Smart Protection for Website Using Machine Learning and Image Processing

by

Sumsil Arafin Pranta
17301174
Mohimen-Al-Tahsin
16201034
Fatin Ishraq Shapnil
17101258
Md. Hazzaz Rahman Antu
16301190
Md Rafidul Islam
16301194

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October, 2020

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sumsil Arafin Pranta

Sumsil Arafin Pranta
17301174

Mohimen-Al-Tahsin

Mohimen-Al-Tahsin
16201034

Fatin Ishraq Shapnil

Fatin Ishraq Shapnil
17101258

Md Hazzaz Rahman Antu

Md. Hazzaz Rahman Antu
16301190

Md Rafidul Islam

Md Rafidul Islam
16301194

Approval

The thesis titled "Smart Protection for Website Using Machine Learning and Image Processing" submitted by

1. Sumsil Arafin Pranta (17301174)
2. Mohimen-Al-Tahsin (16201034)
3. Fatin Ishraq Shapnil (17101258)
4. Md. Hazzaz Rahman Antu (16301190)
5. Md Rafidul Islam (16301194)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 09, 2020.

Examining Committee:

Supervisor:
(Member)

DR. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

DR. Md.Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

DR. Mahbubul Alam Majumdar
Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

The thesis is carried out in complete compliance with research ethics norms, and the codes and practices set by BRAC University. In our thesis we use the data from primary sources. We collect data from different participants and we use our own dataset for our thesis. we are ensuring we use references and in text citations properly. We the five co-authors take full responsibility for the thesis code violations. For solving problems, we read different websites, youtube tutorials, and Questionnaire Free tools. We also took help from our university faculty members. Finally, we declare that we give credit every people from whom we took help. We did not make any fraud able means for completing the thesis. Our work is in compliance with the ethics standard set by BRAC university.

Abstract

The 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 various problems. There have been so many loops and data breaches in web content as a result it's become very easy for the criminals to manipulate those security gaps and inject viruses or other unwanted and harmful contents. Almost every one of us faces the problem of getting trapped in malicious sites or clicking on some popped up advertisements which ended up into absolutely an indecent web page that contain potential threats. As a result, most of our private data is being compromised or got leaked, which create so many problems in both our corporate and private life. So, it's become a top priority to the security analyst and consults to ensure web security as it has become an integral part of every sphere. Keeping these facts in our mind, we have come up with 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 accessing various malicious sites, unwanted link redirection, and pornographic image contents that pop up quite frequently in the time of browsing webpages. To make our system able to detect these problems, we will implement machine learning and image processing algorithms and provide the users a filtered, smooth, and safe user experience.

Keywords: Malicious Site, Machine learning, Link redirection, Image processing

Dedication

We would like to dedicate this research to our parents. Without their support we may not able to do our studies. We also want to dedicate the research to our friends who helped us to do better and most importantly the participants who helped us collecting data from survey. Our supervisor helped us throughout the year. We want to dedicate this to him also.

Acknowledgement

Firstly, we would like to say thanks to our beloved family members whom we will always be indebted to. Secondly, we thank our supervisor DR. Muhammad Iqbal Hossain for all his help and constant guideline. Secondly, we would like to thank all the faculty members and stuffs for providing us with such a wonderful study environment where could develop ourselves as well as this research properly. Our thanks and appreciations also go to people who have willingly helped us out with their abilities in this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

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[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]. According to the report, the business analyst claimed that a company spends approximately 2.4 million USD because of malicious 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.According to the reports from Symantec, there has been a connection between these pornographic images and malicious contents[5].

Not only that. It is also often seen that when we press a link, it sometimes changes the destination and takes us to another link. These kinds of link redirection is com-

monly used to redirect into malicious websites which contains potential threats[6].

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.

1.1 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. 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.

1.2 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.

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.

1.3 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.

1.4 Thesis Structure

Chapter 1 - Introduction where motivation, problem statement, objectives and contribution are discussed.

Chapter 2 - Related works where background, literature review and algorithms are discussed. In the literature review part, we reviewed the previous work on malware detection, link redirection and nudity detection.

Chapter 3 - Our Proposed Approach, here we discussed about our workflow and our methodology to detect the malware, scan link directory and detect nudity content.

Chapter 4 - Experimentation where we discussed about the code used in our work.

Chapter 5 - Includes Result Analysis and Data Visualization, where we visualized our data and analyzed the result obtained from our algorithms.

Chapter 6 - Conclusion and Future works where the conclusion consists of our work till now and future works include the scope of improvement.

Chapter 2

Related Work

We divided related work into background and literature review.

2.1 Background

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

Malicious contents 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 malicious contents varieties, all with the arrangement of causing the most unsettling influence and damage as could be normal under the circum-positions. malicious contents 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 malicious contents [11]. It is anything but difficult to identify known malicious program in a framework yet the issue emerges when the malicious contents is obscure. Since, obscure malicious contents can't be recognized by utilizing accessible known malicious contents marks. Mark based discovery procedures neglects to detect unknown and zero-day assaults. An epic methodology is needed to speak to malicious contents includes successfully to detect obfuscated, unknown, and mutated malicious contents [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 comput-

erized 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.

2.2 Literature Review

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

One of 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).

2.3 Algorithms

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

2.3.1 Logistic Regression

If the dataset is can be separated linearly then applying Logistic regression is the best option. The algorithm uses an S shape curve to detect true or false from the value. basically, this grouping algorithm is utilized for extortion discovery, spam email, malicious, and so on. As it is a sigmoid curve, the strategic curve is not a straight curve.

$$Probabilityp = 1/1 - e^{-}z \ \ where \ \ z = mx + c. \tag{2.1}$$

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. Before applying this algorithm, we need to extract the components. One of the positive sides of this algorithm is that is doesn't require an excessive amount of computational asset and is very much interpretable. It is simpler to actualize, decipher and proficient to prepare. The regression model figures out the probability for one the two classification in the dataset[29]. Unlike linear regression, it's not a straight curve.

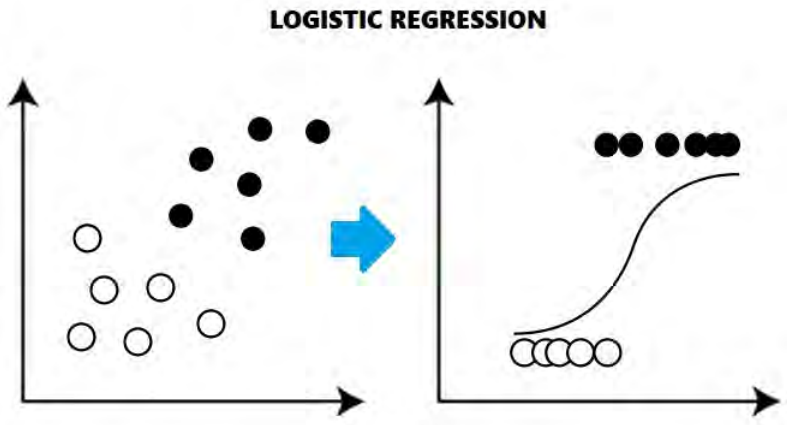


Figure 2.1: Logistic Regression Algorithm

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.

2.3.2 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.

2.3.3 Random Forest

Random forests or random decision forests are an ensemble learning technique for classification, regression and different tasks that work by building a huge number of decision trees at training time and outputting the class that is the method of the classes or mean/average forecast of the individual trees. 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[31]. Scaling of data is not necessary.

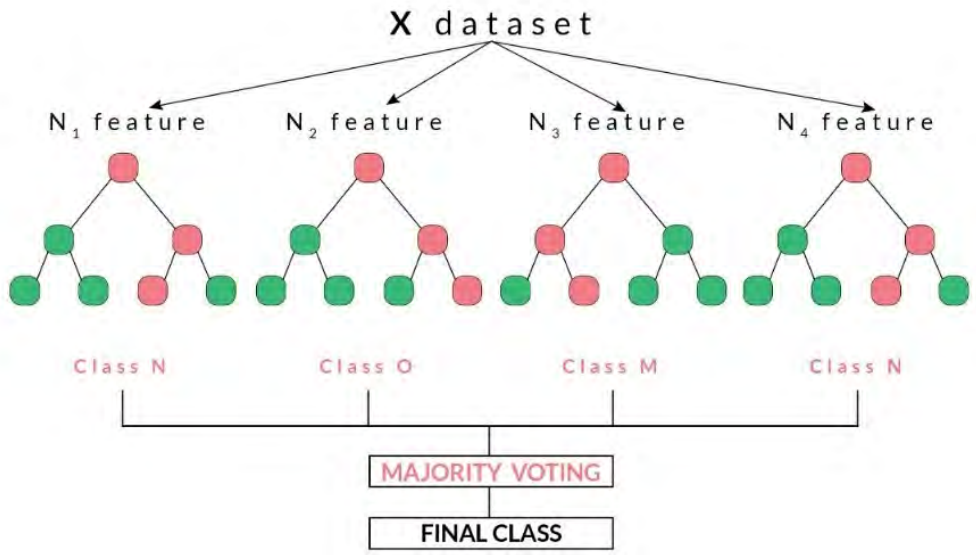


Figure 2.2: Random Forest Algorithm

Some benefits of random forest are like The Random Forest 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.

2.3.4 XGBoost

XGBoost 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. XGBoost is an enhanced appropriated boosting library intended to be exceptionally effective, adaptable and compact. We can use 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. 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 ML procedure that includes joining a few ML models to manufacture a solitary ground-breaking prescient model to receive ideal outcomes. Boosting is an learning algorithm 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.

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.

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. It is difficult to interpret, visualization technique is tough.
2. If parameters not tuned properly, overfitting is possible.

2.3.5 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.

2.3.6 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 BeautifulSoup. This BeautifulSoup 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 have

connected with a web page utilizing developer tools. BeautifulSoup exposes a 15 couple of intuitive functions which can use to explore the HTML. For web scraping with BeautifulSoup library, firstly we install the BeautifulSoup library. At that point, we consequence the library and make a BeautifulSoup object. Then we did a short python code to create a BeautifulSoup object that takes the HTML content we scraped earlier as its input.

Figure 2.3: Web Scraping Method

2.3.7 Convolutional Neural Network (CNN):

CNN is an integral part of neural networks which is widely popular and used algorithm for classifying and analyzing images. It is claimed by the experts that It possesses similar sort of capabilities like the human brain. The idea behind this algorithm was to build a system that can think, learn and give decisions like humans. The deep learning CNN takes input images and trained its system, process it and based on that it classifies under the declared categories.

With each convolutional layer, it needed to specify the number of filters the layer should have. These filters are actually detecting the patterns. 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.

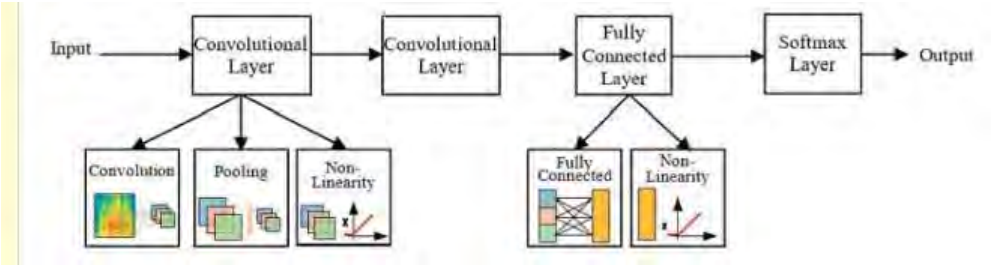


Figure 2.4: Steps of CNN

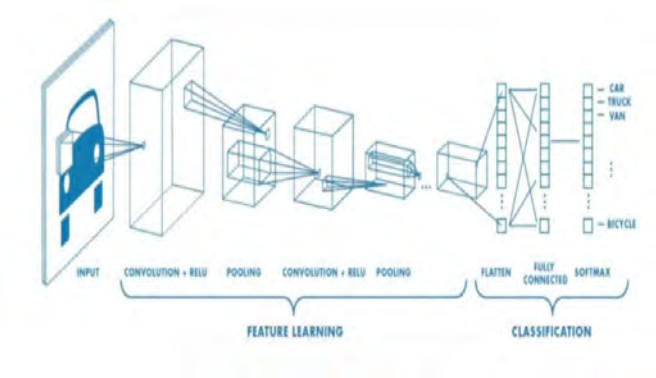


Figure 2.5: How CNN Works

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.

Chapter 3

Our Proposed Approach

3.1 Work plan

For building up our system model, firstly we had to set our work plan properly. Without proper planning our system would not work the way we expect it to do. For ensuring a better web surfing experience, we had to keep in a note about the several parameters which needed to be solved in order to provide a good user experience and from all of those we have noted down three major problems and decided to act according to it. Those are malicious site detection, link redirection, and pornographic image detection.

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

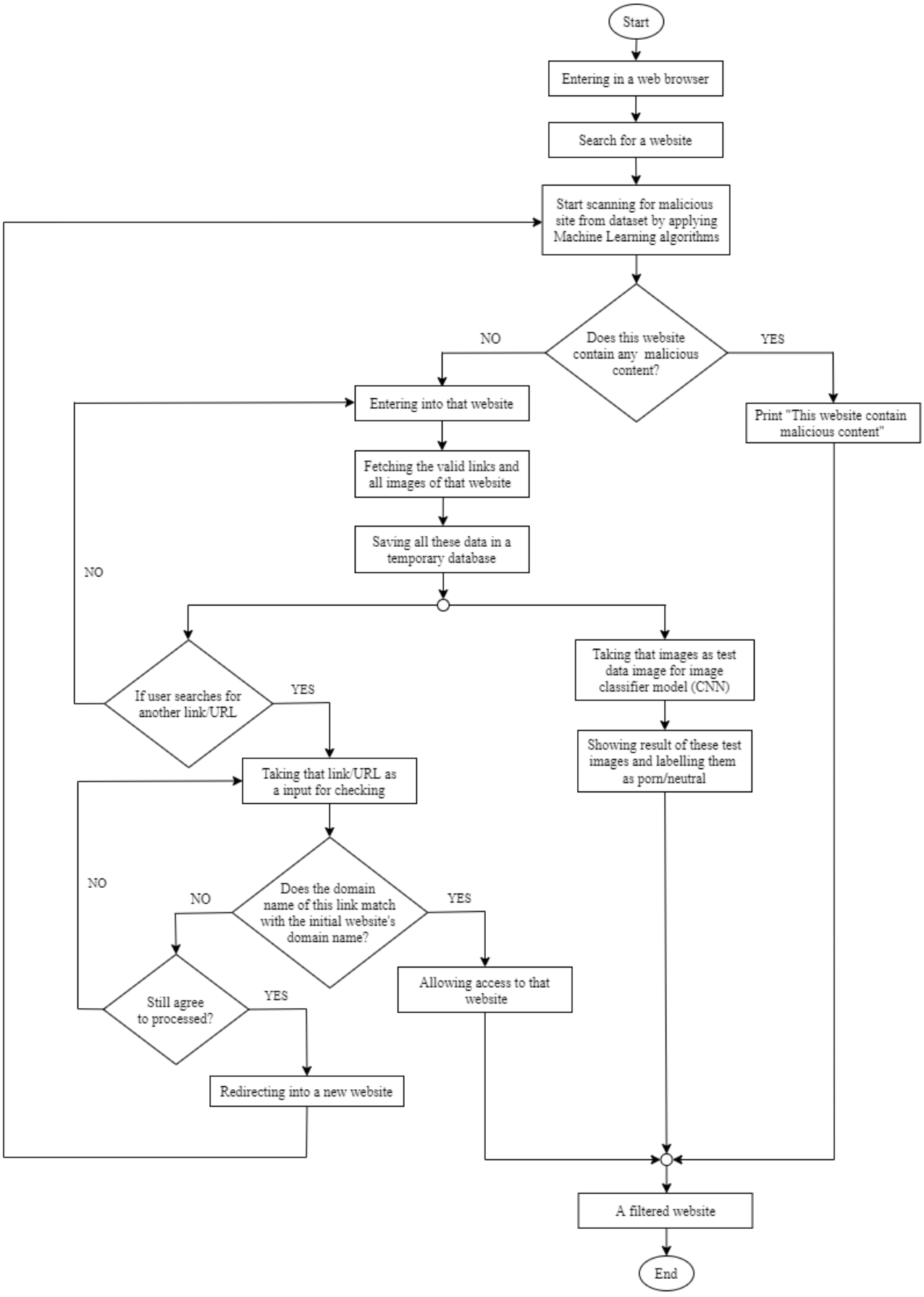


Figure 3.1: Work plan of the research

Figure 3.1 shows the workflow diagram of the complete system. The system will follow the above-mentioned steps to fulfill our requirement.

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

3.1.1 Malicious Site Detection Methodology

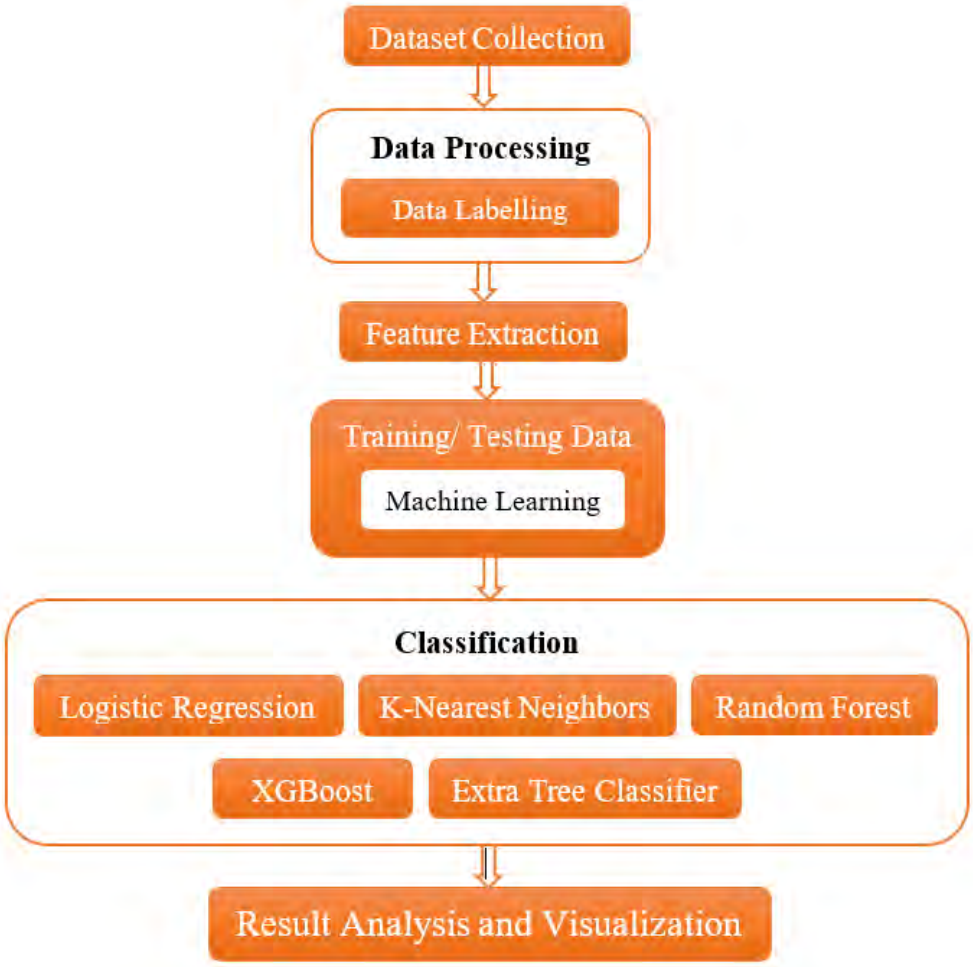


Figure 3.2: Workflow of Malicious Site Detection

Figure 3.2 is workflow of malicious site detection. Here we discuss about the data processing method and feature extraction for our research. The work process of our methodology is offered above to recognize dark malicious content by machine learning.

Dataset Description

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

1	URL:	it is the ID of the URL researched in the study
2	URL_LENGTH:	it is the quantity of characters in the URL
3	NUMBER_SPECIAL_CHARACTERS:	Special characters that we get in the URL, such as, "/", "%", "#", "&", "-", "="
4	CHARSET:	It is a categorical value and its significance is the character-encoding standard (also called character set).
5	SERVER:	It is the operative system of the server got from the packet response.
6	CONTENT_LENGTH:	Content size of the HTTP header.
7	WHOIS_COUNTRY:	This is also a categorical variable, its values are the countries we got from the server response (specifically, our script used the API of Who is).
8	WHOIS_STATEPRO:	These values are the states we got from the server response (specifically, our script used the API of Who is).
9	WHOIS_REGDATE:	Whois_regdate provides the server registration date, so, this variable has date values with format DD/MM/YYY HH:MM
10	WHOIS_UPDATED_DATE:	Through the Whois_updated_date we got the last update date from the server
11	TCP_CONVERSATION_EXCHANGE:	This variable is the number of TCP packets exchanged between the server and our honeypot client
12	DIST_REMOTE_TCP_PORT:	it is the quantity of the ports detected and different to TCP
13	REMOTE_IPS	this variable has the total number of IPs connected to the honeypot
14	APP_BYTES:	number of bytes transferred
15	SOURCE_APP_PACKETS:	packets sent from the honeypot to the server
16	REMOTE_APP_PACKETS:	packets received from the server
17	APP_PACKETS:	the total number of IP packets generated during the communication between the honeypot and the server
18	DNS_QUERY_TIMES:	the number of DNS packets generated during the communication between the honeypot and the server
19	TYPE:	categorical variable, its values represent the type of web page analyzed, specifically, 1 is for malicious websites and 0 is for benign websites

Figure 3.3: Dataset Description

Data Processing

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

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. Featured extraction is a cycle that is basically done to get the most important sources of info and work with them without unnecessary problem. This recoveries from chopping down the time span and furthermore the entanglements that might have emerged from the not all that significant data.

Training/Testing of Machine Learning Classifier

After completing feature selection method, the subsequent stage is testing and training of Machine Learning classifiers. We divided our dataset into training and testing sets. From that point forward, we test and train the dataset in some classification algorithm. Preparing information are utilized to fit and tune our model. Here, test information is spoken to as inconspicuous information to assess the models [33]. For our approach we have used five classification algorithms.

Here are the five classifiers:

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

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

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.

3.1.2 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.

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, BeautifulSoup, 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:** BeautifulSoup is a Python library for getting data out of HTML, XML, and other markup languages. There are a few site pages that show information pertinent to exploration, for example, date or address data, yet that don't give any method of downloading the information straightforwardly. BeautifulSoup 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.

Working Methodology of Scanning Link Directory

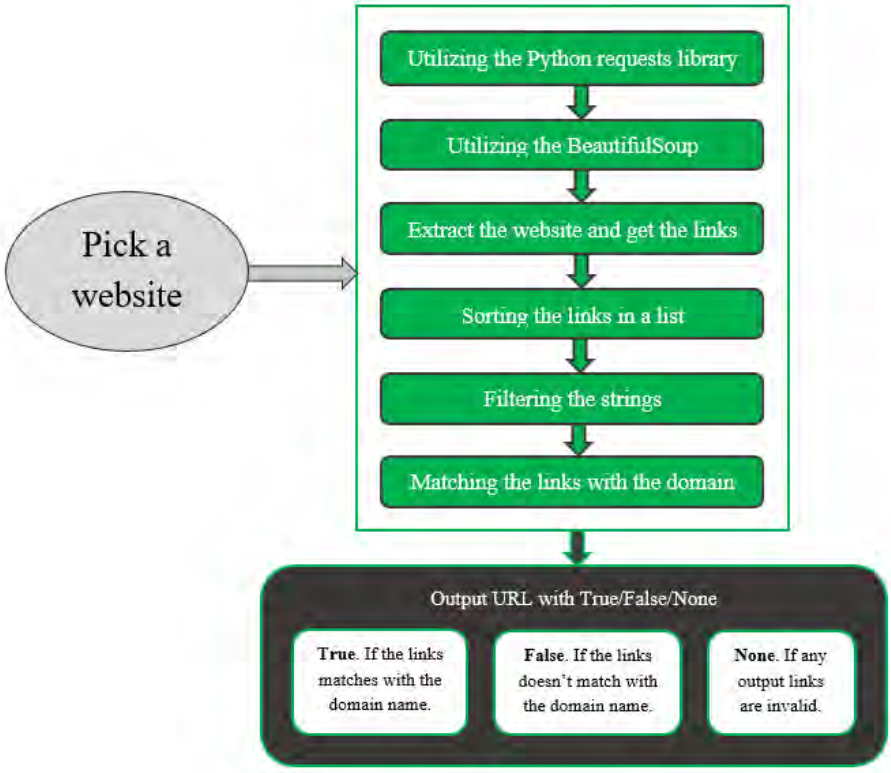


Figure 3.4: Workflow of Link Redirection Methodology

Figure 3.4 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.

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.

3.1.3 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 detection part, we have

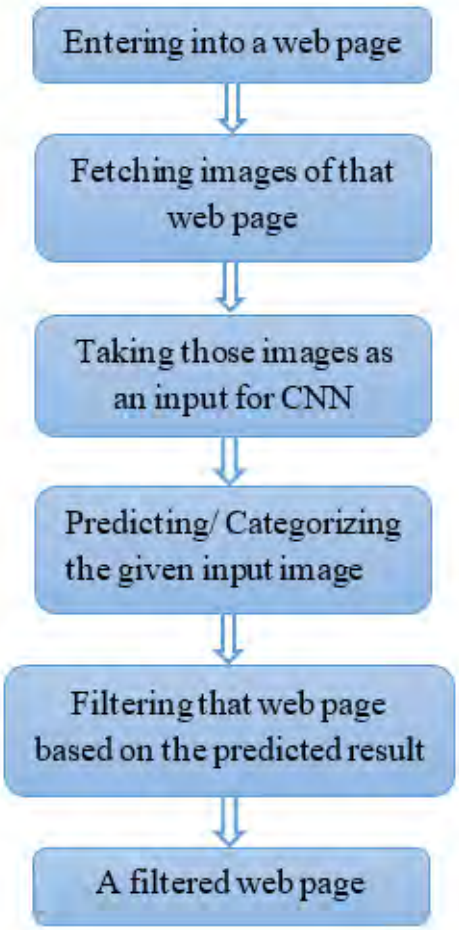


Figure 3.5: Workflow of Nudity Detection

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.

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.

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.

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.

Chapter 4

Experimentations

4.1 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))
```

Figure 4.1: 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

```
#__________KNN
from sklearn.neighbors import KNeighborsClassifier

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

pred3=classifier2.predict(x_test)

#______calculate accuracy for KNN

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

Figure 4.2: K-Nearest Neighbor

Random Forest

We know random forest uses trees. Here Random forest makes 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))
```

Figure 4.3: 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
```

Figure 4.4: 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.

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

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

Figure 4.5: Extra Tree Classifier

4.2 Link Redirection Methodology

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

Figure 4.6: 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()
```

Figure 4.7: 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('.')
```

Figure 4.8: Filtering the String

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

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

Figure 4.9: 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))
```

Figure 4.10: Hypertext Referencer

4.3 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
```

Figure 4.11: 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
```

Figure 4.12: 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.

```
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'])
```

Figure 4.13: Adding layers in the model

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 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)
```

Figure 4.14: Setting Epoch number

Chapter 5

Result analysis and Data Visualization

5.1 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.

5.1.1 Data Visualization

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

Heatmap

In the dataset there are all out 37,000 examples. So to detect the malicious website, we have utilized heatmap device which is a graphical portrayal of data where the individual values included in a matrix are represented as colors. It is a somewhat similar looking a data table from above. It is truly valuable to show a general view of numerical data. The seaborn python package permits the formation of annotated heatmaps which can be changed using Matplotlib tools according to the creator's requirement. In python libraries, there are a bunch of techniques and approaches to visually represent data, however we focused on the utilization of heatmaps.

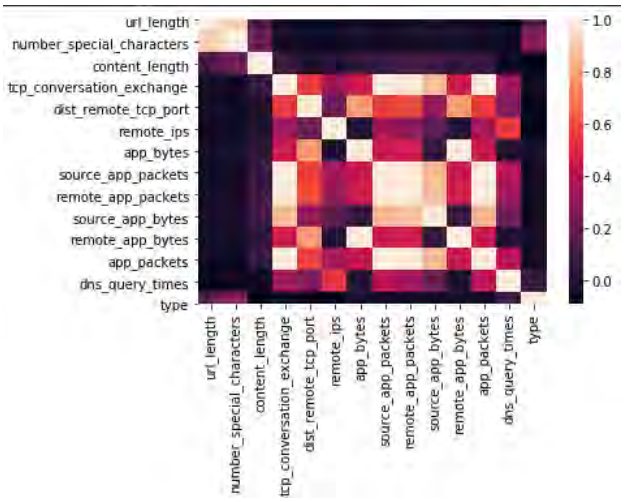


Figure 5.1: Heatmap of input data

Figure 5.1 shows the co-connection between input information as heatmap. From the heatmap, we found that "URL_LENGTH", "NUMBER_SPECIAL_CHARACTERS", "SOURCE_APP_PACKETS" etc. were generally significant as the scale demonstrated that the color was near to 1.0 which spoken to high significance of the component is the matrix and shows the co-relation of input data.

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.

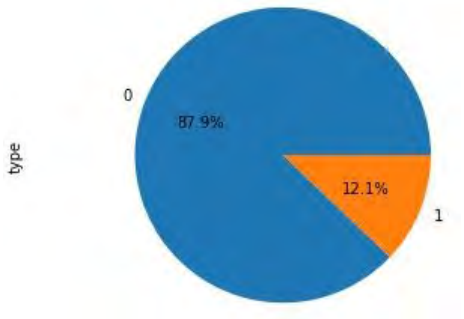


Figure 5.2: Pie Plot

5.1.2 Result analysis

To assess our model, Sensitivity, Specificity, accuracy score metrics are utilized. We likewise assess the ROC curve, AUC, confusion matrix, accuracy score, to choose the best algorithm for malicious web page identification.
We utilized five classifiers to detect malicious site, these five algorithms are K-NN,Logistic Regression, Random Forest, XGBoost and Extra Tree Classifier. We likewise assessed the presentation of the classifiers with the assistance of various metrics which is given bellow.

- Sensitivity: Sensitivity shows the capacity of the classifiers to discover malicious webpage that are really of malicious category.

$$Sensitivity = TruePositive/(TruePositive + FalseNegative) \tag{5.1}$$

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

$$Specificity = TrueNegative/(TrueNegative + FalsePositive) \tag{5.2}$$

Confusion Matrix

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

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

Table 5.1: Table of Confusion Matrix

Here,T.N= True Negative where the algorithm predicted negative and the output is also negative, F.P= False Positive where the algorithm predicted positive and the output is negative, F.N= False Negative where the algorithm predicted negative and the output is also positive, T.P= True Positive where the algorithm predicted positive and the output is also positive.
True Positive Rate (T.P.R): It describes how great the model is at predicting the positive class when thee real result is positive.

$$T : P : R = TruePositive = (TruePositive + FalseNegative) \tag{5.3}$$

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

$$F : P : R = FalsePositive = (FalsePositive + FalseNegative) \tag{5.4}$$

ROC Curve

ROC curve is a graphical plot that represents the diagnostic capacity of a binary classifier framework as its discrimination threshold is changed. This curve shows the connection between clinical sensitivity and specificity for each necessary cut-off. ROC curve plots TPR versus FPR at various classification thresholds. It is a plot of the false positive rate versus the true positive rate for a number of different candidate threshold values somewhere in the range of 0 and 1.

Bringing down the classification threshold classifies more things as positive, hence increase both False Positives and True Positives.

Accuracy Score

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

$$AccuracyScore = \frac{T.P + T.N}{T.P + T.N + F.P + F.N} \tag{5.5}$$

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 5.2: Comparison of values

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

Comparison With Other Papers

Here we compare our system with existing related papers.

Firstly, in paper 1 project name "Application of Hybrid Machine Learning to Detect and Remove Malware", by R. R. Yang, V. Kang, S. Albouq, and M. A. Zohdy, they have use algorithms like Random forest, Naïve Bayes, J48. With all the classifiers they were able to create a hybrid Machine learning algorithm. Among them Random forest gave them the best accuracy and the accuracy was 94% but we were able to achieve 96% and also our best classifier was Random forest.[34]

In second paper similar work for malicious paper name "Empirical Study on Malicious URLDetection Using Machine Learning" Ripon Patgiri(B), Hemanth Katari(B), Ronit Kumar(B), and Dheeraj Sharma(B)National Institute of Technology Silchar, Silchar 788010, Assam, India they use classifiers like Random forest, SVM. They got the best score from random forest and where able to achieve 92% Accuracy which is 4% less than we have.[35]

Lastly, we have compared our project with "Exploring Malware Behavior of Webpages Using Machine Learning Technique: An Empirical Study Alhanoof Faiz Alwaghid and Nurul I. Sarkar". They used Decision Stump, Hoeffding Tree, J48, RF, Random Tree and REPTree. They got the best out from random tree which is 88% which is less than us and show that our work is comparatively good.[36]

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

Table 5.3: Comparison with other Papers

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

Logistic Regression

For Logistic Regression we get an accuracy score of 0.887.

The confusion matrix for Logistic Regression is as follows:

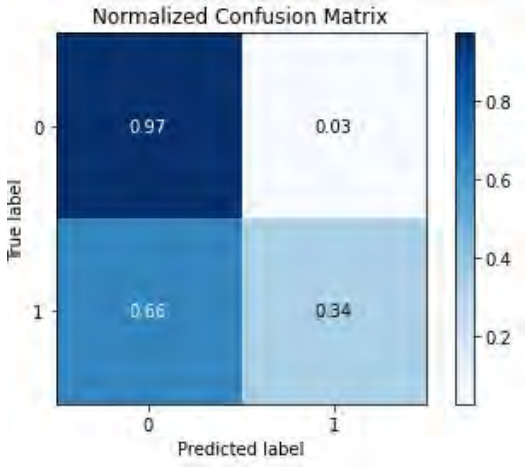


Figure 5.3: Confusion Matrix of Logistic Regression

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:

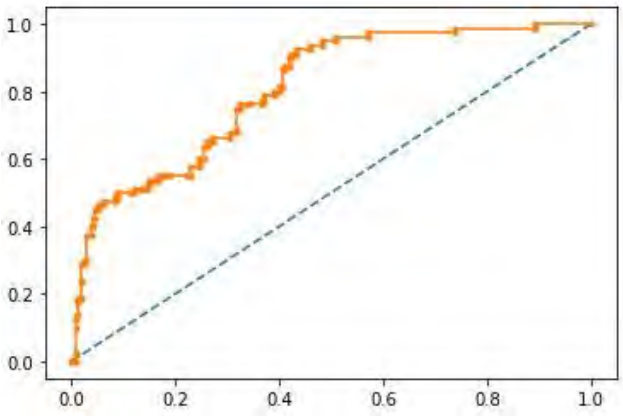


Figure 5.4: ROC Curve of Logistic Regression

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

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

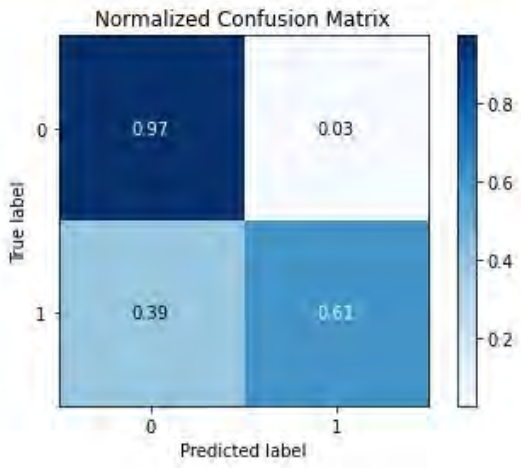


Figure 5.5: Confusion Matrix of K-Nearest Neighbor

matrix, the true negative value is 0.97, the false positive value is 0.03, the false-negative value is 0.39 and the true positive value is 0.61.

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

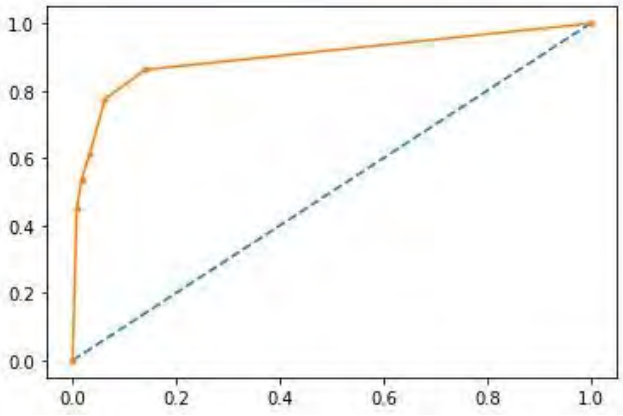


Figure 5.6: ROC curve of K-Nearest Neighbor

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

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 ma-

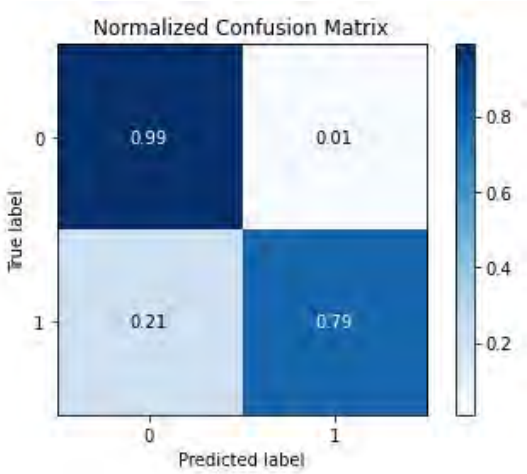


Figure 5.7: Confusion Matrix of Random Forest

trix, the true negative value is 0.99, the false positive value is 0.01, the false-negative value is 0.21 and the true positive value is 0.79.

The ROC curve for Random Forest is as follows:

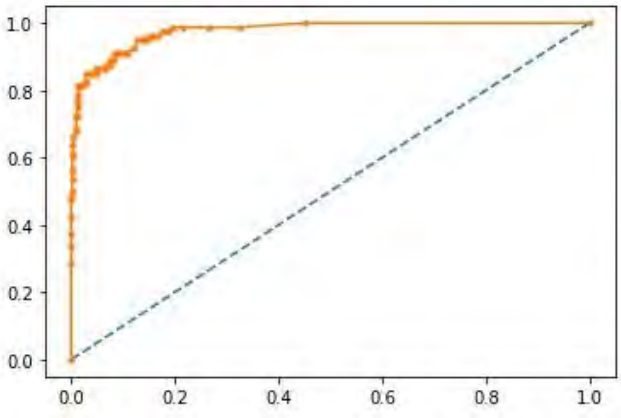


Figure 5.8: ROC curve of Random Forest

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

XGBoost

For XGBoost we get an accuracy score of 0.955.

The confusion matrix for XGBoost is as follows:

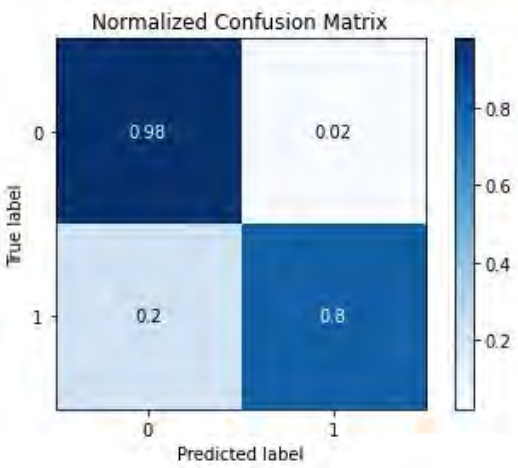


Figure 5.9: Confusion Matrix of XGBoost

Here in the confusion matrix, the true negative value is 0.98, the false positive value is 0.02, the false-negative value is 0.2 and the true positive value is 0.8.

The ROC curve for XGBoost is as follows:

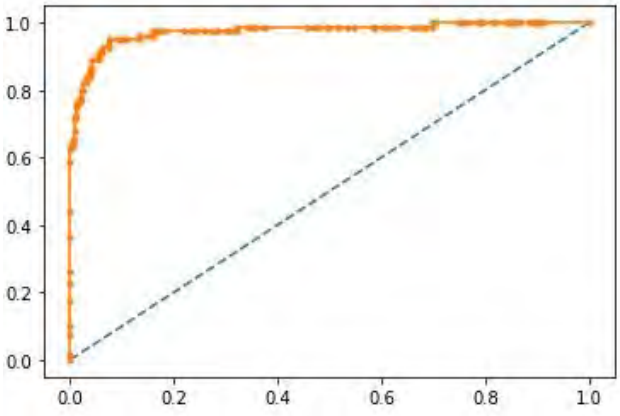


Figure 5.10: ROC curve of XGBoost

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

Extra Tree Classifier

For Extra Tree Classifier we get an accuracy score of 0.912.

The confusion matrix for Extra Tree Classifier is as follows:

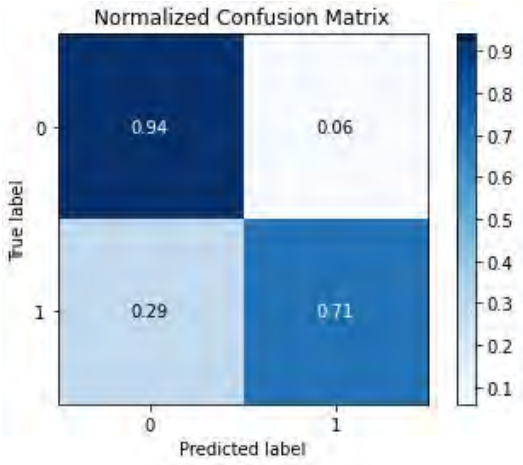


Figure 5.11: Confusion Matrix of Extra Tree Classifier

Here in the confusion matrix, the true negative value is 0.94, the false positive value is 0.06, the false-negative value is 0.29 and the true positive value is 0.71.

The ROC curve for Extra Tree Classifier is as follows:

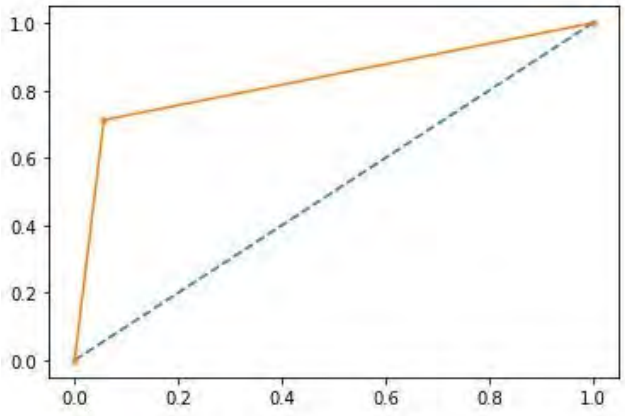


Figure 5.12: ROC curve of Extra Tree Classifier

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

Here, AUC for logistic regression is 0.809 and accuracy is 88.7%. AUC for K-NN 0.90 and accuracy is 95.96%. AUC for Random Forest is 0.977 and accuracy is 96.12%. AUC for XGBoost is 0.975 and accuracy is 95.5%. AUC for Extra Tree Classifier is 0.828 and accuracy is 92.7%. If 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. Random forest has the highest AUC value which is 0.977.

5.2 Link Redirection Methodology

5.2.1 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 5.4: Output of the given input

Table 5.3 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 30 websites, where there were websites of different domain types. We are very lucky that we got our expected result.

Here, "YES" refers to websites that are applicable to our system and "NO" means that these websites are not applicable to our system because of different domain types.

The list of that 30 websites is given below:

	Checked Websites list	YES/NO
1	https://www.atlhca.org	YES
2	https://www.elpais.com	YES
3	https://adobe.com	YES
4	https://www.amazon.co.uk	NO
5	https://wikimedia.org	YES
6	https://netflix.com	YES
7	https://mozilla.org	YES
8	https://nih.gov	YES
9	https://themeforest.net	YES
10	https://microsoft.com	YES
11	https://brazzers.com	YES
12	https://www.bangladesh.gov.bd	NO
13	https://cdc.gov	YES
14	https://github.com	YES
15	http://www.bracu.ac.bd	NO
16	https://archive.org	YES
17	https://aliexpress.com	YES
18	https://cpanel.net	YES
19	https://www.wix.com	YES
20	https://www.calvinklein.us	NO
21	https://mozilla.com	YES
22	https://www. abc.net.au	NO
23	https://www.wikipedia.org	YES
24	https://soundcloud.com	YES
25	https://teamviewer.com	YES
26	https://dribbble.com	YES
27	https://state.gov	YES
28	https://ndtv.com	YES
29	https://www. nhk.or.jp	NO
30	https://www.gucci.com	YES

Figure 5.13: Checked Website for Our System

5.3 Nudity Detection

5.3.1 Data Visualization

To make the output understandable, it is necessary to use different kinds of data visualization techniques. Python has a lot of built in libraries by which we can visualize the output data. By applying data visualization technique, we get a curve model of "Model Accuracy" and "Model Loss" for our image classifier model. Our system also provides a prediction for given input test image.

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

Model Accuracy

Here, it is our Model Accuracy curve

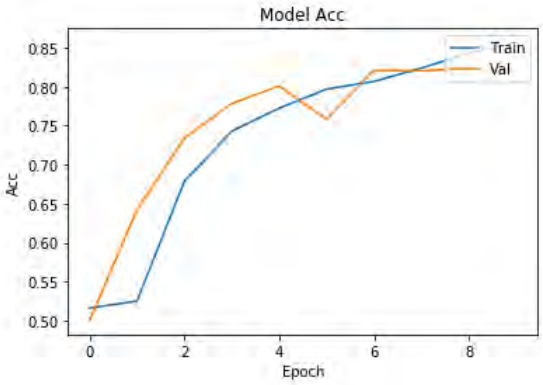


Figure 5.14: Model Accuracy for Our System

Model Loss

Here, it is our Model Loss curve

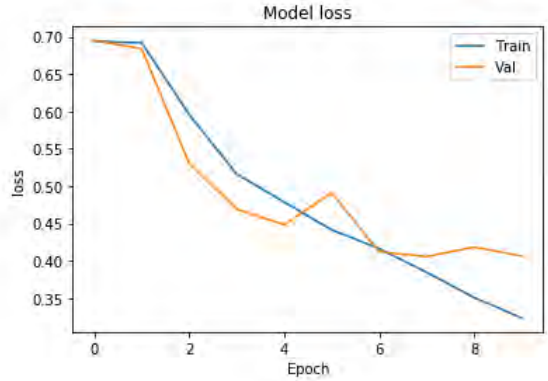
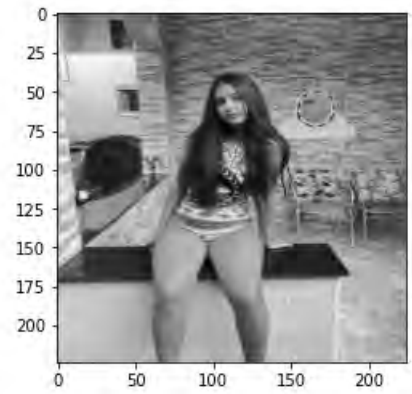


Figure 5.15: Model Loss for Our System

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)
```

Figure 5.16: 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()
```

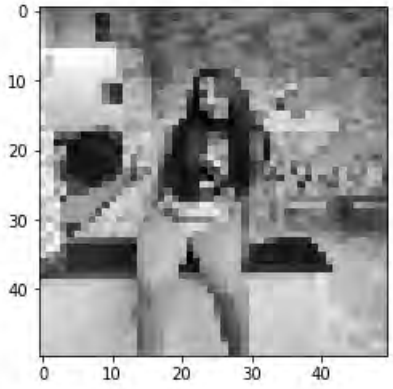


Figure 5.17: Converting Image into GreyScaleImage

```
Epoch 1/10
800/800 [==============================] - 170s 213ms/step - loss: 0.6939 - acc: 0.5159 - val_loss: 0.6950 - val_acc: 0.5000
Epoch 2/10
800/800 [==============================] - 167s 209ms/step - loss: 0.6918 - acc: 0.5250 - val_loss: 0.6838 - val_acc: 0.6420
Epoch 3/10
800/800 [==============================] - 169s 211ms/step - loss: 0.5957 - acc: 0.6787 - val_loss: 0.5303 - val_acc: 0.7340
Epoch 4/10
800/800 [==============================] - 170s 212ms/step - loss: 0.5163 - acc: 0.7430 - val_loss: 0.4692 - val_acc: 0.7785
Epoch 5/10
800/800 [==============================] - 169s 212ms/step - loss: 0.4786 - acc: 0.7725 - val_loss: 0.4483 - val_acc: 0.8010
Epoch 6/10
800/800 [==============================] - 168s 210ms/step - loss: 0.4416 - acc: 0.7969 - val_loss: 0.4911 - val_acc: 0.7580
Epoch 7/10
800/800 [==============================] - 169s 211ms/step - loss: 0.4163 - acc: 0.8067 - val_loss: 0.4120 - val_acc: 0.8210
Epoch 8/10
800/800 [==============================] - 169s 211ms/step - loss: 0.3846 - acc: 0.8239 - val_loss: 0.4058 - val_acc: 0.8205
Epoch 9/10
800/800 [==============================] - 170s 213ms/step - loss: 0.3508 - acc: 0.8431 - val_loss: 0.4184 - val_acc: 0.8230
Epoch 10/10
800/800 [==============================] - 168s 210ms/step - loss: 0.3236 - acc: 0.8569 - val_loss: 0.4063 - val_acc: 0.8205
```

Figure 5.18: 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.

Figure 5.19: Taking a Random Image as an Input

```
In [75]: resize_image = plt.imshow(new_image)
```

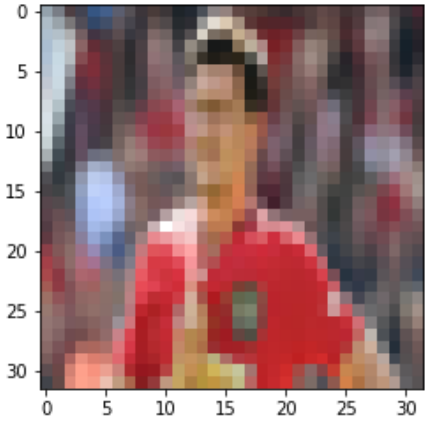


Figure 5.20: Resizing the Image

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.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

The main motto 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 prepare it and classify it under certain categories. So, we decided to implement Convolutional Neural Network for image classification. Fortunately, we also get a good accuracy using CNN to detect indecent 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.

6.2 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.

Bibliography

[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] 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)

[10] 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.

[11] 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.

[12] 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.

[13] 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).

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

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] 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).

[22] 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)

[23] 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.

[24] 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)

[25] 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).

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

[27] 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

[28] 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.

[29] 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.

[30] 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.

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

[32] R. R. Yang, V. Kang, S. Albouq, and M. A. Zohdy, "Application of hybrid machine learning to detect and remove malware",presented at Transactions on Machine Learning and Artificial Intelligence, vol. 3, no. 4, pp. 16–16, 2015.

[33] Ripon Patgiri(B), Hemanth Katari(B), Ronit Kumar(B),and Dheeraj Sharma(B), "Empirical Study on Malicious URLDetection Using Machine Learning" National Institute of Technology Silchar, Silchar 788010, Assam, India

[34] Alhanoof Faiz Alwaghid and Nurul I. Sarkar, "Exploring Malware Behavior of Webpages Using Machine Learning Technique: An Empirical Study"