IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PHISHING WEBSITE DETECTION USING DEEP LEARNING

[1]Arun P, [2]Ravi Teja N, [3]Jayanth Gowda S,[4] T Kesuchand Sushil Kumar,[5] Divya Jyoti Bhuyan

[1] Assistant Professor ,[2]Student, [3] Student, [4] Student, [5] Student

Department of Computer Science and Engineering,

Cambridge Institute of Technology, Bangalore, India

Abstract: The internet's explosive growth has resulted in a rise in cyberthreats, with phishing assaults presenting a serious risk to both people and businesses. Phishing websites aim to trick visitors into disclosing private information, including bank account information and login credentials. Real-time detection of these harmful websites is essential to protecting consumers' privacy and security when they are online. Online security is seriously threatened by the frequency of phishing attempts, in which malevolent actors try to obtain personal information by pretending to be reputable websites. In order to counter this threat, this project presents an extensible and open-source system that uses an artificial neural network (ANN) to detect phishing websites. The goal of the phishing website detection system is to accurately distinguish between legitimate and phishing websites, improving the capacity to safeguard internet users from malicious attacks.

Index Terms **– Phishing, URLs, Features, ANN**

I. INTRODUCTION

Since the early days of the internet, phishing assaults have been a constant threat; the first cases were noted in the 1990s. Because of the rising expertise of cybercriminals and technological breakthroughs, these attacks have undergone substantial evolution over time. Phishing attacks in the past were mostly straightforward, frequently consisting of bogus emails that led victims to phony websites. But as users become more aware of these strategies, attackers started using more advanced ones, including spear phishing, in which they send customized communications to targeted people or organizations. The rise of online banking and e-commerce has provided attackers with new opportunities to exploit. Phishing attacks targeting financial institutions and online retailers have become increasingly common, as attackers seek to steal sensitive information such as credit card details and login credentials. These attacks can have serious consequences for individuals and businesses, leading to financial loss, reputational damage, and legal implications. To combat

the growing threat of phishing attacks, researchers and cybersecurity experts have developed various techniques and tools. One approach that has shown promise is the use of deep learning (ANN algorithm) to detect phishing websites.

Since the early days of the internet, phishing assaults have been a constant threat; the first cases were noted in the 1990s. Because of the rising expertise of cybercriminals and technological breakthroughs, these attacks have undergone substantial evolution over time. Phishing attacks in the past were mostly straightforward, frequently consisting of bogus emails that led victims to phony websites. But as users become more aware of these strategies, attackers started using more advanced ones, including spear phishing, in which they send customized communications to targeted people or organizations. Because phishing websites are constantly changing, it might be difficult to identify them. Even though industry, research, and academic communities have put forth a number of safeguards, phishing assaults still constitute a serious risk ANN algorithms, which use patterns in a website's features to detect malicious intent, present a viable method for differentiating between phishing and legitimate websites.

Overall, it sets the stage for understanding the complex landscape of phishing detection and cybersecurity, emphasizing the critical role of advanced technology like deep learning in combating evolving cyber threats. By addressing the challenges outlined like the absence of a thorough framework for feature extraction and updating a collection of legitimate and phishing websites is the first significant limitation. This leads to incomplete and out-of-date datasets, which lowers the accuracy of phishing detection. Second, the algorithms now in use frequently make use of a large number of features, for which there is scant evidence to demonstrate their applicability in differentiating between legitimate and phishing websites. This makes it harder to interpret the data and adds computational complexity. Last but not least, research-grade datasets are frequently skewed, exhibiting an imbalance in URL- or content-based features. This might result in biased classifiers that are not very good at generalizing to new data. Researchers and cybersecurity experts aim to develop innovative solutions that enhance online security and protect users from the detrimental impacts of phishing attacks. Artificial Neural Networks (ANN) have become an important weapon in the fight against phishing, with the ability to increase accuracy and automate the detection process. artificial neural network (ANN) algorithms are able to accurately identify phishing websites by examining trends in the features of those websites. However, the caliber of the training dataset and the features chosen for analysis determine how effective these algorithms are. In order to create strong and dependable phishing detection systems that can shield users from harmful attacks, it is imperative that these issues be resolved.

II. LITERATURE SURVEY

There are several approaches to detect phishing website detection. some approaches are [1] a deep learning-based method to achieve high-precision detection of phishing websites. The proposed method uses convolutional neural networks (CNN) for high classification to distinguish genuine websites from phishing websites. The evaluated model using data from 6,157 real websites and 4,898 phishing websites. Based on

the results of many experiments, our CNN model has proven to be effective in detecting unknown phishing sites, another method [2] involves based mainly on machine learning to identify real-time phishing websites by considering hybrid features based on URLs and hyperlinks to achieve high accuracy without dependence on third parties. phishing URL detection as a real-case scenario through login URLs [3] as been attempted using logistic regression model combined with time-frequency inverse document frequency (TF-IDF) feature extraction and has achieved an accuracy of 96.50% of URL input data. Phishing website detection as also been achieved by using[4] Paper presents a phishing detection model called PDGAN, which depends only on the Uniform Resource Locator (URL) of the website to achieve high performance and use of Long Short-Term Memory Network (LSTM) network as a generator for phishing URLs and the Convolutional Neural Network (CNN) as a moderator where as another approach involved [5] to detect phishing websites by PhishDet, a new way of detecting phishing websites through Long-term Recurrent Convolutional Network and Graph Convolutional Network using URL and HTML features. PhishDet is the first of its kind, which uses the powerful analysis and processing capabilities of Graph Neural Network in the anti-phishing domain and recorded 96.42% detection accuracy, with a 0.036 false-negative rate. It is effective against zero-day attacks, and the average detection time which is 1.8 seconds could also be considered realistic. The feature selection of PhishDet is automatic and occurs inside the system, this has outperformed similar solutions by achieving a 99.53% f1-score with a public benchmark dataset. However, PhishDet requires periodic retraining to maintain its performance over time, whereas another approach proposed [6] A URL-Based Social Semantic Attacks Detection with Character-Aware Language Model and comparison with LSTM and CNN where, A detection accuracy of 99.65% was recorded in all tests using the characteristic BERT-based detection model obtained from the average performance of 5-fold cross-validation. Considering the model's performance for each group, the CharacterBERT model is the best among our 3 models at detecting social attacks, achieving the best accuracy of 99.90% in the attack-seeking study. The detection of phishing website can also be achieved though[7] two different artificial intelligence (GAN)-based methods for linking phishing and legitimate web samples: Adversarial Autoencoders (AAE) and Wasserstein GAN (WGAN). The goal is to create real synthetic data that leverages real-world data to train products with better performance and increased resistance to attacks. The data is obtained from ten phishing campaign datasets used by AAE (Adversarial Autoencoders) and WGAN (Wasserstein GAN) to create synthetic data. Using real and synthetic data, we show how to achieve classification with better performance and stronger resistance to attacks. The researcher proposes a number of hypotheses and investigate them by trying to show that the synthetic models are indistinguishable from real models, susceptibility of the classifier to adversarial attacks, synthetic models are introduced to reduce resistance when datasets are trained, and big data Classifiers trained on it have better performance. The fact that the AAEs and WGANs are trained from many datasets makes it widespread use, another AI based method involves [8] four meta-learning models (AdaBoost-Extra Tree (ABET), Bag - Extra Tree (BET), Competitive Forest - Extra Tree (RoFBET) and LogitBoost-Extra Tree (LBET)) using additive tree-based classifier offers AI-based meta-learning applications are deployed on a database of phishing websites (now with state-of-the-art features) and their performance is evaluated. The detection accuracy of this model is no less than 97%, and the false alarm rate is minimal, no more than 0.028. Additionally, the proposed model outperformed existing machine learning-based models in detecting phishing attacks and

another method [9] proposed an Efficient An Anti-Phishing Technique to Protect e-Consume is resource request-based phishing discovery (RRPD). This technique may be applied on both the client and server sides and works by examining the resources request features of phishing websites. The web browser may detect phishing sites on the client side by using client RRPD. By examining a tiny sample of web content, server RRPD based on domain name system dataflow can identify suspected phishing sites on the server side, saving bandwidth and processing power to the user utmost, another simple method [10] that can be used is a method to identify phishing websites, using Normalized Compression Distance (NCD) (NCD). This method involves compressing and comparing web pages to identify similarities with known phishing sites, eliminating the need for special removal. Extensive research and development has been done to detect phishing attempts based on specific context, network and URL characteristics. Theoretically, there are many methods, data analysis methods and measurements. This should be done carefully to identify and compare the advantages and limitations of each method and its suitability for different situations.

III. DATASET

A dataset for phishing website detection typically includes features extracted from URLs, such as domain age, length of URL, presence of special characters, HTTPS usage, and more. It may also include labels indicating whether each URL is classified as legitimate or phishing.

Certain features that can be used to classify a website URL as legitimate or phishing is :

- Domain names exist in the Alexa ranking: A list of domain names ranked by the Internet is called Alexa ranking. The majority of phishing websites are hacks into fresh domains or authentic websites. Given that top-ranked domain names often have greater security, it is doubtful that the domain name will be included in the TLD if the phishing assault is launched on a website that has been taken over. The freshly registered domain name will not show up in the Alexa rankings if the phishing website is hosted on it.
- Subdomain length: The length of the URL subdomain. Phishing sites attempt to use their domain as their subdomain to mimic the URL of a legitimate website. Legitimate websites tend to have a short subdomain name.
- URL. length: Phishing URLs tend to be longer than legitimate URLs. Long URLs increase the likelihood of confusing users by hiding the suspicious part of the URL, which may redirect user-submitted information or redirect uploaded web pages to suspicious domain names.
- Prefixes and suffixes in URLs: Phishers trick users by remodeling URLs that look like legitimate URLs.
- Length ratio: Determine the ratio of the path's length to the URL's length; phishing sites frequently have a higher percentage of valid URLs..
- The "@" and "-" counts: within the URL, the "@" mark in the URL forces the browser to disregard earlier inputs and then reroutes users to the links they have typed.
- Punctuation counts: The quantity of "! #$% &" included in the URL, typically, phishing URLs use more punctuation.

- IP address: Instead of using a domain name, the host name portion of the URI utilizes an IP address.
- Port Number: If the URL has a port number, make that it is one of the known HTTP ports (21, 70, 80, 443, 1080, and 8080, for example). Mark the port number as a potential phishing URL if it is not included in the list.

IV. METHODOLOGY

User Interface (UI):

The UI provides a web-based platform for user interaction. It allows users to upload websites for analysis, view analysis results, and provide feedback. The Web Framework of phishing website detection system user interface and web request processing will be developed using Flask, a lightweight and adaptable web framework for Python. Building the system's frontend with Flask is a great option because of its simplicity and ease of use, which enables users to interact with the system through an intuitive web interface. Compared to the Django web framework, Flask is thought to be more Pythonic because the equivalent Flask web application is typically more explicit. Because there is no boilerplate code required to launch a basic app, Flask is also straightforward for beginners to start using. As a well-known Python web framework, Flask is a third-party Python library used in web application development.

Feature Extraction Module:

This module extracts relevant features from websites for phishing detection. It includes components for extracting URL-based features (e.g., domain age, URL length), content-based features (e.g., presence of certain keywords), and server-based features (e.g., server location, SSL certificate). We treat the URL string as a file containing content and use natural language processing (NLP)to extract features. The Feature Extraction Module serves as a critical component in phishing website detection, systematically extracting relevant features from websites to facilitate effective classification. This module encompasses various components tailored to capture diverse aspects of website characteristics. First of all, it has methods for extracting URL-based information, such the length and age of the domain, which might reveal details about the legitimacy of the website's creation and organization. Subsequently, attributes derived from content are retrieved to assess the existence of particular keywords or patterns suggestive of phishing activities. These characteristics provide important hints about the type and purpose of the information on the website. In addition, server-based characteristics like server location and SSL certificate status are taken into account when evaluating the security and dependability of the hosting environment Notably, the module treats the URL string as textual information by utilizing natural language processing (NLP) techniques, which makes it possible to extract significant features in a manner similar to processing text documents. By employing NLP, the module can discern linguistic patterns and semantic cues embedded within the URL, further enriching the feature set for phishing detection. Through a comprehensive amalgamation of URL-based, content-based, and server-based features, the Feature Extraction Module equips the subsequent phases of the detection process with robust and informative input, facilitating accurate classification of phishing websites.

Preprocessing:

Collect a dataset of URLs labeled as phishing or legitimate. Preprocess the URLs to extract features such as domain age, URL length, presence of certain keywords, and server location. Convert these features into a suitable format for input to the ANN, such as numerical values or one-hot encoding. Initially, a dataset containing examples of both phishing and legitimate websites is collected, encompassing various features such as URL length, domain age, presence of HTTPS, and usage of suspicious keywords. The data undergoes cleaning to eliminate irrelevant or redundant features, such as removing missing values or duplicates. Subsequently, feature engineering techniques are applied to enhance the dataset, potentially by extracting domain-related features or analyzing URL structures. Numerical features are normalized or standardized to ensure consistency in scale, preventing any bias towards features with larger values. Numerical representations of categorical variables are encoded so that ANN training is possible.

The dataset is then divided into testing and training sets in order to assess the model. Generally, allocate 70% of the data for training, 15% for validation, and 15% for testing. This ensures that each set of data contains a fair proportion of authentic and phishing URLs..

ANN Module:

ANN algorithms for training classifiers to differentiate between phishing and authentic websites. This module includes components for training, testing, and evaluating classifiers, along with mechanisms for algorithm selection and fine-tuning.

The ANN algorithm used in involves several key components:

- Input Layer: The retrieved features of the URL, including the server location, age of the domain, length of the URL, and the presence of specific keywords, are sent to the input layer. In the input layer, anodes stand in for each feature.

- Hidden Layers: Neurons in the hidden layers of the artificial neural network (ANN) use activation functions and weighted connections to process the input characteristics. These tiers aid in the network's discovery of the data's underlying patterns.
- Output Layer: An assessment of whether a URL is authentic or phishing is generated by the ANN's output layer. A single neuron in the output layer is utilized for binary classification tasks, including phishing URL detection, where a value near 0 denotes a genuine URL and a value near 1 denotes a phishing URL.

Activation Function: Every neuron's activation function uses the weighted sum of its inputs to calculate its output. The sigmoid, tanh, and ReLU (Rectified Linear Unit) functions are common activation functions in artificial neural networks (ANNs)..

Sigmoid Function (σ(x)): The sigmoid function maps any real-valued number to a value between 0 and 1 in the output layer. It's defined by the formula

$$\sigma(x) = 1 / (1 + e^{(-x)}), \quad (1)$$

where 'e' is Euler's number. It's used in neural networks to introduce non-linearity and is often used in the output layer for binary classification tasks.

Hyperbolic Tangent Function (tanh(x)): The tanh function is similar to the sigmoid function but maps input values to a range between -1 and 1. It's defined by the formula

$$\tanh(x) = (e^{(2x)} - 1) / (e^{(2x)} + 1). \quad (2)$$

Like the sigmoid function, tanh introduces non-linearity into the network and is commonly used in hidden layers of neural networks.

Rectified Linear Unit (ReLU): The ReLU function is defined as

$$f(x) = \max(0, x), \quad (3)$$

It indicates that it returns zero otherwise and the input value if it is positive. ReLU's ease of use and effectiveness in training deep neural networks have made it one of the most often used activation functions in deep learning. It expedites convergence during training and aids in solving the vanishing gradient problem.

Training Algorithm:

: The activation function of each neuro is to reduce the error between the expected output and the actual label of the URL, the ANN is trained by varying the weights of the connections between neurons. Usually, backpropagation is used for this, which determines the gradient of the error with respect to the weights and uses optimization strategies like gradient descent to modify them appropriately. To train an artificial neural network (ANN) to detect phishing websites, there are several important procedures that must be taken in order to maximize the network's performance and minimize overfitting. First and foremost, selecting an optimization technique is crucial. Two popular choices are Adam and stochastic gradient descent (SGD). While Adam modifies the learning rate for each parameter based on previous gradients, potentially resulting in faster convergence, SGD updates the model's parameters based on the gradient of the loss function with respect to a subset of the training data. It's critical to keep an eye on the model's performance on the training and validation sets during the training phase. Measures such as the F1-score, accuracy, precision, recall, and loss function value shed light on how well the model differentiates between legitimate and phishing websites which will be discussed in result section. Optimizing performance requires adjusting a few hyperparameters, such as batch size and learning rate. By testing with various variables, the optimal convergence speed to stability ratio can be found. Strategies like early stopping prevent overfitting by monitoring performance on a validation set and stopping training when degradation occurs. Regularization techniques such as L1 or L2 regularization penalize large weights as well. Batch normalization and dropout further enhance the model's stability and generalizability. By closely adhering to these guidelines, an ANN can be trained to identify phishing websites with exceptional accuracy and resistance.

External Interfaces:

include interfaces for gaining access to other resources, like phishing website repositories online or extra datasets for ANN. These interfaces improve the system's ability to detect phishing attempts by providing it with the most recent information. You may want to think about using an external interface that has capabilities such as URL scanning, content and structure analysis, comparison against established phishing patterns, and user feedback methods in order to identify phishing websites. Users could be able to report questionable websites or enter URLs for inspection through this interface, which could be a web application or browser extension. It should give directions on what to do next and give unambiguous indicators of a site's credibility. Accurate detection can also be improved by combining with machine learning models and threat intelligence feed.

Database:

Utilizes SQLite for efficient database management, storing website data, extracted features, and additional metadata such as website URLs and classifications. Table 1 presents the major fields of a table named URL. lightweight, serverless database engine that is ideal for managing the system's dataset of legitimate and phishing websites because of its efficiency and simplicity, which guarantees fast access to data for analysis and categorization.

name	description	sample
url	URL	https://amazom.mhmgmm.rest/mobile/
label	1: phishing 0: legitimate	1
source	Data source	Phish Tank
External id	Unique ID of the same data source	7270002
netloc	netloc	amazom.mhmgmm.rest
Gmt_created	Created date	2021-08-21 01:39:40

Table 1:URL description and structure in database

Notification:

This Module a sends alert messages to users via Telegram and whats app when a URL is classified as phishing or legitimate along with the percentage safety of the URL that the user wants.

V. RESULT

All the experiments were executed on a Asus Vivo book running Hexa-Core Ryzen 5 processor Windows 11operating system. The server has a 500 GB storage capacity. In addition, the test data ratio is 0.25.

In this study, we used a deep learning algorithm to assess the effectiveness of phishing website identification. The goal of the phishing website detection system is to accurately distinguish between authentic and phishing websites, improving the ability to protect internet users from malicious attacks. It does this by providing a

framework for extracting features and keeping an up-to-date dataset of phishing and genuine websites. According to our research, the ANN-based system generated results with a 97.64% accuracy rate.

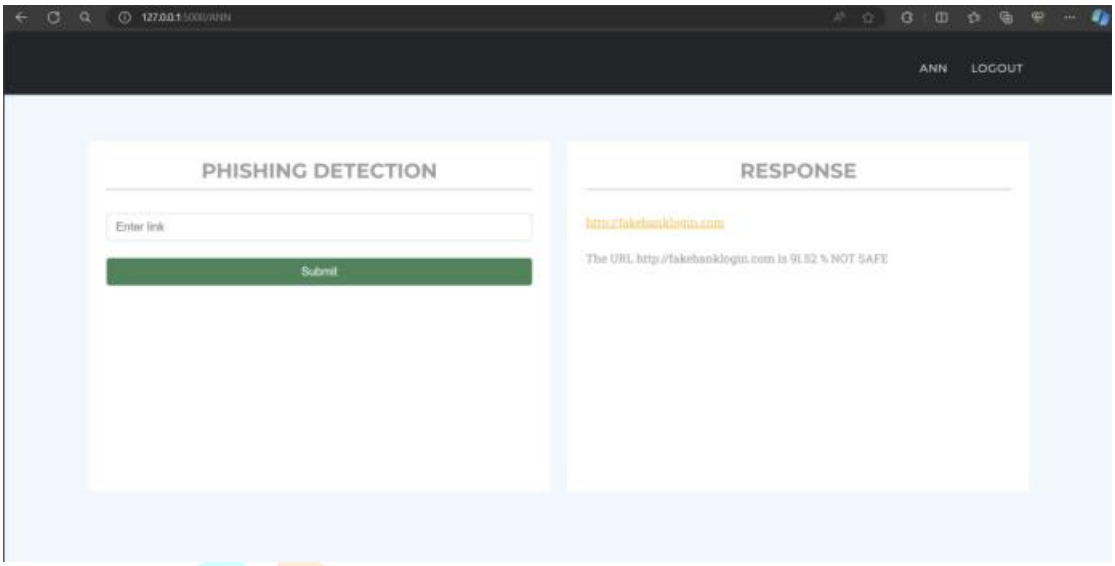

Fig: Detection of phishing website.

The is graphical user interface related to phishing detection system, The interface includes options for entering a link and checking its safety, with the example URL " http://fakebanklogin.com/" shown to be 91.52% not safe, hence advising users not to visit the URL.

We employ typical statistical metrics with accuracy, recall, precision, and validation loss to assess if a deep learning model performs well. Using the ANN approach for continuous testing, we get a validation loss of 0.071. A smaller validation loss denotes higher performance since it shows the model is more adept at reducing errors in its predictions made on unobserved data. The accuracy of 90.97% is essential for assessing the model's overall performance in accurately categorizing reputable and fraudulent websites. A recall of 1.0 indicates that all phishing URLs are accurately identified by the algorithm, with none being missed. A precision of roughly 97.66% indicates how well the model can prevent phishing websites from being mistakenly classified as legitimate websites.

VI. CONCLUSION AND FUTURE SCOPE

To sum up, the Fresh-Phish project is a noteworthy development in the field of phishing prevention and detection. The system can distinguish between secure and phishing websites with accuracy because to its use of advanced feature extraction techniques, machine learning, and artificial neural network algorithms. The creation of an extensible and open-source solution such as Fresh-Phish tackles the increasing risk of phishing assaults and offers a useful resource for cybersecurity practitioners and researchers alike.

The project's modular architecture, which consists of parts like the ANN Module, Feature Extraction Module, and User Interface Module, shows how methodical and thorough the approach to phishing detection is. When

these elements are combined, parsing URLs, extracting pertinent characteristics, and classifying websites may be done quickly and effectively, which eventually results in users receiving alert alerts on time.

The efficacy and usability of the Fresh-Phish project could be improved in the future in a number of ways. To further increase the accuracy of phishing detection, one possible improvement is the inclusion of more sophisticated machine learning and ANN techniques, such as deep learning models. Larger and more varied datasets may be used to train these algorithms so they could recognize the subtle patterns and traits of phishing websites. Furthermore, improving the user interface and user experience could enhance the overall usability of the system. This could involve implementing more interactive and intuitive features, such as real-time feedback on the analysis process and personalized recommendations for users. Overall, these enhancements could make the Fresh-Phish project even more effective in combating phishing attacks and protecting users' online security.

VII. ACKNOWLEDGMENT

We extend our heartfelt gratitude to Mr. Arun P, Assistant Professor in the Department of Computer Science and Engineering at CITech, for his invaluable guidance and impressive technical insights that significantly contributed to the successful completion of our project. Additionally, we would like to express our deep appreciation to our friends and teachers who provided assistance in various technical aspects, enriching our project with their expertise and feedback. Lastly, we acknowledge our parents for their unwavering support and encouragement throughout this journey, serving as a constant source of strength and motivation.

REFERENCES

[1] Suleiman Y. Yerima, Mohammed K. Alzaylaee,” High Accuracy Phishing Detection Based on Convolutional Neural Networks”2020

[2] Sumitra Das Guptta, Khandaker Tayef Shahriar, Hamed Alqahtani, Dheyaaldin Alsalman ,Iqbal H. Sarker1,” Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques”, October 2021

[3] Saad Al-Ahmadi , (Senior Member, IEEE), Afrah Alotaibi ,Omar Alsaleh,” PDGAN:Phishing Detection With Generative Adversarial Networks”.

[4] Saad Al-Ahmadi , (Senior Member, IEEE), Afrah Alotaibi ,Omar Alsaleh,” PDGAN:Phishing Detection With Generative Adversarial Networks”.

[5]Subhash Ariyadasa, Shantha Fernando, Subha Fernando, (Member, IEEE),”Combining Long-Term Recurrent Convolutional and Graph Convolutional Networksto Detect Phishing Sites Using URL and HTML**”.**

[6] May Almousa1,Mohd Anwar, (Senior Member, IEEE),” A URL-Based Social Semantic Attacks Detection With Character-Aware Language Model”.

[7] Hossein Shirazi , Shashika R. Muramudalige , Indrakshi Ray , Senior Member, IEEE,Anura P. Jayasumana ,Member, IEEE, and Haonan Wang," Adversarial Autoencoder Data Synthesis for Enhancing Machine Learning-Based Phishing Detection Algorithms".

[8] Felipe Castaño , Eduardo Fidalgo Fernañdez , Rocío Alaiz-Rodríguez , And Enrique Alegre," PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification".

[9] Guang-Gang Geng, Zhi-Wei Yan China, Jong-Hyouk Lee," An Efficient Anti-phishing Method to Secure eConsume".

[10] Rizka Widyarini Purwanto , Graduate Student Member, IEEE, Arindam Pal ,Senior Member, IEEE, Alan Blair, and Sanjay Jha , Senior Member, IEEE,"PHISHSIM: Aiding Phishing Website Detection With a Feature-Free Tool".