Anonymity and Confidentiality in Website using ML

under supervision of

Mr. Umashankar Sharma

Aditya Mishra (2101321530003)

Anshuman Soni (2101321530007)

Anuj Mishra (2101321530008)

Shivam Singh (2101321530045)

Project title Justification

Phishing is a growing cybersecurity threat where attackers impersonate legitimate websites to steal sensitive information like passwords, credit card numbers, or personal identification data. Traditional methods of detecting phishing websites rely heavily on blacklists or rule-based systems, which often fail to detect new phishing websites in real-time. Machine learning (ML) provides an effective solution as it can detect patterns and characteristics of phishing websites dynamically. By leveraging machine learning, the system can learn from large datasets, identify phishing sites more accurately, and evolve as attackers modify their tactics. This project aims to improve the detection rate and reduce false positives, contributing to a safer online environment.

Objective of the Project

Develop a system to detect Anonymity and secure Confendentiality in website using Phising Website Detection System: Build an ML model capable of identifying phishing websites based on website features (e.g., URL structure, website content, domain age).

1.Evaluate Different Machine Learning Algorithms: Compare the performance of different algorithms (such as Decision Trees, Random Forest, Support Vector Machines, and Neural Networks) to find the most efficient model.

2.Feature Selection and Engineering: Identify key features of phishing websites to improve the model's performance.

3.Implementation of a Real-Time System: Integrate the ML model into a system that can perform real-time phishing detection on websites.

Evaluate Accuracy and Efficiency: Measure the system’s accuracy, sensitivity, and false-positive rate to ensure high performance.

Data Collection

1. Data Collection:

Phishing Website Data: Gather datasets from publicly available sources such as PhishTank, UCI repository, or Kaggle, which contain records of both phishing and legitimate websites.

Legitimate Website Data: Collect data from trustworthy sources for a balanced comparison with phishing websites.

2. Data sets:

- The set of phishing URLs are collected from opensource service called **PhishTank**. This service provide a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly. To download the data: https://www.phishtank.com/developer_info.php. From this dataset, 5000 random phishing URLs are collected to train the ML models.
- The legitimate URLs are obatined from the open datasets of the University of New Brunswick, https://www.unb.ca/cic/datasets/url-2016.html. This dataset has a collection of benign, spam, phishing, malware & defacement URLs. Out of all these types, the benign url dataset is considered for this project. From this dataset, 5000 random legitimate URLs are collected to train the ML models.

Phishing_sites.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Domain	Having_@_	Having_IP	Path	Prefix_suff	Protocol	Redirectio	Sub_doma	URL_Lengt	age_domai	dns_recor	domain_re	http_token	label	statistical_	tiny_url	web_traffic
2	asesoresvelfit.com	0	0	/media/datacredito.co/	0	http	0	0	0	0	0	1	0	1	0	1	1
3	caixa.com.br.fgtsagendesaqueconta.com	0	0	/consulta8523211/principal.php	0	http	0	1	1	0	0	1	0	1	1	0	1
4	hissoulreason.com	0	0	/js/homepage/home/	0	http	0	0	0	0	0	1	0	1	0	0	1
5	unauthorizd.newebpage.com	0	0	/webapps/66fbf/	0	http	0	0	0	0	0	1	0	1	1	0	1
6	133.130.103.10	0	1	/23/	0	http	0	2	0	1	0	1	0	1	0	0	1
7	dj00.co.vu	1	0	/css/	0	http	0	0	2	1	1	1	0	1	1	0	0
8	133.130.103.10	0	1	/21/logar/	0	http	0	2	0	1	0	1	0	1	0	0	1
9	httpssicredi.esy.es	0	0	/servico/sicredi/validarclientes/mobi/index.php	0	http	0	2	2	1	1	1	1	1	1	0	1
10	gamesaty.ga	0	0	/wp-content///yh/en/	0	http	1	0	2	1	0	1	0	1	0	0	1
11	luxuryupgradepro.com	0	0	/ymailNew/ymailNew/	0	http	0	0	0	0	0	1	0	1	0	0	1
12	133.130.103.10	0	1	/1/	0	http	0	2	0	1	0	1	0	1	0	0	1
13	133.130.103.10	0	1	/24/sicredi/psmlld/31/paneid/index.htm	0	http	0	1	2	1	0	1	0	1	0	0	1
14	smscaixaacesso.hol.es	0	0		0	http	0	0	0	1	1	1	0	1	1	0	1
15	133.130.103.10	0	1	/7/SIIBC/siwinCtrl.php	0	http	0	1	0	1	0	1	0	1	0	0	1
16	tinyurl.com	0	0	/kjmmw57	0	http	0	0	0	0	0	0	0	1	0	1	0
17	wrightlandscapes.org	0	0	/no/T/Y1.html	0	http	0	0	0	1	0	1	0	1	1	0	1
18	mautic.eto-cms.ru	0	0	/themes/goldstar/mtbonline/newmandt/	1	http	0	0	2	0	0	1	0	1	0	0	1
19	ginatringali.com	0	0	//al/alibaba21012015/alibaba21012015/666/in	0	http	1	0	2	0	0	0	0	1	1	0	1
20	staticmail.000webhostapp.com	0	0	/	0	https	0	0	0	0	0	0	0	1	0	0	0
21	umeda.com.br	0	0	/bba/BOA/home/	0	http	0	0	0	1	0	1	0	1	1	0	1
22	krishworldwide.com	0	0	/BackUp/under/js/ayo1/index.html	0	http	0	0	2	0	0	1	0	1	0	0	1
23	yahoo.co.in	0	0	/email_open_log_pic.php	0	http	0	2	1	1	0	1	0	1	0	0	2
24	www.avcc.ac.in	0	0	/fonts/1/wropboxp/login.html	0	http	0	1	0	1	0	1	0	1	0	0	2

phishing-urls

Legitimate_sites.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Domain	Having_@_	Having_IP	Path	Prefix_suff	Protocol	Redirectio	Sub_doma	URL_Lengt	age_domai	dns_recor	domain_re	http_token	label	statistical_	tiny_url	web_traffic		
2	www.liquidgeneration.com	0	0	/	0	http	0	0	0	0	0	1	0	0	0	0	2		
3	www.onlineanime.org	0	0	/	0	http	0	0	0	0	0	1	0	0	1	0	1		
4	www.ceres.dti.ne.jp	0	0	/~nekoi/senno/senfirst.html	0	http	0	1	0	1	0	1	0	0	0	0	0		
5	www.galeon.com	0	0	/kmh/	0	http	0	0	0	0	0	0	0	0	0	0	0		
6	www.fanworkrecs.com	0	0	/	0	http	0	0	0	1	1	1	0	0	1	0	1		
7	www.animehouse.com	0	0	/	0	http	0	0	0	0	0	1	0	0	1	0	1		
8	www2.117.ne.jp	0	0	/~mb1996ax/enadc.html	0	http	0	1	0	1	0	1	0	0	0	0	2		
9	archive.rhps.org	0	0	/fritters/yui/index.html	0	http	0	2	0	0	0	1	0	0	0	0	2		
10	www.freecartoonsex.com	0	0	/	0	http	0	0	0	0	0	1	0	0	0	1	2		
11	www.cutepet.org	0	0	/	0	http	0	0	0	2	0	0	0	0	0	0	2		
12	www.taremeparadise.com	0	0	/	0	http	0	0	0	2	0	2	0	0	0	0	2		
13	www.internetdump.com	0	0	/users/pornographite/index1.html	0	http	0	2	2	0	0	1	0	0	0	0	1		
14	darkkaminari.net	0	0		0	http	0	0	0	1	1	1	0	0	1	0	1		
15	www.iei.net	0	0	/~bkos1/velneko.htm	0	http	0	2	0	0	0	1	0	0	1	0	1		
16	www9.kinghost.com	0	0	/fetish/hentaibee/	0	http	0	0	0	2	0	2	0	0	0	1	0		
17	www.jasonmeador.com	0	0	/	0	http	0	0	0	0	0	1	0	0	0	0	1		
18	www.geocities.com	0	0	/kaseychan17/index.html	0	http	0	2	0	2	0	2	0	0	0	0	2		
19	www.angelfire.com	0	0	/journal/coldlemonade/index.html	0	http	0	2	2	0	0	0	0	0	0	0	0		
20	e.webring.com	0	0	/hub	0	http	0	0	2	0	0	0	0	0	0	0	2		
21	www.nemurokinenkan.net	0	0		0	http	0	0	0	1	1	1	0	0	1	0	1		
22	j-heaven.tripod.com	0	0	/library.htm	1	http	0	2	0	0	0	0	0	0	0	0	1		
23	www.angelfire.com	0	0	/poetry/nicolesstories/	0	http	0	0	0	0	0	0	0	0	0	0	0		
24	thesheeparecoming.tripod.com	0	0	/papercrane/	0	http	0	0	0	0	0	0	0	0	0	0	1		

legitimate-urls

Feature Extraction

Feature Extraction consists of:

Extract significant features from website URLs and webpage content that differentiate phishing websites from legitimate ones.

Key features to consider:

- URL-based features: Length of URL, presence of special characters, use of IP address instead of a domain name, presence of suspicious keywords.
- Domain-based features: Domain age, domain name length, registration information, presence of SSL certificates.
- Page-based features: Website content, use of HTTPS, form handling, third-party resources.

The below mentioned category of features are extracted from the URL data:

1. Address Bar based Features

   In this category 9 features are extracted.
2. Domain based Features

   In this category 4 features are extracted.
3. HTML & Javascript based Features

   In this category 4 features are extracted.

So, all together 17 features are extracted from the 10,000 URL dataset.

Feature Extraction

1. Address Bar based Features considered are:

- Domian of URL
- Redirection '//' in URL
- IP Address in URL
- 'http/https' in Domain name
- '@' Symbol in URL
- Using URL Shortening Service
- Length of URL
- Prefix or Suffix "-" in Domain
- Depth of URL

2. Domain based Features considered are:

- DNS Record
- Website Traffic
- Age of DomainEnd
- Period of Domain

3. HTML and JavaScript based Features considered are:

- Iframe Redirection
- Status Bar Customization
- Disabling Right Click
- Website Forwarding

Feature Extraction for Phishing Website Detection

1. Using URL Shortening Services
•**Objective**: Identify phishing attempts using shortened URLs.
•**Method**: Detect URL shortening services like "bit.ly" or "tinyurl.com" through keyword matching. These services obscure the true destination, often used by attackers.
2. Existence of HTTPS Token
•**Objective**: Avoid misleading security indicators in the domain.
•**Method**: Flag URLs containing “HTTPS” in places other than the protocol (e.g., "httpssecure.com").
3. Abnormal URL
•**Objective**: Validate the legitimacy of domains.
•**Method**: Cross-check domains against WHOIS databases to verify registration details. Abnormal or unregistered domains raise suspicion.
4. Google Index
•**Objective**: Confirm the URL is indexed by search engines.
•**Method**: Use Google search APIs to verify if the URL exists in Google's database. Non-indexed sites are more likely to be phishing.
5. Website Traffic
•**Objective**: Assess website popularity.
•**Method**: Analyze Alexa rank data. Legitimate sites usually have higher traffic, while phishing sites rank low or are absent.

6. Domain Registration Length

- **Objective**: Detect short-lived domains.
- **Method**: Phishing domains often register for under a year. WHOIS checks identify short registration durations.

7. Age of Domain

- **Objective**: Identify newly created domains.
- **Method**: WHOIS records reveal the domain's age. New domains are a common indicator of phishing attempts.

8. DNS Record

- **Objective**: Verify domain validity.
- **Method**: DNS queries ensure the existence of legitimate records. Missing or invalid records suggest phishing activity.

9. Statistical Report

- **Objective**: Leverage threat intelligence databases.
- **Method**: Match URLs and IPs against lists of known phishing entities. Regularly updated datasets enhance accuracy.

10. Long URLs

- **Objective**: Detect unusually long URLs.
- **Method**: URLs exceeding a standard character limit (e.g., >75 characters) are flagged as suspicious.

11. @ Symbol in URL

- **Objective**: Identify domain obfuscation techniques.
- **Method**: The "@" symbol redirects users to different pages and is typically used in phishing URLs.

12. Double Slashes (//) in Path

•**Objective**: Detect abnormal URL formatting.

•**Method**: URLs with "//" in unexpected places (other than "http://") are flagged as suspicious.

13. Subdomains

•**Objective**: Identify excessive subdomains.

•**Method**: Count the number of dots in the URL. Phishing URLs often use multiple subdomains to confuse users (e.g., "login.bank.com.fake.com").

14. IP Address Usage

•**Objective**: Detect direct IP addresses in URLs.

•**Method**: URLs using raw IPs instead of domain names are flagged, as legitimate sites typically avoid this practice.

Implementation

The script is designated to detect phishing websites by extracting specific features from URLs. The key steps involved:

1.**Feature Extraction Class**:

- The class FeatureExtraction defines methods for various URL characteristics, such as:
  - **Protocol** (e.g., HTTP, HTTPS)
  - **Domain**: Checks if the domain is an IP address or a typical domain name.
  - **URL Length**: Longer URLs are often used in phishing sites.
  - **Subdomains**: More than three subdomains might indicate a phishing attempt.
  - **Redirection and Symbols**: Phishing sites often use tricks like @ symbols or redirect symbols (//).
  - **Tiny URLs**: Shortened URLs often lead to phishing sites.

2.**WHOIS Lookup**: The script uses the whois library to fetch domain registration details such as the domain's age and expiration date, which help assess if a website is legitimate. Short-lived or recently created domains are red flags.

3.**Web Traffic Analysis**: It checks the popularity of a site based on its Alexa ranking, as legitimate sites tend to have higher traffic.

4.**CSV File Generation**: After processing each URL from a file, the script stores the extracted features in a DataFrame and saves the data into a CSV file labeled as "phishing-urls.csv."

5.**Labeling**: The URLs are labeled based on the extracted features, helping in the classification of the website as **legitimate**, **phishing**, or **suspicious**.

This approach can be used to train machine learning models to automate phishing detection.

1. Data Collection:
•Collect URL data from different sources, including legitimate and phishing websites. Files like legitimate_urls.txt and 1000-phishing.txt provide the URLs used for training.
2. Feature Extraction (Class FeatureExtraction):
•**Protocol**: Extract whether the URL uses "http" or "https".
•**Domain**: Extract the domain name of the URL.
•**Path**: Extract the path from the URL.
•**Having IP**: Check if the domain contains an IP address.
•**URL Length**: Measure the length of the URL to classify it as suspicious or legitimate.
•**@ Symbol**: Check for the presence of "@" in the URL.
•**Redirection**: Detect if the URL contains a "//" after the protocol.
•**Subdomains**: Count the number of subdomains in the URL.
•**Shortening Services**: Detect URLs shortened using services like bit.ly, goo.gl.
•**Web Traffic**: Use Alexa rank to check the popularity of the domain.
•**Domain Registration Length**: Extract domain registration and compare the validity.
•**Age of Domain**: Calculate the age of the domain based on registration date.
•**DNS Records**: Check DNS information for additional classification.
•**Statistical Report**: Conduct a statistical analysis on the URL to detect anomalies.
•**HTTPS Token**: Check if the URL starts with "https://" for security validation.
3. Data Labeling:
•URLs are labeled as:
 •0: Legitimate
 •1: Phishing
 •2: Suspicious
4. Model Training:
•Use RandomForestClassifier or DecisionTreeClassifier for the model to classify URLs as phishing or legitimate based on extracted features.
5. Web Interface:
•The app.py file likely serves the model using a web framework (e.g., Flask) and includes UI files like home.html, _navbar.html, and about.html for user interaction.
6. Model Saving and Loading:
•After training, save the model to a file like RandomForestModel.sav for future use.
7. Prediction:
•For new URLs, the system extracts features, processes them, and uses the trained model to classify the URL as phishing or legitimate.

METHODOLOGY

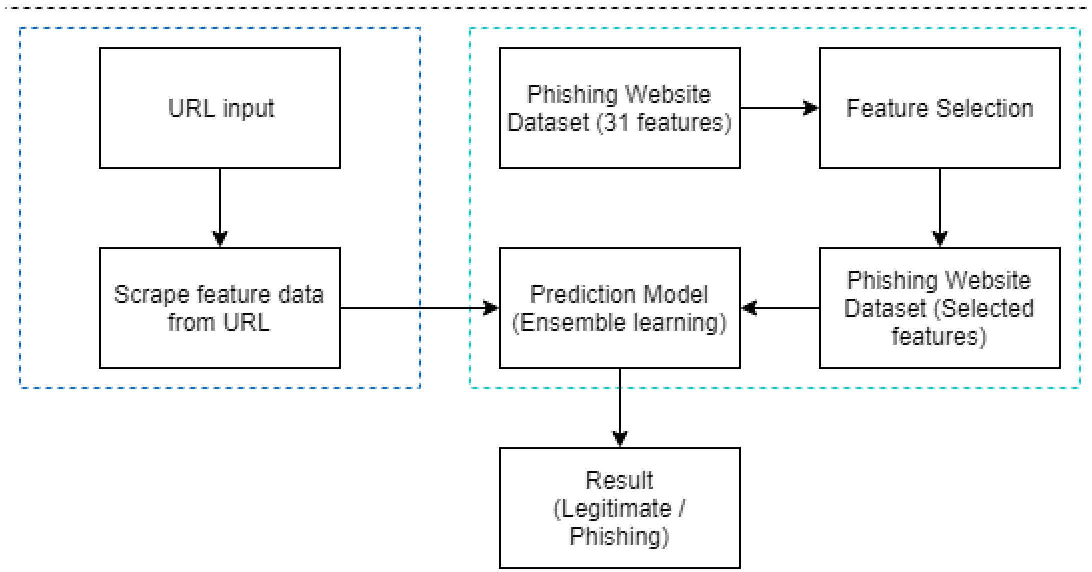

Fig. Flowchart of the Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning

Technologies Used:

Detecting phishing websites using machine learning algorithms typically involves using supervised learning techniques. Some common machine learning algorithms and methods used for phishing website detection include:

1. Decision Trees: Decision trees can be effective for phishing detection as they can capture complex relationships between features. Ensemble methods like Random Forests and Gradient Boosting Machines (GBM) can also be used for improved performance.

2. Random Forest: Random Forests are a powerful ensemble learning algorithm that combines multiple decision trees to improve accuracy and robustness. They are particularly effective for handling high-dimensional data and are widely used in various applications, including classification and regression tasks.

METHODOLOGY

Before stating the ML model training, the data is split into 70-30 i.e., 7000 training samples & 3000 testing samples. From the dataset, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0). The supervised machine learning models (classification) considered to train the dataset in this project are:

- Decision Tree
- Random Forest
- Multilayer Perceptrons
- XGBoost
- Autoencoder Neural Network
- Support Vector Machines

All these models are trained on the dataset and evaluation of the model is done with the test dataset.

Testing

In the **Phishing Website Detection** project, testing the model involves the following steps:

1. **Data Splitting**: The dataset is divided into training and testing sets, typically using an 80-20 or 70-30 ratio. This ensures that the model is evaluated on unseen data.
2. **Model Evaluation**: After training the classifier (e.g., RandomForest or DecisionTree), the model is tested using the testing dataset. Performance is evaluated using metrics such as:
    - Accuracy
    - Precision
    - Recall
    - F1-Score
3. **Cross-validation**: To ensure robustness, cross-validation techniques like k-fold cross-validation may be applied to assess the model's performance across multiple subsets of the data.
4. **Confusion Matrix**: A confusion matrix helps visualize the classification results, showing true positives, false positives, true negatives, and false negatives.
5. **Performance Visualization**: The performance metrics, such as accuracy and F1-score, are plotted to assess how well the model is generalizing.
6. **Model Tuning**: Hyperparameters of the classifier can be adjusted (e.g., number of trees in Random Forest) to improve the model's performance on the test set.

Conclusion related to Project

The project focuses on detecting phishing websites using a Random Forest Classifier. It employs 14 feature extraction methods, such as URL length, traffic ranking, and subdomain analysis. After merging the legitimate and phishing URL datasets, the data is shuffled to avoid overfitting and split into training (70%) and testing (30%) datasets. Feature importance analysis shows that URL length and web traffic are the most significant predictors. A bar plot is used to illustrate the ranked feature importances, providing a clearer understanding of their influence on phishing detection.

LITERATURE SURVEY

[1] Marwa Abd Al Hussein Qasim, Dr. Nahla Abbas Flayh : Using six criteria based on URL parameters such as the subdomain, principal domain, Page rank, Alexa rank, path domain, and Alexa reputation, this article suggests a novel method for identifying phishing websites. The method focuses on evaluating how closely a phishing site's URL resembles the URL of a reliable website and also takes into account the site's ranking as a crucial component in determining its validity. The approach was tested using data from PhishTank and DMOZ, and the authors showed that it could identify over 97% of phishing sites.

Key findings :-

- Subdomain and Principal Domain: Analyzing these components helps in identifying suspicious patterns that are common in phishing attempts.
- PageRank: The study found that about 95% of phishing sites have no PageRank, suggesting that low or absent PageRank values can be indicative of phishing activity.
- Alexa Rank: Websites with an Alexa rank greater than 100,000 were classified as suspicious or phishing, while those ranked lower were deemed legitimate.
- Path Domain and Alexa Reputation: These elements further refine the classification process by evaluating the trustworthiness associated with the URL path and overall site reputation.

2Irfan Siddavatam, Rishikesh Mahajan : Introduced the use of decision trees , SVM, Random Forest to classify phishing websites based on features like URL length, domain age, presence of special characters, and security certificate information. Their approach demonstrated improved detection rates compared to traditional methods but faced challenges in identifying phishing sites.

Key findings :-

- URL Length: Longer URLs often indicate phishing attempts.
- Domain Age: Newly registered domains are frequently associated with phishing.
- Special Characters: The presence of unusual characters can signal malicious intent.
- Security Certificate Information: Lack of valid SSL certificates is a common trait of phishing sites.

LITERATURE SURVEY

[13] Ankit Kumar Jain & B. B. Gupta : The proposed strategy utilizes an Innovative methodology for defending counteract phishing attempts by incorporating a URL and DNS matching module with a white list of trusted websites that are automatically up-dated based on each user's browsing history. This method offers quick retrieval speeds, high rates of detection, and alerts users to not disclose personal information when at-tempting to access a website, not on the white list. Additionally, hyperlink properties are utilized to verify the validity of a website by retrieving hyperlinks from the source code and applying them to the phishing detection method. The performance of this strategy was evaluated using data from reputable sources such as Stuffgate, Alexa, and PhishTank and achieved an accuracy rate of 89.38 %.

Key findngs :-

- Detection and Alert Mechanism:
    - The system offers quick retrieval speeds and high detection rates of phishing attempts. Users are alerted not to disclose personal information when attempting to access websites that are not on the whitelist, providing an additional layer of protection.
- Utilization of Hyperlink Properties:
    - The method leverages hyperlink properties by retrieving hyperlinks from the source code of websites. This information is then applied to the phishing detection process, further validating the legitimacy of a site.
- Performance Evaluation:
    - The performance of this strategy was rigorously evaluated using data from reputable sources such as Stuffgate, Alexa, and PhishTank. The approach achieved an accuracy rate of 89.38%, indicating its effectiveness in identifying phishing sites.

[14] M. Aydin and N. Baykal : Throughout this experiment, phishing websites were detected using subset-based feature selection methods based on URL attributes. A da-taset comprising both legitimate and phishing URLs was obtained from Google and PhishTank, and multiple features were retrieved from URLs. The usefulness of two classification algorithms—Naive Bayes and Sequential Minimal Optimization (SMO)—for identifying phishing websites was investigated in this study. The results showed that SMO performed bette r than Naive Bayes for phishing detection, with an accuracy rate of 95.39%. The SMO algorithm also had another benefit in that it made use of more chosen features overall. The accuracy rate of the Naive Bayes method, in contrast, was 88.17% while using the same quantity of chosen features.

Key findings :-

- Performance Results:
    - The results indicated that the SMO algorithm outperformed Naive Bayes, achieving an impressive accuracy rate of 95.39% in detecting phishing websites.
    - In contrast, the Naive Bayes method yielded an accuracy rate of 88.17%, despite utilizing the same number of selected features.
- Feature Selection Benefits:
    - The SMO algorithm demonstrated a significant advantage by employing a larger number of selected features overall, which contributed to its superior performance in distinguishing between phishing and legitimate websites.

REFERENCES

1Marwa Abd Al Hussein Qasim, Dr. Nahla Abbas Flayh, “Phishing Website Detection Using Machine Learning: A Review” June 2023Wasit Journal of Pure sciences 2(2):270-2812(2):270-281

2 Rishikesh Mahajan, Irfan Siddavatam ,”Phishing Website Detection using Machine Learning Algorithms” , International Journal of Computer Applications, Volume 181 - Number 23 Year of Publication: 2018

3S. A. Anwekar and V. Agrawal, "PHISHING WEBSITE DETECTION USING MACHINE LEARNING ALGORITHMS."

4 S. Jain, "Phishing Websites Detection Using Machine Learning," Available at SSRN 4121102.

5A. Lakshmanarao, P. S. P. Rao, and M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021: IEEE, pp. 1164-1169.

6L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Web-site Detection," IEEE Access, vol. 10, pp. 1509-1521, 2021.

7 A. D. Kulkarni and L. L. Brown III, "Phishing websites detection using machine learn-ing," 2019.

8I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur, "A novel machine learning ap-proach to detect phishing websites," in 2018 5th International conference on signal pro-cessing and integrated networks (SPIN), 2018: IEEE, pp. 425-430.

REFERENCES

9M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," in 2018 6th International Symposium on Digital Forensic and Security (ISDFS), 2018: IEEE, pp. 1-5.

10X. Zhang, Y. Zeng, X.-B. Jin, Z.-W. Yan, and G.-G. Geng, "Boosting the phishing detection performance by semantic analysis," in 2017 IEEE international conference on big data (big data), 2017: IEEE, pp. 1063-1070.

11W. Fadheel, M. Abusharkh, and I. Abdel-Qader, "On Feature selection for the prediction of phishing websites," in 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017: IEEE, pp. 871-876.

12A. A. Ahmed and N. A. Abdullah, "Real time detection of phishing websites," in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Con-ference (IEMCON), 2016: IEEE, pp. 1-6.

13A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list," EURASIP Journal on Information Security, vol. 2016, no. 1, pp. 1-11, 2016.

14M. Aydin and N. Baykal, "Feature extraction and classification phishing websites based on URL," in 2015 IEEE Conference on Communications and Network Security (CNS), 2015: IEEE, pp. 769-770.

Thank You !