A Combinational Approach of Hybrid Model BiLSTM-CNN-GRU to Improve the Detection rate of Click jacking in Websites

MSc Research Project
Msc Cyber Security

Student ID: 21215898

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Udhaya Thirunavukarasu

Student ID: 21215898

Programme: MSc Cyber Security **Year:** 2023-2024

Module: MSc Academic Internship

Supervisor: Vikas Sahni

Submission Due Date: 31 Jan 2024

Project Title: A Combinational Approach of Hybrid Model BiLSTM-CNN-GRU to Improve the Detection rate of Click jacking in Websites

Word Count:6314 **Page Count:**20

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.

Signature: ……Udhaya Thirunavukarasu……

Date: ……28/01/2024……

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):	

A Combinational Approach of Hybrid Model BiLSTM-CNN-GRU to Improve the Detection rate of Click jacking in Websites

UDHAYA THIRUNAVUKARASU
21215898

Abstract

The growing incidence of clickjacking attacks poses a severe threat to web security. Clickjacking is still a concern today because of how dynamic it is and how well it can get around traditional defenses. Despite earlier studies in this area, there is still much to be learned about clickjacking hits because it is unclear how far the attack can go in gathering the victim's personal information.

The primary focus is to provide a more accurate and efficient detection method to detect attacks. In order to identify clickjacking attempts with fewer false positives and false negatives, So, BILSTM hybrid layer of CNN+GRU technique is used, that classifies the malicious Phishing content present on the webpage and will be highlighted. This hybrid deep learning model was compared with Convolutional Neural Network (CNN) model with respect to their performance metrics like Accuracy and statistical calculation of Specificity and Sensitivity metrics. The results demonstrate the BILSTM model's enhanced skills in distinguishing between benign and phishing URLs, providing a significant breakthrough in clickjacking detection

1 Introduction

Clickjacking is addressed in “Security Misconfiguration” in 5th rank in OWASP top10[1],which involves tricking people into unintentionally interfered with malicious elements that are hidden within the website.Figure 1 shows good example of a malicious webpage is layered over a legitimate website. When a user clicks on the legitimate website, they are actually clicking on the malicious webpage and can be redirected to any website the attacker desires as stated by MarcoBalduzzi(2010)[1]. As per Fahani et al(2014) a study has been made in identifying a clickjacking worn that can propogate themselves hidden in social media page.

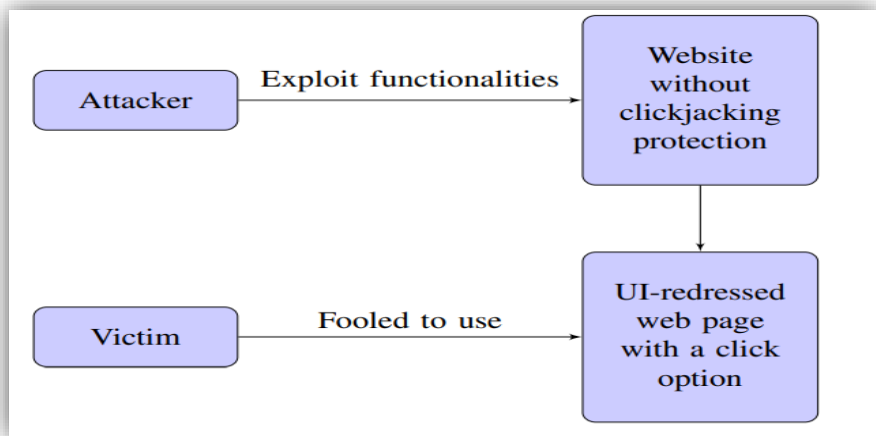

Figure 1- Click jacking

[1]https://owasp.org/www-community/attacks/Clickjacking

While recent research efforts have explored the application of Convolution Neural Network (CNN) techniques for clickjacking detection. Yet this approaches have achieved a promising 10% false rate in the work done by Hariram.K, (2023), there is still ample room for improvement to enhance detection accuracy further.

1.1 Research Question

1. In the identification of clickjacking instances, how does the Convolution Neural Network (CNN) model compare to the Bidirectional LSTM (BILSTM) model with such a hybrid layer of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU)?
2. What are the subtle variations between the CNN and BILSTM models' sensitivity (recall) and specificity, and how do these metrics contribute to the models' robustness in reliably recognising real positives and true negatives in the context of clickjacking?

The aim of this research is to detect clickjacking occurrences in URLs using deep learning models, especially Convolutional Neural Network (CNN) and Bidirectional LSTM (BILSTM) with a hybrid layer of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). The primary purpose is to assess these models' ability to distinguish between benign and phishing URLs, with a focus on phishing detection. The study intends to examine the models' performance in terms of accuracy, sensitivity, specificity, and other important metrics using a dataset obtained from several URL categories, including benign, spam, phishing, and malware URLs.The findings are meant to provide useful insights into the world of cybersecurity, laying the groundwork for the creation of effective and adaptable clickjacking detection techniques to improve online security.

2 Related Work

2.1 Non Machine Learning Research Studies on detecting Clickjacking

According to Lin-Shung Huang *et al.* (2012), The visual integrity and temporal integrity of user interactions with vulnerable UI elements are protected by a number of strategies used by “InContext” which a web browser application.“InContext” imposes pauses before sending user actions to sensitive elements in order to guarantee temporal integrity, especially when the pointer enters the element. However this solution can bring a significant defense from clickjacking but not on all browsers that will support this strategy. This gave a idea of using machine learning techniques for defense

R. Sinha and D. Uppal (2024), have addressed a innovative strategy of leveraging Java's Aspect-Oriented Programming (AOP) to simplify the addition of HTTP headers. AOP enables the establishment of aspects and advices to intercept do get post method calls, ensuring the automatic inclusion of the X-Frame-Options header at the necessary locations. This AOP strategy in improving server-side defense against clickjacking attacks quickly and effectively. But this cant completely mitigate the issues and still faces the issues with more sophisticated hidden Iframes. But this gave rise to the idea of feature extraction of the dataset that is used in the current research.

The approach of BeEF framework by B. Lundeen and J. Alves-Foss (2012), was first created to address particular clickjacking vulnerability. It is a standalone tool to mitigate number of attcaks, the contained div surrounding the iframe, and an input element to detect when a click occurs are the only three fundamental HTML elements required to carry out the assault. That after the attack is over, the page's entire clickjacking setup—including the iframe—is erased.

The document object model of the clickjacked page will therefore no longer include any evidence of the attack after the second click. This type of defense have a lot of limitations when comes to more sophisticated phishing attacks.
As per commenced by K. S. Rao*et al.* (2016),XBuster was proposed defense against all known XSS attack vectors as well as clickjacking in this study field. They implemented a Firefox extension that simulates process HTTP requests, XBuster divides each parameter into JavaScript and HTML contexts. The HTTP response is then examined for each H context match. The rate of false negatives and false positives is influenced by the latter, which is a critical design element. In this current research, the HTML content will be further featured for the analysis for labeling.

2.2 Machine Learning Related research Studies on Detecting clickjacking

The Extreme Learning Machine (ELM) has been used by Yashodha Patil (2021),to find clickjacking iframes. The research methodology is described in depth using ELM and SVM. The CSS technology of HTML and the Extreme Learning Machine algorithm are employed.The performance metrics of the SVM model is also evaluated and has derived the result of 91% on 1.5 seconds time period. This performances metrics are used in current research project to evaluate the metrics on hybrid model. Applying deep learning model is predominant in predictions for phishing.
As per Hariram, K. (2023) used applied Convolution Neural Network (CNN) method approach to find the Clickjacking iframes on a website. Malicious iframes on a webpage are highlighted using the HTML CSS attribute. The site contents are particularly URLs that were collected using a web scraping technique from the webpage URLs. After he evaluation, The accuracy eventually reached 91% after multiple test instances but have a false positive of 10% .This research is the base idea of the current research and adopting a hybrid model to overcome with better accuracy.
Research done by Jain et al.,(2022) recognised smartphones' vulnerability to phishing attempts, motivating the creation of APuML, an Anti-Phishing strategy based on Machine Learning that used the Random Forest classifier to reach a detection accuracy of 93.85 per. In parallel, As Shin et al., 2022) addressed the phishing challenge by proposing an ensemble approach using various machine learning models such as Random Forest, K-Nearest Neighbor, and AdaBoost, resulting in a significant 6 per cent improvement over single models and the detection of 141 additional malicious URLs. On the other side according to Wit et al., (2022) employed machine learning to identify mobile malware on Android smartphones, stressing the usefulness of the Random Forest classifier and getting an F1 score of 0.73 across 10 Mobile Trojan subtypes. Furthermore, Ahmed et al., (2023) recognised the importance of phishing attack detection and introduced PhishCatcher, a client-side defence mechanism based on machine learning, specifically the Random Forest classifier, which demonstrated an impressive accuracy and precision of 98.5per cent in classifying phished and legitimate URLs. On the other side Rehman et al(2013)proposed a browser solution for clickjacking with success rate of 76% but the succession rate was so low.

2.3 Research studies on using Hybrid model for detections

The hybrid CNN-BiLSTM approach used by N. Gandhi*etal*(2021) was successful in identifying SQL injection by obtaining information from diverse queries using convolutional layers. Data is fed into a max pooling after convolution to extract high-level features. Bi-LSTM is used to process data chronologically to provide a better level of understanding. More accuracy is offered by a hybrid approach for SQL injection based on CNN-BiLSTM than by any other machine learning algorithm mentioned. N. Gandhi*et al* (2021) proposal

reduces the frequency of SQL injection attacks by foreseeing them utilizing the hybrid CNN-BiLSTM. This hybrid model shows a good approach in detection of the SQL attacks, So for this current research work, It'll be more perfect to adopt this combination of hybrid model for the clickjacking classification.

Sinha, J. and Manollas, (2020)proposed a model that combines multiple layers of bi-directional LSTM (Bi-LSTM) and a 1-Dimensional convolutional neural network (1-D CNN).This model's results show promising potential real-time usage for intrusion detection systems.

2.4 Comparison of previous Works

Privious Papers	Methodology	Key Contribution
Click jacking Attacks and Defence(Lin-Shung Huang et al., 2012)	Custom defence Strategy (Incontext)Application	Visual integrity and temporal integrity protection.
Clickjacking existing Defenses and Some Novel(R. Sinha et al., 2014)	Aspect-Oriented Programming (AOP) with Java	Simplified addition of HTTP headers for server-side defense.
Practical Clickjacking with BeEF (B. Lundeen, 2012)	BeEF framework	Demonstrates clickjacking vulnerability using BeEF framework.
A combined browser defense against XSS and clickjacking (K. S. Rao et al., 2016)	Firefox extension simulating XBuster	Provides defense against XSS and clickjacking.
Detection of Clickjacking Attacks using the Extreme Learning Machine algorithm (Yashodha Patil ,2021)	Extreme Learning Machine (ELM)	Uses ELM for clickjacking iframe detection.
Detection of Clickjacking using Convolutional Neural Network (Hariram, K. , 2023)	Convolutional Neural Network (CNN)	Highlights malicious iframes using CNN.
A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks (N. Gandhi, J. Patel, R. Sisodiya, N. Doshi and S. Mishra, 2021)	CNN-BiLSTM	Successful identification of SQL injection using the hybrid model.
Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection (Sinha, J. and Manollas ,2020)	CNN-BiLSTM	Efficient model combining CNN and Bi-LSTM for network intrusion detection.
Proposed: BiLSTM- CNN-GRU Hybrid for Clickjacking	BiLSTM-CNN-GRU	Aims to improve clickjacking detection accuracy using spatial and sequential information

		provided by CNN and BiLSTM.

2.5 Traditional Models for Click jacking Detection

While the three studies, done by (Saini et al., 2019), (Possemato et al., 2018) and (Roy et al., 2023), focus on different cybersecurity risks and difficulties, they all have one thing in common: they all focus on the vulnerabilities and possible abuses connected with sophisticated technology. The researchers identify and address rising vulnerabilities in the digital ecosystem in each scenario, demonstrating the longevity of certain security problems. Especially, both Saini et al. and Possemato et al. focus on failure of present defence measures against given types of attacks—clickjacking in the one incident and mobile-based UI attacks in the second. Relatedly, Roy et al. prove the vulnerability of Large Language Models (LLMs) to misuse in making phishing attacks, highlighting the larger issue of urbane language models' possible malicious usage. Added to this, Jaber and Fritsch emphasize the emergent usage of artificial intelligence (AI) in cyber-attacks and propose for the creation of attack model platforms to detect and prevent AI-powered risks. Likewise, Eskandari et al(2018). Investigate the modern routine of in-browser crypto currency mining, aiming on Monero via platforms such as Coinhive. Together studies, done by Jaber and Fritsch,(2019) and Eskandari et al., (2018), include new trends and troubles in the territory of cyber security. Both sections emphasize the altering nature of cyber threats and the necessity for proactive countermeasures. Both answers stress the need to recognize and react to new cyber risks, whether they be AI-driven assaults or bitcoin mining via web browsers. Besides, both studies propose innovative techniques to focus the highlighted struggles. Both Shu et al(2018). and Santander et al(2020). Acknowledge the inadequacy of traditional security measures in the admit of quickly progressing technology, as well as the significance of continually advance security systems. Wang et al (2023). And Lv et al (2023). Investigate specific security challenges in the mobile app and online domains, emphasizing the vital significance of these policies in the larger perspective of cyber security.

2.6 Research niche:

The current clickjacking defense approaches and machine learning models expose merits but face little of restrictions. The period towards hybrid models, like BiLSTM, this will contribute to this development, the proposed Combinational Approach combines BiLSTM-CNN-GRU, aiming for enhanced accuracy in identifying clickjacking.

3 Research Methodology

3.1.1 Research Steps:

The primary step involves loading a widespread dataset containing 800 URLs from the benign class plus 950 URLs from the phishing class. Leveraging the urllib library, the subsequent process focuses on feature extraction and labeling. The extracted features, pivotal for training the subsequent model, are meticulously saved into a CSV file for efficient organization and accessibility. Feature extraction is performed based on two categories: Address Bar-based Features and HTML &and JavaScript-based[2.]

[2]https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-Learning-Techniques

In the figure-2, shows how the raw data is then transformed into a format through thorough data pre-processing, which involves cleaning, normalization and splitting the dataset. Then It is given into the training model (BiLSTM with hybrid layer of CNN and GRU) and it will predict if it is malicious and out and the Output phase is resulted in GUI page where the phishing content is highlighted.

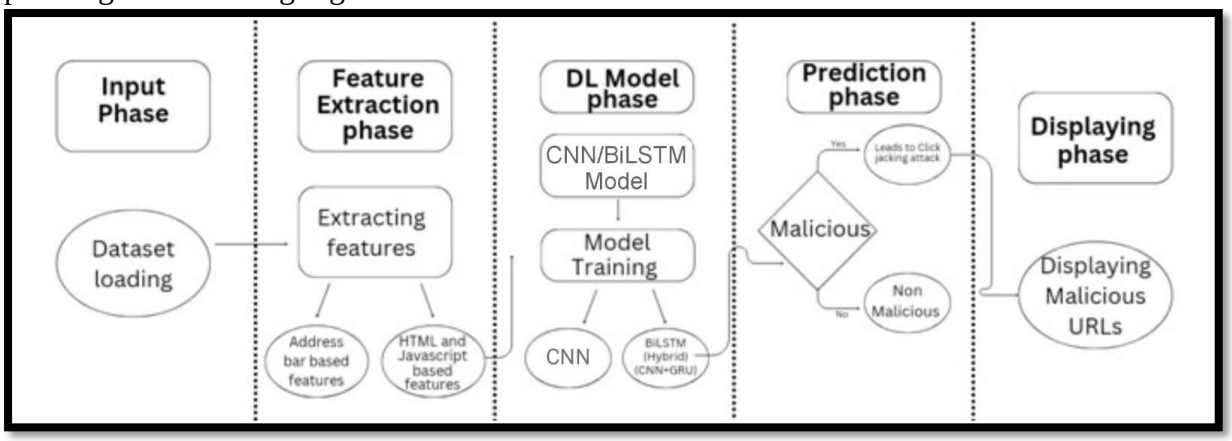

Figure 2 - Flow Diagram

3.2 Feature Extraction (12 Features)

3.2.1 Address Bar Attribute Section

length_of_url	Computes the length of the URL, aiding in detecting potential phishing attempts utilizing lengthy URLs to obscure suspicious content
http_has	Checks for the presence of "http/https" in the domain part of the URL, a crucial indicator as phishers may manipulate the "HTTPS" token to deceive users
suspicious_char	Identifies the presence of the '@' symbol in the URL, a tactic used by phishers to redirect browsers and conceal the genuine address.
prefix_suffix	Checks for the presence of '-' in the domain part of the URL, a rare occurrence in legitimate URLs, indicating potential phishing activity.
dots	Counts the number of dots in the URL, offering insights into URL complexity
slash	| Examines the presence of "//" in the URL, a potential sign of redirection, with the position of "//" being crucial in determining the type of URL (HTTP or HTTPS)
sub_domain	Detects the presence of a sub domain in the URL, an important characteristic for distinguishing legitimate URLs from potential phishing URLs
ip_contain	Identifies the presence of an IP address in the URL, a red flag for potential phishing attempts.

3.2.2 Domain Attributes Section

Domain Length	It is calculated by counting the number of characters in the domain name. Longer domain names may be indicative of suspicious URLs, as attackers might use elongated

	domains to deceive users
Number of Sub domains	A higher number of sub domains might suggest a more complex URL structure, and attackers may manipulate sub domains for malicious purposes.
Use of Hyphens in Domain	Hyphens are commonly used in phishing URLs to mimic legitimate domains.
Presence of Top-Level Domain (TLD)	A valid TLD is a crucial component of legitimate URLs. Its absence or uncommon TLDs may raise suspicions about the authenticity of the URL.

3.2.3 Features Extracted Based on JavaScript and HTML Section

IFrameRedirection	Identifies the use of the "iframe" tag for potential phishing, especially when employed invisibly without frame borders.
Status Bar Customization	Detects changes to the status bar using JavaScript, a common tactic to display fake URLs to users.
Disabling Right Click	| Identifies the disabling of right-click functionality using JavaScript, a method employed by phishers to prevent users from viewing and saving webpage source code.
Website Forwarding	Analyzes the number of times a website has been redirected, with phishing websites often displaying multiple redirects, contrasting with legitimate websites that redirect minimally

3.3 Dataset used:

The Dataset was published by Cybersecurity $(CIC)^2$, enhances the research's dependability and credibility. The dataset obtained from the University of New Brunswick (UNB).

3.4 Model Used:

Model Training part to provide the deep learning model with the capacity to correctly distinguish and categorize URLs. The procedure begins with the importing of required libraries, laying the groundwork for following processes.

The dataset loaded, and a delicate data cleaning process begins, attending null values and removing unnecessary columns to increase the dataset's suitability for model training. Then pre-processing processes are accepted to further refine the data. To get perceptions into the dataset's features, visualization techniques are used. Pursuing that, the dataset is thoroughly split into training, testing, and validation sets in an 80:10:10 ratio to confirm a well-balanced distribution. The model training comprises the use of both Convolutional Neural Network (CNN) and Bidirectional LSTM (BILSTM) architectures, the last of which is added by a hybrid layer that combines Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU).

This advanced pattern is fundamental for detecting difficult patterns in data. The training results are thoroughly assessed, which involves the creation of a Confusion Matrix, a Classification Report, and the estimation of Specificity and Sensitivity metrics.

3.5 List of Models

1. **CNN (Convolution Neural Network)**
2. **BILSTM (Bidirectional LSTM) with Hybrid Layer (CNN+GRU)**

3.6 Data Visualization

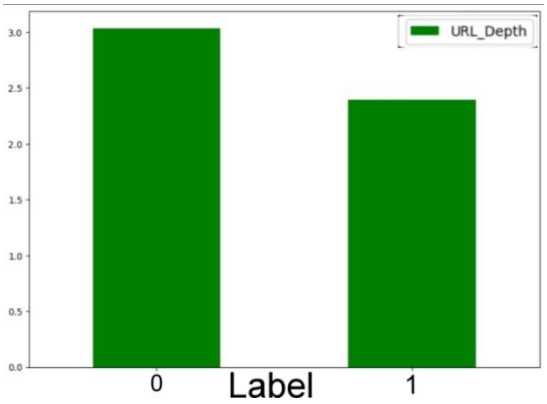

Figure 3 - Bar Plot - Average URL Depth by Label

In the Figure-3, showcases bar plot chart demonstrating the average URL depth sorted by labels. In this circumstance, the word "URL depth" classically refers to the ordered structure of a URL, indicating the number of levels or directories within the URL. Each bar in the plot corresponds to a specific label, which could signify different categories or classes of URLs, such as benign, phishing, spam, or malware. The height of each bar represents the average URL depth for URLs belonging to that particular label

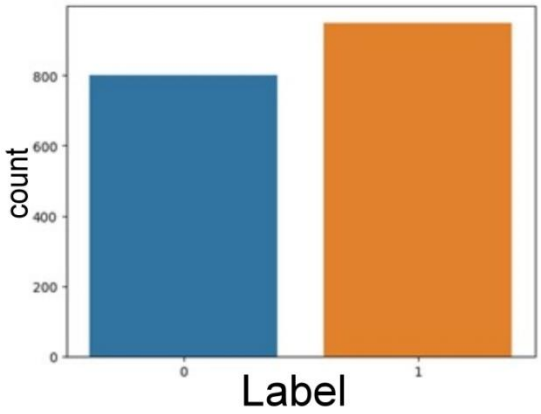

Figure 4 - Plot - Distribution of Records in Each Target Class (0: Benign, 1: Phishing)

In the Figure-4, presents a Count plot depicting the distribution of records in each target class, where class 0 corresponds to benign URLs, and class 1 represents phishing URLs. The x-axis of the plot is labeled with the target class values (0 and 1), designating the different categories, while the y-axis enumerates the count of records. The count values range from 0 to 800 and are visually represented above each respective bar on the plot. Each bar on the plot corresponds to one of the target classes, and its height represents the number of records belonging to that class.

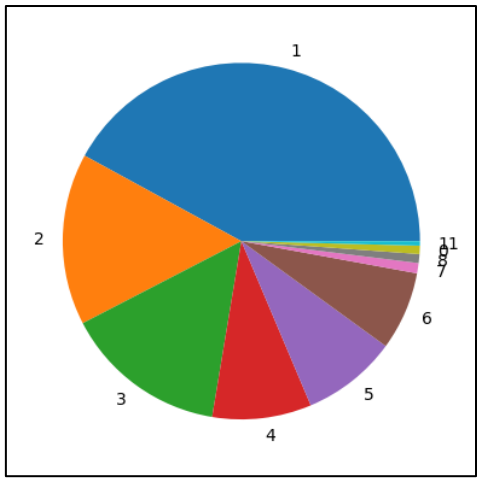

Figure 5 - Pie Chart - Distribution of URL Depth Values

In the Figure-5, features a Pie Chart representing the distribution of URL depth values within the dataset. Each segment of the pie corresponds to a unique URL depth category, with the size of each segment proportional to the value count of URLs falling within that specific depth range.

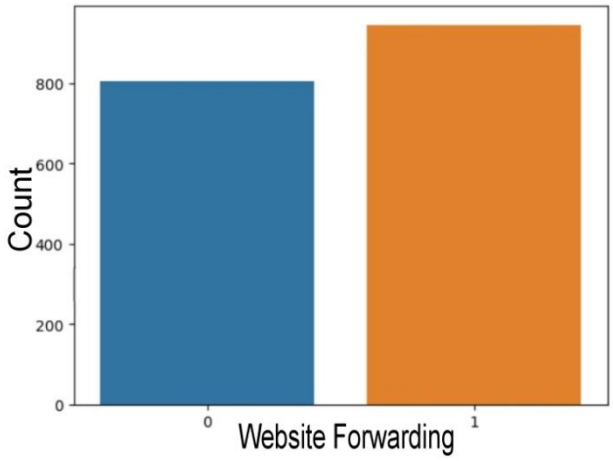

Figure 6- Count Plot - Distribution of Website Forwarding (0: No, 1: Yes)

In the Figure-6, exhibits a Count Plot portraying the distribution of the "Website Forwarding" feature. The x-axis of the plot is labeled with the values 0 and 1, representing the binary labels where 0 signifies no website forwarding, and 1 indicates the presence of website forwarding. The y-axis enumerates the count of occurrences for each label.

4 Design Specification:

This design specification combines data preparation, deep learning, as well as a graphical user interface (GUI). Python is the major technology and design option, with versions above 3.7 providing compatibility and access to the most recent language features. The implementation environment includes Google Colab, a cloud-based collaboration platform, and Visual Studio Code, a flexible code editor that offers a blend of collaborative development and powerful individual coding capabilities. Google Colab integration enables collaborative model training and experimentation, while Visual Studio Code provides a fast and adaptable programming environment.

Python libraries are vital in allowing the system's numerous functionalities. Pandas is used for effective data processing, and Matplotlib and Seaborn are used for complete data expose. Scikit-learn is the go-to package for machine learning tasks, granting a comprehensive set of algorithms and kits for model formation and evaluation. TensorFlow and Keras are used to construct deep learning models, especially Convolutional Neural Network(CNN) and Bidirectional LSTM (BILSTM) using a hybrid layer combining Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU).

These deep learning models excel in collecting sequential and geographical patterns within URL data, enhancing the system's capacity to distinguish between dangerous and benign (Legitimate) URLs. Beautiful Soup's[3] inclusion enables web scraping functionality, which extracts URL characteristics. For GUI development, Flask, a lightweight web framework, is used to create an easy interface allowing users to interact with the clickjacking categorization module. Users may enter URLs into the GUI, which causes the model to anticipate and visibly indicate the existence of clickjacking components.

5 Implementation

Convolutional Neural Network (CNN) and hybrid model (CNN-BILSTM-GRU) models are trained on the preprocessed dataset, and they are supplemented with a hybrid layer that combines Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). TensorFlow and Keras are the primary model development frameworks, exploiting their capabilities for creating, training, and assessing complicated neural network designs.

To accomplish robust performance and overview, the models are trained utilizing an 80:10:10 split of train-test-validation. In addition, the execution includes the creation of a Graphical User Interface (GUI) with Flask. This interface yields it easy for clients to engage with the clickjacking classification module. Users can present URLs, which bases the algorithm to predict whether a clickjacking attempt is hitting.

The GUI allows the user to control the model's features without lacking substantial technical knowledge. Visual Studio Code is utilized as the primary code editor throughout the application, offering a beneficial environment for development, debugging, and version management. Google Colab, which permits for real-time collaboration on Jupyter notebooks, enables the collaborative mechanisms of the implementation.

In the applied graphical user interface (GUI), users are exhibited with a home page that functions as the entry point to two apparent sub-pages, each designed to showcase peculiar traits of the clickjacking detection model. The first sub-page simulates a Original webpage containing a malicious iFrame, allowing users to observe how the model identifies and handles such potentially harmful elements. On the second sub-page it is named "Clickjacking Results". The GUI utilizes BeautifulSoup[3] to scrape all the links present on this page, and the extracted URLs undergo feature extraction. These features are then fed into the Bilstm hybrid model for prediction. If a predicted phishing URL is detected, a red border is dynamically added to the corresponding section of the clickjacking web app, providing a visual indication of the potentially highlight red boarders on the malicious content.

[3]https://oxylabs.io/blog/web-scraping-for-machine-learning

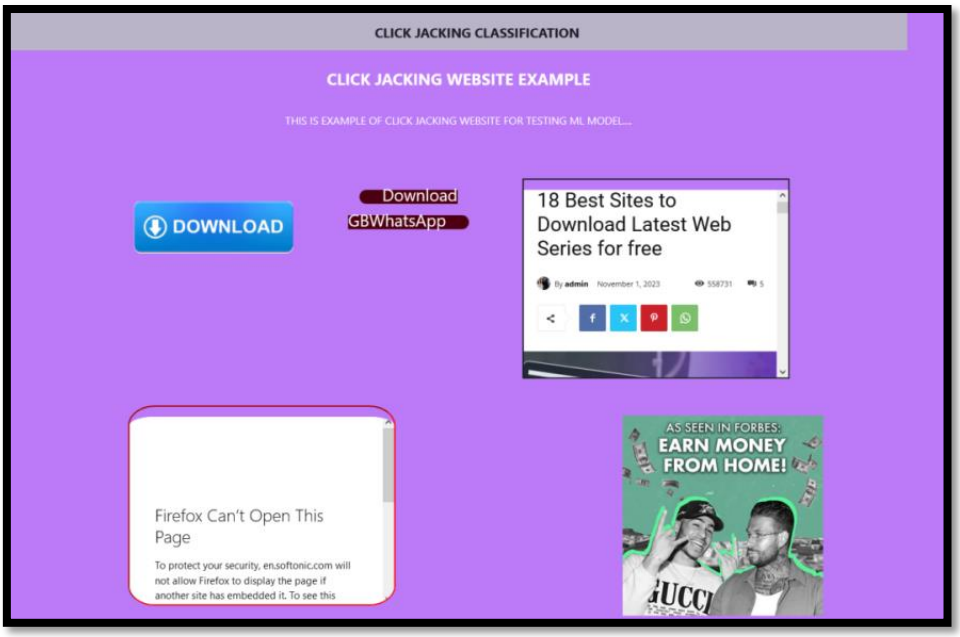

Figure 7 - Model Output Screen: Clickjacking Classification

In figure-7, The Output webpage is interactive demonstration not only allows users to witness the model's real-time performance but also serves as an effective tool for conveying the practical application of the clickjacking detection research. The screen provides an example of a clickjacking website, featuring deceptive content such as "download this," "download GB WhatsApp," and "earn money from home," which are common phrases associated with clickjacking attempts. This visualization is a valuable demonstration of the model's ability to identify and classify potentially malicious URLs indicative of clickjacking activities

6 Evaluation

6.1 CNN (Convolutional Neural Network)

The evaluation of the CNN model is predicting clickjacking demonstrates its robust performance in discerning visual patterns indicative of potential threats[4]. Leveraging convolutional layers for effective feature extraction, the CNN model showcases commendable accuracy in distinguishing between normal and click jacked iframes in URLs. Sensitivity and specificity, vital metrics for assessing the model's ability to correctly identify clickjacked and normal URLs, respectively, exhibit competitive values.

The model excels in capturing visual cues and spatial hierarchies, making it adept at detecting subtle manipulations in the layout of web pages associated with clickjacking attempts. While CNN's sequential processing capabilities may not match those of dedicated sequential models like BiLSTM, its resilience to variations in visual patterns underscores its reliability for clickjacking detection, especially in scenarios where visual features play a pivotal role.

[4]https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306

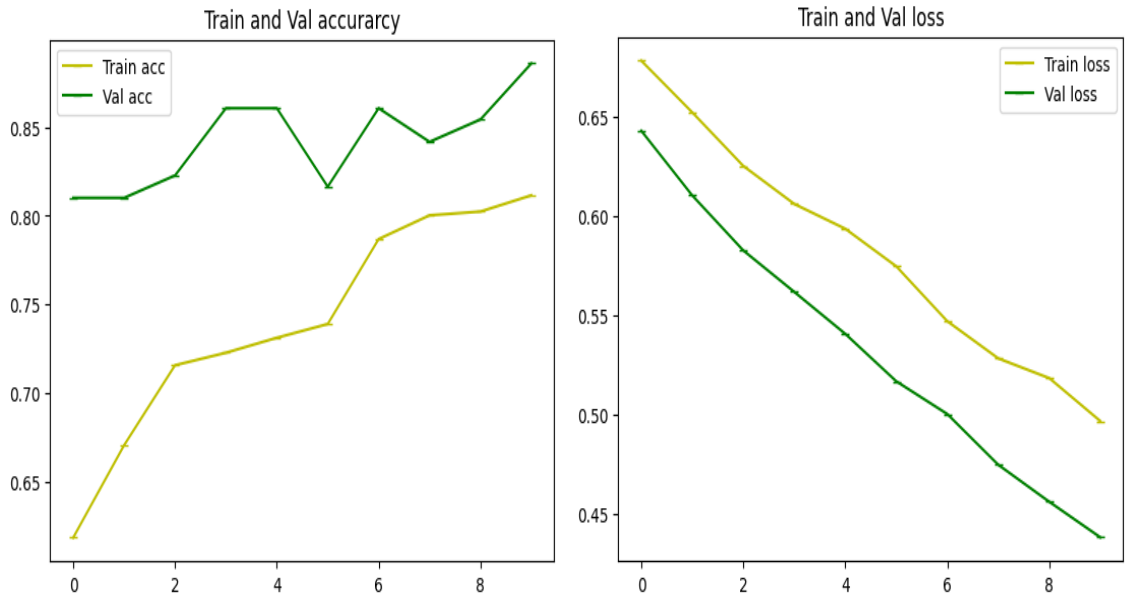

Figure 8 - CNN Accuracy graphFigure 9- CNN Loss graph

After ten epochs of training, in the figure-8, the CNN model had a validation accuracy of 90%. The validation accuracy is a statistic that analyses the model's performance on a distinct dataset that was not used during training, revealing information about its ability to generalize to new data. The acquired accuracy indicates that the CNN model makes predictions on the validation set with an accuracy of around 90% percent.

In the figure-10, training loss decreases monotonically over the course of training. This shows that the model is learning the training data well. However, the training loss does not reach zero, showing that the model is not perfectly learning the training data.

6.2 BILSTM (Bidirectional LSTM) with Hybrid Layer CNN+GRU

BILSTM improves higher capabilities of CNN. This bidirectional method expands the model's comprehension of situation and its ability to sense nuanced patterns in sequence data[5]. This bidirectional processing is supplemented further by the hybrid layer of CNN+GRU, which combines spatial and sequential characteristics vital to clickjacking detection.

The bidirectional character of BILSTM is succeededusing two sets of memory cells, one managing the sequence from beginning to end and the another in reversal. Because of this bidirectional flow, the model may obtain dependencies from both the past and the future.

The hybrid layer contains features of a convolutional neural network (CNN) for spatial feature extraction and gated recurrent unit (GRU) for more temporal modeling. The hybrid layer of BILSTM is very actual at extracting complex designs associated with clickjacking challenges. The bidirectional processing confirms a comprehensive comprehension of URL sequences, while the hybrid layer permits the model to describe both sequential and spatial data characteristics.
As a conclusion, BILSTM with a hybrid layer is a robust solution for detecting complicated clickjacking behaviors that may imply URL structure modification in various directions.

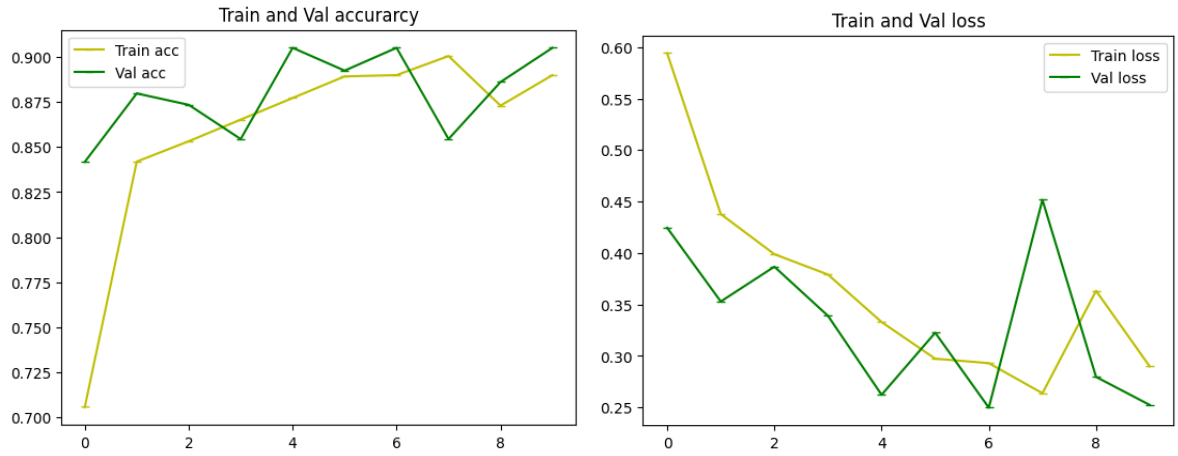

Figure 10 - Hybrid BiLSTM Accuracy GraphFigure 11 - Hybrid BiLSTM Loss Graph

The BILSTM model, which had been advanced using a hybrid layer integrating CNN and GRU, done a validation accuracy of 93% with 10 training epochs. This meaningful improvement in accuracy over the CNN model exhibits that bidirectional processing[5], structured with the hybrid layer's spatial and temporal feature extraction resources, has substantially improved the model's capacity to acknowledge patterns within the sequential URL data. The validation set accuracy of 93% percent proves a strong performance in categorizing clickjacking cases. In the figure-11, this hybrid deep learning model is better at extracting features from the web page data and classifying them as normal or malicious.

6.3 Classification and Confusion matrix

The Precision, recall, and F1-score measures[6] were used to estimate the classification functioning of the deep learning models, which involved CNN and BILSTM with a hybrid layer of CNN+GRU.

	precision	recall	f1-score	support
0	0.88	0.93	0.90	80
1	0.93	0.89	0.91	95
accuracy			0.91	175
macro avg	0.91	0.91	0.91	175
weighted avg	0.91	0.91	0.91	175

Figure 12- Classification report of CNN Model

The classification report for the CNN model displays an accuracy of 91%, with a precision, recall, and F1-score around 0.91 for both classes(benign and phishing). This indicates that the

[5]https://medium.com/@nutanbhogendrasharma/sequence-prediction-with-bidirectional-lstm-model-fc8b94b3357

[6]https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

model performs well in classifying instances of both classes, demonstrating a balanced performance. The weighted mean metrics are also around 91%, showing that the model generalizes well across the entire dataset.

	precision	recall	f1-score	support
0	0.90	0.94	0.92	80
1	0.95	0.92	0.93	95
accuracy			0.93	175
macro avg	0.92	0.93	0.93	175
weighted avg	0.93	0.93	0.93	175

Figure 13 - Classification report of BiLSTM hybrid Model

The BILSTM model's accuracy, recall, and F1-score values were drastically higher for both the benign (90 percent, 94 percent, and 92 percent) and phishing (95 percent, 92 percent, 93 percent) classes. This presents that the BILSTM model, with its bidirectional processing and hybrid layer involving CNN and GRU outperformed other models in constantly identifying URLs, especially in differentiating phishing URLs. These metrics show that the model performs well in categorizing instances of both classes, demonstrating a balanced and accurate performance. The weighted average metrics are also near 93%.

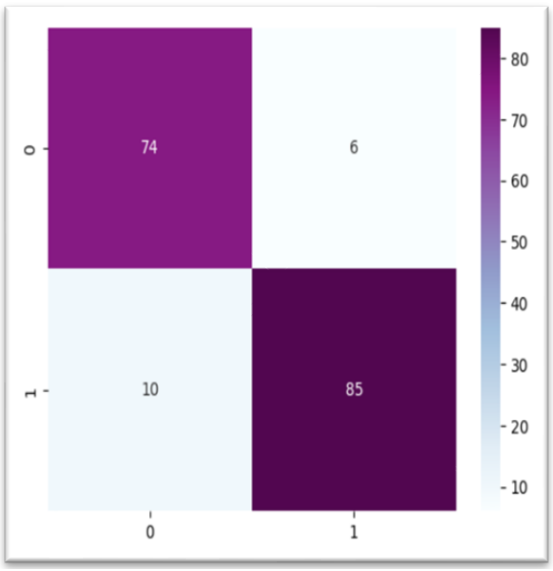

Figure 14 - CNN Confusion matrix

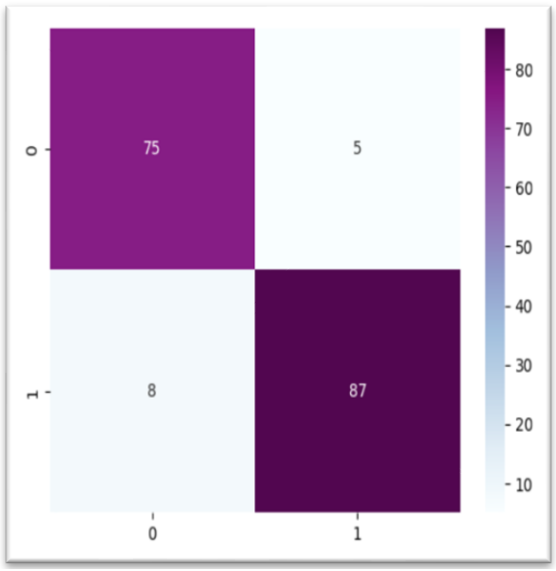

Figure 15 - BiLSTM Hybrid confusion matrix

The two confusion matrices exhibit both CNN and BiLSTM models can be valuable for clickjacking detection. BiLSTM with hybrid CNN-GRU model is the more accurate of the two models.

- The CNN model has an accuracy of 80% and a precision of 91.5789% for the vulnerable class
- The BiLSTM with hybrid CNN-GRU model has an accuracy of 90% and a precision of 95% for the vulnerable class.

The BiLSTM with hybrid CNN-GRU model is able to capture both local and long-range dependencies in web page content, which gives it a more comprehensive view of the page.

Table 6.3.1: Comparison Table: Model Metrics Comparison

Model	Accuracy	Support	Precision	Recall	F1-Score
CNN	0.91	175	93	92	93
BILSTM (with CNN+GRU)	0.93	175	95	96	96

Table 6.3.2: Comparison of Sensitivity and Specificity for CNN and BILSTM with Hybrid Layer (CNN+GRU)

Model	Class	Sensitivity	Specificity
CNN	0	0.89	0.92
	1	0.92	0.89

Model	Class	Sensitivity	Specificity
BILSTM (CNN+GRU Hybrid Layer)	0	0.91	0.93
	1	0.93	0.91

With Table 6.3.2,

- CNN shows specificities of 92% and 89% for classes 0 (Benign) and 1 (Phishing), respectively, and sensitivity of 89% for class 0 (Benign) and 92% for class 1 (Phishing).
- BiLSTM hybrid with 93% and 91% for classes 0 (Benign) and 1 (Phishing), respectively, and sensitivity of 91% for class 0 (Benign) and 93% for class 1 (Phishing)
- BiLSTM hybrid model has a slight edge in both measures.

6.4 Discussion

The BILSTM model with the hybrid layer outperforms the CNN in terms of overall accuracy, obtaining an accuracy of 93% as opposed to 91% for CNN. This indicates increased efficiency for the hybrid model since it shows a better success rate in correctly identifying situations.

Both models exhibit good performance in terms of specificity and sensitivity, with the BiLSTM hybrid model owning a modest advantage in both categories. However, which model to select may be determined by your particular application requirements. The BiLSTM

hybrid model may be a desirable solution if the goal is to lower the danger of misclassifying phishing websites as benign. The CNN model, on another hand, may be a preferable replacement if the goal is to extend the accuracy of locating benign websites.

7 Conclusion and Future work

The aim of this project was to design and examine deep learning models for clickjacking detection using URL features and compare CNN and BILSTM with a CNN+GRU hybrid layer, established good results in classifying between benign and phishing URLs. The best performance was the BILSTM model, which attained the maximum accuracy of 93 percent. This proves the efficiency of bidirectional processing and the use of a hybrid layer in detection complicated patterns associated by clickjacking attempts. Advanced sensitivity and specificity values further reinforce the BILSTM model's resilience, revealing its ability to effectively identify both benign and phishing occurrences. The findings of this learning help to develop clickjacking detection algorithms, giving a responsible solution for protecting users from dangerous online activity.

There are specific limitations for this model. It mostly relies on extracted URL characteristics, and their efficacy may be crushed by developing clickjacking tactics. Likewise, the dataset utilized for training and estimation may not contain all feasible scenarios, possibly restricting the models' general ability to real-world situations. Furthermore, while the performance values are helpful, they do not give insights into model interpretability, which is a significant part of understanding the decision-making process.

Future study may address the noted flaws and improve the proposed models. Discovering a larger limit of clickjacking approaches on more different datasets might help to advance model generalization. Investigating the model's interpretability might guide to a better understanding of the factors that influence predictions. Additionally, investigating ensemble approaches or containing new contextual information may enhance model performance. Forthcoming research should focus on constant adaptation to evolving clickjacking strategies and the combination of real-time detection systems to ensure the models stay effective in dynamic online settings.

References

Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. 2010. *A solution for the automated detection of clickjacking attacks*. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS '10). Association for Computing Machinery.

M. R. Faghani and U. T. Nguyen, "A study of clickjacking worm propagation in online social networks," Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, USA, 2014, pp. 68-73, doi: 10.1109/IRI.2014.7051873.

Hariram, Kishore (2023) *Detection of Clickjacking using Convolutional Neural Network.* Masters thesis, Dublin, National College of Ireland.

Huang, L.-S., Moshchuk, A., Wang, H. J., Schecter, S., & Jackson, C. (2012). *Clickjacking: Attacks and Defenses*. In Proceedings of the 21st USENIX Security Symposium (USENIX Security 12) (pp. 413-428)

R. Sinha, D. Uppal, D. Singh and R. Rathi, "Clickjacking: Existing defenses and some novel approaches," 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, India, 2014, pp. 396-401, doi: 10.1109/ICSPCT.2014.6884934.

B. Lundeen and J. Alves-Foss, "Practical clickjacking with BeEF," 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 2012, pp. 614-619, doi: 10.1109/THS.2012.6459919.

K. S. Rao, N. Jain, N. Limaje, A. Gupta, M. Jain and B. Menezes, "Two for the price of one: A combined browser defense against XSS and clickjacking," 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 2016, pp. 1-6, doi: 10.1109/ICCNC.2016.7440629.

Patil, Yashodha (2020) *Detection of Clickjacking Attacks using the Extreme Learning Machine algorithm.* Masters thesis, Dublin, National College of Ireland.

N. Gandhi, J. Patel, R. Sisodiya, N. Doshi and S. Mishra, "A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE),

Sinha, J. and Manollas (2020) *Efficient deep CNN-BILSTM model for network intrusion detection: Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, ACM Other conferences*.

Sonmez, Yasin & Tuncer, Turker & Gökal, Hüseyin & Avci, Engin. (2018). Phishing web sites features classification based on extreme learning machine. 1-5. 10.1109/ISDFS.2018.8355342.

Saini, A., Gaur, M.S., Laxmi, V. and Conti, M., 2019. You click, I steal: analyzing and detecting click hijacking attacks in web pages. *International Journal of Information Security*, *18*, pp.481-504.

Possemato, A., Lanzi, A., Chung, S.P.H., Lee, W. and Fratantonio, Y., 2018, October. *Clickshield: Are you hiding something? towards eradicating clickjacking on android*. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1120-1136).

Roy, S.S., Thota, P., Naragam, K.V. and Nilizadeh, S., 2023. From Chatbots to PhishBots?--Preventing Phishing scams created using ChatGPT, Google Bard and Claude. *arXiv preprint arXiv:2310.19181*.

Jaber, A. and Fritsch, L., 2022, October. Towards ai-powered cybersecurity attack modeling with simulation tools: Review of attack simulators. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 249-257). Cham: Springer International Publishing

Eskandari, S., Leoutsarakos, A., Mursch, T. and Clark, J., 2018, April. A first look at browser-based cryptojacking. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 58-66). IEEE.Garand, A. (2023) *What is Clickjacking & How Do I prevent it?,Sucuri Blog*.

Santander, C.J.M., Moreno, H. and Alvarez, M.B.H., 2020, October. The evolution from Traditional to Intelligent Web Security: Systematic Literature Review. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-9). IEEE.

Lv, Y., Shi, W., Zhang, W., Lu, H. and Tian, Z., 2023. Don't trust the clouds easily: the insecurity of content security policy based on object storage. *IEEE Internet of Things Journal*.

Shu, K., Wang, S., Le, T., Lee, D. and Liu, H., 2018, November. Deep headline generation for clickbait detection. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 467-476). IEEE.

Wang, T., Zhang, T., Zhang, W., Li, X., Zhao, T. and Wang, Y., 2023, August. Mobile application-oriented interface hijacking security defense method. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)* (pp. 950-953). IEEE

Jain, A.K., Debnath, N. and Jain, A.K., 2022. APuML: An Efficient Approach to Detect Mobile Phishing Webpages using Machine Learning. *Wireless Personal Communications*, *125*(4), pp.3227-3248.

Shin, S.S., Ji, S.G. and Hong, S.S., 2022. A Heterogeneous Machine Learning Ensemble Framework for Malicious Webpage Detection. *Applied Sciences*, *12*(23), p.12070.

Panman de Wit, J.S., Bucur, D. and van der Ham, J., 2022. Dynamic detection of mobile malware using smartphone data and machine learning. *Digital Threats: Research and Practice (DTRAP)*, *3*(2), pp.1-24.

Altamimi, A.B., Ahmed, M., Khan, W., Alsaffar, M., Ahmad, A., Khan, Z.H. and Alreshidi, A., 2023. PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning. *IEEE Access*.

U. U. Rehman, W. A. Khan, N. A. Saqib and M. Kaleem, "On Detection and Prevention of Clickjacking Attack for OSNs," 2013 11th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 2013, pp. 160-165, doi: 10.1109/FIT.2013.37.