Smart Protection for Website Using Machine Learning and Image Processing

Sumsil Arafin Pranta, Mohimen-Al-Tahsin, Fatin Ishraq Shapnil,Md. Hazzaz Rahman Antu,Md Rafidul Islam, Dr. Muhammad Iqbal Hossain
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
Email:{sumsil.bracu,mohimen.bracu,fatin.bracu,hazzaz.bracu,rafidul.bracu}@gmail.com, iqbal.hossain@bracu.ac.bd

***Abstract*—Internet has become a basic and one of the most important tools for regular life. This ascent in the boundless utilization of innovation carried with it an ascent in cybercrime. There have been so many loops and data breaches in web content as a result it's become very easy for the criminals to break those security breaches of web contents and inject virus and other unwanted and harmful content. Almost every one of us face the problem of getting trapped in phishing sites or clicking on some popped up advertisements which ended up in to absolutely an indecent web pages that contain potential malicious data. As a result, most of our very private data is being compromised or got leaked, which create so many problems in both our corporate and private life. So, it's become top priority to the security analyst and consults to ensure the security as cyber security has become an integral part in every sphere. Keeping these facts in our mind, we have come up with an idea of a smart protection concept for web sites that we browse regularly in everyday life. Our aim is to build a model that will prevent us from various malicious threat, unwanted instead link redirection and unwanted and indecent contents that pop up quite regularly in the time of browsing or surfing in webpages. To make our software efficient we will integrate machine learning and image processing method to detect harmful content accurately**

***Index Terms*—Malicious Data, Machine learning, Link redirection, Image processing**

I. Introduction

The Morris Worm is broadly taken into consideration to be 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 try measuring the Internet. To try this, he created software that might implant itself in UNIX computer systems as it traveled round 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 bug39;s replication technique before sending it into the wild. But at present rise in the huge use of technology brought with it a huge rise in cybercrime. As the number of users increase so does the critical infrastructure of the cyber community and subsequently, the amount and variety of cyber-attacks increase as well [1]. This number of cyber-attacks is increasing daily with an exponential rate. According to the Symantec Internet Security Threat report 2019, the use of destructive malware by different groups increased by about 25% in 2018 [2]. Almost every one of us face the problem of getting trapped in phishing sites or clicking on some popped up advertisements which ended up in to absolutely an indecent web pages that we did not desire. As a result, most of our very private data is being compromised or got leaked, which create so many problems in both our corporate and private life. 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 [3]. These attacks are also causing tremindous economic loss. In Accenture report, they are saying that a company on average costs 2.4 million us dollar because of malware attack [4]. So, it's become top priority to the security analyst and consults to ensure the security as cyber security has become an integral part in every sphere.

There have been so many loops and data breaches in web content as a result it's become very easy for the criminals to break that security breach of web contents and inject virus and other unwanted and harmful content. Sometimes they use nude pictures and animated video to attract user. When these unwanted content is popping up to the web browser, then the user attracted by this pornographic content and press this from their attraction. But these pornographic content can also additionally result in pages with malicious threats. It is clearly provided in a recent report provided by Symantec [5] that there is a direct relationship between these pornography and dissemination of malware.

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 URL redirection is normally used as part of phishing attacks that confuse visitors about which web site they are visiting.[6] Because modern browsers always display the original URL in the address bar, the threat is lessened. However, redirects also can take a person to sites that will otherwise attempt to attack in different ways. For example, a redirect may take a person to a site that would attempt to trick them into downloading antivirus software and installing a Trojan of a few kind instead.

Moreover, the random popping up of websites in our browser may be an indication of many things on our computer. We also see many online popping up advertisements when we open our browser or when we are in any website. These online

ads come in a form of banners, text, pop- ups, images, or in transitional format. Advertisers began implementing greater intrusive techniques to attract user's attention [7]. They lively motion pictures or images, additionally 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 [8].
To reduce distraction, malware infections, unwanted link directory, smart protection system can be the great option. Though some of us use preinstalled pop-up blockers. Using smart protection system, the pages that users access are secure and free from internet threats, such as malware and phishing scams, also it is free from unwanted link redirection which can be designed to trick users into providing personal records.

II. Motivation

There have been so many loops and data breaches in web content. As a result it's become very easy for the criminals to break that security breach of web contents and inject virus and other unwanted and harmful content. Sometimes they use nude pictures and animated video to attract user. When these unwanted content is popping up to the web browser, then the user 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 dissemination of malware. The random popping up of websites in our browser may be an indication of many things on our computer. We also see many online popping up advertisements when we open our browser or when we are in any website. Not only that, we face many more problems while browsing any website. Sometimes it is seen that 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 the another page instead. It ensures visitors don't emerge as on a 404 page and instead find something relevant to what they were mainly looking for. Keeping these facts in our mind, we have come up with an idea of a smart protection concept for web sites that we browse regularly in everyday life. Using smart safety, we ensure that the pages that users access are secure and free from internet threats, such as malware, phishing scams, also it is free from unwanted link redirection which can be designed to trick users into providing personal records. So, our aim is to build a model that will prevent us from various malicious data and unwanted and indecent contents that pop up quite regularly in the time of browsing or surfing in webpages. Our supervisor helps us learn about the advancement of machine learning algorithms. He also suggests us to use some techniques which play an important rule to reduce link redirection of a website.

III. Problem Statement

We rely on the web surfing or browsing through all the day for any random daily basic needs. So, any kind of problem or uneasiness regrading web surfing can have a serious impact on productivity and peace of mind as they become intolerable. In fact, some major problems such as malicious data, malware hamper our privacy by compromising our private data without our will.
These malware is such a program planned exclusively to purpose problems and as days pass more malware is made however the old existing ones. Antivirus is regularly uninformed of any new infection or malware that is being spread through the web and when an answer comes, clients have just been influenced by the new infection. This pernicious malware is the cause of losing billions of dollars simply as information. These cybercriminals will take a normal 33 billion records by 2023 as shown by an ongoing record from Juniper Exploration [9].
Again, we see many inappropriate Ad and indecent pictures in between the web contents which is very uncomfortable. So, we think nudity detection is constantly being a significant issue of web indexes, person to person communication sites and other web channels. Yet, this issue is turning out to be more genuine these days since 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 [10].
Moreover, these inappropriate things cover the web contents as well. Then we also face link redirection to an undesired web destination almost every time we surf from one web address to another. These are the most common problems that everyone faces but hardly there 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.

IV. Objectives and Contributions

The problems we have discussed earlier, needed different algorithms from different section like we if we have to deal with unwanted image detection then we have to implement image processing method. Then for detecting malicious contents and ensuring web security we will use 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 our mind we came up with a possible solution of introducing a model contains 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 the possible algorithms to run our software. But we have made our goals very clear about what we are going resolve. First of all, we will secure our web browsing from malware and malicious date by implementing 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 a track of the URL links of a requesting web site. As all the information of a website kept in the HTML file of that web page. So we will trace the links and scan for all the unnecessary directory.
Furthermore, we will implement image processing algorithm to detect inappropriate and indecent pictures or Ad among all web contents. We will implement 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.
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.

V. Background

There are three wings in our thesis. The wings are malware detection and blocking, scanning invalid link directory and nudity detection. No works had been done before which has the combination of all threes.
Malware is the total name for different malicious software variations, including viruses, ransomware and spyware. Shorthand for malicious software, malware typically involves code made by cyber attackers, planned to make wide damage data and systems or to increment unapproved induction to an association. Malware is routinely passed on as an association or report over email and requires the customer to tap on the association or open the record to execute the malware. Malware has truly been a danger to individuals and relationship since the mid-1970s when the Creeper virus at first appeared. Starting now and into the foreseeable future, the world has been persevering through a surge from a colossal number of different malware varieties, all with the arrangement of causing the most unsettling influence and damage as could be normal under the circum-positions. Malware and its 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 malware [11]. It is anything but difficult to identify known malicious program in a framework yet the issue emerges when the malware is obscure. Since, obscure malware can't be recognized by utilizing accessible known malware marks. Mark based discovery procedures neglects to detect unknown and zero-day assaults. An epic methodology is needed to speak to malware includes successfully to detect obfuscated, unknown, and mutated malware [12].
Sometimes we face problem like we are in a website and it is entering another website having different domain. This websites or links having different domain are called the invalid link directory. In our thesis, we detect those false or invalid links and give warning to the user about that invalid link. Identification of malevolent site page's strategies incorporates black-list and white-list strategy are utilized. In any case, the black list and white list advancement is vain if a particular URL isn't in list [13].
Nudity detection methodologies assume a significant function in arrangements zeroing in on controlling admittance to improper substance. These systems normally apply channel or comparable methodologies so as to identify nakedness in computerized pictures [14]. In order to achieve our goal, we need to introduce an algorithm of image classifier that can classify an image and can label it as safe or not safe. There's has been number of works published on pornographic image detection over the time. Earlier people used to measure nudity based on a human structure model [15], [16], [17]. There we have seen in most of the cases they have used simple background and completely naked people which make it very simple to detect and classify them however, in reality the images and their background are very complex in nature. So, those primitive methods don't work efficiently as per the expectations. There are some different chips away at explicit picture identification have distributed where they picked skin tone to recognize bare pictures against ordinary pictures[18] [19].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, color moment, histogram [20], [21]. Yet, it likewise can't give the normal outcome. We chose to utilize CNN for picture order. The convolutional neural organization (CNN) as one of the techniques in profound learning has indicated prevalence in grouping these pictures. The CNN can beat the human arrangement exactness in the ImageNet Large Scale Visual Recognition Competition (ILSVRC).

VI. Literature Review

There are some extensive literatures on malware attack in a website using image processing and machine learning. The relevant papers which most influenced our work will be mentioned here. Through many approaches nudity can be detected.
One of pioneering work is done by Dragos Gavrilut¸ Mihai Cimpoes¸ Dan Anton and Liviu Ciortuz [22]. In this paper, they presented a machine learning system for malware detection planning to get as barely any fake positives as possible, by utilizing a simple and a basic multi-stage combination (cascade) of various adaptations of the perceptron algorithm. They were extremely near the objective, in spite of the fact that they actually have a non-zero fake positive rate.
In another research paper done by E. Rokkathapa and S. Kanrar [23], proposed a method if a huge amount of malicious file has been context to identify the best classifier and maximizing QoE for malware detection source. 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 modeled under supervised learning. In this research paper around 3000 datasets were experimented among which 1400 were malicious programs.
This type of work is done by Ammar Yahya Daeef, R. Badlishah Ahmad, Yasmin Yacob, Ng Yen Phing [24], This paper creates recognition framework with a wide security scope utilizing URL includes just which is relying upon users straightforwardly manage 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 process in order to expand the detection performance in a real time. The simulation results of the proposed framework demonstrated phishing URLs recognition exactness with 93% and gave online process of a singular URL in average season of 0.12

second.

Similar kind of work is also done by Cho Do Xuan, Hoa Dinh Nguyen and Tisenko Victor Nikolaevich [25], they utilized AI algorithms to characterize URLs dependent on the features and conducts of URLs. The features are extricated from static and dynamic conducts 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 utilized, Support vector machine (SVM) and Random forest (RF).

There is an interesting work done by Murali R. Bala, K. Aakash, S. Anand, Sekar A. Chandra [26]. This paper examines and executes a novel technique to channel questionable indecent pictures showed in websites, as this issue stays a fascinating issue to be tended to with regards to the present situation. The algorithm works with human body area utilizing shape, shading and picture focused pixel scanning analysis. The output of pixel checking approach are broke down and enhanced integrated filter is intended to improve the exhibition. The idea of pixel filtering approach is tested and shown in real time utilizing a web-based interface.

Adult content on a website can be detected by a pixel-based approach presented by Garcia, Revano, Habal, Contreras, Enriquez [27]. For this, all the multimedia files are being processed, segmented and filtered to analyze skin-colored 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 color count and considered as pornographic in nature which aims to classify images and video frames that may contain nudity. With pixel wise kin detection with image processing for this pornography filtering, precision of 90.33% and accuracy of 80.23% were obtained.

Clayton Santos, Eulanda M. dos Santos, Eduardo Souto [28] proposed a nudity detection algorithm which offers a system for nudity discovery in images, which is separated into two modules: (1) filter skin identification; and (2) image zoning. The target of image zoning module is to partition pictures into independent separate parts dependent on the presumption that a nude image presents higher measure of skin pixels in its focal locale. It is imperative that these features are the most regular data utilized in works managing nudity location in pictures. At last, nudity recognition is performed utilizing SVM (Support Vector Machines).

VII. Algorithms

We use several algorithms for our research. Those are explained in this section

A. Logistic Regression

If the dataset is can be separated linearly than Logistic regression is the best . The algorithm uses an S shape curve to detect true or false from the value. Essentially, this is a grouping calculation utilized for extortion discovery, spam email, malware and so forth. Strategic curve is certainly not a straight one. It is a sigmoid curve. Probability p = 1/1-e^{-}z $where$ $z = mx + c$. The condition ensures that the indicator is somewhere in the range of 0 and 1, where the greater qualities are 1 and lower esteems are treated as 0. After component extraction, we applied this calculation. The benefit of utilizing this calculation is that it doesn't require an excessive amount of computational asset and is amazingly interpretable. It is simpler to actualize, decipher and proficient to prepare. A relapse model figures the class enlistment probability for one of the two orders in the informational index [29]. However, the Logistic curve is not a straight curve like linear regression.

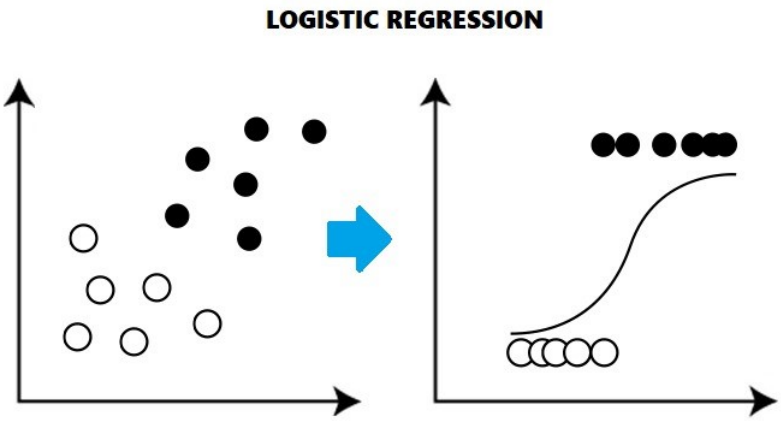


Fig. 1: Logistic Regression Algorithm

Pros and Cons of Logistic Regression:

Pros:

1) It trains very quickly and you can implement it very easily.
2) It gives very good accuracy for simple data sets.
3) Very much effective when the data is linearly separable.

Cons:

1) Used to predict discrete functions.
2) It can't handle missing values unlike RF which is immune to it as its underlying are decision trees.

B. K-Nearest Neighbors (K-NN)

K Nearest Neighbors can predict points in near proximity. This classifier classifies new cases based on a similarity of previously stored available cases. K-NN is very easy to implement and doesn't need any training period which is the main advantage of this algorithm. This algorism needs no tanning before making predictions, new data are often added seamlessly which is able to not impact the accuracy of the algorithm, new information are frequently added consistently which can not affect the exactness of the calculation.. On the contrary, though it is easy to implement, it is a slow algorithm and doesn't work properly with large dataset. It needs the scaling of data. Without scaling, it generates wrong

predictions. K-NN can't deal with missing value problems. Unlike almost any other algorithm, K-NN works due to its deeply rooted mathematical theories. The initial steps are to change the information focuses into include vectors or their numerical criticalness when implementing K-NN. Then the algorithm works by calculating the difference between these point's mathematical values [30].

Pros and Cons of KNN:

Pros:

1) Simple implementation.
2) Very easy to implement for multi class problem.
3) Makes no prior assumption of the data.

Cons:

1) K-NN is a slow algorithm.
2) Predict time is very high as the model finds the distance with every data point.

C. Random Forest

One of the best supervised machine learning algorithm is random forest. Random forest can be used for both classification and regression. Decision tree is a non-parametric AI calculation that has a tree-like diagram structure that is utilized for AI forecast, the leaf in a decision tree relates to an objective class, and every hub in a decision tree speaks to a quality. Random forest creates the forest with several trees. It is very flexible, can produce high accuracy and even with missing data it can give us the main result. Scaling of data is not necessary. Some benefits of random forest are like The Random Forest

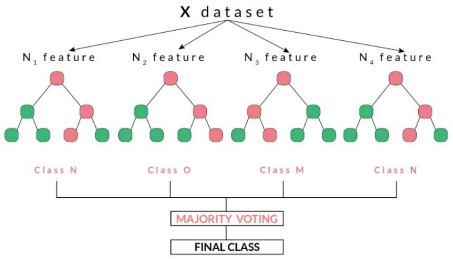


Fig. 2: Random Forest Algorithm

classifier is capable of handling missed values and the Random Forest classifier can be based on categorical values.

Pros and Cons of Random Forest:

Pros:

1) We can use Random Forest algorithm to solve both types of problems which is classification and regression.
2) In Random Forest we do not have to do much preprocessing.
3) It can estimate missing data very effectively.

Cons:

1) It creates a lot of trees, so it is quite memory hungry and increases complexity.
2) It is very hard to interpret.

D. XGBoost

XG boost is extraordinary and that implies it is a major machine teaching classifier with heaps of parts fortunately each part is entirely straightforward and easy to understand. XG Boost is an enhanced appropriated boosting library intended to be exceptionally effective, adaptable and compact. We can use XG Boost for data can also be optimize. It will ensure that over fit is not there. It also helps not to grow the tree in certain level. In order to run the algorithms we need to know about some of the pre requisites. Firstly Ensemble alludes to a gathering of individuals or things, Ensemble is an AI procedure that includes joining a few ML models to manufacture a solitary ground-breaking prescient model to get ideal outcomes. Boosting is an ensemble-based learning calculation that changes over feeble learn students to solid assessors which includes preparing ML models consecutively in a steady progression wherein in every emphasis model attempts to address the mistake made by the model in the past cycle. Gradient Boosting is a boosting procedure wherein in every cycle the new indicator is worked to fit on the pseudo-residuals of the past indicator.

Three primary types:

1) Boosting algorithm includes the learning rate.
2) Gradient Boosting with sub-testing at line, section, and frag- Inadequate Aware utilization with the modified treatment of missing data esteem. Fragment per split levels.
3) L1 and L2 regularization can be used to regularize gradient boosting.

XGBoost alters the boosting algorithm dependent on GBDT (Gradient Boosting Decision Trees). This algorithm is made out of various relapse trees, and the last result is an additional substance blend of the choice aftereffects of all subtrees. Notwithstanding, the glaring forbiddance of GBDT is the need to use the additional stumble of the n-1th tree when preparing the nth tree, which makes GBDT hard to be passed on the spread structure. In order to address this issue, XGBoost uses Taylor extension on the setback work.[32]

Pros and Cons of XGBoost:

Pros:

1) Less feature engineering required.
2) It can handle large sized datasets well.
3) Very fast to interpret.

Cons:

1) Difficult interpretation, visualization is tough.
2) Overfitting is possible if parameters not tuned properly.

E. Extra Tree Classifier

Extra Trees Classifier is a kind of gathering learning procedure, which totals the consequences of numerous de-associated choice trees gathered in a "forest" to show its classification result. In a random forest, we grow multiple trees such that each tree comprises of the square root of the total number

of features that are present. Therefore, what exactly is the difference between an extra tree classifier and a random forest the main difference between a random forest and extra tree classifier lies in the fact that instead of computing the locally optimal split for a feature combination a random value is selected for the split for the extra tree. So again, the whole idea is rather than not spending time and finding out the best splitting point. We randomly pick up a point and split based on that this leads to more diversified trees and less splitters to evaluate when training and extremely random forest. When extra classifiers where tested with the readily available data sets, we observed that when we had noisy features in our data set extra tree classifiers seemed to outperform random forest. However, when all the features are relevant and when you train both extra classifier as well as a random forest both methods seem to have achieved the same performance. From a computational perspective, the unpredictability of the tree developing technique is, expecting adjusted trees, on the request for N Log N as for learning test size, like most other tree developing systems. Be that as it may, given the effortlessness of the hub parting system we anticipate that the consistent factor should be a lot littler than in other troupe based techniques which locally advance cut-focuses. The parameters K, naming and M have various impacts: K decides the quality of the quality determination process, naming the quality of averaging yield clamor, and M the quality of the change decrease of the gathering model total. These parameters could be adjusted to the issue points of interest in a manual or a programmed way (for example by cross-approval). In any case, we want to utilize default settings for them to boost the computational points of interest and self-governance of the strategy. Segment 3 examinations these default settings as far as power and sub optimality in different settings. To indicate the estimation of the principle parameter K, we will utilize the documentation ETK, where K is supplanted by 'd' to state that default settings are utilized, by star to signify the best outcomes acquired over the scope of potential estimations of K, and by 'cv' if K is balanced by cross-approval.

F. Web Scraping

Web scraping is a technique to fetch and extract data from websites. While surfing on the web, many websites don't allow the user to save data for personal use or extracting data from a HTML document. Web scraping is the process of extracting and creating a structured representation of data from websites. For web scraping, we need to access the HTML of the webpage and extract useful information/data from it. There are several libraries that we can use for web scraping. Among these, here we use Beautiful Soup. This Beautiful Soup is a python library which takes care of extracting data from a HTML document. Basically, it is used as a parser for the HTML. It allows us to interact with HTML in a similar way to how you would interact with a web page using developer tools. Beautiful Soup exposes a couple of intuitive functions which can use to explore the HTML. For web scraping with Beautiful Soup library, firstly we install the Beautiful Soup library. Then, we import the library and create a Beautiful Soup object. Then we did a short python code to create a Beautiful Soup object that takes the HTML content we scraped earlier as its input.

Fig. 3: Web Scraping Algorithm

G. Convolutional Neural Network (CNN):

Convolutional Neural Network (CNN) is an artificial neural network that has so far been most popularly used for analyzing images. Although image analysis has been the most widespread use of CNN's it can also be used for other data analysis or classification problems as well. Most generally we can think of a CNN as an artificial neural network that has some type of specialization for being able to pick out or detect patterns and make sense of them.

With each convolutional layer, it needed to specify the number of filters the layer should have. These filters are actually detecting the patterns. For example, a single image has multiple edges, shapes, textures, objects etc. So, one type of quote pattern that a filter could detect could be edges and images. Filters may be able to detect specific objects like Eyes, Ears, hair, fur, feathers, scales or even beaks.

To summarize, Convolutional Neural Network employ complex techniques in order to make the right prediction. They are a great tool to be used, particularly when you intend to apply Machine Learning algorithms to images.

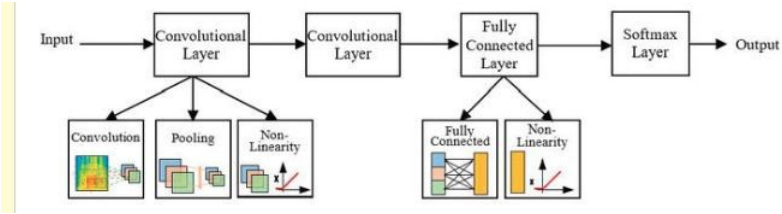


Fig. 4: Steps of CNN

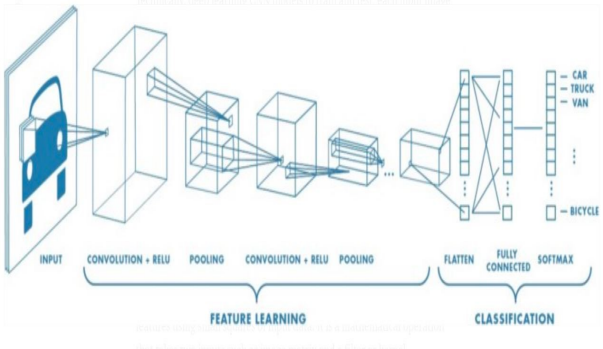


Fig. 5: How CNN Works

Pros and Cons of CNN:

Pros:

1) Good accuracy in image detection.
2) Minimizes computation compared to a regular neural network.
3) Good at handling image classification.

Cons:

1) Data requirements are leading to over fitting under fitting.
2) Network is a bit too slow and complicated if you just want a good pre-trained model.

VIII. Work plan

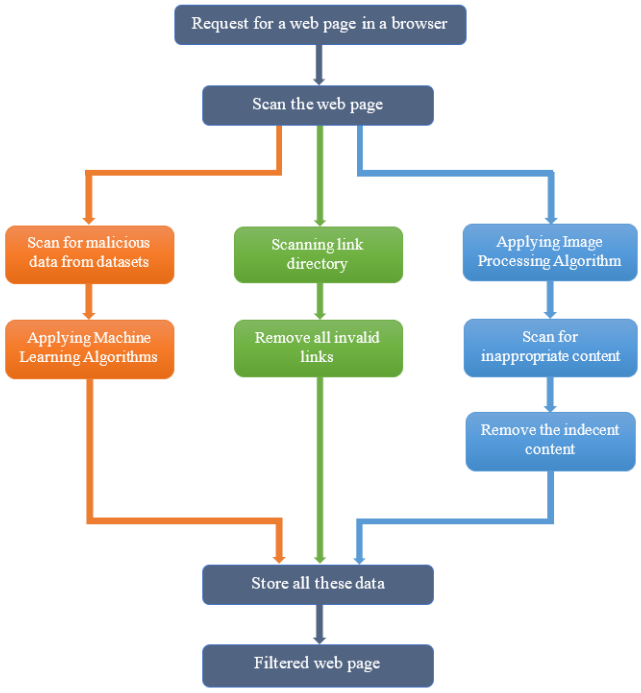


Fig. 6: Work plan of the research

Figure 6 shows the workflow diagram of the complete system. The system will follow the above-mentioned steps to fulfill our requirement.

For the research purpose, firstly we set our work plan. Without a proper planning it would not be done. We had to read several research papers and we had to find out a proper solution that we can get a safe and good browsing experience in any website.

Basically, our system ensures overall security of websites by the help of machine learning and image processing. Usually whenever, we request for web content. We often see lots of inappropriate and indecent contents in the web pages and there is hardly any software which deals with problems like inappropriate Ad blocking, link tracing, malware detection, indecent content detection etc. Our software deals with these problems in real time and generates user-friendly and filtered web contents. We will implement image process algorithms to detect unwanted pictures and Ads. Then, we also keep the track of the URL links which will stop the issues like invalid, undesired link redirection to trap sites or fishing sites. For these, we will train our system using machine learning algorithm. For malware detection from website, we used Logistic regression, K-Nearest Neighbors (K-NN), Random forest algorithm, XGBoost and Extra Tree Classifier. For link redirection part we use web scraping to extract data from websites. We use Beautiful Soup library for this case. Then we did some string checking to get some possible output. To detect unwanted pictures, we use Convolutional Neural Network.

A. Malicious Site Detection Methodology

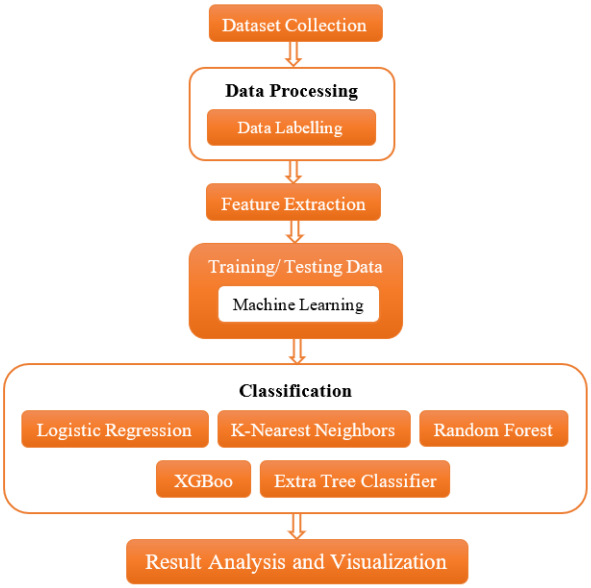


Fig. 7: Workflow of Malicious Site Detection

Figure 7 is workflow of malware 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 malware by machine learning. Our methodology incorporates a few stages to make it more precise, proficient and compelling. Our methodology incorporates preprocessing of the feature selection, Training/Testing information in machine learning classification algorithm to recognize malware or benevolent. In the wake of applying the arrangement calculation, we dissected and pictured the result.

1) Dataset: 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 immense number of highlights and properties which are import. Besides, there are around 37000 examples in the dataset.

Dataset Description

The data we are using in the dataset are represented in in columns. Each column portray a different standards.

URL is the mysterious distinguishing proof of the URL examined in the investigation. The ”URL” contains all favorable,

spam, phishing and malware URLs. URL LENGTH is the quantity of scorch acters in the URL. It decides the conceivable favorable URL. Some character may bear some danger, to distinguish them this is utilized. CHARSET is a clear cut worth and its significance is the character encoding standard (additionally called character set) to distinguish the adequate characters. CONTENT LENGTH speaks to the substance size of the HTTP header. WHOIS COUNTRY is an all out factor, its qualities are the nations we got from the worker reaction (explicitly, our content utilized the API of Whois). REMOTE IPS is a variable which has the absolute number of IPs associated with the honeypot. SOURCE_APP_PACKETS shows parcels sent from the honeypot to the server REMOTE_APP_PACKETS demonstrates bundles got from the worker. APP_PACKETS is the absolute number of IP parcels produced during the correspondence between the honeypot and the worker. TYPE is a clear cut variable, its qualities speak to the kind of page investigated, explicitly, 1 is for malicious sites and 0 is for benign sites. A portion of the sections are a higher priority than the others.

All the segments here aside from "TYPE" are functioning as contribution to the dataset that will be utilized in calculations. The yield we will have will include Boolean worth. Not many of these information sources have connections among them, particularly the ones with comparative names. The information likewise has unfilled or invalid qualities. Among the information sources "CONTENT_LENGTH" has the vacant or invalid qualities.

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

Fig. 8: Dataset Description

2) Dataset Description: In data processing, data is mined in such way that it changes crude 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.

3) Feature Extraction: There were principally 20 input, however while extraction we made sense of 2 of them were pointless and we worked with rest 18. Highlight extraction is a cycle that is basically done to get the most important sources of info and work with them without superfluous problem. This recoveries from chopping down the time span and furthermore the entanglements that might have emerged from the not all that significant data.

4) Training/Testing of Machine Learning Classifier: After the element determination measure, the subsequent stage is preparing and testing of AI classifiers. We split our dataset into preparing and testing sets. From that point forward, we train and test the dataset in some characterization calculations. Preparing information are utilized to fit and tune the models. Test information are spoken to as inconspicuous information to assess the models [33]. Test information is used to see how well the machine can predict new answers subject to preparing. In our dataset, we utilized eighty percent information as preparing information and twenty percent as testing information. In the wake of parting the dataset, we train our model.

We have used five classification algorithms for our approach. The five classifiers are

- Logistic Regression
- K-Nearest Neighbor (KNN)
- Random Forest
- XGBoost
- Extra Tree Classifier.

These are the classification algorithms we used to detect malicious site.

5) Result Analysis and Visualization: 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" curve for our final visual representation.

B. Link Redirection Methodology

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 parse data from any website. But there are some websites which do not give

permission or access to parse their data with the web scraping tools.

1) Web Scraping Tools: Various programing languages can be used for web scraping. Here, we have used python language because we can use python for various kind of web scraping tools. The web scraping tools are called web scrapers. There are many types of web scraper. For example, ParseHub, Beautiful Soup, Scrapy, Octoperse etc.

- 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.
- BeautifulSoup: Beautiful Soup 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 command 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.

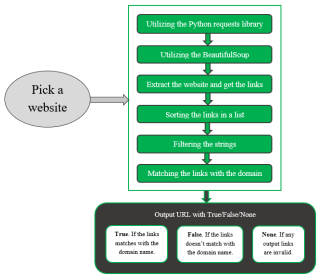


Fig. 9: Workflow of Link Redirection Methodology

2) Working Methodology of Scanning Link Directory: Figure 9 is workflow of link redirection methodology. Here we discuss about how this part of our model works.

When someone search for a website, from then on our system will start working. When our system start scraping the web. It sends a request to the server that's hosting the page we specified using Python request library. Basically, our code downloads that page's source code, just as a browser would. The BeautifulSoup library extract 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.

3) Steps to Scan Invalid Link Directory: For our system we used Python code to scan invalid link directory. We used 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:

1) 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)
2) Import BeautifulSoup: We can utilize the BeautifulSoup 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 BeautifulSoup class to parse our document.

   from bs4 import BeautifulSoup
   bSoup = BeautifulSoup(page.content, 'html.parser')
3) Now the BeautifulSoup will extract the website and get all the links that are included in that particular website. For example, we are taking www.netflix.com as the input.

   inputt = 'https://www.netflix.com'
4) 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')
5) Now it will filter the string. String means the link that is being used as input.

   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()
6) After filtering, it will match the links with 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 user wants to enter that link, there will be no problem to enter. That means user can enter where

he/she wanted to enter as a destination link. But False shows when URL domain are redirected to a different domain and None used for invalid links. In our system, false or invalid links will give warning to the user about unexpected link redirection.

C. Nudity Detection:

In our regular web surfing experience, we all faced a common issue of popping out of some unwanted pictures every now and then. In the past ten years, spreading negative content over internet such as pornography has increased significantly. This quick ascent of spreading disgusting pictures or recordings has driven numerous individuals sharing their protective measures without understanding its results. Moreover, sometimes these pictures seemed cover our web content and create an uncomfortable situation. Most of the cases, these images contain potential threats and also redirect to an unwanted advertisement of an inappropriate web page. To avoid these pictures, we have to manually shut them down which is very disturbing and time consuming. Even if we cancel them out, they pop up again and again periodically. So, if there a system exists which can dynamically detect these unwanted pictures then the user experience would become very fluent, smooth and eye pleasing. That's the motivation of us to introduce a model that can detect these sort indecent contents on its own and filter that particular searched web page based on the filtering algorithm. For completing the unwanted image

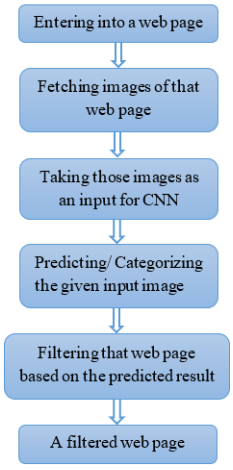


Fig. 10: Workflow of Nudity Detection

detection part, we have built a flowchart above based on that we will describe how each process will work sequentially. First of all, when a user enters a webpage, our system will fetch all the image files from that web page in real-time using web scrappers. Those fetched images will work as an input for CNN image classifier and CNN classifier would predict that image of being porn or natural. Based on that prediction result our system would filter that web page as result the user will get a filtered web page.

1) Our Approach for Image Classification: To reach that expected goal and solve these problems we came up with an idea to detect pornographic images using Convolutional neural networks. CNN is generally utilized these days for picture arrangement and acknowledgment on account of its high exactness. CNN follows a various leveled model that deals with building an organization, similar to a pipe, lastly gives out a completely associated layer where all the neurons are associated with one another and the yield is handled. There are two types of basic deep learning approaches that are convolutional and recurrent. The convolutional method is mostly used in cases related to computer vision like all sorts of image classification tasks.

2) Dataset: For image classification, we have 20,000 pictures as input where half of them are natural pictures and the rest of them are indecent pictures. We have categorized our dataset into two parts one is porn or indecent pictures and the other one is natural or normal picture. We have collected our dataset from Kaggle. We didn't have to apply any preprocessing techniques. For processing our dataset we've just sorted out the images based on image clarity and context. As we know CNN takes directly images as input and works efficiently so our maximum works relate to processing weren't needed.

3) Result Analysis and Visualization: After training and testing our dataset we get our accuracy model and loss model and by applying the data visualization technique of curve representation we get a graph of "model accuracy" and "model loss". Then for a given image, our system provides a prediction about that image.

IX. Malicious Site Detection Methodology

Logistic Regression 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 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

classifier3=LogisticRegression(random_state=42)
classifier3.fit(x_train,y_train)


pred2 = classifier3.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))
```

Fig. 11: Logistic Regression

K-Nearest Neighbor
In this section, we have used KNN where classifiers determine the region and the neighbors starting 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 find the close neighbors

Random Forest
We know random forest uses trees. Here Random forest makes

```
#__________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))
```

Fig. 12: K-Nearest Neighbor

trees form 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 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 takes the average of the outputs by different trees. 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.

```
#________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))
```

Fig. 13: Random Forest

XGBoost

Here we are using XGBoost for data can also be optimize. It will ensure that over fit is not there. It also helps not to grow the tree in certain level.

```
from sklearn.metrics import classification_report,  accuracy_score
from xgboost import XGBClassifier
model8 = XGBClassifier(learning_rate=0.5, max_depth=9, min_child_weight=3, n_estimators=100, nthread=1, subsample=1.0)
model8.fit(x_train, y_train)
pred4=model8.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
```

Fig. 14: XGBoost

Extra Tree Classifier

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

X. Link Redirection Methodology

Here "import requests" is basically making a request to a web page. We are importing Beautiful-Soup library by

```
[ ] 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))
```

Fig. 15: Extra Tree Classifier

adding this line "from bs4 import BeautifulSoup". We're also importing regular expression as "import re" and time function returns the number of seconds passed since epoch.

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

Fig. 16: Import BeautifulSoup

By this we are checking input with domain links that we got from the website.

```
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()
```

Fig. 17: Checking Input with domain links

We are filtering the string to get the domain name and later we are using it to match with the output 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('.')
```

Fig. 18: Filtering the String

In this line we are giving an input to get links from that particular website.

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

Fig. 19: Taking Input from User

"HREF" is a Hypertext reference. This HTML code utilized to make a connect to another page. The HREF is an attribute of the anchor tag, which is additionally utilized 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 connected list and printing the links with 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.

```
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))
```

Fig. 20: Hypertext Referencer

XI. Nudity Detection

Importing all the dependencies like tensorflow, keras, Conv2D, matplotlib

```
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense,Flatten,Conv2D,MaxPooling2D,Dropout
from tensorflow.keras import layers
from keras.utils import to_categorical
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
```

Fig. 21: Importing all Dependencies

Setting the directory of our data into the variable "DATADIR" and define categories into two section one is neutral and other is porn. Here neutral is referring that the image is safe for work and porn is referring that the particular test image is not safe for work.

```
DATADIR= "E/train"
CATEGORIES = ['neutral','porn']
for category in CATEGORIES:
  path =  os.path.join(DATADIR, category)
  for img in os.listdir(path):
    img_array = cv2.imread(os.path.join(path,img), cv2.IMREAD_GRAYSCALE)
    plt.imshow(img_array, cmap="gray")
    plt.show()
    break
```

Fig. 22: Setting the Directory of Data

The sequential model is used for a plain slack of layer where each and every layer has one input and one output layer. It groups a linear stack of layers into a tf.keras. Model and provides training and interference features on Sequential Model. It doesn't work properly when the model has multiple inputs or outputs. The Conv2D layer creates a convolution kernel which is convolved with the input layer in order to produce a tensor output. The operation Max Pooling basically calculates the largest value from matrix of each layer and by this pooling operation every time the input matrix cut down into half of its original size. It is a building block for CNN and it progressively reduce the size in each operation. The dropout layer helps to prevent overfitting issue in the model. It can be implemented after every dense layer. As our model is a binary classifier so we have used binary_crossentropy method.

Keras deep learning library includes three functions, model. Fit is one of them. Epoch is the complete pass through of the training data where the dataset is passed forward and backward through the neural network in every iteration. Here the epoch

```
In [17]: model = tf.keras.models.Sequential([
             tf.keras.layers.Conv2D(16 , kernel_size=(3,3), activation="relu"),
             tf.keras.layers.MaxPooling2D((2,2)),
             tf.keras.layers.Conv2D(32, kernel_size=(3,3), activation="relu"),
             tf.keras.layers.MaxPooling2D((2,2)),
             tf.keras.layers.Conv2D(64, kernel_size=(3,3) , activation="relu"),
             tf.keras.layers.MaxPooling2D((2,2)),
             tf.keras.layers.Conv2D(128, kernel_size=(3,3) , activation="relu"),
             tf.keras.layers.MaxPooling2D((2,2)),
             tf.keras.layers.Flatten(),
             tf.keras.layers.Dense(200, activation="relu"),
             tf.keras.layers.Dropout(.2, input_shape=(2,)),
             tf.keras.layers.Dense(100, activation="relu"),
             tf.keras.layers.Dense(50, activation="relu"),
             tf.keras.layers.Dense(1, activation="sigmoid")
         ])

         model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'
```

Fig. 23: Adding layers in the model

no is set to 10 which we have to tweak to get a better result. Verbose basically provides a GUI.

```
history = model.fit(train_generator,
                                epochs=10, #Change the number of epochs
                                verbose=1,
                                validation_data=validation_generator)
```

Fig. 24: Setting Epoch number

XII. Malicious Site Detection Methodology

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.

A. Data Visualization

So as to work with any dataset, it is imperative to envision 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.

1) Heatmap: In the dataset there are all out 37,000 examples. So as to visualize the malicious website data set, we have utilized heatmap device which is a two-dimensional graphical portrayal of the matrix and shows the co-connection of information. Figure 23 shows the co-connection between input information as heatmap.

A heat map is a graphical representation of data where the individual values are expressed as colors. Heat maps provide an effective visual summary of information because

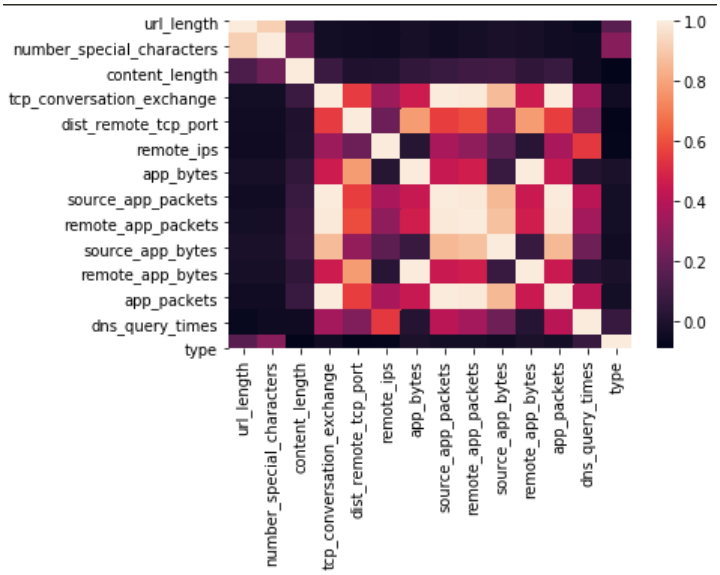


Fig. 25: Heatmap of input data

they synthesize data and then present it in a pictorial form, which means it uses a system of color-coding to represent different values. Analytics tools can provide metrics to show which data are used most but they can lack detail when it comes to understanding how it engages. Heat maps can give a more comprehensive overview of how things are behaving. To visualize the malicious website dataset we have used heat map tool which is a two-dimensional graphical Representation o From the heatmap, we can know the importance level of features in the dataset. From the heatmap, we found that "URL_LENGTH", "NUMBER_SPECIAL_CHARACTERS", "SOURCE_APP_PACKETS" etc. were most important as the scale indicated that the color was close to 1.0 which represented high importance of the feature is the matrix and shows the co-relation of input data.

2) Pie Plot: Pie plot is one of the most widely used forms of visualizing data, it is used to plot the pie chart. In the pie chart, the percentage of data in each category is represented as a slice of the circle. A pie chart is utilized when attempting to work out the structure of something. In the event that we have categorical data, at that point utilizing a pie chart would work truly well as each slice can speak to an alternate classification. With the help of pie chart statistical data can be easily represented. Here in the pie plot, 0 represents benign and 1 represents malware data.

B. Result analysis

To assess our model, Sensitivity, Specificity, exactness score measurements are utilized. We likewise assess the ROC bend, AUC, disarray grid, exactness score, to choose the best calculation for vindictive site page identification.

We utilized five classifiers to characterize the malware and generous information, they are Logistic Regression, K-Nearest Neighbor (K-NN), Random Forest, XGBoost and Extra Tree Classifier. We likewise assessed the presentation of the classifiers with the assistance of various measurements which is given bellow.

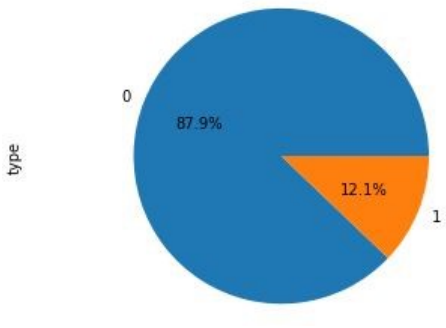


Fig. 26: Pie Plot

- Sensitivity:Sensitivity shows the capacity of the classifiers to discover malicious webpage that are really of malicious category. Sensitivity = True Positive / (True Positive + False Negative)
- Specificity: Specificity shows the capacity to accurately recognize benign websites that are without the state of malicious websites.
  Specificity = True Negative / (True Negative + False Positive)

Confusion Matrix

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

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

TABLE I: Table of Confusion Matrix

True Negative (T.N) implies that the algorithm predicted negative and the outcome is really negative, False Positive (FP) implies that the algorithm predicted positive however the outcome is really negative, False Negative (FN) implies that the algorithm predicted negative yet the outcome is really sure and True Positive (TP) implies that the algorithm predicted positive and the outcome is actually positive.

True Positive Rate (T.P.R): TPR is also known as recall. It shows the ability of our classifier to detect malware. T:P:R = TruePositive=(TruePositive + FalseNegative)

False Positive Rate (F.P.R): FPR shows that the possibility of benign_les wrongly classified as malware. F:P:R = FalsePositive=(FalsePositive + FalseNegative)

ROC Curve

Receiver Operating Characteristic curve (ROC Curve) is utilized to plot the positive rate rate against the false positive rate. With the assistance of ROC Curve, we can discover the capacity of a classifier to separate between various classes. The Area Under Curve (AUC) is the region under the ROC Curve, with the assistance of AUC score we get how well the model is performing.

Accuracy Score

Accuracy score is the most widely recognized measurement for assessing a model. We utilized sklearn Accuracy score metrics to discover the accuracy score for various algorithms. The formula for finding the accuracy score is as follow:

$$\text{AccuracyScore} = \frac{T.P+T.N}{T.P+T.N+F.P+F.N}$$

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

The results obtained from the different algorithm is described as follows :

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%

TABLE II: Comparison of values

In comparison, we see a precision of 96.12% utilizing our Random forest algorithm and 89.05% utilizing Logistic Regression and 91.91% utilizing K-NN. From this, the outcome that can be drawn is that by utilizing supervised machine learning benign and malicious websites can be related to demonstrated accuracy. The result depicts that among the three algorithms the Random forest is performing the best.
The results obtained from the different algorithm is described as follows:

1) Logistic Regression: For Logistic Regression we get an accuracy score of 0.887.

The confusion matrix for Logistic Regression is as follows:

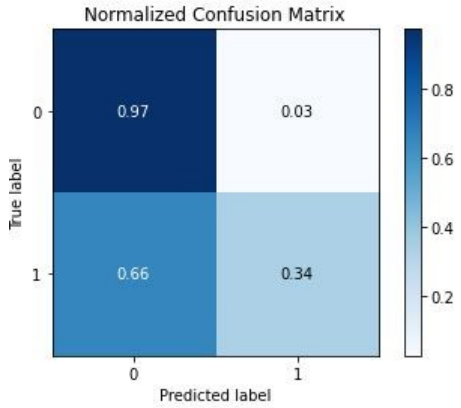


Fig. 27: Confusion Matrix of Logistic Regression

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.

The ROC curve for Logistic Regression is as follows:

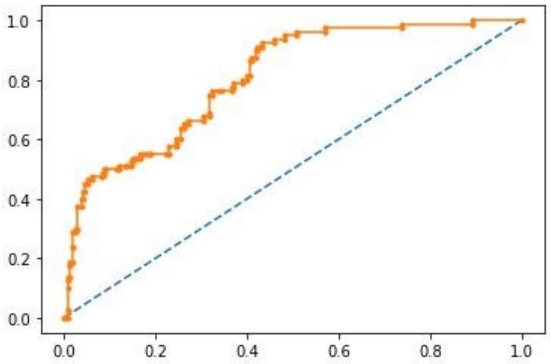


Fig. 28: ROC Curve of Logistic Regression

For Random Forest we get the AUC (Area Under Curve) score 0.809.

2) 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.

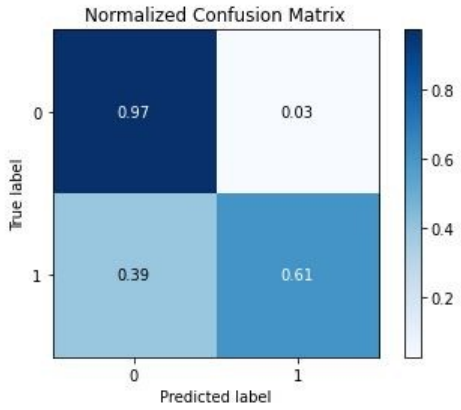


Fig. 29: Confusion Matrix of K-Nearest Neighbor

The ROC curve for K-Nearest Neighbor is as follows:

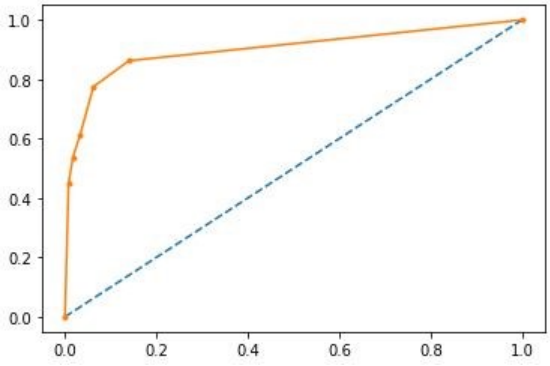


Fig. 30: ROC curve of K-Nearest Neighbor

For K-Nearest Neighbor we get the AUC (Area Under Curve) score 0.900.

3) 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.

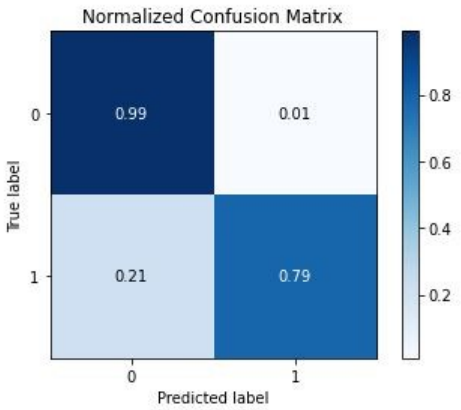


Fig. 31: Confusion Matrix of Random Forest

The ROC curve for Random Forest is as follows:

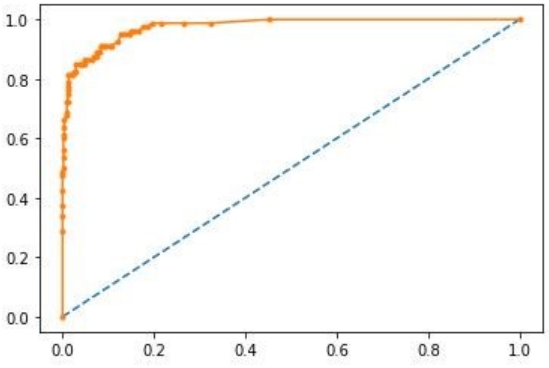


Fig. 32: ROC curve of Random Forest

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

4) XGBoost: For XGBoost we get an accuracy score of 0.955.

The confusion matrix for XGBoost is as follows:

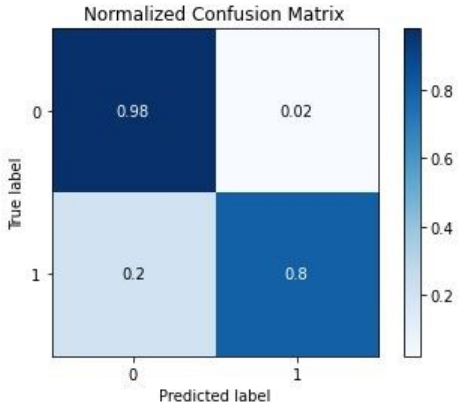


Fig. 33: Confusion Matrix of XGBoost

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:

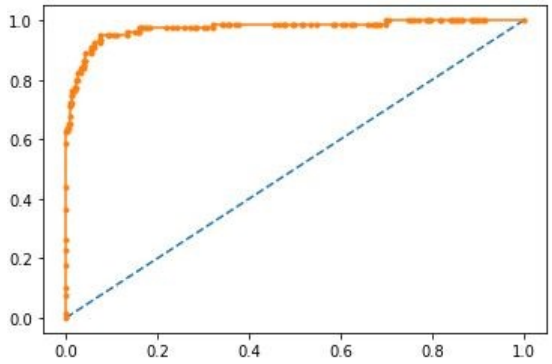


Fig. 34: ROC curve of XGBoost

For XGBoost we get the AUC (Area Under Curve) score 0.975.

5) 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:

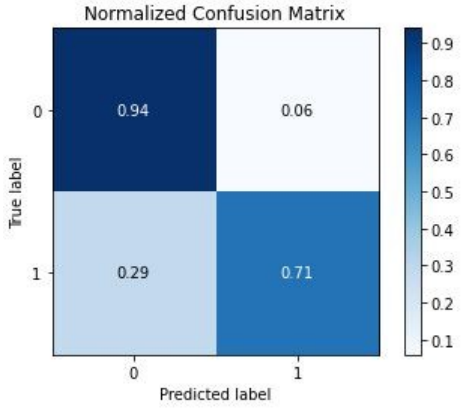


Fig. 35: Confusion Matrix of Extra Tree Classifier

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:

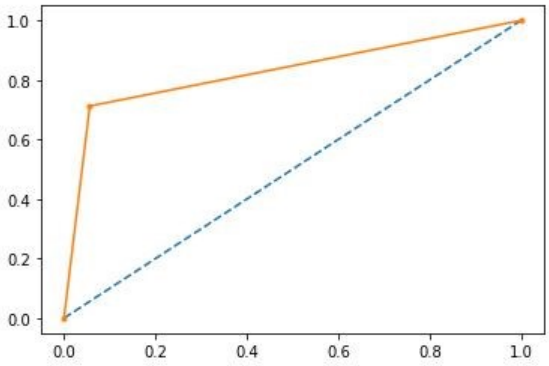


Fig. 36: ROC curve of Extra Tree Classifier

For Extra Tree Classifier we get the AUC (Area Under Curve) score 0.828.

Area Under Curve (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 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. From our value the highest AUC is 0.977 for Random Forest algorithm.

XIII. Link Redirection Methodology

A. Result analysis

We are using Link Redirection Methodology to scan the invalid links which have 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 give permission for 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 bellow:

Link	Result
/login	None
tel:1-844-505-2993	None
https://help.netflix.com/support/412	True
https://help.netflix.com	True
/youraccount	None
https://media.netflix.com/	True
http://ir.netflix.com/	True
https://jobs.netflix.com/jobs	True
/redeem	None
/gift-cards	None
/watch	None
https://help.netflix.com/legal/termsofuse	True
https://help.netflix.com/legal/privacy	True
https://help.netflix.com/legal/privacy#cookies	True
https://help.netflix.com/en/node/2101	True
https://help.netflix.com/contactus	True
https://fast.com	False
https://help.netflix.com/legal/notices	True
https://www.netflix.com/browse/genre/839338	True

TABLE III: Output of the given input

Table III is the output of given input Here, we can see three types of result (True, None, False). If the result is True, our system will let the user get into the link. If the result is None, it means that is an invalid link and our system will automatically block that link. If the result is False, it means the link is valid but it has a different domain. Then our system will send a warning massage to the user and ask if he/she wants to continue to enter a link that has 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 plan to extend this work field.

We have tested our system on 25 websites. The list of that 25 websites is given below:

https://adobe.com
https://facebook.com
https://www.youtube.com
https://netflix.com
https://mozilla.org
https://nih.gov
https://themeforest.net
https://microsoft.com
https://brazzers.com
https://cdc.gov
https://github.com
https://wikimedia.org
https://archive.org
https://aliexpress.com
https://www.yahoo.com
https://cpanel.net
https://www.wix.com
https://mozilla.com
https://www.over-blog.com
https://www.wikipedia.org
https://whitehouse.gov
https://teamviewer.com
https://dribbble.com
https://state.gov
https://ndtv.com

Fig. 37: Checked Website for Our System

XIV. Nudity Detection

A. Data Visualization

So as to work with any dataset, it is imperative to envision it, python has a lot of libraries with the assistance of which we can undoubtedly visualize the data. We use our accuracy model and loss model and by applying the data visualization technique of curve representation we get a graph of "model accuracy" and "model loss". Then for a given image, our system provides a prediction about that image.

The model accuracy and model loss curve of our CNN model is given below:

Model Accuracy

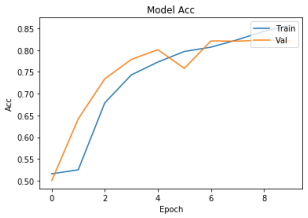


Fig. 38: Model Accuracy for Our System

Model Loss

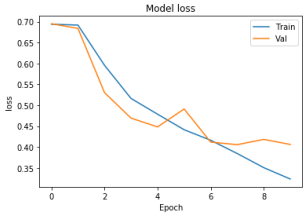
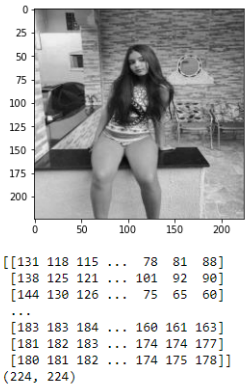


Fig. 39: Model Loss for Our System

Result analysis



```
[[131 118 115 ...  78  81  88]
 [138 125 121 ... 101  92  90]
 [144 130 126 ...  75  65  60]
 ...
 [183 183 184 ... 160 161 163]
 [181 182 183 ... 174 174 177]
 [180 181 182 ... 174 175 178]]
(224, 224)
```

Fig. 40: Representing the Image into 2D Array

```
In [132]: IMG_SIZE = 50
          new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
          plt.imshow(new_array, cmap='gray')
          plt.show()
```

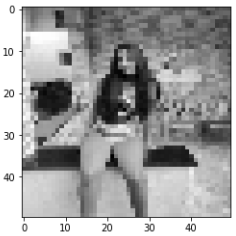


Fig. 41: Converting Image into GreyScaleImage

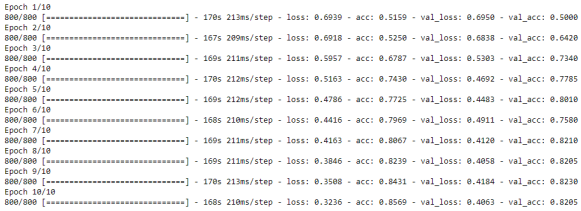


Fig. 42: Result After Model Training

After training and testing our dataset we get an accuracy of 0.8569 and loss of 0.3236. So, from this stat, we can see our model accuracy is approximately around 85%, and model loss around 32%. For testing purposes, we have only put the value of epoch at 10 which we will tweak later into bigger no to see the fluctuation.

Fig. 43: Taking a Random Image as an Input

Fig. 44: Resizing the Image

neutral: 84.4578958644583%
porn: 15.5422542355417%

Here, we can see the prediction result of the test image. The test image is referring a player and it is expected to be shown as a neutral image. The result prediction exactly refer that the given image is a neutral.

XV. Conclusion

The main goal of our project is to build a model that helps a user to experience a filtered and safe web surfing. In order 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 such malicious site, unwanted link redirection and nudity detection part. For now, 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 which can help user from any unwanted link redirection. Fortunately, we applied our process to (“.com”, “.net”, “.org”, “.gov”) these domain types. We used web scraping technique to parse data from website. Then we matched the input domain name with the output domain name. Here we got three types of result (True, None, False). If the result is True, our system will let the user get into the link. If the result is None, it means that is an invalid link and our system will automatically block that link. If the result is False, it means the link is valid but it has a different domain. Then our system will send a warning massage to the user and ask if he/she wants to continue to enter a link that has a different domain. Finally, we worked with nudity detection part. As our plan is to filter the unusual and unwanted pictures which pop up every now and then while web browsing. So, we decided to implement such model which is dynamic in terms of making judgment whether that popped up image is containing nudity or any sort of unwanted contents. As the task seems to be very complex, we had to introduce a model which can take an image as input then process it and classify it under certain categories. So, we decided to implement Convolutional Neural Network for image classification. This Deep Learning algorithm requires a ConvNet which is much lower as compared to other classification algorithms. While in primitive methods filters are hand- engineered, with enough training, ConvNets

have the ability to learn these filters/characteristics by own. Fortunately, we also get a good accuracy using CNN to detect indecent picture/image. To include, we can proudly say that instead of having so many obstacles, we have tried our best to integrate three totally different wings all together.

XVI. Future Works

Apart from all of the approaches that we have taken so far, there are still exist more rooms for improvements. 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 which can be fixable and very much implementable. Then, for filtering the unwanted images we've seen nowadays the web images are very complex in nature in terms of background and a huge number of objects in it which make it harder for our present system to detect and categorize them. Like there are some pictures that 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 are planning to implement a hybrid algorithm which will combine more than one algorithm to perform more efficiently and perfectly in our malicious link detection part. Finally, as the problems are nowadays getting complicated and critical so the solutions also need to be updated to provide a better result.

References

[1] Wyatt Yost, Chetan Jaiswal, "MalFire: Malware Firewall for Malicious Content Detection and Protection", presented at 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 19-21 Oct 2017

[2] "2019 internet security threat report," https://www.symantec.com.

[3] Gallup, inc, "cybercrimes remain most worrisome to americans," presented at gallup.com, 09 Nov 2018.

[4] "Cybertech europe 2017 i accenture," presented at i accenture. https://www.accenture.com

[5] Symantec, "Symantec Internet Security Threat Report –Trends for 2010", presented at Dispo -nível em, June 2011.

[6] "Unvalidated Redirects and Forwards Cheat Sheet". Open Web Application Security Project (OWASP). 21 August 2014.

[7] V. Krammer, Taylor Francis "An Effective Defense against Intrusive Web Advertising", presented at Sixth Annual Conference on Privacy, Security and Trust, 14 Mar. 2008

[8] Elliott L. Post, and Chandra N Sekharan, "Comparative Study and Evaluation of Online Ad-blockers" presented at 2015 2nd International Conference on Information Science and Security (ICISS), 14-16 Dec. 2015.

[9] 10 cyber security facts and statistics for 2018, "https://us.norton.com/internetsecurity-emerging-threats-10-facts-about-todays-cybersecurity-landscape-that-you-should-know.html"

[10] Chirag Ahuja; Anurag Singh Baghel; Gotam Singh," Detection of nude images on large scale using Hadoop", Published in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)

[11] Dragos Gavrilut¸ Mihai Cimpoes¸ Dan Anton and Liviu Ciortuz, "Malware Detection Using Machine Learning", presented at on International multiconference on Computer Science and Information Technology pp. 735–741, April 2009.

[12] E. Rokkathapa and S. Kanrar, "A novel approach for predicting the malware attack", presented at International journal of computer applications (0975 – 8887), vol. 181, no. 45, Mar. 2019.

[13] Wu Liu,Ping Ren, Ke Liu, Hai-xin Duan, "Behavior-Based Malware Analysis and Detection", published at 2011 First International Workshop on Complexity and Data Mining.

[14] Om Prakash Samantray ; Satya Narayan Tripathy ; Susanta Kumar Das, "A study to Understand Malware Behavior through Malware Analysis", Published in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN).

[15] M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In Computer VisionłECCV'96, pages 593–602. Springer, 1996.

[16] D. A. Forsyth and M. M. Fleck. Identifying nude pictures. In Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on, pages 103–108. IEEE, 1996.

[17] D. A. Forsyth and M. M. Fleck. Automatic detection of human nudes. International Journal of Computer Vision, 32(1):63–77, 1999.

[18] L. Duan, G. Cui, W. Gao, and H. Zhang, "Adult Image Detection Method Base-on Skin Color Model and Support Vector Machine," Comput. Vision, 5th Asian Conf. on. ACCV 2002. Proceedings., no. January, pp. 1–4, 2002.

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

[20] J. S. Lee, Y. M. Kuo, P. C. Chung, and E. L. Chen, "Naked image detection based on adaptive and extensible skin color model," Pattern Recognit., vol. 40, no. 8, pp. 2261–2270, 2007.

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

[22] Rajesh Kumar; Xiaosong Zhang; Hussain Ahmad Tariq; Riaz Ullah Khan," Malicious URL detection using multi-layer filtering model", Published in 2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).

[23] Clayton Santos; Eulanda M. dos Santos; Eduardo Souto," Nudity detection based on image zoning", Published in: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)

[24] S. Dreiseitl and L. Ohno-Machado, regression and artificial neural network classification models: A methodology review", Journal of biomedical informatics, vol. 35, no. 5-6, pp. 352359, 2002.

[25] K-nearest neighbors (knn) algorithm for machine learning, https://medium. com / capital - one -tech / k - nearest - neighbors - knn - algorithm - for - machine - learning-e883219c8f26, (Accessed on 12/24/2019)

[26] How the random forest algorithm works in machine learning, https://dataaspirant. com/2017/05/22/random-forest-algorithm-machine-learing/, (Accessed on 12/24/2019).

[27] S. Hutchinson and U. Karabiyik, analysis of spy applications in android devices", 2019.

[28] Ammar Yahya Daeef , R. Badlishah Ahmad , Yasmin Yacob , Ng Yen Phing, "Wide scope and fast websites phishing detection using URLs lexical features", presented at 2016 3rd International Conference on Electronic

[29] Design (ICED), 11-12 Aug. 2016. Cho Do Xuan, Hoa Dinh Nguyen and Tisenko Victor Nikolaevich, "Malicious URL Detection based on Machine Learning", presented at (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 1, 2020.

[30] Murali R. Bala , K. Aakash , S. Anand , Sekar A. Chandra, "Intelligent Approach to Block Objectionable Images in Websites", presented at 2010 International Conference on Advances in Recent Technologies in Communication and Computing, 16-17 Oct. 2010.

[31] Manuel B. Garcia, Teodoro F. Revano, Beau Gray M. Habal, Jennifer O. Contreras, John Benedic R. Enriquez, presented at "A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm", IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018.

[32] Clayton Santos, Eulanda M. dos Santos, Eduardo Souto, presented at "NUDITY DETECTION BASED ON IMAGE ZONING", presented at The 11th International Conference on Information Science, Signal Processing and Their Applications: Special Sessions, 2012.

[33] Chapter 6: Model training with machine learning - data science primer, https: //elitedatascien- ce.com/model-training, (Accessed on 12/07/2019).