National College of Ireland

Detection of Clickjacking Attacks using the Extreme Learning Machine algorithm

MSc Internship
MSc Cyber Security 2019-2020

Yashodha Patil
Student ID: x19102437

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Miss. Yashodha Subhash Patil……

Student ID: x19102437…

Programme: MSc Cyber Security… **Year:** 2019-2020….

Module: MSc Internship ……

Supervisor: Prof. Vikas Sahni……

Submission Due Date: 17/08/2020……

Project Title: Detection of Clickjacking Attacks using the Extreme Learning Machine Algorithm……

Word Count: 7458 **Page Count 17**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Clickjacking Attacks using the Extreme Learning Machine algorithm

Yashodha Patil
x19102437

Abstract

Clickjacking attack is one of the emerging web-based attack. In clickjacking, the user is tricked to click on transparent iframes placed over the web elements. This may lead to unwanted operations without the user's knowledge. Even though, clickjacking is one of the major points of discussion, it is still unclear, how and to what extent the attacker might use this attack to lure the user and gain the user data. Therefore, in this paper we have proposed a solution that identifies malicious links which are being used for clickjacking attacks. In this, Extreme Machine Learning (ELM) technique is used, that classifies the malicious links present on the webpage and these are displayed on the webpage using the HTML CSS property. Only one type of malicious link is identified i.e. phishing links present on the webpage. Hence, phishing dataset is used. The Extreme Learning Machine model was compared with the support vector machine learning model with respect to their performance metrics. The training time required by the ELM model is less compared to SVM model.

1 Introduction

Web application is created with multiple static HTML pages and dynamic HTML pages to make the website more user friendly and interactive. Many web applications are developed by merging the multiple site content provided by different sources [1]. The increased number of web application has increased the number of the web-based attacks, one of which is Clickjacking. Clickjacking was discovered by Jeremiah Grossman and Robert Hansen in year 2008 [19]. The clickjacking attack is implemented through the iframes on the webpage by loading the malicious script in the transparent iframes which is placed on top of the legitimate webpage or on the web elements. Using the CSS (Cascading Style Sheet) property, the iframe opacity level is set to very low to make the iframe transparent that it can be barely visible [20]. The user is tricked in a way to click on the malicious frame that causes an unintentional activity without his/her knowledge. The attacker carries out such types of attacks with an intention of money transfer, blog posting, forum message, redirection to the fake social networking sites to gain credentials and many more malicious activities which can be triggered by mouse click [1].

The fig:1 shows an example of a clickjacking attack. The Win big frame is visible to the user and the fund transfer page is hidden and placed in such a way that the confirm button overlaps the click to win button. When user clicks on this "Click to win" button the user redirects to a page for the money transfer. This is happening because the user has actually clicked on the "confirm" button. This Confirm fund transfer frame is made so transparent that no one can notice.

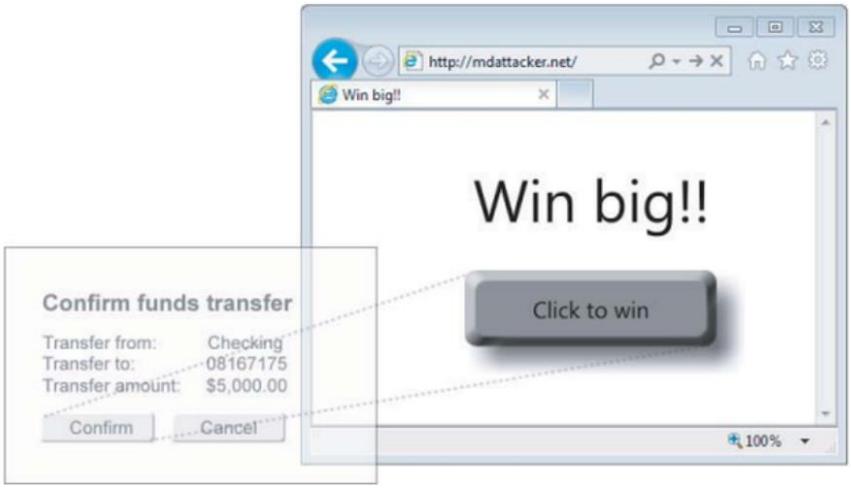

Figure 1 : Transparent Iframe used for Clickjacking attack. [1]

The attacker intentions and to what extent the attacker might use the clickjacking to perform the attacks is completely unknown. Some of the clickjacking attack that took place at large scale in real time are, a botnet named ”Chameleon” was designed to click on the advertisement posted on the website, fooling the advertisement owner by showing the interest on the product, but in real the attacker is getting paid based on the number of advertisement hits [21]. Another real time example of clickjacking was that it was used to spread the twitter message across the Twitter network [1].

There are different types of clickjacking like, the attack takes place through the iframe by taking the advantage of social engineering. Another type of attack is done by clicking on link, search action takes place and parallelly the query is passed to the attacker. The email messages which supports HTML are also used to spread the clickjacking by placing the malicious hyperlinks over the legitimate ones [12].

The proposed model can be used for various websites where machine learning techniques can be used to identify the malicious link present on the webpage and using CSS property those links which are used for the clickjacking are detected and then displayed on the webpage. This model verifies the URLs based on the real time web-based features including Domain Name, SSL, web traffic and web hosting provider.

The aim of the research is to detect the malicious links that an attacker has placed on the webpage with an intention such as to grab the user credentials, to download the malware on the victim’s machine that are available on the webpage. The extreme learning machine technique is used for the classification of the links. Furthermore, the list of malicious URLs is created and the links whose opacity is set at very low and whose margin property is fixed at the position are displayed on the webpage.

[1] https://prashantshakti.wixsite.com/security/post/clickjacking-attack-and-prevention

The Research Question proposed for this research is “How emerging machine learning techniques can be utilised to detect the most common web-based clickjacking attacks?”

The Extreme Machine Learning (ELM) technique is used in predicting the malicious URLs. The ELM is feedforward neural network and it is fast and efficient in binary classification. The ELM classifier is trained on the phishing dataset which contains vectors that are defined for the malicious activity and produces the output in less time. The prediction result concluded that the ELM classifier performed well with respect to time taken by model for learning compared to supervised machine learning model.

Various tool, techniques, models and algorithms from non-machine learning technique to the machine learning technique are researched, studied and compared in the literature review section to choose the technique that satisfy our Research question.

This report is organised in various sections. Section 2 is the Related work in which various technologies are researched and studied from non-machine to machine learning techniques. In section 3, the Research Methodology of our research is described in detail, the system architecture is explained in the section 4, the implementation and evaluation are described in section 5 and 6 respectively and finally in section 7 we conclude our research.

2 Related Work

This section describes about different approaches that have been carried out earlier in the field of clickjacking attacks. As discussed in section 1, many previous researches are studied to gain knowledge of the work done in this field and also to know what challenges are still open in the same field. Through detailed research, different tools and techniques are studied from non-machine learning techniques to the machine learning techniques.

The author in the paper [1] has proposed the automated solution for the detection of clickjacking attacks. In this approach, coordinates of all clickable elements are collected from the rendered webpage on a real browser. For the detection, two browser plugins are developed, “ClickIDS” and “NoScript”. The ClickIDS analyses the web elements over the webpage and detect the clickjacking attack and searches for any web elements overlapped on the same click coordinates in a particular region or position on the webpage. NoScript is a Mozilla browser plugin that generates an alert when web elements are overlapped which is detected by the mouse click and then generates an alert for the web elements and is stored in the SQL database. Millions of webpages are analyzed for the evaluation. This method performs well on the detection of the clickjacking attacks on clickable events; however, it turns to be a limitation as this technique cannot detect the attack over the non-clickable elements. Also, the NoScript is browser dependent plugin.

Using the position of the clickable coordinates on the webpage another research is done by the author K. Joylin Bala et al. [2]. The 3 components (Tracking agent, Detection agent, and action agent) are used for detecting the clickjacking attack. The tracking agent counts and saves all the clickable web elements coordinates on the hard disk before the page gets load. Once the page is loaded this saved positional clickable element are compared with the loaded page through the detection agent and if there is any mismatch found in the count,

then an alert is triggered by the action agent. Using this method, the clickjacking attack can be detected effectively by comparing the clickable elements, but the position of mismatched clickable elements is not displayed on the page so it can increase the work to find this element. Also, the computational time may also increase for comparison with the increase in the web elements.

Krishna Chaitanya T et al. [3] has created an extension which is similar to the NoScript Mozilla web browser extension [1] for google chrome to detect the spam attacks which are carried out using clickjacking attack. The same origin check rule is added to this extension. This feature checks for the transparent Iframe origin on the webpage. Using Java/CSS same origin chrome extension is created. This extension is placed on the attack page to check how effectively this extension works on real time. This extension detects the transparent Iframe present on the webpage but this approach has no full functionality as that of the NoScript extension [1], it is just a mimic of the NoScript extension, thus it may not provide full functionality for the detection of different clickjacking attacks.

The application which mounts the clickjacking attack on the web application whose same origin policy is set, the attacker cannot directly render the web page content. To perform an attack, the attacker has to assume the layout of the webpage. To detect such attacks on the webpage the author [4] has proposed a technique which describes the back-end statistical analysis of the page. The author has proposed two processes, 1st a button is placed on the website in a random manner where there are no other web elements present and then all the missed clicks and actual clicks are recorded. Another refinement process is the BUCKETISATION, in which all the user clicks are recorded on a single click and saved in the form of groups or buckets to perform the statistical analysis. For the social networking sites the buckets might act as an identity of an account for Follow and Like buttons. The application checks if any clicks are missed based on the bucket then it can state the clickjacking attack is taken place. Though this technique shows effective result on the detection of the clickjacking attack, it is not suitable for the complex applications.

A browser-based solution is developed by the author Ubaid Ur Rehman et al. [5] to detect clickjacking attacks performed using Facebook widgets present in the webpage. The two google chrome extension are created, the Zscaler Likejacking prevention detects all the hidden web widgets available on the webpage and the Cursor Spoofing and Clickjacking Prevention (CSCP) extension popups a message dialog box, when the user clicks on the like and the follow button on the webpage. The author has carried out 10 days experiment on 10 PC's and compared the attacks success rate of the user who are clickjacked before using the extension with the success rate of post deployment of this mechanism. From the analysis, it is observed that the CSCP has well performed as the success rate of attack is reduced from 76.7% to 10%. This method is only applicable to the social networking sites such as Facebook.

Marco Cova et al. [6] performs the dynamic analysis of the Javascript code of the webpage to detect the drive-by-download attack. The Drive-by-download attack is a web application attack. The approach is to detect and analyze the Javascript code. A JSAND tool is developed that identifies the Javascript code for maliciousness based on the features whose values are evaluated using the anomaly detection techniques of the legitimate websites and on the machine learning techniques. The accuracy is calculated based on over 140,000

webpages. This work is not relevant in detecting clickjacking attacks, but using in-depth dynamic analysis of the webpages, many clickjacking attempts can be detected [8].

In the paper [7], the author has developed a tool known as “Prophiler” to analyse the web page at a larger scale. The web crawler is used to collect the HTML pages and by using the feature extraction technique, the features based on web page content and URL based features are extracted. To extract the web page content feature, the parser is used that extracts the feature from the HTML and from the JavaScript code. The supervised machine learning technique is used to classify the webpage into benign or malicious. After classification, webpage is transferred to the dynamic analysis tool that decides if the obtained result is correct or is false indication. The approach is very inspirational but it is time consuming as the result is checked twice.

In the paper, Analysis and Detection of Clickjacking on Facebook [8], clickjacking is detected on the social networking site “Facebook”. Recent comments made on the webpage are collected and stored in the file and URLs are extracted from this file. The features from this URL are extracted and converted into the feature vector to create the dataset matrix and fed into the supervised machine learning algorithm to classify the links as malicious, clickjacking and the benign. Out of 8360 data points 118 are defined as the clickjacking attack and 71 of which were correctly predicted and remaining were false positives.

This approach is very effective for detecting clickjacking attacks on the Facebook, this approach motivated us to detect the clickjacking attack through the machine learning algorithm.

Youngsang Shin et el [9] classifies the URL as a spam or the legitimate one. The URL is retrieved from the comments available on the blog. The SVM classifier classifies the URL based on the features extracted. This method is limited for the extraction of links from the blog comments.

The author [10] in this paper has described the technique to detect the malicious URL using the machine learning technique. The author suggested that the supervised, unsupervised or the semi supervised machine learning algorithm can be used for the classification. The malicious URL is predicted based on the feature extracted and converted into vector space to give input to the machine learning algorithm. The model is trained using the blacklisted URL dataset and depending on this learning, it predicts the URL as benign or malicious. The supervised machine learning algorithm is used for evaluation. Four different categories of features are extracted from the URL for the classification which is effective.

A survey was conducted by Dr. Jitendra Agrawal et. al [11] for the detection of the malicious webpage using classification techniques. A various supervised machine learning algorithm were used for the classification of the links into the different attacks. For the large size of the web, this technique is challenging because for the feature extraction process various techniques and tools are used.

Sha_ Ahmed [12] conducted real time detection of the malicious webpage using supervised and unsupervised machine learning algorithm. The data is gathered from the phishing tank database and from Alexa. The features are extracted for all the URLs obtained from both the source and converted into the vector space representation and then URLs are classified into phishing and legitimate ones. Among all the model, the SVM performed very well comparatively.

Anjali B. Sayamber [13] collected the data such as benign, spam url and phishing urls from various source. The accuracy of this model is compared with SVM and Naïve Bayes. The Naïve Bayes performed well for detecting and classifying the URLs into different category such as spam, benign and phishing. Comparing this paper with the paper [12], the data used in [12] were only taken from the phishing tank which contain only the phishing data while in paper [13], malicious links are collected from the jwspamspy, phishing tank and DNS-BH. Thus, the performance of the models in the paper [12] [13] shows different result based on the different combination of dataset are used.

In the previous papers [11] [12] [13], the URL is collected from various database where they are marked as blacklisted and good URL's are also collected from the classification. The author [14] has proposed a solution to extract web content from the HTML webpage. The HTML page is converted in the Document Object model tree and the features are extracted and the rule is generated to extract the informative content from the HTML page.

The author in the paper [15] has described about various web scrapping technique to extract the content of the HTML webpage by converting the unstructured data into structured data of the web and suggested some of the tools and libraries to scrape the data.

In the paper [16], the researcher has collected the phishing URL and the benign URL from the phishing tank database and DMZ database. The python libraries are used to extract the feature of the URL. Classification of the URL is performed using multiple supervised machine learning algorithm and with the MATLAB Neural Network. The accuracy of both the supervised machine learning and the neural network is compared. The decision tree, supervised machine learning algorithm achieves accuracy of 96.18% which is the better than other supervised learning algorithms. Almost 98.16%, accuracy is achieved from the MATLAB neural network. The MATLAB neural network has performed well comparatively.

Several machine learning model such as SVM, K-means, Neural Network, Self-organising Map model are used by the author Andrew H. Sung et. al in the paper [17] for efficient prediction of the website emails. Multiple experiment is carried out with multiple models in order to check the accuracy of the model. Almost, 97.99% of accuracy is achieved by the Neural Network model that detects the phishing emails.

Hüseyin Gökal [18] et al used the ELM classifier for the classification purpose. The classification of the extreme learning machine is a type of the Neural Network. This classifier classifies the phishing website based on 30 different extracted features. For the result, the ELM neural network is compared with SVM and the Naive Bayes Supervised machine algorithms. The ELM has shown the best performance compared to this algorithm with an accuracy rate of 95.34%. The performance of the model is calculated based on the validation value which is the ratio of data estimated correctly to all the data from the dataset.

From the research it is observed that, there are limitations and challenges in terms of browser, techniques, tools for the non-machine learning techniques. For the machine learning, the ELM classifier is a type of neural network techniques which is used for detection of malicious links has given good accuracy compared to the other supervised algorithms. Detection of the malicious links that are used for the clickjacking attacks using different machine learning algorithm is explained.

3 Research Methodology

The attacker's main attraction is to target the victims by using social engineering technique, the attackers trick the user to visit their domain using various method [8]. One of the methods used is where the user is tricked to visit the malicious websites by using double framing methods or by hiding this malicious links in the images, so when user clicks on this frames which has malicious links it redirects the user to the websites which are used by an attacker to perform the attacks such as phishing, spam, malware [13].

This section provides an insight of how the clickjacking attacks on the client side can be detected and displayed to the user over the web. The primary goal is to identify such malicious URLs available on the webpage and to make them visible to the user, using two techniques i.e. the extreme learning machine algorithm and the CSS property of the HTML. The Extreme Learning Machine (ELM) is a neural network used with a single layer feedforward neural network. [22]. The phishing dataset is used to train the model and the result is predicted based on these learnings. The detailed description of the research methodology is explained in the sub sections 3.1, 3.2, 3.3, 3.4.

3.1 URL Extraction:

Web Scrapping is a technology that is used for extracting the content from the websites. It is also known as the Web data extraction, Web harvesting and Web data scrapping. There are many options such as API (Application Interface Programming), to scrape the data, but API is not used here as many webapps do not support API[27].The web scrapping methodology is used for parsing the HTML webpages i.e. the relevant content such as URL, text, keywords, email, etc from the HTML code is extracted [15]. For our thesis, we have scraped the HTML pages of the web application first and then we have extracted only hyperlinks from the scrapped content which is further used for classification.

3.2 Feature Extraction and Feature Pre-processing:

Once we have all the hyperlinks of the HTML (obtained from web scrapping) we can then extract the various types of features of these hyperlinks. Following is the list of the features that are need to be extracted from the web.

3.2.1 Features based on their Abnormal webpage scripts: [18].

a) Request URL: This feature checks whether the webpage has any external domain objects such as images, videos.

b) URL of anchor: From this feature we examine whether the website and the anchor tag <a> has different domain names.

c) Links in tags such as script, meta, Link: This feature helps to analyse whether all these tags present in the webpage belong to the same domain.

d) SFH: It stands for Server Form handler. Using this feature, we can get the details whether the domain submitted to SFH is similar to the domain name of the webpage.

e) Submitting information to Email: The availability of server side scripts such as "mail()" or the client side script such as mailto() on the webpage can assumed to be that the webpage is suspicious.

f) Abnormal URL: This feature can extract the website information through WHOIS database.

3.2.2 Features of the Address Bar:

a) Having IP address: If the website is having the IP address instead of the domain name, we can say that website is used to perform malicious activity.

b) URL Length: This feature calculates the length of the URL and state whether the URL is satisfying the minimum length or not.

c) Use of URL shortening applications: This feature extracts whether this application is used to shorten the link.

d) Contains "@": This feature extract whether the "@" symbol is available or not.

e) Using "//": the position of the double slash is examined and output is produced.

f) Contains "-" in domain names: The dash symbol used by the attacker to separate the domain name by inserting this symbol in between the domain name.

g) having multiple sub domains: The rule for extracting this feature is to eliminate the top-level domain, second level domain and the www and then the number of dots are calculated in order to classify whether the link is malicious or not.

h) HTTPS: This feature extracts the presence of HTTPS along with the certificate issued details.

i) Length of the domain registered: The life of the website's domain is verified. Usually it should be more than equal to 1 year.

j) Favicon available: It is the unique graphic icon that belongs to the websites. It checks whether the favicon is loaded from internal or external domain and gives the output.

h) Nonstandard Port: It validates particular service is coming from the specific server.

3.2.3 Features extracted based on JavaScript and HTML:

a) Forwarding of Website

b) Customisation of the Status Bar

c) Right click disabled: Verifies the mouse click.

d) Pop up Window

e) IFRAME Redirection: It extracts if the webpage is using iframes or not.

3.2.4 Features of Domain:

a) Domain age: The domain age is verified and o/p is produced.

b) DNS Record: This feature extracts the DNS record of the website.

c) Web Traffic: The domain traffic is checked and the information is then extracted.

d) PageRank: It is the important feature as the legitimate websites may have good rank compared to the malicious ones.

e) Google Index: This feature extracts the information whether the website is available on the Google' Index.

The Feature Pre-processing technique is used where all the information is transformed into the numerical vector space before giving it as an input to the machine learning algorithm. [11] Through this numerical vector space, it is not possible to classify which links are malicious, so for classification purpose ELM machine learning algorithm [12] was used.

3.3 Extreme Learning Machine (ELM):

The ELM is a single hidden layer feed forward neural network in which only one hidden layer is there between the input layer and the output layer and weights are chosen between the input layer and the hidden layer. [26]. The parameters such as activation function, threshold value and weight must have assigned the appropriate values that ensures the high-performance learning rate of the model. The ELM model learns faster compared to the gradient based algorithms and also, the ELM does not face the difficulties that the gradient based algorithm faces such as learning rate, stopping criteria, local minima [25], thus increases the performance ratio. In order to activate the neuros in the hidden layers of the ELM model, various function such as sigmoid, activation function have to be used. [18]. The fig: 2 shows the structure of the ELM based neural network consist of Input Node, Hidden Nodes, Activation function and the output node. [24]

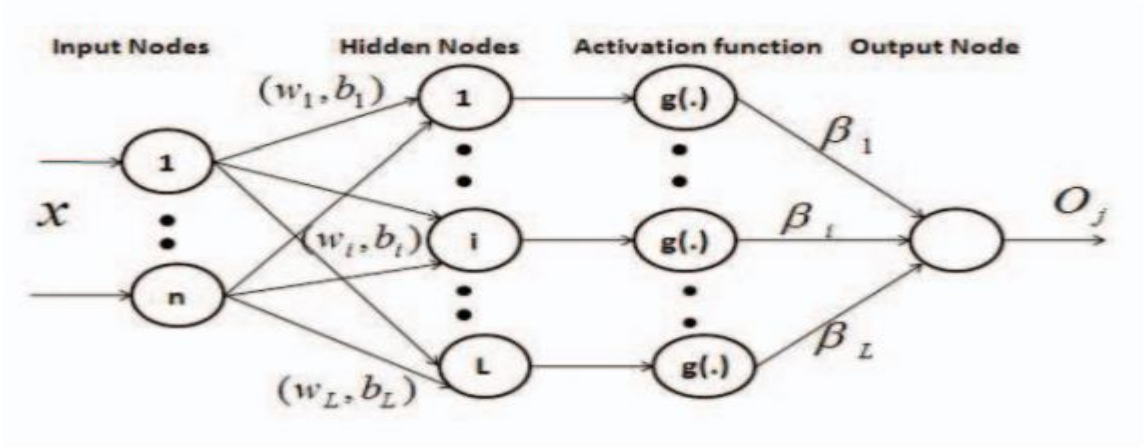

Figure 2: Working of ELM model

The ELM Classifier is used for the binary classification. The ELM classifier detects the malicious URL based on learning which the model has adopted using the dataset. The ELM classifier performs well compared to the other supervised machine learning algorithms. [18]. Hence ELM is used for the classification of the URLs. This algorithm is trained on the dataset so that the model can train and behave accordingly. Once the data is trained, the numerical vector space which is collected in array format from extracted features are given as an input to the model and after evaluation the result is calculated to check the performance of the model.

a) Dataset Description:

To perform the clickjacking attack, the attacker with a malicious intent can embed the links into webpage with an intention to perform the phishing attacks, the Drive-by-download attacks, spear phishing attacks and many more. For research purpose, the phishing dataset which is available publicly on website "https://archive.ics.uci.edu/ml/datasets/Phishing+Websites" is used to train the model and classifies the links as a phishing or legitimate. The dataset contains 11055 data with 30 different features.

b) **Model Training:**

The ELM model is trained on the phishing dataset. The dataset is split into 80% for training purpose and 20% for testing purpose of the model to evaluate the performance. The model learns on the dataset and perform accordingly to predict the result based on the feature extracted in the form of numerical vector space.

3.4 Highlighter:

In this subsection of the research, the detected malicious links are displayed on the webpage which are used for clickjacking. As the machine learning model identifies which URL links are malicious and the list of such malicious website links are provided. The attackers might use the opacity property to take the advantage of the HTML property and the Iframe property to perform an attack. The attackers might use the CSS position property and the z-index property to place the fake or the malicious iframe or website link on the real website. [19].

In the research, same technique is used to show such malicious links to the user by highlighting them. For highlighting, opacity of the detected links is increased to 100% so that all the hidden links are visible and to show the overlapped frame, the z-index property of this links and the alignment of this links are manipulated. Using the cascading style sheet property, these links are made visible to the user. To display all these changes, a dummy webpage is created where all the malicious links are highlighted and made visible to the user. This approach is adopted so that changes are visible on the dummy HTML, keeping the actual webpage safe by not making any change in it. Through this approach, the links used for the clickjacking are made visible and the user is prevented from being clickjacked.

3.5 Model Evaluation:

There are many metrics available to calculate the performance of the model, since our model ELM is a classification model, we are going to use only the classification metrics such as Accuracy, Precision, F1 score and recall. The confusion matrix is not a performance metrics, instead it used to calculate the True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) which is further used in metrics calculations [23]. The metrics are as follows:

1) Accuracy:

   It is defined as the number of correct predictions divided by the total predictions observed and multiplied by 100. Accuracy formula is:

   $$Accuracy = ((TP+TN)/(TP+FP+FN+TN) *100)$$

2) Recall:

   It calculates actual positive results are correctly identified from True Positive and False Negative samples. The formula for Recall is: [23]

   $$Recall = TP/(TP+FN)$$

3) Precision:
It calculates the positive results divided by number of True positive and False positive that are predicted.

Precision = TP/(TP+FP)

4) F1-Score: [23]
It defines how precisely the classifier is working by considering the precision and the recall result. The performance of the model is directly proportional to the F1 score.

F1 Score = 2*(Precision*Recall/Precision + Recall)

4 Design Specification:

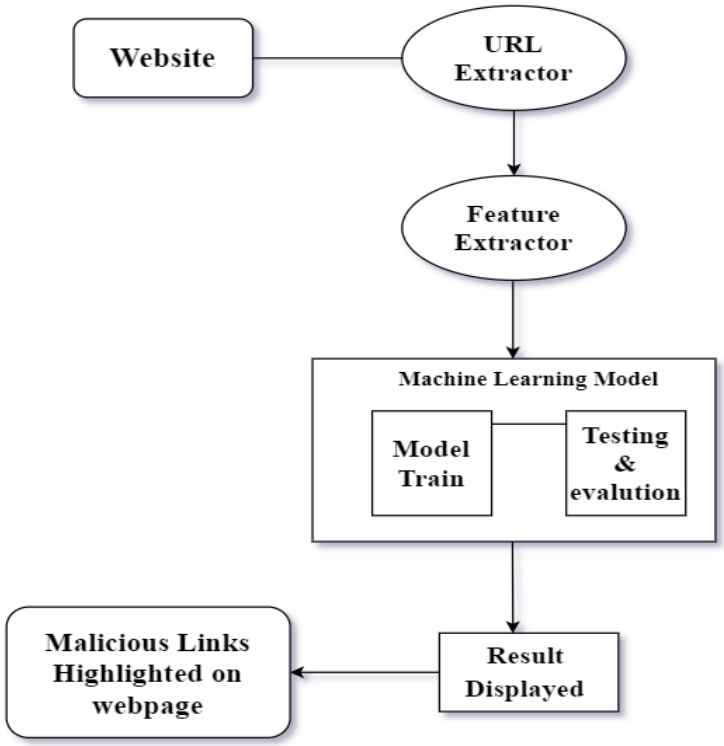

Figure 3: System Architecture

Above is the system architecture that helps in detecting the clickjacking attacks that are conducted by the attacker on a webpage. The system detects the clickjacking attacks by detecting the malicious link available on the page. The Web scrapper scrapes the whole HTML page for us and from the whole HTML page only URLs are extracted, that's why the terminology URL extractor is given. Then the features (such as Features for Domain, Features based on their Abnormal webpage scripts, Features of the Address Bar, Features extracted for JavaScript and HTML, Features extracted based on JavaScript and HTML) are extracted from this input. In the machine learning model, the ELM classifier is trained on

80% percent of the dataset. The model evaluates the o/p based on 2 facts, 1st is training achieved using the dataset and 2nd is through the array of numerical values given as an input to the model for prediction. If the result shows the list of malicious URLs, then using highlighter system these links are highlighted on the dummy webpage which is a replica of the original webpage.

5 Implementation

In this section, different tools and techniques are used for implementation of the research is explained.

The python Django web framework is used to develop the web application. This framework is one of the best open source web frameworks which includes the component to develop a secure and maintainable website. In this application multiple iframes and hyperlinks are added on the HTML page and their opacity is kept at "0" so that it won't be visible to the users. The URL extraction is carried out on this web application to extract the content from the HTML page.

Using web scrapping technology, the HTML page is scrapped to extract the URL using python3. Python has a large number of libraries, from which the request and the BeautifulSoup library are used. Library in python language is very useful and important, because just by importing the libraries all the functionality associated with them can be utilised just by loading the library. The request library makes a request to the web page for the information and to receive the response from the server, the get() function is used. Some libraries do not come preinstalled in python so there is a need to install manually. The BeautifulSoup library is installed using command "pip3 install beautifulsoup4".

The BeautifulSoup is used to parse the HTML page. From this parsed data, only the hyperlinks and iframes links are extracted. As some of the hyperlinks are used for the internal object therefore such hyperlinks are excluded from the list of URLs that we found in the URL extraction process and a new URL list is obtained. The feature extraction process is carried out which takes the input one at a time from the new URL list and all the extracted features output are stored in the numerical vector space representation. The output in numerical vector is accumulated in another list in the array format and fed into the Extreme Learning Machine algorithm for detecting the phishing links.

The python libraries such as pandas, NumPy, sklearn and pickle were used for Extreme Learning Machine algorithm. The NumPy library works best with the arrays. Data cleaning process is carried out in order to check any "null values" or "NA" values in the dataset. The sklearn python package was used for the Extreme Learning Machine algorithms. The dataset is checked for the class imbalance, but it was found to be balanced. The correlation measures the dependency of the variable on each other. The correlation between the independent variable and the dependent variable is identified.

After data processing, the dataset is then split into training and testing part. The "train_test_split" package from sklearn is loaded for the ELM model. The model is trained on the training dataset and that is used for the future prediction. The ELM classifier is trained,

and based on this learning the classifier can classify the phishing links. The trained model is saved in the pickle file so that it can be used later to make predictions. The numerical vector array of the extracted features is given as an input to the Extreme learning model that classifies the link as a phishing and list of phishing links is obtained.

To display the links on the HTML page, a dummy page is created. To create a dummy page, a HTML page obtained through the "beautifulsoup" is converted in to a string to make the further changes. The phishing link list available through the ELM model is compared with the list of links obtained from the URL extraction and on the matched links, the opacity value is set to 100% and for the hidden iframes their position is updated, so that the hidden frames alignment get changed and it is displayed properly on the webpage to the user.

6 Evaluation

In order to detect the clickjacking attack on the websites the Extreme Learning Machine algorithm and the HTML's CSS technology is used. For evaluating the performance of the Extreme Learning Machine (ELM) model, various metrics such as precision recall, F1 score have been used. The confusion matrix is also evaluated to compute this performance metrics. For the comparative analysis, the performance metrics of the SVM model is also evaluated. The performance of both the models is compared with respect to accuracy, precision recall, F1 score and Time.

6.1 Evaluation result of ELM:

The experiment is carried out for the ELM model to check the performance behaviour of the model which is used in the research for the detection. The confusion matrix for the ELM model is shown in the Table 1. It represents the TP, FP, TN, FN values.

Confusion Matrix			
TP	TN	FP	FN
884	1139	111	77

Table 1: Confusion matrix for the ELM model

Based on the confusion matrix the performance of the ELM model is calculated. The performance metrics values are shown in the Table 2.

Accuracy	**Precision**	**Recall**	**F1-score**	**Training Time (in sec)**
91.40%	91.00%	95.00%	93.00%	0.34

Table 2: Performance matrix of the ELM model

The overall accuracy of the ELM model obtained is 91.40%, the time taken for the model to train is 0.34 sec. The below graph shows the ROC (Receiver operating characteristics) curve showing the performance of the model, with respect to the true positive rate and the false positive rate of the model. Approximately, 91% of roc_auc_score for the ELM classifier is achieved.

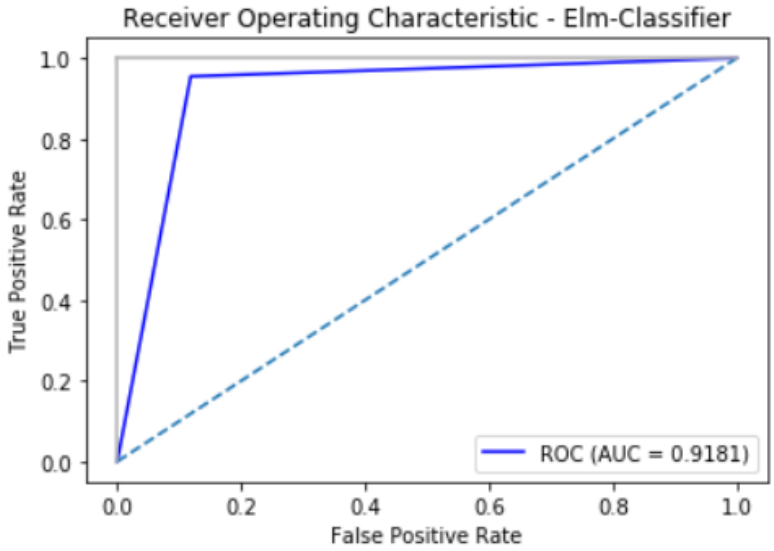

Figure 4 : ROC curve

6.2 Evaluation result of SVM

The SVM model is one of the best models for the binary classification. So, for comparison purpose, the SVM is chosen. The confusion matrix obtained for the SVM model is shown in Table 3. It contains, True Positive, False Positive, True Negative and False Negative values.

Confusion Matrix			
TP	**TN**	**FP**	**FN**
882	1154	113	62

Table 3: Confusion matrix for the SVM model

The values obtained from the Confusion matrix are used to evaluated the performance of the SVM model. The performance metrics such as accuracy, Time, Precision, recall and F1 score are evaluated. The Table 4 shows the performance metrics of the SVM model.

Accuracy	**Precision**	**Recall**	**F1-score**	**Training Time (in sec)**
92.00%	91.00%	95.00%	93.00%	1.1 sec

Table 4: Performance matrix of the SVM model

6.3 Discussion

The extreme learning machine algorithm is compared with support vector machine model with respect to their performance metrics. SVM is one of the best supervised models used for binary classification, thus it is used for comparison. From the evaluation result of both the models, it is observed that the overall accuracy achieved by the extreme learning machine is 91.40 % and the accuracy of the support vector model is 92%, the difference in the accuracy is not much between the models. The time taken by the ELM model is very less i.e. 0.34 sec whereas SVM model took time i.e. 1.1 sec to train the model, hence it is observed that the training time taken by the ELM classifier is much less compared to the SVM. The ELM model and SVM model both performed well in terms of the accuracy, precision, recall, F1-score, but the time taken by the SVM model to train the model is more. Hence it can be said that the Extreme learning machine model (ELM) is much faster than the SVM model.

7 Conclusion and Future Work

The attackers use the malicious links as their best weapon to get control over the victim's system by making the user to fall prey to such links. The clickjacking is carried out by the attacker, so that the user may fall for phishing, spam and many more attacks. This paper proposed an approach to detect the malicious links used for clickjacking attacks using extreme learning algorithm and CSS property on the HTML webpage. After performing an experiment, the iframes and the links which are used to perform clickjacking are displayed on the webpage successfully with the help of ELM classifier that detects such malicious links and then using the opacity property, position property of the CSS, this links are highlighted and displayed on a webpage. All the links which are predicted as malicious are displayed on the webpage only if these links are used in Iframes and in href tags. The overall accuracy of the ELM model is 91.40% which is a good accuracy in itself, so the overall proposed solution is effective.

The detection of advertisement links which are actually used in real time scenario to perform clickjacking is a limitation encountered during this research. However, in the future work, the advertisement links used for the clickjacking will be detected. For this a real time server can be created and the advertisement links will be added on this webpage from this server. Furthermore, a web browser extension can be created by which the detection and highlighting of this clickjacking will be performed on a single click of extensions.

References

[1] Balduzzi, Marco & Egele, Manuel & Kirda, Engin & Balzarotti, Davide & Kruegel, Christopher. (2010). A solution for the automated detection of clickjacking attacks. 135-144. 10.1145/1755688.1755706.

[2] Bala, Kavita, and Dr E. Babu Raj. EFFECTIVE APPROACH TO DETECT CLICKJACKING ATTACKS. 2016, https://www.semanticscholar.org/paper/EFFECTIVE-APPROACH-TO-DETECT-CLICKJACKING-ATTACKS-Bala-Raj/873defae4652d90659eeddaaaa252b8d3bff0326

[3] T, Krishna Chaitanya, et al. "Analysis and Detection of Modern Spam Techniques on Social Networking Sites." 2012 Third International Conference on Services in Emerging Markets, 2012, pp. 147–52. IEEE Xplore, doi:10.1109/ICSEM.2012.28.

[4] Hill, Brad Andrew. Adaptive User Interface Randomization as an Anti-Clickjacking Strategy.

[5] Rehman, Ubaid Ur, et al. "On Detection and Prevention of Clickjacking Attack for OSNs." 2013 11th International Conference on Frontiers of Information Technology, 2013, pp. 160–65. IEEE Xplore, doi:10.1109/FIT.2013.37.

[6] Cova, Marco & Krügel, Christopher & Vigna, Giovanni. (2010). Detection and analysis of drive-by-download attacks and malicious JavaScript code. Proceedings of the 19th International Conference on World Wide Web, WWW '10. 281-290. 10.1145/1772690.1772720.

[7] Canali, Davide, et al. "Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages." Proceedings of the 20th International Conference on World Wide Web, Association for Computing Machinery, 2011, pp. 197–206. ACM Digital Library, doi:10.1145/1963405.1963436.

[8] Deiana, Giulia. Analysis and Detection of Clickjacking on Facebook. UCL, 29 Apr. 2015, http://www0.cs.ucl.ac.uk/staff/g.stringhini/papers/deiana_clickjacking.pdf.

[9] Shin, Youngsang & Myers, Steven & Gupta, Minaxi & Radivojac, Predrag. (2015). A link graph-based approach to identify forum spam. Security and Communication Networks. 8. 10.1002/sec.970.

[10] Sahoo, Doyen & Liu, Chenghao & Hoi, Steven. (2017). Malicious URL Detection using Machine Learning: A Survey.

[11] Agrawal, Dr. Jitendra, et al. "Malicious Web Page Detection through Classification Technique: A Survey." International Journal of Computer Science And Technology, vol. Vol. 8, no. Issue 1, Mar. 2017, http://www.ijcst.com/vol8/1/15-dr-jitendra-agrawal.pdf.

[12] Ahmed, Shafi. Real Time Detection of Malicious Webpages Using Machine Learning Techniques. 2015.

[13] B.Sayamber, Anjali & Dixit, Arati. (2014). Malicious URL Detection and Identification. International Journal of Computer Applications. 99. 17-23. 10.5120/17464-8247.

[14] Nethra, K., et al. "Web Content Extraction Using Hybrid Approach." SOCO 2014, 2014. Semantic Scholar, doi:10.21917/IJSC.2014.0099.

[15] Saurkar, Anand V., et al. An Overview on Web Scraping Techniques and Tools. 2018, https://www.semanticscholar.org/paper/An-Overview-on-Web-Scraping-Techniques-and-Tools-Saurkar-Pathare/132ce5f7831e05f73430dcda7e98777734e51577.

[16] Namasivayam, Bhuvana Lalitha. “Categorization of Phishing Detection Features And Using the Feature Vectors to Classify Phishing Websites.” Undefined, 2017, https://www.semanticscholar.org/paper/Categorization-of-Phishing-Detection-Features-And-Namasivayam/7f1d9fbbe0ff4c9ab9a0e93cbd30ed26fe5c1609.

[17] Basnet, Ram & Mukkamala, Srinivas & Sung, Andrew. (2008). Detection of Phishing Attacks: A Machine Learning Approach. 10.1007/978-3-540-77465-5_19.

[18] Y. Sönmez, T. Tuncer, H. Gökal and E. Avcı, "Phishing web sites features classification based on extreme learning machine," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-5, doi: 10.1109/ISDFS.2018.8355342.

[19] Selim, Haysam & Tayeb, Shahab & Kim, Yoohwan & Zhan, Justin & Pirouz, Matin. (2016). Vulnerability Analysis of Iframe Attacks on Websites. 1-6. 10.1145/2955129.2955180.

[20] Shahriar, & Devendran,. (2014). Classification of Clickjacking Attacks and Detection Techniques. Information Security Journal: A Global Perspective. 23. 10.1080/19393555.2014.931489.

[21] HachmanMarch 21, Mark, and 20133 Min Read. “Chameleon Clickjacking Botnet Stealing $6 Million a Month.” Dice Insights, 21 Mar. 2013, https://insights.dice.com/2013/03/21/chameleon-clickjacking-botnet-stealing-6-million-a-month/.

[22] Cao, Jiuwen & Lin, Zhiping. (2015). Extreme Learning Machines on High Dimensional and Large Data Applications: A Survey. Mathematical Problems in Engineering. 2015. 10.1155/2015/103796.

[23] Minaee, Shervin. “20 Popular Machine Learning Metrics. Part 1: Classification & Regression Evaluation Metrics.” Medium, 28 Oct. 2019, https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce.

[24] S. Baraha and P. K. Biswal, "Implementation of activation functions for ELM based classifiers," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2017, pp. 1038-1042, doi: 10.1109/WiSPNET.2017.8299920.

[25] H. Rong, G. Huang and Y. Ong, "Extreme learning machine for multi-categories classification applications," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 1709-1713, doi: 10.1109/IJCNN.2008.4634028.

[26] Várkonyi, Dániel, and Krisztián Buza. Extreme Learning Machines with Regularization for the Classification of Gene Expression Data. Telekom Innovation Laboratories, http://ceur-ws.org/Vol-2473/paper11.pdf.

[27] Web Scraping Using Python: A Step By Step Guide. https://www.octoparse.com/blog/web-scraping-using-python. Accessed 15 Aug. 2020