I. J. Computer Network and Information Security, **2025, 1, 39-56**
Published Online on February 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2025.01.04

Enhancing Web Security through Machine Learning-based Detection of Phishing Websites

Najla Odeh*
Palestine Technical University Kadoorie / Computer Science Department, Faculty of Information Technology, Tulkarm, P.O Box 305, Palestine
E-mail: najlaa.odeh@ptuk.edu.ps
ORCID iD: https://orcid.org/0000-0003-1089-9243
*Corresponding Author

Derar Eleyan
Palestine Technical University Kadoorie / Computer Science Department, Faculty of Information Technology, Tulkarm, P.O Box 305, Palestine
E-mail: d.eleyan@ptuk.edu.ps
ORCID iD: https://orcid.org/0000-0001-8876-7019

Amna Eleyan
Manchester Metropolitan University / Department of Computing and Mathematics, Manchester M15 6BH, United Kingdom
E-mail: a.eleyan@mmu.ac.uk
ORCID iD: https://orcid.org/0000-0002-2025-3027

Received: 04 July 2023; Revised: 30 October 2023; Accepted: 22 December 2023; Published: 08 February 2025

Abstract: The rise of cyberattacks has led to an increase in the creation of fake websites by attackers, who use these sites for advertising products, transmit malware, or steal valuable login credentials. Phishing, the act of soliciting sensitive information from users by masquerading as a trustworthy entity, is a common technique used by attackers to achieve their goals. Spoofed websites and email spoofing are often used in phishing attacks, with spoofed emails redirecting users to phishing websites in order to trick them into revealing their personal information. Traditional solutions for detecting phishing websites rely on signature-based approaches that are not effective in detecting newly created spoofed websites. To address this challenge, researchers have been exploring machine-learning methods for detecting phishing websites. In this paper, we suggest a new approach that combines the use of blacklists and machine learning techniques such that a variety of powerful features, including domain-based features, abnormal features, and abnormal features based on URLs, HTML, and JavaScript, to rank web pages and improve classification accuracy. Our experimental results show that using the proposed approach, the random forest classifier offers the best accuracy of 93%, with FPR and FNR as 0.12 and 0.02, with a Precision of 90%, Recall of 97% an F1 Score of 93%, and MCC of 0.85.

Index Terms: Web Security, Phishing, Machine Learning, Cyberattacks, Fake Websites, Blacklists.

1. Introduction

The rapid expansion and accessibility of the Internet have led to a shift in consumer behavior from traditional shopping to e-commerce. However, this shift has also brought about an increase in cyberattacks, with attackers employing strategies such as phishing to obtain sensitive credentials from unsuspecting users. Phishing is a method of identity theft that deceives individuals into disclosing confidential information on fraudulent websites, including credit card numbers, passwords, and personal details [1].

Phishing attacks have garnered significant attention from security researchers due to their potential to exploit users by creating counterfeit websites that mimic legitimate ones. Despite users' ability to identify these fake pages by scrutinizing URLs, the busy nature of online activities sometimes causes them to overlook such distinctions, making them susceptible to falling into the attackers' traps. The repercussions of phishing attacks extend beyond individual users, affecting online shopping trust and causing financial losses to both businesses and individuals. Worryingly, phishing serves as the initial step in 90% of reported cyberattacks, as indicated by Verizon's Data Breach Study Report [2]. in

This work is open access and licensed under the Creative Commons CC BY 4.0.

report by APWG [3] 262,704 phishing attacks were reported in Q1 2018, up from 233,613 in Q4 2017. Fig. 1. shows the 2019 Phishing Report for the third quarter.

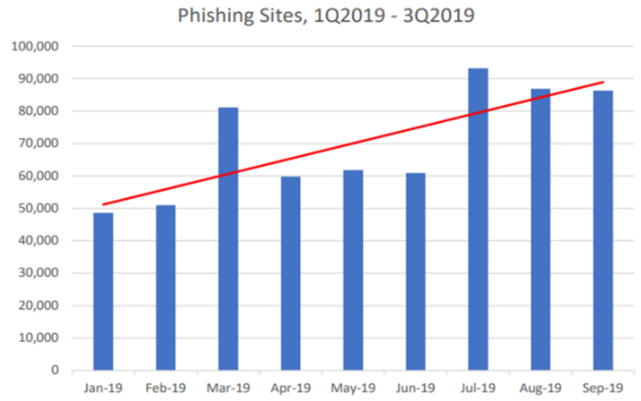


Fig.1. Phishing report for the third quarter of the year 2019 [4]

In modern phishing attacks, the aim extends beyond the acquisition of login credentials to encompass the infection of victims' computers with various forms of malware [5, 6]. These attacks employ multiple channels, including the internet, SMS, and phone. The internet serves as a medium for attackers to communicate with victims through emails, malicious websites, instant messaging, and social networking. Additionally, the prevalence of smart mobile devices has led attackers to adopt techniques such as smishing and vishing, which exploit SMS and voice communication [7, 8].

Criminals of phishing attacks use online tools such as HTTrack, Wget, Jsoup, and Selenium to construct fraudulent websites that initiate phishing attacks. Anti-phishing software often identifies these imitation websites by searching for identity components like keywords, copyright information, and anchor connections found on legitimate websites. To evade detection, attackers typically replicate the layout of legitimate login pages while redirecting victims to malicious web pages. Signature matching is a widely employed technique for identifying phishing websites, wherein each incoming request is compared against a database of blacklisted websites. PhishTank and Google Safe Browsing are two services that maintain such blacklists of websites associated with phishing and malware.

Several solutions have been developed to detect phishing websites and safeguard users from falling victim to these assaults. Among the strategies that have been used previously: the list-based strategy focuses on maintaining a catalog of known phishing websites and then cross-referencing the user's request with this list. If there is a match, the request is either denied or blocked. However, this method has disadvantages such as a high rate of false negatives, insufficient coverage, and significant maintenance expenses. It is also hampered by the constant emergence of new phishing sites, making it difficult to keep the list up to date [9-11].

Another way is the content-based strategy, which examines a web page's textual and visual content for phishing indicators such as false domain names, counterfeit logos, grammatical errors, or requests for sensitive information. Although capable of detecting novel phishing sites not previously identified, this strategy may generate false positives, necessitate large processing resources, and be vulnerable to evasion tactics [9-12]. Furthermore, distinguishing phishing websites from authentic ones might be difficult due to situations where phishing websites contain actual content [11]. The URL-based method investigates the structure and attributes of a web page's URL, such as its length, domain age, IP address, and the existence of special characters. While this strategy is excellent in detecting phishing websites quickly and efficiently based on URL features, it may be subject to obfuscation or redirection strategies. A heuristic-based technique involves assessing the validity of a web page using a set of established rules or criteria. These regulations could apply to SSL certificates, WHOIS information, or external links. Although this system captures some common phishing patterns and offers a score or rating to each web page, it may miss subtle or inventive phishing attempts [9, 13].

Phishing attacks target computer users for five key reasons [14]: 1. Users don't have a basic understanding of Uniform Resource Locators (URLs), 2. don't know which websites can be trusted, 3. don't know the exact location of the page due to redirection or hidden URLs, 4. the URL has many possible options, or some pages are accidentally entered, and 5. don't know how to distinguish a phishing website or a legitimate one.

In this study, we propose a comprehensive approach for detecting and preventing phishing websites that incorporates the use of blacklist techniques in addition to machine learning techniques. thereby enabling more accurate identification of such websites. Our primary objective is to develop a system with enhanced capability for detecting and categorizing phishing websites and then preventing them. We incorporate blacklist testing, drawing upon the GSB and PhishTank databases, and using machine learning algorithms, we extract thirty distinct features from websites, including domain-based attributes, abnormal features, HTML and JavaScript characteristics, and URL-based attributes. The proposed approach aims to bolster internet users' security and shield them from the compromise of their personal and financial information. To assess the effectiveness of our approach, we conduct a comparative analysis of seven machine learning models: Naive Bayes, Decision Tree, K-Nearest Neighbor, Support Vector Machine, Random Forest, Logistic Regression, and AdaBoost. This research holds the potential to advance website security and safeguard sensitive data from malicious attacks. We aspire for this study to raise awareness about privacy and security concerns and inspire further exploration in this field.

The remainder of this paper is structured as follows: Section 2 investigates into related work within the domain of phishing website detection. Section 3 presents our proposed method for detecting phishing websites using a machine learning algorithm, detailing the features utilized. Lastly, Section 4 showcases the experimental results and engages in a comprehensive discussion.

2. Review of Previous Studies

Various phishing website detection techniques utilizing machine learning algorithms and website features are examined in this section.

The authors [15] propose a framework (PhishTime) for tracking the efficacy of anti-phishing blacklists over time. The authors launch new phishing websites in a systematic manner to assess the performance and consistency of blacklists. The study identifies flaws in the anti-phishing ecosystem and makes recommendations to better protect users from modern phishing attacks.

Chiew and Chang [16] proposed a method that depends on the website logo. They extracted the logo and submitted it to Google's image search engine to learn the determination of a suspect website. They were able to determine the legitimacy of a website by contrasting the website with the search engine results.

A strategy for categorizing webpages based on website identity keywords was described by Tan and Chiew [17]. They gave this technique the moniker "PhishWho." Using the N-gram model, they were able to extract the keywords from the website. The aim website field is assessed using search engine results based on these keywords. The validity of a website is then determined by comparing these findings to the suspicious domains.

This paper [18] presents a content-agnostic method for predicting phishing domains based on data from Certificate Transparency Logs and passive DNS records. The study demonstrates the utility of this analysis by training a classifier with unique features and achieving low false positive rates as well as high precision and recall in predicting phishing domains.

The use of heuristics-based solutions for detecting phishing websites is discussed in this paper [19]. It investigates a variety of features, such as non-content-based, content-based, and visual similarity-based features, as well as DNS information and the registers used to register the site. To predict phishing websites, the paper proposes a general solution based on heuristics over these features.

Sinha [20] proposes a dataset with 198 features extracted from phishing websites as an empirical approach to phishing detection. The dataset is analyzed using machine learning and deep learning models, with Random Forest detecting phishing pages with high accuracy. The paper emphasizes the importance of capturing a diverse set of features in order to detect phishing effectively.

Bhattacharyya and others [21] describe PhishSaver, a desktop application that detects phishing attacks by combining blacklists and heuristic features. The application employs the Google Safe Browsing blacklist as well as several heuristics, including footer links, zero links in the body of HTML, copyright content, title content, and website identity. The paper emphasizes PhishSaver's ability to detect zero-hour phishing attacks and its faster performance when compared to visual-based assessment techniques.

The purpose of this paper [22] is to investigate the use of machine learning for phishing detection and to highlight the limitations of blacklists in detecting zero-day attacks. The research focuses on phishing URLs and domains associated with Italian organizations, and the models based on pre-trained encoders and Convolutional Neural Networks produce promising results.

Moghimi and Varjani [23] suggested a method for detecting phishing websites that combined SVM and a decision tree algorithm. The decision tree rules were used to detect phishing websites aimed at banking applications.

Mohammad, Thabtah, and McCluskey [24] conducted research on various features that can be used to distinguish between malicious and benign websites. They suggested a 17-feature rule-founded classification for detecting phishing websites.

Varshney, Misra, and Atrey [25] presented a method for detecting phishing websites that use the Google search engine. To create the search string, the domain of the suspicious URL is appended to the name of the website. The website's legitimacy is then determined by comparing this domain to the top search engine results.

On the login screen, some users have been noted to enter fictitious credentials before entering legitimate ones, according to Srinivasa Rao and Pais [26]. They suggested a method for detecting phishing assaults that automates this human tendency and use heuristic filtering.

Jain and Gupta [27] extracted 19 features from the website's URL and source code and used machine learning techniques from the WEKA toolkit, including SVM, Random Forest (RF), Neural Network, Logistic Regression, and Naive Bayes (NB).

A nonlinear regression technique is used by Babagoli, Aghababa, and Solouk [28] in recent work to identify phishing websites. For model training, they used SVM meta-heuristic methods and harmony search. They came to the conclusion that the harmony search yields more accurate findings.

A real-time system using classification methods and features based on natural language processing (NLP) was proposed by Sahingoz, and Buber [29]. According to experimental findings, the Random Forest classifier has the highest accuracy of 97% when employing NLP-based characteristics.

Doke, and Khismatrao [30] talk about how difficult it is to recognize phishing websites. In order to identify between authentic and phishing websites, the article suggests a novel approach based on machine learning algorithms that can extract different information. The suggested system is a web browser add-on that aids users in identifying phishing websites.

The research [31] suggests PhishHaven, an ensemble machine learning-based detection system that can distinguish between phishing URLs created by artificial intelligence and those created by humans. The system uses lexical analysis to extract features and to improve its speed, it introduces URL HTML Encrypting and a URL Hit method. The system uses multi-threading to carry out the classification in parallel, resulting in real-time detection. The final classification is made via an impartial voting method. The suggested technique is assessed theoretically and empirically and achieves good accuracy and precision in identifying both artificial intelligence (AI)-generated and straightforward phishing URLs. The system's drawback is that it can only identify AI-generated phishing URLs that resemble DeepPhish in terms of lexical attributes and pattern usage.

Machine learning techniques are being developed by researchers in [32] that make use of a variety of features broken down into a webpage, URLs, and HTML-based features. For the purpose of classification, it is suggested that all of these qualities be combined. The findings demonstrate that URL-based variables are the most useful for categorizing web pages, and the suggested method, which employs a random forest as the classifier, has an accuracy of 99.5% with false positive and false negative rates of 0.006 and 0.005, respectively.

This study [33] focuses on the issue of phishing attacks, which take advantage of human weaknesses to obtain consumers' personal information online. The researchers suggest a unique method for phishing identification that merely makes use of nine lexical elements based on URL characteristics to address this problem. The method is evaluated using 11,964 instances of valid and phishing URLs from the ISCXURL-2016 dataset. With the Random Forest algorithm, the method obtains an accuracy of 99.57 percent.

In [34] describes a new hybrid approach that uses six algorithm methods—blacklisted, lexical and host, content, identity, identity similarity, visual similarity, and behavioral—to identify and block phishing URLs. The study evaluates the performance of machine learning and deep learning models, such as CART, SVM, KNN, MLP, and CNN, in identifying phishing URLs by looking at 37 features retrieved from these methods. According to the study, the new hybrid solution is very effective and accurate at analyzing URL stress from a variety of angles, with the CNN model and MLP model both scoring higher accuracy levels of 97.945 and 93.216 respectively. The report advises putting into practice the new hybrid strategy to stop phishing attempts.

A technique for identifying phishing short URLs is put out by Xie, Li, and others [35]. When a user clicks on a short URL, the method uses a hierarchical hidden Markov model with a two-layer structure to represent the link-jumping process, which aids in the detection of phishing short URLs. The observation sequence's average log-likelihood probability is computed. The usefulness of the strategy is tested using actual Weibo datasets, and the results of the experiment support this claim.

The method for identifying phishing websites described in this research [36] uses URL analysis and includes login page URLs from both legitimate and phishing classes. When we analyzed several machine learning and deep learning techniques, we discovered that models developed with authentic login page URLs have a significant false-positive rate. Additionally, they examined the most recent phishing domains and produced a brand-new dataset called PILU-90K that contains up-to-date, authentic login URLs from the real world as well as phishing URLs. Their method used a logistic regression model with TF-IDF feature extraction and had a high accuracy rate of 96.50%. Their method provides a number of benefits, including independence from outside services, the ability to identify login websites, and the utilization of real-world, up-to-date statistics.

Based on earlier studies, it is evident that current methods for recognizing fraudulent websites only take into account a single sort of characteristic, making it challenging to recognize sophisticated phishing websites created by knowledgeable attackers. Therefore, it is important to choose the most efficient set of heuristic features that include both generic and obfuscation-based traits. In this paper, we present a method for detecting and preventing phishing websites by looking at their URL extracting a set of attributes, and then classifying them as authentic or phishing websites. Specifically, we rely on the use of blacklists such as Google Safe Browsing (GSB) and PhishTank to identify potentially malicious websites based on their URL. Then, in order to further assess the website and establish its authenticity, we extract a complete collection of 30 features, including domain-based features, anomalous features, HTML and JavaScript-based features, and features based on URLs.

We compared various machine learning models in order to assess the efficacy of our suggested solution. The performance of the Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), and Adaboost algorithms were specifically assessed. Our findings demonstrate that the suggested approach is effective in detecting phishing websites with an excellent percentage.

3. Proposed Method

This section presents the design of an approach for detecting and preventing phishing websites, as shown in Fig .2.

The approach involves three stages. In the first stage, the requested URL's hash is created and compared against a list of blacklisted domains in the Google Safe Browsing (GSB) database. If a match is found, the website is considered a phishing website. If not, the approach proceeds to the second stage, where the website is compared to the Phishtank database. A match in the Phishtank database indicates that the website is a phishing website. In the event that there is no match between the site link and the previous databases, the site can be considered safe initially, and because phishing websites are temporary and may not be updated in the databases, there is a possibility that some websites may remain phishing even if they appear to be legitimate. To solve this problem, we move to the third stage, which is to use machine learning algorithms to classify websites as legitimate or phishing.

The features that will be used to classify websites into legitimate or phishing websites are a mix of general and obfuscation features the layout is shown in Fig. 4. A total of 30 attributes were used, including page, URL, domain, and HTML features that are a mix of general and obfuscation characteristics. The following section depth into more descriptions of these suggested features. A feature vector is created by numerically representing all of the features extracted from each website. Each attribute generates a -1 or a 1, indicating phishing and legitimate websites respectively. The training cases are built to construct the seven machine-learning models. Experimental results show that the suggested approach, when using the random forest classifier, achieves an accuracy of 93%, with FPR and FNR values of 0.12 and 0.02. The approach also achieves high precision, recall, F1 score, and MCC, proving its usefulness in detecting phishing websites.

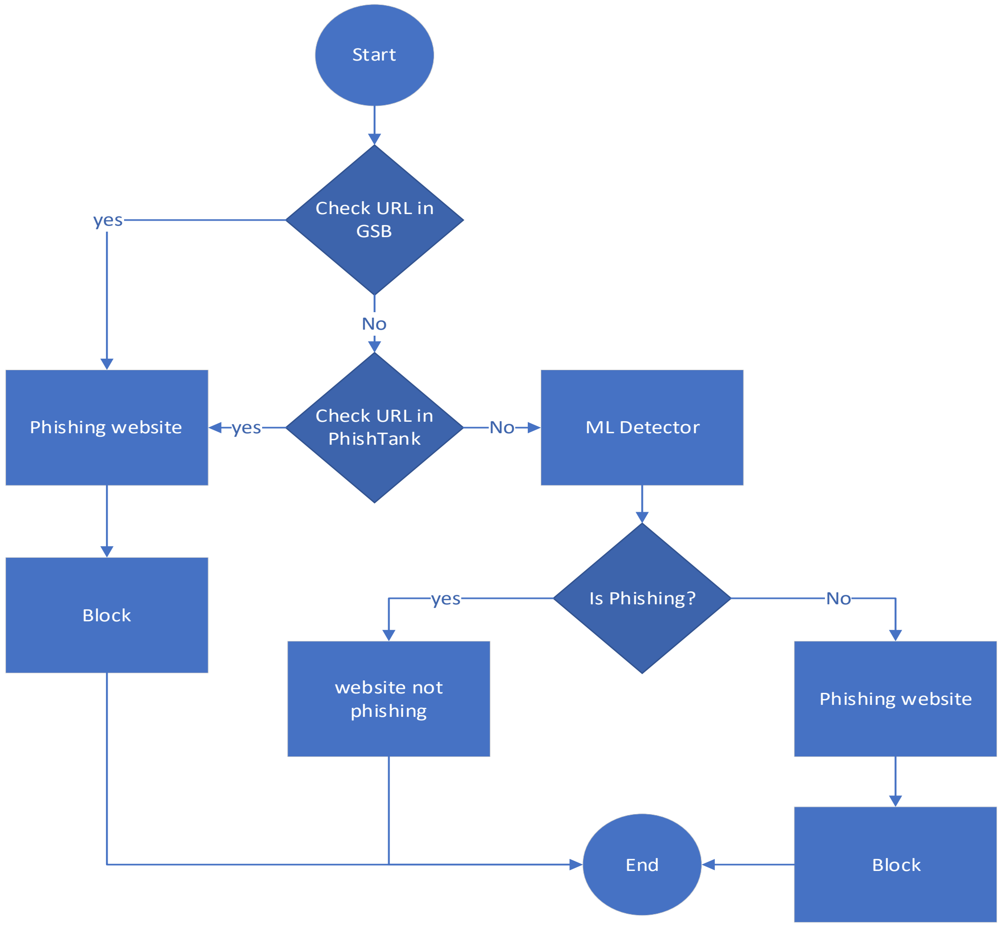
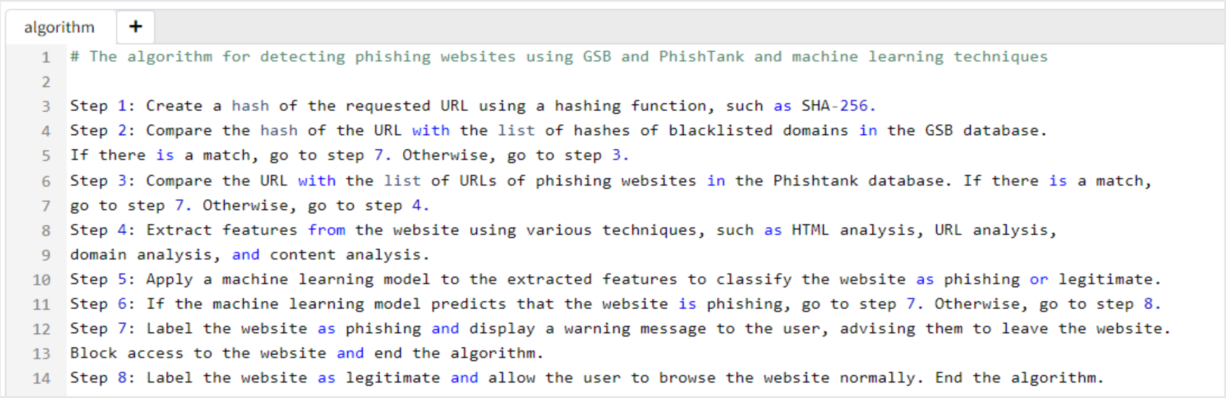


Fig.2. Flowchart of the proposed approach

The flowchart shows how a website can be detected as phishing or legitimate using Google Safe Browsing (GSB) API and PhishTank database and machine learning techniques. The flowchart has nine steps, starting from when a user visits a website and ending with either blocking access to the website if it is phishing or allowing the user to browse the website if it is legitimate. Fig. 3. Shows the phishing website detection and prevention algorithm.

The novelty of this methodology lies in combining GSB API and PhishTank database with machine learning techniques to achieve high accuracy and efficiency in detecting phishing websites. GSB API provides a fast and reliable way to check if a website is known to be malicious. In contrast, machine learning techniques provide a strong and adaptive way to deal with new or unknown phishing websites which we may not be able to detect using GSB and PhishTank. This methodology can enhance web security and protect users from falling victim to phishing attacks.



```
# The algorithm for detecting phishing websites using GSB and PhishTank and machine learning techniques

Step 1: Create a hash of the requested URL using a hashing function, such as SHA-256.
Step 2: Compare the hash of the URL with the list of hashes of blacklisted domains in the GSB database.
If there is a match, go to step 7. Otherwise, go to step 3.
Step 3: Compare the URL with the list of URLs of phishing websites in the Phishtank database. If there is a match,
go to step 7. Otherwise, go to step 4.
Step 4: Extract features from the website using various techniques, such as HTML analysis, URL analysis,
domain analysis, and content analysis.
Step 5: Apply a machine learning model to the extracted features to classify the website as phishing or legitimate.
Step 6: If the machine learning model predicts that the website is phishing, go to step 7. Otherwise, go to step 8.
Step 7: Label the website as phishing and display a warning message to the user, advising them to leave the website.
Block access to the website and end the algorithm.
Step 8: Label the website as legitimate and allow the user to browse the website normally. End the algorithm.
```

Fig.3. Phishing website detection and prevention algorithm

3.1. *Feature Extraction*

To categorize websites, we employed 30 features, including domain-based features, anomalous features, HTML and JavaScript features, and features based on URLs. Fig. 4. shows the features that were used in this research.

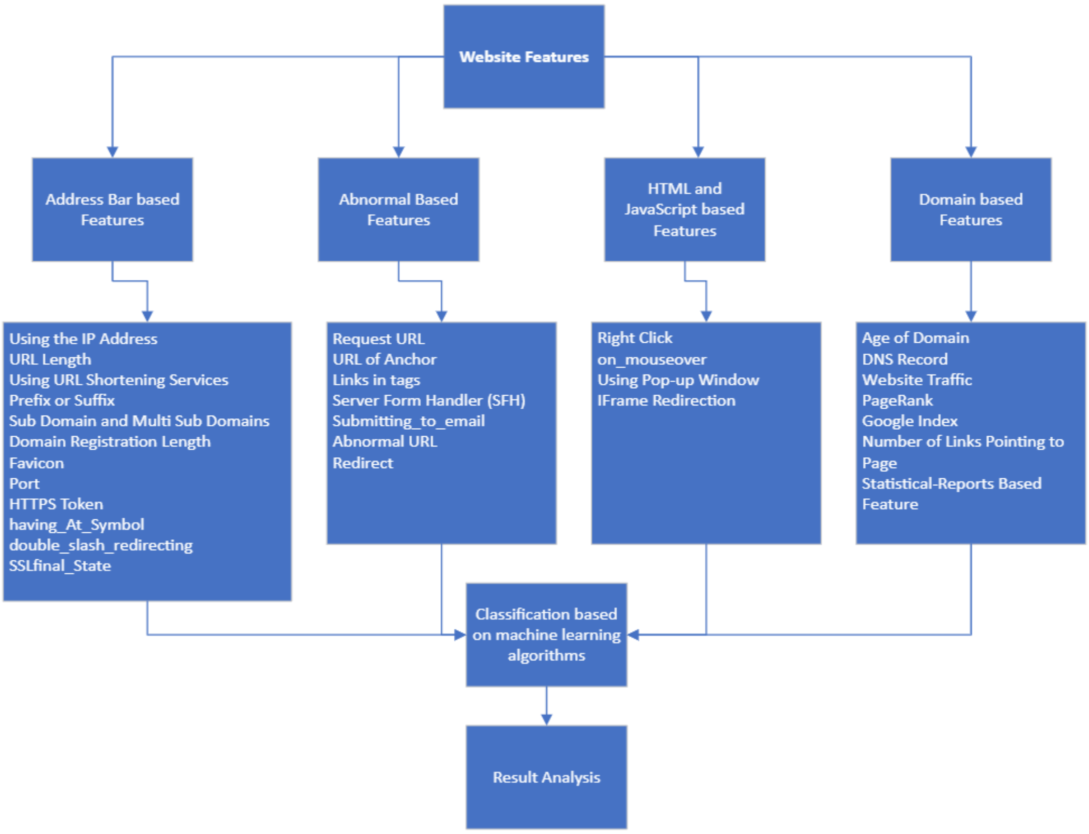


Fig.4. Features extracted to classify websites

The features that are extracted to classify websites as phishing or legitimate can include domain-based features, anomalous features, HTML and JavaScript features, and features based on URLs. These features are used to examine the behavior of the website and detect any malicious activity or purpose that might be dangerous. For example, a phishing website's URL may have minor differences from a real website's URL or extra subdomains or pathways. The source code of a phishing website could include containing suspicious JavaScript code or hidden forms that are used to steal user information. A phishing website may send users to several pages during their session to collect more data or start the download of harmful software. A phishing website's security protocol might not employ SSL or might use SSL certificates that are invalid. These features may include domain-specific characteristics like the domain's age and length of registration as well as the domain name itself. structural elements including the number of hyperlinks, and the existence of pop-up advertisements. A website's hosting location, the inclusion of security features like SSL encryption, and the website's standing among users and in search engine results are examples of additional features. the presence of subdomains, the number of letters or digits in the URL, and its length are all URL-based characteristics. These characteristics can be used to train machine learning algorithms that categorize websites according to how likely they are to be phishing. The quality, variety, and amount of the training dataset, along with its representativeness, all affect how well the algorithm performs. Here's more detail of the features set used in our research:

- Address bar features

To correctly identify phishing attempts, special characters such as @,"", /, _, -, and the length of the input URLs must be included in this IP address. It also has a URL length. The length of a domain registration is also considered. Unusual port numbers in URLs (for example, non-standard ports like 8080) can indicate phishing. While looking for "https://" in the URL can help identify secure connections, phishers may use deceptive URLs with HTTPS tokens to appear secure. Multiple consecutive slashes in a URL can indicate an attempt to trick users into thinking they are on a legitimate site when they are being redirected elsewhere. The final state of an SSL certificate can reveal discrepancies or issues with its validity, such as expired certificates or mismatched domains, which are common in phishing scenarios.

- Abnormal Features

Abnormal-based features are critical for detecting phishing websites. Here are a few examples: Analyzing the URLs to which a webpage sends requests can reveal unusual or suspicious destinations, such as external domains associated with phishing. Examining the URLs linked in anchor tags can aid in the detection of deceptive links that redirect users to phishing sites rather than their intended destinations. Examining HTML tags for unusual or hidden links, particularly those leading to unexpected locations, can be a strong indicator of phishing attempts. It is critical to monitor how forms on a website interact with server handlers. Forms that send user input to email addresses rather than secure server-side scripts can expose phishing schemes.

- HTML and JavaScript Features

HTML and JavaScript features are critical in detecting phishing websites. Phishing sites, for example, may disable the right-click context menu in order to prevent users from inspecting elements or opening links in new tabs. Suspicious use of JavaScript's "onmouseover" events, which cause actions to be performed when the mouse cursor hovers over elements, can be indicative of phishing attempts. When hovering over certain elements, for example, unexpected pop-ups or hidden forms can be triggered. Pop-up windows are frequently used by phishing websites to collect sensitive information. Iframes can be used by phishers to redirect users to malicious content while hiding the URL in the browser's address bar. Detecting iframe redirection and cross-origin requests can raise concerns about a website's legitimacy.

- Domain features

Because phishing websites have such a short lifespan, this code examines the domain's age. It investigates DNS records, web traffic, and page rank. Because phishing pages are visited infrequently, the index will be 0 or 1 at most. It also looks at the Google index and the number of pages that link to that page. The Statistical report contains all of the feature information.

Table 1. Feature extraction

Name of Feature	Description Rule Feature	Value
IP Address	IF $\begin{cases} \text{If The Domain Part has an IP Address} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$	(-1, 1)
URL Length	IF $\begin{cases} URL\ length < 75 \rightarrow \text{Legitimate} \\ else\ if\ URL\ length \geq 75\ and\ \leq 75 \rightarrow Suspicious \\ otherwise \rightarrow \text{Phishing} \end{cases}$	(-1, 1)
URL Shortening Services	IF $\begin{cases} \text{TinyURL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$	(-1, 1)
Prefix or Suffix	IF $\begin{cases} \text{Domain Name Includes } (-) \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$	(-1, 1)
Sub Domain and Multi Sub Domains	IF $\begin{cases} \text{Dots In Domain} = 1 \rightarrow \text{Legitimate} \\ \text{Dots In Domain} = 2 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$	(-1, 1)
Domain Registration Length	IF $\begin{cases} \text{Domains Expires on} \leq 1 \text{ years} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$	(-1, 1)
Favicon	IF $\begin{cases} \text{Favicon Loaded From External Domain} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$	(-1, 1)

Port	IF { Port # is of the Preffered Status → Phishing Otherwise → Legitimate	(-1, 1)
HTTPS Token	IF { HTTP Token in Domain Part of The URL → Phishing Otherwise → Legitimate	(-1, 1)
Having At Symbol	IF { Url Having @ → Phishing Otherwise → Legitimate	(-1, 1)
Double slash redirecting	IF { The Position of the Last Occurrence of "//" in the URL > 7 → Phishing Otherwise → Legitimate	(-1, 1)
SSL State	IF { Use https and Issuer Is Trusted and Age of Certificate ≥ 1 Years → Legitimate Using https and Issuer Is Not Trusted → Suspicious Otherwise → Phishing	(-1, 1)
Request URL	IF { % of Request URL < 22% → Legitimate %of Request URL ≥ 22% and 61% → Suspicious Otherwise → feature = Phishing	(-1, 1)
URL of Anchor	IF { % of URL Of Anchor < 31% → *Legitimate* % of URL Of Anchor ≥ 31% And ≤ 67% → Suspicious Otherwise → Phishing	(-1, 1)
Links in tags	IF { % of Links in " < Meta > "," < Script > " and " < Link>" < 17% → Legitimate % of Links in < Meta > "," < Script > " and " < Link>" ≥ 17% And ≤ 81% → Suspicious Otherwise → Phishing	(-1, 1)
Server Form Handler (SFH)	IF { SFH is "about: blank" Or Empty → Phishing SFH Refers To A Different Domain → Suspicious Otherwise → Legitimate	(-1, 1)
Submitting to email	IF { Using "mail()" or "mailto:" Function to Submit User Information → Phishing Otherwise → Legitimate	(-1, 1)
Abnormal URL	IF { The Host Name Is Not Included In URL → Phishing Otherwise → Legitimate	(-1, 1)
Redirect	IF { Redirect Page ≤ 1 → Legitimate of Redirect Page ≥ 2 And < 4 → Suspicious Otherwise → Phishing	(-1, 1)
On mouseover	IF { onMouseOver Changes Status Bar → Phishing It Does′t Change Status Bar → Legitimate	(-1, 1)
Right Click	IF { Disabled → Phishing Otherwise → Legitimate	(-1, 1)
Pop-up Window	IF { Popoup Window Contains Text Fields → Phishing Otherwise → Legitimate	(-1, 1)
IFrame Redirection	IF { iframe → Phishing Otherwise → Legitimate	(-1, 1)
Age of Domain	IF { Age Of Domain ≥ 6 months → Legitimate Otherwise → Phishing	(-1, 1)
DNS Record	IF { no DNS Record For The Domain → Phishing Otherwise → Legitimate	(-1, 1)
Website Traffic	IF { Website Rank < 100,000 → Legitimate Website Rank > 100,000 → Suspicious Otherwise → Phish	(-1, 1)
Page Rank	IF { PageRank < 0.2 → Phishing Otherwise → Legitimate	(-1, 1)
Google Index	IF { Indexed by Google → Legitimate Otherwise → Phishing	(-1, 1)
Number of Links Pointing to Page	IF { Of Link Pointing to The Webpage = 0 → Phishing Of Link Pointing to The Webpage > 0 and ≤ 2 → Suspicious Otherwise → Legitimate	(-1, 1)
Statistical Reports Based Feature	IF { Host Belongs to Top Phishing Ips or Top Phishing Domains → Phishing Otherwise → Legitimate	(-1, 1)

- Prefix Suffix: If a website has a hyphen (-), between the domain name and its extension it could be a sign of fakery. This practice is commonly used to incorporate keywords or trademarks into links.
- Having Sub Domain: if a website has a number of subdomains (more than three) it may indicate an attempt to deceive visitors by adding an extra layer of complexity.
- SSLfinal State: This feature reveals the status of the website's SSL (Secure Sockets Layer) certificate. This certificate validates the website's identity and secures the user's data. If the website has a valid and matching certificate for its domain name, it is trustworthy. If the website lacks a certificate, or if it has an expired or mismatched one, it is likely that it is a forgery.
- Domain registration length: This characteristic indicates the length of registration for the website's domain name. If the website has been registered for a short time (less than a year), it is possible that it is a forgery, as hackers prefer not to commit to domains for long periods of time.
- Favicon: This is the icon for a website that appears in the address bar or tabs. If the website employs an external icon from another source, it is possible that it is a forgery, as this is a method of adding false credibility.
- port: The port number that the website uses to connect to the server is shown by this feature. If the website uses

non-standard ports (not 80 or 443), it is possible that it is a forgery, as this is a method of evading detection by security software.

- HTTPS token: This characteristic specifies whether or not the word https appears in the domain name. If the website utilizes this word in the domain name, it is possible that it is a forgery, as this is a method of fooling visitors into thinking the website is secure and encrypted.
- Request URL: This feature represents the percentage of external links requested by the website from other sources. If the website requests a high number of external links (greater than 61%), it is possible that it is a forgery, as this is a method of loading undesired or dangerous content.
- URL of anchor: This feature shows the percentage of links that have anchor tags that redirect visitors to the same website or internal pages. If the website includes a few of these links (less than 31%), it is possible that it is a forgery, as this is a strategy to minimize interaction with the user and keep him on the same page.
- Links in tags: This feature displays the percentage of links that include HTML tags like meta, script, and link. If the website includes a high percentage of these links (greater than 81%), it is possible that it is a forgery, as this is a method of loading undesired or dangerous content.
- SFH: This feature shows the presence of a hidden field in the website's submit form. If the website employs a hidden field, it is possible that it is a forgery, as this is a method of collecting data from visitors without their knowledge.
- Submitting to email: This feature shows the presence of a mailto property in the website's submission form. If the website employs this attribute, it is possible that it is a forgery, as this is a method of sending the visitor's data to an unknown email address.
- Abnormal URL: This feature shows the occurrence of a discrepancy between the website's domain name and server name. If the website utilizes a server address that does not match its domain name, it is possible that it is a forgery, as this is a method of concealing the website's true identity.
- Redirect: This feature displays the number of redirects used by the website when it is visited. If the website employs a large number of redirections (more than four), it is possible that it is a forgery, as this is a method of misleading the visitor and changing the page that they see.
- On mouseover: This attribute shows the presence of an onmouseover attribute in the website's links. When the cursor passes over a link, this attribute is utilized to modify its appearance or content. If the website employs this attribute, it is possible that it is a forgery, as this is a method of displaying a different text or image than the actual link.
- Right Click: The presence of a right-click feature on the website is indicated by this feature. When the right mouse button is clicked, this attribute is used to disable or change its purpose. If the website employs this attribute, it is possible that it is a forgery, as this is a method of preventing the visitor from scrutinizing or copying a link or an image.
- popup window: This attribute indicates that a pop-up window is present on the page. This is a window that appears on the screen without the user's request. If the website employs a pop-up window, it is possible that it is a forgery, as this is a method of distracting the user or forcing them to click on something.
- Iframe: This feature shows that an iframe element is present on the webpage. This is an element that allows another page to be embedded within the current page. If the website employs an iframe element, it is possible that it is a forgery, as this is a method of loading external or malicious material without the user's knowledge.
- Age of domain: This attribute indicates the age of the website's domain name. If the website was founded recently (within the last 6 months), it is possible that it is a forgery, as hackers prefer to utilize fresh and unfamiliar domains.
- DNS Record: This feature indicates whether or not the website has a DNS (Domain Name System) record. This is a record that provides information such as the domain name, IP address, and other technical information. If the website lacks a DNS record, it is possible that it is a forgery because it is not formally registered or recognized.
- Web traffic: This feature shows the amount of web traffic received by the website. If the website has little traffic (less than 20%), it is possible that it is a forgery, because it is not well-known or trusted.
- Page Rank: This feature shows the page rank assigned to the website by Google's search engine. This is a ranking used to assess the quality and importance of a website based on the amount and quality of links pointing to it. If the website has a low page rank (less than 0.2), it is possible that it is a forgery, as this suggests that it is not appreciated or sought.
- Google Index: This characteristic denotes the website's presence in Google's index, which is a list of websites that Google has examined and saved. If a website is not indexed by Google, it is likely that it is a forgery because it is unknown or inefficient.
- Links pointing to page: This feature displays the number of links pointing to the website from other websites. If the website has few links connecting to it (less than 2), it is possible that it is a forgery, as this suggests poor popularity or reliability.
- Statistical report: The presence of a statistical report about the website from a reliable source, such as Alexa or SimilarWeb, is indicated by this feature. If the website offers a statistical report, it is likely to be legitimate because it has been analyzed and validated by a third party.

4. Experimental Results

The classification findings employing characteristics (such as domain-based features, abnormal features, abnormal features based on URLs, and HTML and JavaScript features) are shown in this section.

4.1. Algorithm Used

In this study, we compared the performance of seven classifiers utilized as machine-learning methods for the suggested system: Naive Bayes (NB), Decision Tree (DT), K Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), logistic regression, and Adaboost. Table 2 shows a comparison of the advantages and disadvantages of each algorithm.

NB: The Naïve Bayes algorithm is a probabilistic classification technique based on the theorem of Bayes. Given the class label, it assumes that all features in the data are independent of one another. It computes the likelihood of a specific class for a given set of features and selects the class with the highest likelihood as the predicted class [38].

Equation (1) is used to calculate the later probability of class c for a dataset with feature vector X = (x_1, x_2, ... x_n), given predictor attribute x.

$$P(c|x) = \frac{P(X|C)\,P(c)}{P(c)} \tag{1}$$

P(x) is the prior probability of the predictor attribute, P(c) is the prior probability of class c, and P(x|c) is the probability of the predictor attribute given class c. P(c) is the prior probability of class c.

DT: is a Supervised method of learning that can be used for both classification and regression problems, but it is most commonly used for classification. It is a tree-structured classifier in which internal nodes represent dataset features, branches represent decision rules, and each leaf node represents the result. A Decision tree has two nodes: the Decision Node and the Leaf Node. Decision nodes are used to make decisions and have multiple branches, whereas Leaf nodes are the results of those decisions and do not have any additional branches [39].

KNN: As one of the most essential and effective algorithms for data separation, it has the potential to become the primary choice for implementation, particularly when the given data is quite ambiguous. Evelyn Fix and Joseph Hodges invented this algorithm in 1951 for discriminant examination. The K-NN algorithm belongs to the supervised type of learning technique and is regarded as one of the most user-friendly algorithms in Machine Learning. Although it is suitable for classifying and regressing, it is primarily used for classifying objects [40]. The Euclidean distance between two points with coordinates (x, y) and (a, b) is calculated using (2).

$$Dist((x, y), (a, b)) = \sqrt{(x-a)^2 + (y-b)^2} \tag{2}$$

SVM: Is a powerful machine learning algorithm that can be used for linear or nonlinear classification, regression, and even outlier detection. Text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection are all tasks that SVMs can perform. Because they can handle high-dimensional data and nonlinear relationships, SVMs are adaptable and efficient in a wide range of applications [41].

RF: Is a type of Supervised Machine Learning Algorithm that is commonly used in classification and regression problems. It constructs decision trees from various samples and uses their majority vote for classification and average for regression. Leo Breiman and Adele Cutler created it. To generate predictions or classifications, it employs an ensemble of multiple decision trees. The random forest algorithm produces a more accurate result by combining the outputs of these trees. Its widespread popularity stems from its user-friendliness and adaptability, which allow it to effectively handle classification and regression problems. The algorithm's strength is its ability to handle complex datasets while minimizing overfitting, making it a valuable tool for a variety of predictive tasks in machine learning [42].

Logistic Regression: which falls under the Supervised Learning technique, is a commonly used Machine Learning algorithm. It is used to predict a categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable and is therefore suitable for predicting categorical or discrete values. This algorithm is important as it can provide probabilities and classify new data using both continuous and discrete datasets [43].

AdaBoost: Also called Adaptive Boosting, is a technique in Machine Learning used as an Ensemble Method. The most common estimator used with AdaBoost is decision trees with one level which means Decision trees with only 1 split. These trees are also called Decision Stumps [44].

4.2. Dataset

The suggested method for the classification of URLs as legitimate and phished web pages has been tested using a dataset. There are 11055 websites in the dataset utilized in the trials, some of which were used for phishing and others legitimate. We employ a set of publicly accessible phishing websites from the UCI machine learning repository. 30 attributes are provided for every website that is part of the dataset. The list covers various features such as URL length, Iframes usage, popups by website, domain registration age, etc. We mention the features of the dataset used in the experiments and evaluation in Table 1. We give each feature in the dataset a value of -1 if it is a phishing website and a value of 1 if it is a legitimate website.

Table 2. Comparison of the advantages and disadvantages of each algorithm

Algorithm	Advantages	Disadvantages	Ability to handle numerical data	Ability to classify websites
Naïve Bayes	- Simple to implement and train. - Adaptable to a wide range of tasks. - Effective on classification data.	In some difficult cases, it may not be accurate.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
Decision Tree	- Simple to learn and interpret. - Works well with category and numerical data. - Can deal with non-linear connections.	- Overfitting is possible. - Biased trees based on training data may result.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
K-Nearest Neighbor	- Simple and uncomplicated to use. - It may be applied to classification or regression problems. - Works well with small datasets.	- Computationally costly. - May be affected by the distance metric used and the number of neighbors.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
Support Vector Machine	- Works well with both linear and nonlinear data. - Excellent in high-dimensional spaces. - Can handle datasets with a large number of features.	- Needs careful kernel function selection. - May be sensitive to hyperparameter selection.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
Random Forest	- It is capable of dealing with both classification and regression problems. - It is capable of handling high-dimensional datasets with a large number of features. - It can manage missing data and retain accuracy even when values are absent.	- It is computationally costly and time-consuming, especially when dealing with huge datasets. - It might not be as easy to interpret as other models. - It might not perform well on imbalanced datasets.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
Logistic Regression	- Simple and efficient. - Works well with binary classification tasks. - Provides a probabilistic interpretation of outcomes.	- Non-linear data may not perform well. - Careful feature selection is required.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.
Adaboost	- It is compatible with a wide range of basic classifiers. - Performs well on classification challenges. - Decreases the risk of overfitting.	- It could be sensitive to noisy data - Computationally costly - Prone to outliers.	Works well with numerical data and can be used to accurately classify websites.	Can be used to classify websites with good accuracy.

4.3. *Evaluation Parameters Used*

In our experiment, we used k-fold cross-validation, a commonly used technique for calculating plausible predictions on unobserved data, to assess the efficacy of the proposed system. To make sure that every piece of pertinent information in the training set was retained in this investigation, we performed 5-fold cross-validation. This technique includes randomly dividing the initial sample into five equal-sized sub-samples. Four of the five sub-samples are utilized for training, while the final sub-sample is kept as validation data for testing. Each of the sub-samples is used as the validation data once during the course of the five iterations of this process. To determine how well the algorithm performed over the full dataset, the outcomes of these tests are then summed.

Using evaluation metrics including the False Positive Rate, False Negative Rate, Precision, Recall, F-score, Accuracy, and Matthews Correlation Coefficient, we compared different classifiers. According to Table 4, these parameters are calculated using the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) fields of the confusion matrix shown in Table 3.

Table 3. Confusion matrix

Class	Phishing	Legitimate
Phishing	TP	FP
Legitimate	FN	TN

FPR: This is the percentage of erroneously identifying legitimate websites, it is calculated by the following (3):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \tag{3}$$

FNR: It is the percentage of phishing websites that were wrongly classified, it is calculated by the following (4):

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP+FN}} \tag{4}$$

Precision: It evaluates how accurate a model is. It is a possibility that a true outcome will be correctly classified, it is calculated by the following (5):

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{5}$$

Recall: The model's ability to correctly predict positives from real positives is indicated by the model recall score, it is calculated by the following (6):

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{FN+TP}} \tag{6}$$

F-Measure: It is the precision and recalls harmonic mean. It gives a quick way to compare classifiers and is between 0 and 1, , it is calculated by the following (7):

$$\mathrm{F-Measure} = \frac{2*\mathrm{TP}}{2*\mathrm{TP+FN+FP}} \tag{7}$$

Accuracy (%): It is the percentage of both legitimate and phishing websites that have been accurately detected, it is calculated by the following (8):

$$\mathrm{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FN+TN+FP}} * 100 \tag{8}$$

MCC: It is used to assess and contrast the binary classification performance of machine learning algorithms. It ranges in value from 1 to -1 and assesses the correlation between labels on the expected and actual data, it is calculated by the following (9):

$$\mathrm{MCC} = \frac{\mathrm{TP\ X\ TN - FB\ X\ FN}}{\sqrt{\mathrm{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}} \tag{9}$$

Table 4. Fields of confusion matrix with description

Field	Description
TP	Number of websites that have been correctly identified as phishing.
TN	Number of websites that have been correctly identified as legitimate.
FP	Number of legitimate websites that were mistakenly labeled as phishing.
FN	Number of phishing websites that were mistakenly labeled as legitimate.

A number of evaluation metrics, including accuracy, precision, recall, F-measure, and the Matthews Correlation Coefficient, are computed using these fields. We assess the algorithm's performance using these measures, and then make the required changes to enhance it.

5. Results and Discussions

According to equations (3) to (9), Table 5 shows the values for the proposed features' accuracy, precision, recall, F1 Score, and MCC. In order to categorize phishing websites according to feature groupings, the seven classifiers are used. We can understand how each feature affects classification using the information in this table. The findings show that the RF algorithm delivers the most precise classification with the lowest FPR and FNR. Furthermore, the MCC value is quite near 1, indicating a perfect correlation between the actual and projected classes.

The performance of the suggested set of features was evaluated, and the results were gathered for the seven classifiers that were taken into consideration based on accuracy. Fig. 5. makes clear that RF, with a 93% accuracy rate, and DT, with a 91% accuracy rate, both attained the maximum accuracy for this set of features.

The results show in Fig. 5. the accuracy with which various machine learning algorithms categorize websites into legitimate and phishing websites. According to the data, the accuracy of the Random Forest (RF) algorithm is 93%, followed by that of the decision tree (DT) at 91%, and then the AdaBoost algorithm, which offers 90% accuracy. While Support Vector Machine (SVM) and logistic regression techniques offer 82% and 84% accuracy, respectively,

respectively, K-Nearest Neighbor (KNN) algorithm offers 87% accuracy. The results indicate that DT, RF, and Adaboost algorithms are more effective in identifying phishing websites in comparison to other algorithms. The Naive Bayes (NB) method delivers the lowest accuracy of 61%.

Table 5. Performance of different classifiers

Evaluation Classifier	FPR	FNR	Accuracy	Precision	Recall	F1 Score	MCC
Naive Bayes	0.24	0.49	62%	72%	50%	59%	0.26
Decision Tree	0.13	0.05	91%	88%	94%	91%	0.81
K Nearest Neighbor	0.19	0.06	88%	85%	93%	89%	0.75
Support Vector Machine	0.32	0.04	83%	78%	95%	86%	0.66
Random Forest	0.12	0.02	93%	90%	97%	93%	0.85
Logistic Regression	0.28	0.05	85%	80%	94%	87%	0.69
Adaboost	0.17	0.03	90%	87%	96%	91%	0.80

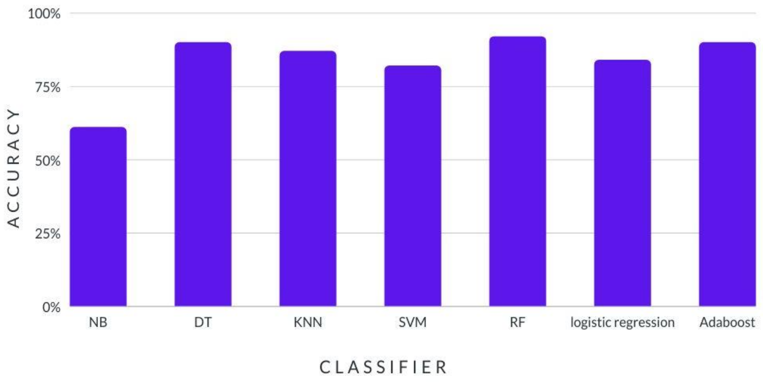


Fig.5. Accuracy results of algorithms

The results show in Fig. 6. the precision is defined as the ratio of true positive predictions to all positive predictions (including true positive and false positive predictions). Therefore, the quantity of false positive predictions decreases as precision increases. In this instance, the RF algorithm has the highest precision score of 90%, demonstrating that it is the most accurate classifier for this task and has the fewest false positives. The DT and Adaboost algorithms both function well, scoring 88% and 87% in terms of precision, respectively. SVM has the lowest accuracy score (78%), indicating that it is less accurate than the other classifiers and has a larger rate of false positives.

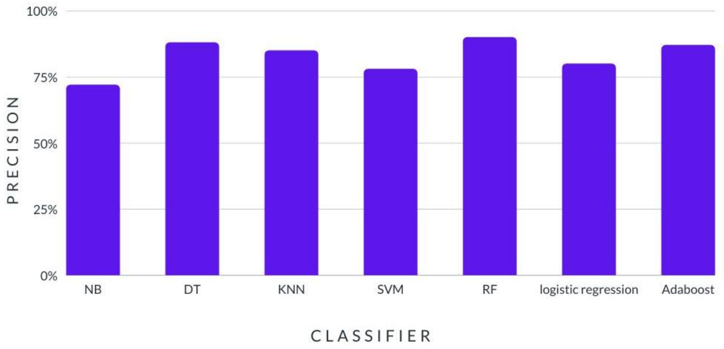


Fig.6. Precision results of algorithms

The results show in Fig. 7. the recall values for various machine learning algorithms that were used to categorize websites into legitimate and phishing websites. Recall gauges how well a model can distinguish between positive cases (phishing websites) and true positive instances. The results show that the RF method, with a recall value of 97%, has the highest recall rate, followed by Adaboost, SVM, DT, and logistic regression, all of which have recall values of 96%, 95%, 94%, and 95%, respectively. The NB algorithm had the lowest recall value, at 50%, indicating that it was not very good at correctly identifying phishing websites.

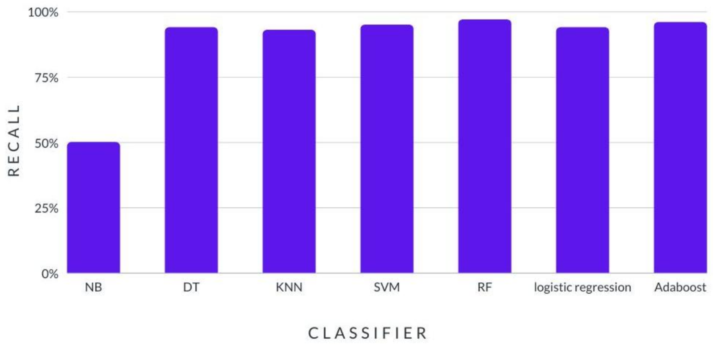


Fig.7. Recall results of algorithms

The results show in Fig. 8. the F1 scores for different machine learning algorithms that categorize websites as legitimate or phishing sites. The harmonic means of recall and precision, known as the F1 score, is used to assess the algorithm's overall accuracy. According to the findings, the Random Forest (RF) algorithm has the highest F1 score (93%) followed by Decision Tree (DT) and Adaboost, both of which have scores of 91%. The lower F1 ratings of the other algorithms, such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression, range from 59% to 87%. The most accurate method for categorizing websites as real or phishing is the RF algorithm.

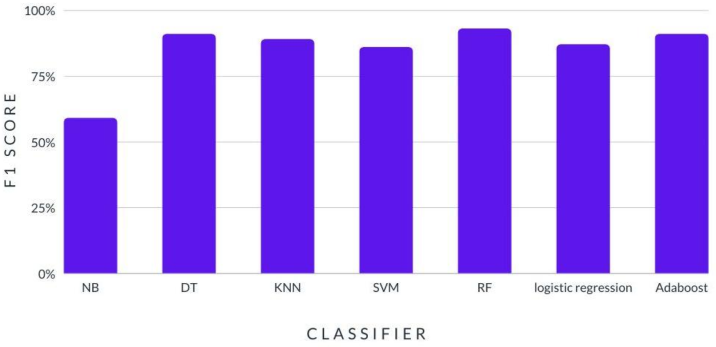


Fig.8. F1 scores results of algorithms

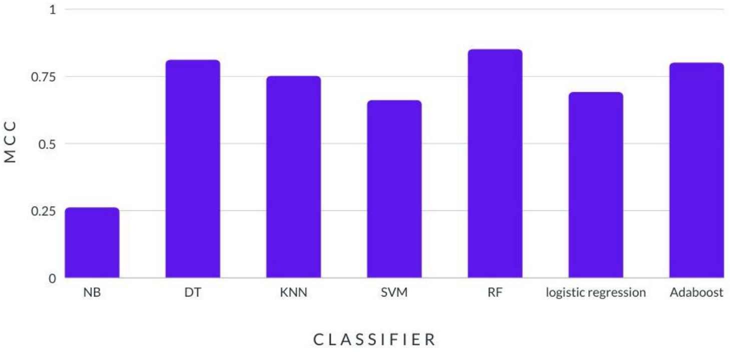


Fig.9. MCC results of algorithms

The Matthews correlation coefficient (MCC), which ranges from -1 to 1, is a gauge of the accuracy of binary categorization. Inverse predictions are represented by a coefficient of -1, whereas perfect predictions are represented by a value of +1. The MCC scores in the results above Fig. 9. range from 0.26 to 0.85. The decision tree (DT) method and random forest (RF) algorithm have the highest MCC scores of 0.81 and 0.85, respectively, showing that these two algorithms are effective at predicting whether a website is real or phishing. The naïve Bayes (NB) algorithm has the lowest MCC score (0.26), which shows that it struggles to accurately forecast how to classify websites. With MCC scores ranging from 0.66 to 0.80, the KNN, SVM, logistic regression, and Adaboost algorithms are somewhat successful in predicting the categorization of the websites.

The following Fig. 10. shows a comparison of the Accuracy, Precision, Recall, F1 score, and MCC results for the seven algorithms used in the search. It shows that the Random Forest algorithm is the best.

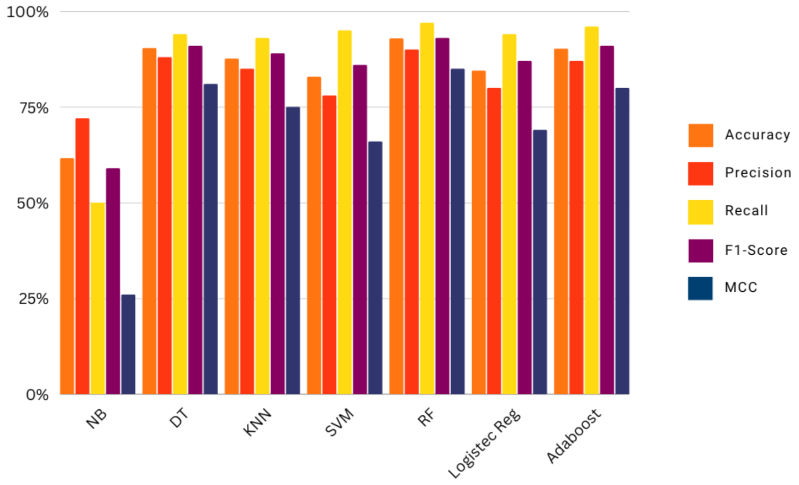


Fig.10. Comparison of algorithms

The resulting graph Fig. 11. help to identify the most relevant features for the model, which can be useful for feature selection, model interpretation, and improving overall model performance. We sort the features based on their importance score using the Random Forest algorithm and plot them in descending order to show which features have the most impact on the target variable.

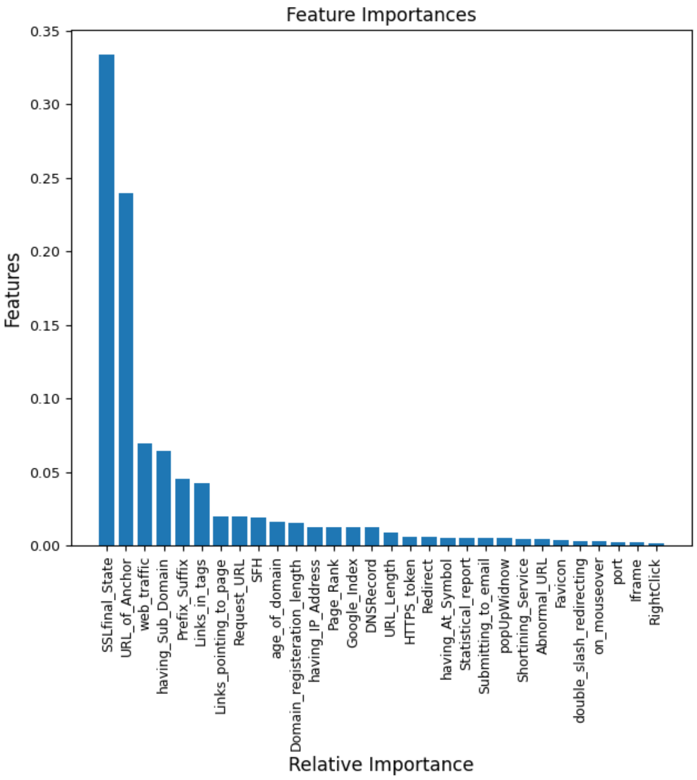


Fig.11. Most relevant features for the model

6. Conclusions

In this study, we presented a strong method for detecting phishing websites by using Blacklist databases and machine learning techniques with an array of powerful features, including Page feature, URL, HTML, JavaScript, and domain-based data. The outcomes of our research have demonstrated that the RF classifier achieves the highest accuracy at 93%, accompanied by FPR and FNR rates of 0.12 and 0.02, respectively. Furthermore, the decision tree model exhibited excellent performance with an accuracy of 91%, albeit with slightly elevated FPR and FNR rates of 0.13 and 0.05.

The evolving landscape of cyberattacks demands a proactive approach to countering sophisticated phishing strategies. As cybercriminals continually improve their methods, it becomes imperative to develop increasingly strong and effective anti-phishing systems. Our investigation is an important step towards addressing this challenge. However, the journey does not end here.

In the future, our research will initiate on exploring the potential of deep learning algorithms and expanding the feature set to enhance the classification accuracy of phishing website detection on a broader scale. We aspire to strengthen the protection of individuals against falling victim to malicious online activities. We believe that our future endeavors will contribute significantly to the ongoing battle against cyber threats.

As the digital landscape continues to evolve, collaboration between researchers, security practitioners, and policymakers will be pivotal in fortifying our online defenses. By aligning our efforts, we can collectively shape a more secure digital realm, mitigating the risks posed by phishing attacks and safeguarding the integrity of online interactions.

References

[1] Singla, S., Gandotra, E., Bansal, D., & Sofat, S. “A novel approach to malware detection using static classification”, International Journal of Computer Science and Information, Vol.13, No.3, pp.1-5, 2015.

[2] Enterprise, V. "Verizon 2018 data breach investigations report", 2018. [Online]. Available: https://verizon.com/business/resources/reports/2018-data-breach-digest.pdf

[3] APWG. "Phishing Activity Trends Report, 1st Quarter 2018", 2018. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf

[4] APWG. "Phishing Activity Trends Report 4th Quarter 2019", 2019. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q3_2019.pdf

[5] Gandotra, E., Bansal, D., & Sofat, S. “Malware intelligence: beyond malware analysis”, International Journal of Advanced Intelligence Paradigms, Vol.13, No.1-2, pp.80-100, 2019. DOI: 10.1504/IJAIP.2019.099945

[6] Sharma, A., Gandotra, E., Bansal, D., & Gupta, D. “Malware capability assessment using fuzzy logic”, Cybernetics and Systems, Vol.50, No.4, pp.323-338, 2019. DOI: 10.1080/01969722.2018.1552906

[7] Chiew, K.L., Yong, K.S.C., & Tan, C.L.J.E.S.w.A. “A survey of phishing attacks: Their types, vectors, and technical approaches”, Expert Systems with Applications, Vol.106, pp.1-20, 2018. DOI: 10.1016/j.eswa.2018.03.050

[8] Gandotra, E., & Sofat, S.J.I.J.o.N.-G.C. “Tools & Techniques for Malware Analysis and Classification”, International Journal of Next-Generation Computing, Vol.7, No.3, pp.176-197, 2016.

[9] Federal Trade Commission. "How to Recognize and Avoid Phishing Scams", https://consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams (accessed Aug. 1, 2023).

[10] Security Gladiators. "How to Detect a Phishing Email Attack and Scam: Tips and Methods" https://securitygladiators.com/threat/phishing/detection/ (accessed Aug. 15, 2023).

[11] Krupalin, V.A., Sriramakrishnan, G.V., & Daniya, T. “A Survey and Taxonomy of Anti-Phishing Techniques for Detecting Fake Websites”. 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp.601-604, 2022. DOI: 10.1109/ICIRCA54612.2022.9985744

[12] Agrawal, V. "WhatAPhish: Detecting Phishing Websites" https://towardsdatascience.com/whataphish-detecting-phishing-websites-e5e1f14ef1a9 (accessed sept. 10, 2023).

[13] Shahrivari, V., Darabi, M. M., & Izadi, M. “Phishing detection using machine learning techniques”, arXiv preprint arXi, Vol.2009, No.11116, 2020.

[14] Volkamer, M., Renaud, K., Reinheimer, B., & Kunz, A. “User Experiences of Torpedo: Tooltip-Powered Phishing Email Detection”, Computers & Security, Vol.71, pp.100-113, 2017. DOI: 10.1016/j.cose.2017.02.004

[15] Oest, A., Safaei, Y., Zhang, P., Wardman, B., Tyers, K., Shoshitaishvili, Y., & Doupé, A. “PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists”, 29th USENIX Security Symposium (USENIX Security 20), pp. 379-396, 2020.

[16] Chiew, K. L., Chang, E. H., & Tiong, W. K. “Utilization of website logo for phishing detection”, Computers & Security, Vol.54, pp.16-26, 2015. DOI: 10.1016/j.cose.2015.07.006

[17] Tan, C.L., Chiew, K.L., & Wong, K.J.D.S.S. “PhishWHO: Phishing Webpage Detection via Identity Keywords Extraction and Target Domain Name Finder”, Decision Support Systems, Vol.88, pp.18-27, 2016. DOI: 10.1016/j.dss.2016.05.005

[18] AlSabah, M., Nabeel, M., Boshmaf, Y., & Choo, E. “Content-Agnostic Detection of Phishing Domains Using Certificate Transparency and Passive DNS”, Proceedings of the 25th International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 446-459, 2022. DOI: 10.1145/3545948.3545958

[19] Torrealba A, L. and Bustos-Jiménez, J. “Detecting Phishing in a Heuristic Way (Abstract)”, 2021.

[20] Sinha, J., & Sachan, M. “PhishX: An Empirical Approach to Phishing Detection”, 2022. DOI: 10.1145/1122445.1122456

[21] Bhattacharyya, S., kumar Pal, C., & kumar Pandey, P. “Detecting Phishing Websites, a Heuristic Approach”, International Journal of Latest Engineering Research and Applications (IJLERA), Vol.2, No.03, pp120-129, 2017.

[22] Ranaldi, L., Petito, M., Gerardi, M., Fallucchid, F., & Zanzotto, F.M. “Machine Learning Techniques for Italian Phishing Detection”, in Italian Conference on Cybersecurity, Rome, Italy 2022.

[23] Moghimi, M., and Varjani, A.Y.J.E.s.w.a. “New Rule-Based Phishing Detection Method”, Expert systems with applications, Vol.53, pp.231-242, 2016. DOI: 10.1016/j.eswa.2016.01.028

[24] Mohammad, R.M., Thabtah, F., & McCluskey, L.J.I.I.S. “Intelligent Rule-Based Phishing Websites Classification”, IET Information Security, Vol.8, No:3, pp.153-160, 2014. DOI: 10.1049/iet-ifs.2013.0202

[25] Varshney, G., Misra, M., & Atrey, P. K. “A Phish Detector Using Lightweight Search Features”, Computers & Security, Vol.62, pp.213-228, 2016. DOI: 10.1016/j.cose.2016.08.003

[26] Srinivasa Rao, R., and Pais, A.R. “Detecting Phishing Websites Using Automation of Human Behavior”, In Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security, pp.33-42, 2017. DOI: 10.1145/3055186.3055188.

[27] Jain, A.K., and Gupta, B.B. “Towards Detection of Phishing Websites on the Client Side Using a Machine Learning-Based Approach”, Telecommunication Systems, Vol.68, pp.687-700, 2018. DOI: 10.1007/s11235-017-0414-0

[28] Babagoli, M., Aghababa, M.P., and Solouk, V.J.S.C. “Heuristic Nonlinear Regression Strategy for Detecting Phishing Websites”, Soft Computing, Vol.23, No.12, pp4315-4327, 2019. DOI: 10.1007/s00500-018-3084-2

[29] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. “Machine Learning-Based Phishing Detection from URLs”, Expert Systems with Applications, Vol.117, pp.345-357, 2019. DOI: 10.1016/j.eswa.2018.09.029

[30] Doke, T., Khismatrao, P., Jambhale, V., & Marathe, N. J. I. W. C. “Phishing-Inspector: Detection & Prevention of Phishing Websites”, ITM Web of Conferences, Vol.32, No.03004, 2020. DOI: 10.1051/itmconf/20203203004

[31] Sameen, M., Han, K., and Hwang, S.O. “PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System”, IEEE Access, Vol.8, pp.83425-83443, 2020. DOI: 10.1109/ACCESS.2020.2991403

[32] Gandotra, E., and Gupta, D. “Improving Spoofed Website Detection Using Machine Learning”, Cybernetics and Systems, Vol.52, No.2, pp169-190, 2021. DOI: 10.1080/01969722.2020.1826659

[33] Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. “A Novel Approach for Phishing URLs Detection Using Lexical-Based Machine Learning in a Real-Time Environment”, Computer Communications, Vol.175, pp.47-57, 2021. DOI: 10.1016/j.comcom.2021.04.023

[34] Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G., & Alghamdi, A. “Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network”, Wireless Communications and Mobile Computing, pp.1-24, 2021. DOI: 10.1155/2021/8241104

[35] Xie, B., Li, Q., and Wei, N. “Phishing Short URL Detection Based on Link Jumping on Social Networks”, In ITM Web of Conferences, Vol. 47, pp. 01009, 2022. DOI: 10.1051/itmconf/20224701009

[36] Sánchez-Paniagua, M., Fernández, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. “Phishing URL Detection: A Real-Case Scenario Through Login URLs”, IEEE Access, Vol.10, pp.42949-42960, 2022. DOI: 10.1109/ACCESS.2022.3168681

[37] Mohammad, R., and McCluskey, L. “Phishing Websites. UCI Machine Learning Repository”, UCI Machine Learning Repository, 2015. DOI: 10.24432/C51W2X.

[38] Wickramasinghe, I., & Kalutarage, H. “Naive Bayes: Applications, Variations, and Vulnerabilities: A Review of Literature with Code Snippets for Implementation”, Soft Computing, Vol.25, No.3, pp2277–2293, 2021. DOI: 10.1007/s00500-020-05297-6.

[39] JavaTPoint. "Decision Tree Algorithm in Machine Learning" https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed Nov. 10, 2023).

[40] Bansal, M., Goyal, A., & Choudhary, A. “A Comparative Analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short-Term Memory Algorithms in Machine Learning”, Decision Analytics Journal, Vol.3, pp.100071, 2022. DOI: 10.1016/j.dajour.2022.100071.

[41] GeeksforGeeks. "Support Vector Machine SVM Algorithm" https://www.geeksforgeeks.org/support-vector-machine-algorithm/ (accessed Nov. 10, 2023).

[42] Analytics Vidhya. "Understand Random Forest Algorithms with Examples" https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (accessed Nov. 10, 2023).

[43] JavaTPoint. "Logistic Regression in Machine Learning" https://www.javatpoint.com/logistic-regression-in-machine-learning (accessed Nov. 10, 2023).

[44] Analytics Vidhya. "Master the AdaBoost Algorithm: Guide to Implementing & Understanding AdaBoost"

Authors’ Profiles

Najla Odeh, Palestine Technical University Kadoorie / Computer Science Department, Faculty of Information Technology, Tulkarm, Palestine (ORCID ID https://orcid.org/0000-0003-1089-9243).

Received the B.S. degree in computer information systems from the Al-Quds Open University, Tulkarm, Palestine, in 2011. She is currently working toward an M.S. degree in Computer Science at Palestine Technical University Kadoorie PTUK. She was a Programmer and web designer at IDEX Company, Tulkarm, Palestine. From 2011 to 2015. Currently, she is working as Technical Support at PTUK. Her specific areas of research interest mainly focus on machine learning, network technologies, information systems, and network security.

Professor Derar Eleyan, Palestine Technical University Kadoorie / Computer Science Department, Faculty of Information Technology, Tulkarm, Palestine (ORCID ID https://orcid.org/0000-0001-8876-7019).

Derar Eleyan has a good relevant diversity experience in academics and industry. He is the manager of the Erasmus+ project “Pathway in Forensic Computing” and an associate professor in information systems. Eleyan is currently working as the president assistant for international academic cooperation. He served five years as an assistant professor at Birzeit University in the Department of Computer Science teaching a variety of courses at the undergraduate and postgraduate levels. He has served as an external reviewer to some conferences in information science, information systems, and business process modeling. His research interests focus on System dynamics, software quality, Information Systems, Business Process Modelling, Customer service and satisfaction and return on investment, information technology management, IT project management, Quality of Service, Academic quality and performance evaluation, Business and computer ethics, Software testing quality assurance, Usability and e-commerce. He is a member of some societies as the British Computer Society (BCS), the Institute for Learning (IFL), Systems Dynamics Society (SDS).

Dr. Amna Eleyan, Manchester Metropolitan University / Department of Computing and Mathematics, Manchester, United Kingdom (ORCID ID https://orcid.org/0000-0002-2025-3027).

Amna Eleyan (Member, IEEE) received a Ph.D. degree in software engineering from The University of Manchester. She is a Lecturer with the Department of Computing and Mathematics, at Manchester Metropolitan University. Her research interests include computer networks and security, web services, the IoT, and machine/deep learning. She is a fellow of the Higher Education Academy.

How to cite this paper: Najla Odeh, Derar Eleyan, Amna Eleyan, "Enhancing Web Security through Machine Learning-based Detection of Phishing Websites", International Journal of Computer Network and Information Security(IJCNIS), Vol.17, No.1, pp.39-56, 2025. DOI:10.5815/ijcnis.2025.01.04