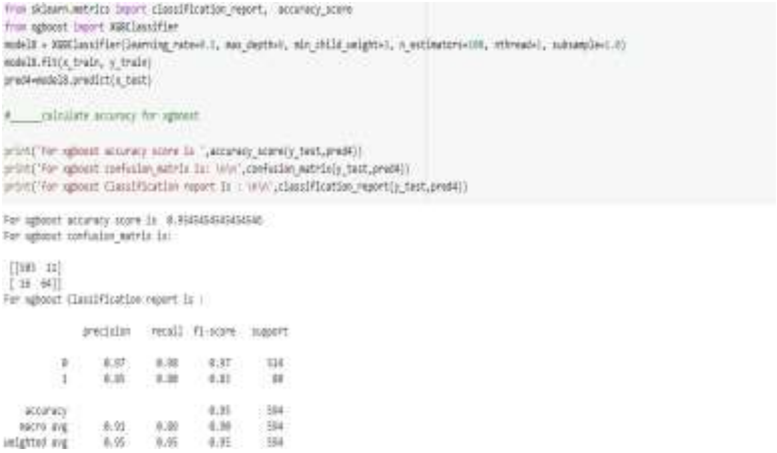
```
#_______Random forest
from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier(n_estimators=100)
rfc.fit(x_train,y_train)
pred1=rfc.predict(x_test)



#____calculate accuracy for random forest


print('For Random Forest accuracy score is ',accuracy_score(y_test,pred1))
print('For Random Forest confusion_matrix is: \n\n',confusion_matrix(y_test,pred1))
print('For Random Forest Classification report is : \n\n',classification_report(y_test,pred1))
```

Figure 4.3: Random Forest

So the same random forest algorithm or random forest classifier can be used for both classification and regression tasks random forest classifier will handle the missing values and maintain accuracy.

XGBoost

Here we are using XGBoost for data can also be optimized. It will ensure that overfit is not there. It also helps not to grow the tree at a certain level.



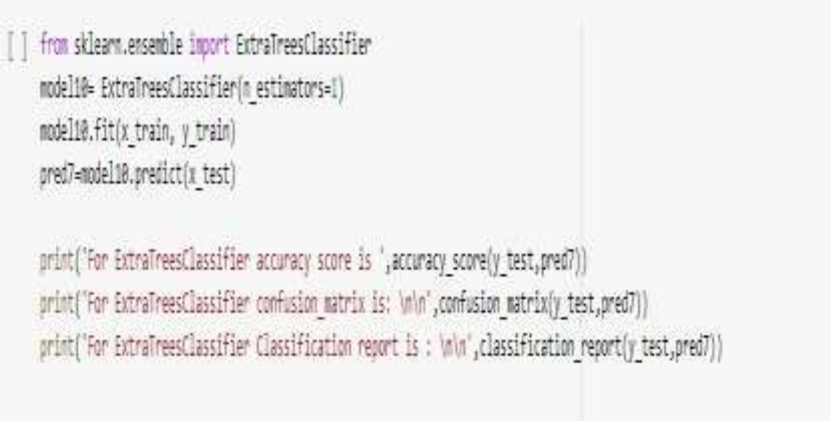
```
from sklearn.metrics import classification_report,  accuracy_score
from xgboost import XGBClassifier
model18 = XGBClassifier(learning_rate=0.1, max_depth=4, min_child_weight=1, n_estimators=100, nthread=1, subsample=1.0)
model18.fit(x_train, y_train)
pred4=model18.predict(x_test)

#_____calculate accuracy for xgboost

print('For xgboost accuracy score is ',accuracy_score(y_test,pred4))
print('For xgboost confusion_matrix is: \n\n',confusion_matrix(y_test,pred4))
print('For xgboost Classification report is : \n\n',classification_report(y_test,pred4))
```

```
For xgboost accuracy score is  0.9545454545454546
For xgboost confusion_matrix is:

 [[503  11]
 [ 16  64]]
For xgboost Classification report is :

              precision    recall  f1-score   support

           0       0.97      0.98      0.97       514
           1       0.85      0.80      0.83        80

    accuracy                           0.95       594
   macro avg       0.91      0.89      0.90       594
weighted avg       0.95      0.95      0.95       594
```

Figure 4.4: XGBoost

Extra Tree Classifier

Then we used an extra tree classifier. The main difference here is the splitter is randomly selected here. Random forest is used locally to optimize splitter but Extra Tree Classifier uses random splitters leading to more diversified trees.

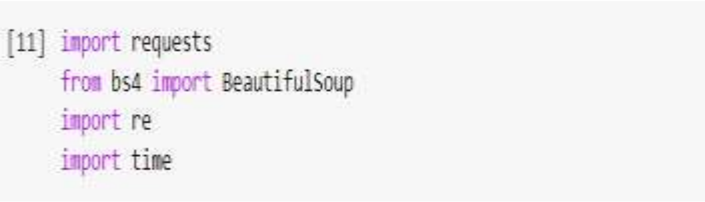


```
[ ] from sklearn.ensemble import ExtraTreesClassifier
    model10= ExtraTreesClassifier(n_estimators=1)
    model10.fit(x_train, y_train)
    pred7=model10.predict(x_test)

    print('For ExtraTreesClassifier accuracy score is ',accuracy_score(y_test,pred7))
    print('For ExtraTreesClassifier confusion_matrix is: \n\n',confusion_matrix(y_test,pred7))
    print('For ExtraTreesClassifier Classification report is : \n\n',classification_report(y_test,pred7))
```

Figure 4.5: Extra Tree Classifier

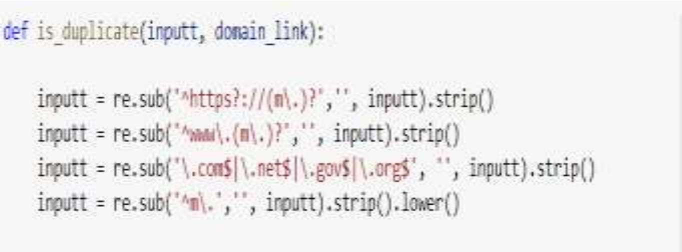
Link Redirection Methodology

Here “import requests” is basically making a request to a web page. We are im- porting Beautiful-Soup library by adding this line “from bs4 import BeautifulSoup”. We’re also importing regular expressions “import re” and the time function returns the number of seconds passed since the epoch.



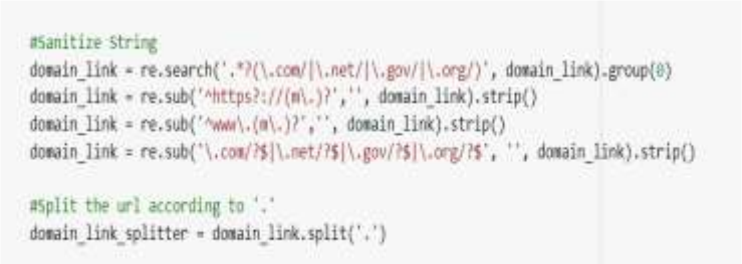
```
[11] import requests
     from bs4 import BeautifulSoup
     import re
     import time
```

Figure 4.6: Import Beautiful Soup



```
def is_duplicate(inputt, domain_link):

    inputt = re.sub('^https?://(m\.)?','', inputt).strip()
    inputt = re.sub('^www\.(m\.)?','', inputt).strip()
    inputt = re.sub('\.com$|\.net$|\.gov$|\.org$', '', inputt).strip()
    inputt = re.sub('^m\.','', inputt).strip().lower()
```

Figure 4.7: Checking Input with domain links

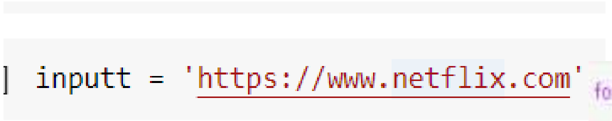


```
#Sanitize String
domain_link = re.search('.*?(\.com/|\.net/|\.gov/|\.org/)', domain_link).group(0)
domain_link = re.sub('^https?://(m\.)?','', domain_link).strip()
domain_link = re.sub('^www\.(m\.)?','', domain_link).strip()
domain_link = re.sub('\.com/?$|\.net/?$|\.gov/?$|\.org/?$', '', domain_link).strip()

#Split the url according to '.'
domain_link_splitter = domain_link.split('.')
```

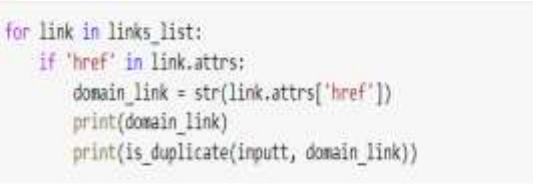
Figure 4.8: Filtering the String

By this, we are checking input with domain links that we got from the website. (figure 4.7)
We are filtering the string to get the domain name and later we are using it to match with the output links.
In this l,ine we are giving input to get links from that particular website.

Figure 4.9: Taking Input from User



```
] inputt = 'https://www.netflix.com'
```



```
for link in links_list:
    if 'href' in link.attrs:
        domain_link = str(link.attrs['href'])
        print(domain_link)
        print(is_duplicate(inputt, domain_link))
```

Figure 4.10: Hypertext Reference

“HREF” is a Hypertext reference. This HTML code is utilized to make a connection to another page. The HREF is an attribute of the anchor tag, which is additionally used to recognize segments inside a report. The HREF includes two components: one is URL, which is the real link, and the clickable content that shows up on the page, called the “anchor text”. In this segment, we also utilized a connected list and printed links, including whether it matches with the space or not with ‘true’ and ‘false’. So, we can say that this “HREF” attribute indicates the URL of the page the link goes to.

6. Results and Discussion

Results

We set up a server and a client with the help of the Python Flask. On the client-side, we implemented our machine learning classifier. The data got by the client is passed to the trained classifier, the trained classifier then predicts the label of the data. In our case, it took around 8 seconds for the classifier to classify the data in real-time. The following figure shows the output of the classifier on the client-side.

So as to work with any dataset, it is important to visualize it, python has a lot of libraries with the assistance of which we can undoubtedly imagine the information. We have used python libraries like seaborn, mathplotlib, scikitplot, etc to plot heatmap, pie chart etc.

To assess our model, Sensitivity, Specificity, accuracy score metrics are utilized. We likewise assess the ROC curve, AUC, confusion matrix, accuracy score, to choose the best algorithm for malicious web page identification.

We utilized five classifiers to detect malicious sites, these five algorithms are K- NN, Logistic Regression, Random Forest, XGBoost and Extra Tree Classifier. We likewise assessed the presence of the classifiers with the assistance of various metrics which is given below.

Sensitivity: Sensitivity shows the capacity of the classifiers to discover a malicious webpage that is really of malicious category.

Sensitivity = True Positive/ (True Positive + False Negative) (5.1)

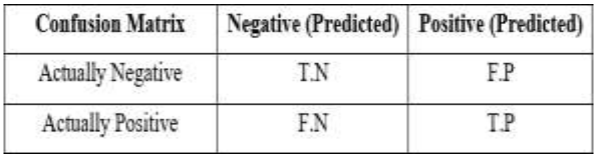
Specificity: Specificity shows the capacity to accurately recognize benign web- sites that are without the state of malicious websites.

Specificity = True Negative/ (True Negative + False Positive) (5.2)

6.1 Confusion Matrix

One of the most broadly utilized strategies for assessing the machine learning algorithm is the confusion matrix.

Table 5.1: Table of Confusion Matrix



Confusion Matrix	Negative (Predicted)	Positive (Predicted)
Actually Negative	T.N	F.P
Actually Positive	F.N	T.P

Here, T.N= True Negative where the algorithm predicted negative and the output is also negative, F.P= False Positive where the algorithm predicted positive and the output is negative, F.N= False Negative where the algorithm predicted negative and the output is also positive, T.P= True Positive where the algorithm predicted positive and the output is also positive.

True Positive Rate (T.P.R): It describes how great the model is at predicting the positive class when the real result is positive.

T : P : R = TruePositive = (TruePositive + FalseNegative) (5.3)

False Positive Rate (F.P.R): It is called the false alarm rate as it sums up how regularly a positive class is predicted when the real outcome is negative.

F : P : R = FalsePositive = (FalsePositive + FalseNegative) (5.4)

ROC Curve

ROC curve is a graphical plot that represents the diagnostic capacity of a binary classifier framework as its discrimination threshold is changed. This curve shows the connection between clinical sensitivity and specificity for each necessary cut-off.

ROC curve plots TPR versus FPR at various classification thresholds. It is a plot of the false positive rate versus the true positive rate for a number of different candidate threshold values somewhere in the range of 0 and 1. Bringing down the classification threshold classifies more things as positive, hence increasing both False Positives and True Positives.

Accuracy Score

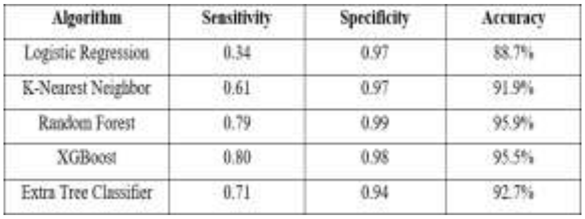
Accuracy score is a metric that is the most recognized classification model. For binary classification, it can be calculated in terms of positive and negative. The formula for finding the accuracy score is as follow:

Accuracy Score = T.P + T.N
T.P + T.N + F.P + F.N   (5.5)

Here, T.P= True Positive, T.N = True Negative= False positive, F.N= False Negative

The results obtained from the different algorithm is shown in table 5.2.

Table 5.2: Comparison of values



Algorithm	Sensitivity	Specificity	Accuracy
Logistic Regression	0.34	0.97	88.7%
K-Nearest Neighbor	0.61	0.97	91.9%
Random Forest	0.79	0.99	95.9%
XGBoost	0.80	0.98	95.5%
Extra Tree Classifier	0.71	0.94	92.7%

In comparison, we can see the accuracy of Logistic Regression is 88.7%, the accuracy of K-NN algorithm is 91.9%, accuracy of Random Forest algorithm is 95.9%, the accuracy of XGBoost is 95.5%, the accuracy of Extra Tree Classifier is 92.7%. From this, the result portrays that among the five algorithms the Random forest is performing the best.

Comparison with Other Papers

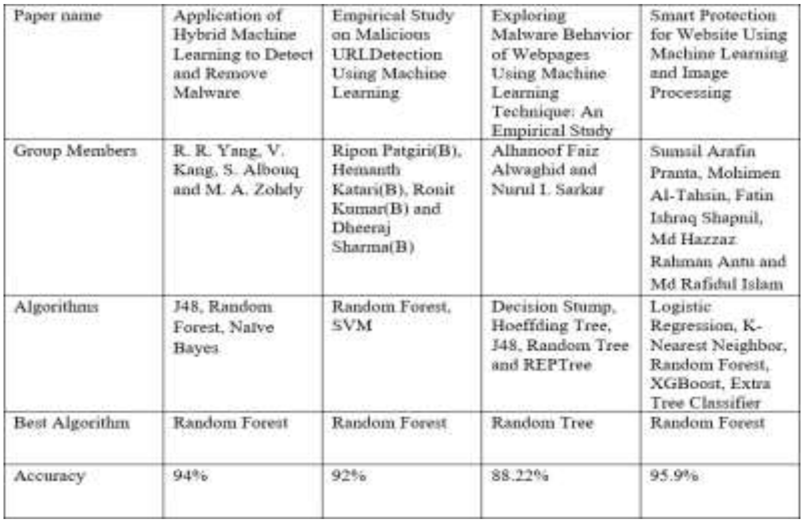
Firstly, in paper one project named "Application of Hybrid Machine Learning to Detect and Remove Malware", by R. R. Yang, V. Kang, S. Albouq, and M. A. Zohdy, they have used algorithms like Random forest, Na¨ıve Bayes, J48. With all the classifiers, they could create a hybrid Machine learning algorithm. Among them, Random Forest gave them the best accuracy, and the accuracy was 94%, but we were able to achieve 96%, and also our best classifier was Random forest (Alloghani et al., 2020)

In the second paper similar work for the malicious paper name "Empirical Study on Malicious URLDetection Using Machine Learning", Ripon Patgiri(B), Hemanth Katari(B), Ronit Kumar(B), and Dheeraj Sharma(B) from the National Institute of Technology Silchar, Silchar 788010, Assam, India they use classifiers like Random Forest, SVM. They got the best score from the random forest and achieved 92% Accuracy, which is 4% less than we have (Basit et al., 2020)

Lastly, we have compared our project with “Exploring Malware Behavior of Web- pages Using Machine Learning Technique: An Empirical Study Alhanoof Faiz Al- Waghid and Nurul I. Sarkar”. They used Decision Stump, Hoeffding Tree, J48, RF, Random Tree and REPTree. They got the best out of the random tree, which is 88% less than us and shows that our work is comparatively good (Alotaibi and Alotaibi, 2020).

The results obtained from the different algorithms are described as follows:

Table 5.3: Comparison with other Papers



Paper name	Application of Hybrid Machine Learning to Detect and Remove Malware	Empirical Study on Malicious URLDetection Using Machine Learning	Exploring Malware Behavior of Webpages Using Machine Learning Technique: An Empirical Study	Smart Protection for Website Using Machine Learning and Image Processing
Group Members	R. R. Yang, V. Kang, S. Albouq and M. A. Zohdy	Ripon Patgiri(B), Hemanth Katari(B), Ronit Kumar(B) and Dheeraj Sharma(B)	Alhanoof Faiz Alwaghid and Nurul I. Sarkar	Sumsil Arafin Pranta, Mohimen Al-Tahsin, Fatin Ishraq Shapnil, Md Hazzaz Rahman Antu and Md Rafidul Islam
Algorithms	J48, Random Forest, Naïve Bayes	Random Forest, SVM	Decision Stump, Hoeffding Tree, J48, Random Tree and REPTree	Logistic Regression, K-Nearest Neighbor, Random Forest, XGBoost, Extra Tree Classifier
Best Algorithm	Random Forest	Random Forest	Random Tree	Random Forest
Accuracy	94%	92%	88.22%	95.9%

Logistic Regression

For Logistic Regression, we get an accuracy score of 0.887. The confusion matrix for Logistic Regression is as follows:

Here in the confusion matrix, the true negative value is 0.97, the false positive value is 0.03, the false-negative value is 0.66, and the true positive value is 0.34.

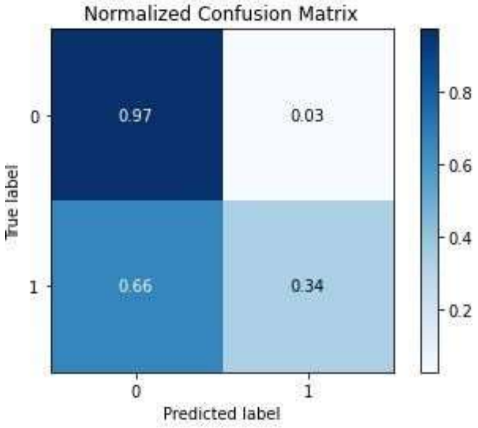


Figure 5.3: Confusion Matrix of Logistic Regression

The ROC curve for Logistic Regression is as follows. For Random Forest we get the AUC (Area Under Curve) score 0.809.

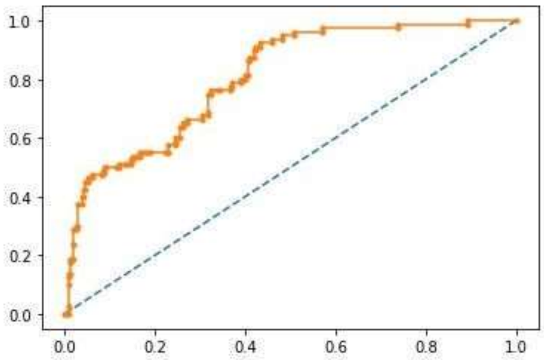


Figure 5.4: ROC Curve of Logistic Regression

K-Nearest Neighbor

For K-Nearest Neighbor, we get an accuracy score of 0.919. The confusion matrix for K-Nearest Neighbor is as follows: Here in the confusion. matrix, the true negative value is 0.97, the false positive value is 0.03, the false-negative value is 0.39, and the true positive value is 0.61. The ROC curve for K-Nearest Neighbor is as follows:
For K-Nearest Neighbor, we get the AUC (Area Under Curve) score of 0.900.

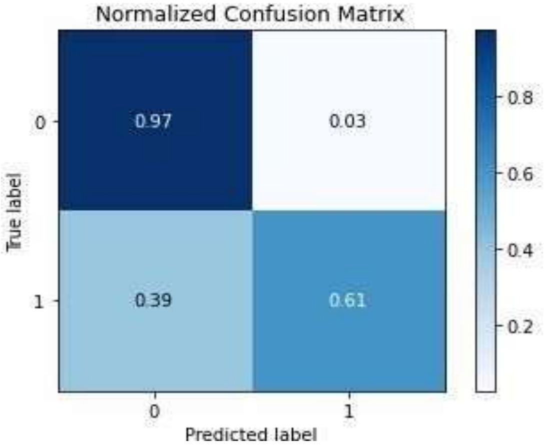


Figure 5.5: Confusion Matrix of K-Nearest Neighbor

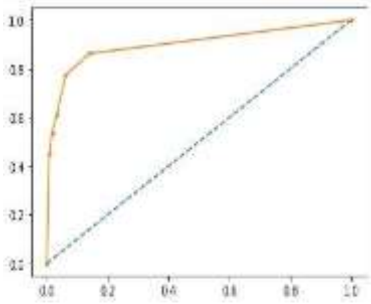


Figure 5.6: ROC curve of K-Nearest Neighbor

Random Forest

For Random Forest we get an accuracy score of 0.9596. The confusion matrix for Random Forest is as follows: Here in the confusion matrix, the true negative value is 0.99, the false positive value is 0.01, the false-negative value is 0.21 and the true positive value is 0.79. The ROC curve for Random Forest is as follows:

For Random Forest we get the AUC (Area Under Curve) score 0.977.

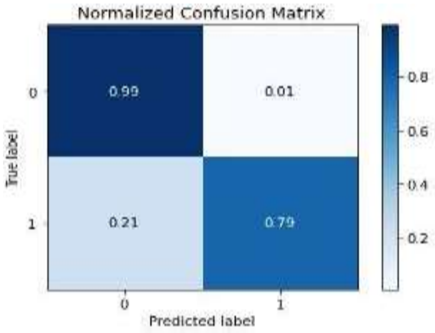


Figure 5.7: Confusion Matrix of Random Forest

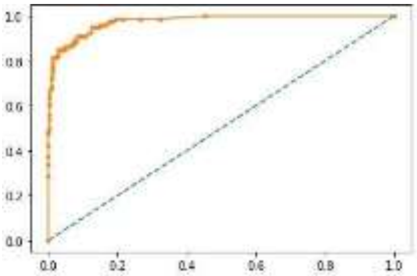


Figure 5.8: ROC curve of Random Forest

XGBoost

For XGBoost we get an accuracy score of 0.955. The confusion matrix for XGBoost is as follows:

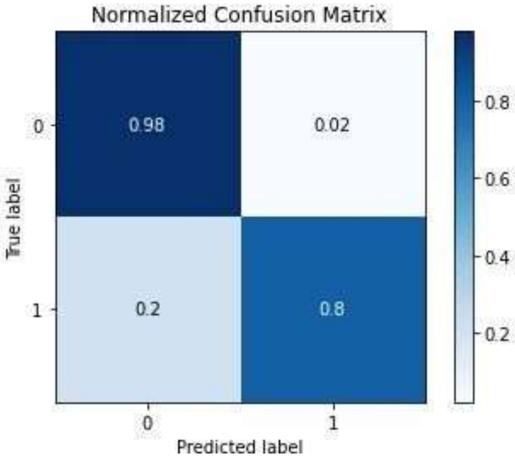


Figure 5.9: Confusion Matrix of XGBoost

Here in the confusion matrix, the true negative value is 0.98, the false positive value is 0.02, the false-negative value is 0.2, and the true positive value is 0.8. The ROC curve for XGBoost is as follows:
For XGBoost we get the AUC (Area Under Curve) score 0.975.

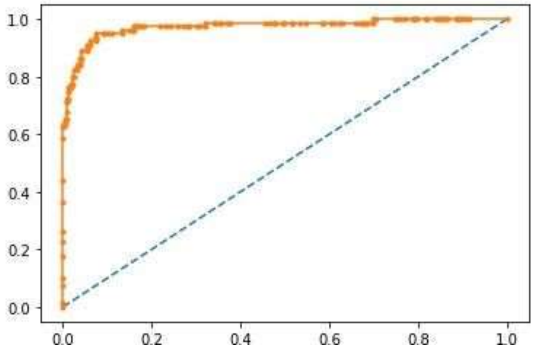


Figure 5.10: ROC curve of XGBoost

Extra Tree Classifier

For Extra Tree Classifier we get an accuracy score of 0.912. The confusion matrix for Extra Tree Classifier is as follows:

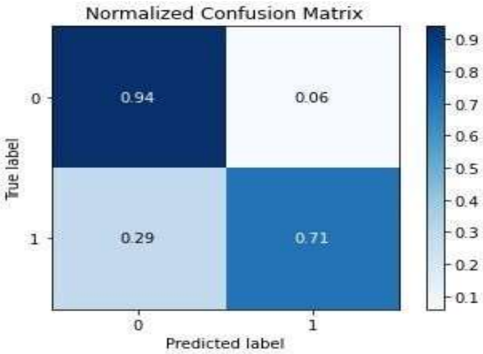


Figure 5.11: Confusion Matrix of Extra Tree Classifier

Here in the confusion matrix, the true negative value is 0.94, the false positive value is 0.06, the false-negative value is 0.29 and the true positive value is 0.71. The ROC curve for Extra Tree Classifier is as follows: For Extra Tree Classifier we get the AUC (Area Under Curve) score 0.828.

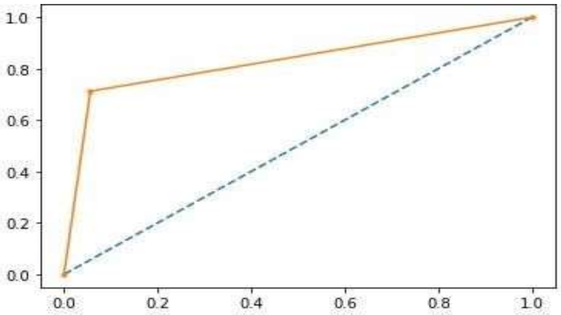


Figure 5.12: ROC curve of Extra Tree Classifier

Here, AUC for logistic regression is 0.809 and accuracy is 88.7%. AUC for K- NN 0.90 and accuracy is 95.96%. AUC for Random Forest is 0.977 and accuracy is 96.12%. AUC for XGBoost is 0.975 and accuracy is 95.5%. AUC for Extra Tree Classifier is 0.828 and accuracy is 92.7%. If the AUC value of a prediction is 100% wrong has an AUC value of 0.0 and if it is 100% right, then it is 1.0. The random forest has the highest AUC value which is 0.977.

6.2 Link Redirection Methodology

We are using Link Redirection Methodology to scan the invalid links with a different domain. Some websites don't provide permission for web scrapping. But many most websites allow web scrapping. Our system works on those websites which permit web scrapping.

For example, https://www.netflix.com. This website allows web scrapping. So, we will get an output from this input with a result of True/ False/ None. Here, the output is given below:

Table 5.4: Output of the given input

Table 5.4 is the output of the given input. Here, we can see three types of results (True, None, False). If the result is True, our system will let the user get into the link. If the result is None, it is an invalid link, and our system will automatically block that link. If the result is False, the connection is valid, but it has a different domain. Then our system will send a warning message to the user and ask if they want to continue entering a link with a different domain.

Our system works in the following domain types:

.com
.org
.gov
.net

In the rest of the domain types, it does not work. In future, we have a plan to extend this work field. We have tested our system on 30 websites, where there were websites of different domain types. We are fortunate that we got our expected result. Here, "YES" refers to websites that apply to our system and "NO" means that these websites do not apply to our system because of different domain types. The list of that 30 websites is given below: Table 5.5: Checked Website for Our System.

7. Conclusion and Future Work

The main motto of our project is to build a model that helps a user experience a filter together and safe web surfing. To achieve that goal, we made a workflow that describes how things are going to be executed. According to the model, the task is divided into three parts malicious site, unwanted link redirection and nudity detection. We have managed to implement the malicious site detection part with the algorithms like KNN, Random Forest, Linear Regression, XGBoost and Extra Tree Classifier algorithm for evaluating the accuracy.

Among these five algorithms, we got the higher accuracy of 95.96algorithm. In our second wing, we build a system that can help the user avoid unwanted link redirection. Fortunately, we applied our process to (".com", ".net", ".org", ".gov") these domain types. We used the web scraping technique to parse data from the website. Then we matched the input domain name with the output domain name. Here we got three types of results (True, None, False). If the result is True, our system will let the user get into the link. If the result is None, it is an invalid link, and our system will automatically block that link. If the result is False, the connection is valid, but it has a different domain. Then our system will send a warning message to the user and ask if they want to continue to enter a link with a different domain. Finally, we worked with the nudity detection part as we plan to filter the unusual and unwanted pictures which pop up now and then while web browsing. So, we decided to implement such a model, which is dynamic in terms of making a judgment whether that popped up image contains nudity or any unwanted content. As the task seemed very complex, we had to introduce a model that could take an image as input, then prepare it and classify it under certain categories. So, we decided to implement Convolutional Neural Network for image classification. Fortunately, we also get a good accuracy using CNN to detect indecent pictures/images. To include, we can proudly say that instead of having so many obstacles, we have tried our best to integrate three different wings altogether.

Future Work

Apart from all of the approaches that we have taken so far, there are still exist more room for improvement. First of all, we have seen our system has some limitations regarding link redirection detection. We have seen our model don’t give expected results for some URLs that can be fixable and very implementable. Then, for filtering the unwanted images, we’ve seen that web images are very complex in terms of background and many objects, making it harder for our present system to detect and categorize them. Some pictures are highly sexually provocative rather than just typical skin exposures porn content, which our system couldn’t detect. As a result, those sexually provocative pictures got unfiltered. So, in future, we will build a model that can deal with these shortcomings and detect the body gestures, intentions, postures. Based on that, it will provide a result. Furthermore, we plan to implement a hybrid algorithm that will combine more than one algorithm to perform more efficiently and perfectly in our malicious link detection part. Finally, as the problems nowadays get complicated and critical, the solutions also need to be updated to provide a better result.

8. References

[1] W. Yost and C. Jaiswal, "MalFire: Malware Firewall for Malicious Content Detection and Protection," in *2017 IEEE 8th Annual Ubiquitous Com- puting, Electronics and Mobile Communication Conference (UEMCON)*, 2017.

[2] Symantec.com, "2019 internet security threat report," 2019. [Online]. Available: https://www.symantec.com. [Accessed 8 December 2021].

[3] i. Gallup, "cybercrimes remain most worrisome to americans," 2018. [Online]. Available: www.gallup.com. [Accessed 8 December 2021].

[4] Accenture.com, "Cybertech europe 2017 i accenture," 2018. [Online]. Available: https://www.accenture.com. [Accessed 8 December 2021].

[5] Symantec, "Symantec Internet Security Threat Report –Trends for 2010," Dispo -n´ıvel em, 2011.

[6] Open Web Application Security Project (OWASP, "Unvalidated Redirects and Forwards Cheat Sheet," 2014.

[7] V. Krammer and F. Taylor, "An Effective Defense against Intrusive Web Advertising," in *Sixth Annual Conference on Privacy, Security and Trust*, 2008.

[8] E. L. Post and C. N. Sekharan, "Comparative Study and Evalua- tion of Online Ad-blockers," in *2015 2nd International Conference on Information Science and Security (ICISS)*, 2015.

[9] D. Gavrilut, M. Cimpoes, D. Anton and L. Ciortuz, "Malware Detection Using Machine Learning," in *International multiconference on Computer Science and Information Technology*, 2009.

[10] E. Rokkathapa and S. Kanrar, "A novel approach for predicting the malware attack," *International journal of computer application,* vol. 181, no. 45, 2019.

[11] W. Liu, P. Ren, K. Liu and H.-x. Duan, "Behavior-Based Malware Analysis and Detection," in *2011 First International Workshop on Complexity and Data Mining*, 2011.

[12] O. P. Samantray, S. N. Tripathy and S. K. Das, "A study to Understand Malware Behavior through Malware Analysis," in *IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019.

[13] M. Fleck, D. Forsyth and C. Bregler, "Finding naked people," *Computer VisionI-ECCV'96,* p. 593–602, 1996.

[14] D. Forsyth and M. Fleck, "Identifying nude pictures," *Applications of Computer Vision,* 1996.

[15] D. Forsyth and M. Fleck, "Automatic detection of human nudes," *International Journal of Computer Vision,* vol. 32, no. 1, p. 63–77, 1999.

[16] L. Duan, G. Cui, W. Gao and H. Zhang, "Adult Image Detection Method Base-on Skin Color Model and Support Vector Machine," *Comput. Vision,* pp. 1-4, 2002.

[17] C. Jeong, J. Kim and K. Hong, "Appearance-based nude image detection," *Proc. Int. Conf. Pattern Recognit,* vol. 4, pp. 467-470, 2004.

[18] J. Lee, Y. Kuo, P. Chung and E. Chen, "Naked image detection based on adaptive and extensible skin color model," *Pattern Recognit.,* vol. 40, no. 8, p. 2261–2270, 2007.

[19] H. A. Rowley, "Large Scale Image-Based Adult-Content Filtering," *Proc. First Int. Conf. Comput. Vis. Theory Appl,* pp. 290-296, 2006.

[20] R. Kumar, X. Zhang, H. A. Tariq and R. U. Khan, "Malicious URL detection using multi-layer filtering model," in *International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2017.

[21] A. Y. Daeef, R. B. Ahmad, Y. Yacob and N. Y. Phing, "Wide scope and fast websites phishing detection using URLs lexical features," in *2016 3rd International Conference on Electroni*, 2016.

[22] C. D. Xuan, H. D. Nguyen and T. V. Nikolaevich, "Malicious URL Detection based on Machine Learnin," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 11, no. 1, 2020.

[23] M. R. Bala, K. Aakash, S. Anand and S. A. Chandra, "Intelligent Approach to Block Objectionable Images in Websites," in *2010 Interna- tional Conference on Advances in Recent Technologies in Communication and Computing*, 2010.

[24] M. B. Garcia, T. F. Revano, B. G. M. Habal, J. O. Contreras and J. B. R. Enriquez, "A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm," in *IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Man- agement (HNICEM)*, 2018.

[25] C. Santos, E. M. dos Santos and E. Souto, "Nudity detection based on image zoning," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, 2012.

[26] Y. Pan, F. Sun, Z. Teng, J. White, D. C. Schmidt, J. Staples and L. Krause, "Detecting web attacks with end-to-end deep learning," *Journal of Internet Services and Applications,* vol. 10, no. 16, pp. 1-22, 2019.

[27] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client–side using machine learning based approach," *Telecommunication Systems,* vol. 68, no. 4, pp. 687-700, 2018.

[28] G. Xiang, H. J. C. P. Rose and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security,* vol. 14, no. 2, pp. 1-28, 2011.

[29] S. Smadi, N. Aslam and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learnin," *Systems,* vol. 107, pp. 88-102, 2018.

[30] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications,* vol. 107, pp. 88-102, 2018.

[31] S. Kumar, A. Faizan, A. Viinikainen and T. Hamalainen, "Mlspdmachine learning based spam and phishing detection," in *International Conference on Computational Social Network*, 2018.

[32] K. Qin and Y. Wen, "Semi-supervised Classification of Concept Drift Data Stream Based on Local Component Replacement," in *International CCF Conference on Artificial Intelligence*, 2018.

[33] S. Sountharrajan, M. Nivashini, S. Shandilya, E. Suganya, A. Banu and M. Karthiga, "Dynamic Recognition of Phishing URLs Using Deep Learning Techniques," in *Advances in Cyber Security Analytics and Decision Systems*, 2020.

[34] M. Alloghani, D. Al-Jumeily, A. Hussain, J. Mustafina, T. Baker and A. Aljaaf, ""Implementation of machine learning and data mining to improve cybersecurity and limit vulnerabilities to cyber attacks," in *Nature-Inspired Computation in Data Mining and Machine Learning*, 2020.

[35] A. Basit, M. Zafar, X. Liu, A. Javed, Z. Jalil and K. Kifaya, "A comprehensive survey of AI-enabled phishing attacks detection technqiues," *Telecommunication Systems,* pp. 1-16, 2020.

[36] B. Alotaibi and M. Alotaibi, "Consensus and Majority Vote Feature Selection Methods and a Detection Technique for Web Phishing," *J. Ambient Intell Human Comput.,* 2020.

INFO

***Corresponding Author:* Chetanpal Singh, Department of Information System Holmes Institute, Australia.**

***How to cite this article*: Chetanpal Singh, Ishtiaque-Al-Mahmood, Mohimem-Al-Tahsin, Novel Approach to Secure Websites with Machine Learning Classifiers, Asian. Jour. Social. Scie. Mgmt. Tech.2022; 4(2): 152-176.**