I.J. Wireless and Microwave Technologies, **2025, 3, 37-53**
Published Online on June 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2025.03.03

Detection and Prevention of Phishing Short URLs Using Machine Learning and Blacklist Approaches

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

Sherin Hijazi
Palestine Technical University – Kadoorie, Computer Science Department, Faculty of Information Technology, Tulkarm, P.O Box 305, Palestine
E-mail: s.hijazi@ptuk.edu.ps
ORCID iD: https://orcid.org/0000-0003-2411-5681

Received: 18 December, 2024; Revised: 09 January, 2025; Accepted: 05 March, 2025; Published: 08 June, 2025

Abstract: Phishing attacks are a common and serious issue in our digital age, short uniform resource locators are frequently used in these attacks to trick unwary visitors into visiting malicious websites. Short uniform resource locators are often used to hide a link's true destination, making it harder for visitors to establish whether a link is legitimate or phishing. Due to this, individuals and organizations attempting to protect themselves from phishing attempts have a significant problem. This research introduces a novel system that integrates machine learning algorithms with a blacklist approach to enhance phishing detection. The system's objective is to support transparency protect user privacy, and increase the precision and efficiency of identifying phishing attacks hidden behind Short URLs, thereby granting users real-time protection against phishing attacks. The findings demonstrate that the proposed system is highly effective. Many machine learning algorithms were used and compared, Gradient Boosting emerged as the best algorithm among those tested, with an excellent accuracy rate of 97.1%. This algorithm outperformed other algorithms in distinguishing between legitimate and phishing uniform resource locators, demonstrating its strong capabilities in the face of the growing threat landscape of phishing attacks via short uniform resource locators. By addressing gaps in prior research, particularly in detecting phishing using short URLs, this study provides a valuable contribution to cybersecurity.

Index Terms: Cybersecurity, Machine Learning Algorithms, Short URLs, Security and Privacy, Phishing Attacks

1. Introduction

Social media networks are rapidly evolving and have a vast user base that frequently utilizes these platforms. Unfortunately, many of these users have a limited understanding of privacy and security and are not fully aware of the risks involved in sharing personal information online. Hackers can easily obtain users personal information from social networks, making these platforms a prime target for online fraud. As a result, social networks have become a major venue for cybercriminals to launch their attacks [1]. One of the attacks that users may be exposed to is phishing. Despite extensive research on phishing, many existing solutions fail to adequately address the complexities of phishing attacks involving short URLs. This study aims to bridge this gap by introducing a system that combines blacklists and machine learning to enhance phishing detection accuracy.

1.1 Phishing Attacks

With the exponential growth of the internet and the widespread utilization of email and online services, the threat of phishing attacks has become an increasingly pervasive concern within the domain of cybersecurity. Phishing can be

classified as a form of cyberattack wherein the attacker takes advantage of social engineering techniques to execute identity theft [2]. These attacks exploit human vulnerabilities and frequently incorporate meticulously crafted messages that entice users to click on malicious links or download harmful attachments. Given the rising complexity of phishing attacks, conventional security measures are struggling to keep pace, necessitating innovative and adaptable solutions to effectively counter this danger.

The term "phishing" can be traced back to the term "fishing," as the operation of phishing bears similarity to fishing in that the attacker uses a "bait" and "fish" to lure the victim and extract personal or confidential information about them [3]. The individual who perpetrates the phishing attack is commonly referred to as a "phisher". The primary objective of phishing is to obtain sensitive and confidential information from victims, including but not limited to usernames, passwords, credit card numbers, and monetary fund's [2].

According to Verizon's data breach study report, 90% of cyberattacks begin with phishing [4]. A report by APWG 262,704 phishing attacks were reported in Q1 2018, up from 233,613 in Q4 2017 [5]. Phishing URLs are on the increase, and it poses a frightening threat to both organizations and consumers. According to Statista research [6], the number of unique base URLs of phishing websites increased by nearly 3.7 times in Q1 2021 compared to Q1 2020. URL phishing is a popular vector of infection used by attackers due to its high success rate and low cost. Fig. 1 shows the URL phishing growth from 2013 to 2022.

Fig. 1. URL phishing growth from 2013 to 2021 [6].

Phishing is a common vector for cybercrime and hacking, and it is successful because it effectively exploits human vulnerabilities. Cybercriminals employ a range of strategies to create fraudulent emails that tempt recipients to perform destructive actions such as clicking on false links. Cybercriminals employ techniques such as generating a sense of urgency while promising a reward, sending notifications, using an authoritative tone, demonstrating shared interest, and consistently employing certain attack strategies [7, 8].

1.2 Short URLs

URL shortening is the process of converting a lengthy URL into a shorter alternative that redirects to the original long URL [9]. The first URL shortening service, TinyURL, was introduced by Kevin Gilbertson in 2002 to simplify sharing links on his website [10]. TinyURL is still widely used today, along with other popular URL shorteners like bitly.com and cutt.ly.

In recent years, there has been a notable significant rise in the popularity of shortened URLs. They provide a convenient means to share lengthy web addresses on social media platforms, advertisements, and messaging services with limitations on character count. The attraction of URLs lies in their brevity, which facilitates their remembrance and dissemination. It also prepares short URL services are a prime example of a service that frequently provides analytics as public information for their shortened URLs. Some URL shortening providers, such as Bitly, offer the ability to view real-time click traffic of a given short URL, including information about referrers and the countries that are referring to it. This feature can provide users with valuable insights into how their short URLs are being used and can be used to optimize their online presence [11]. A user can establish as many unique short URLs that connect to the same long URL as he likes, and a long URL can be linked to numerous short URLs. However, the adoption of URL shortening was slow until it became popular on social media networks. Nowadays, URL shorteners are almost necessary due to character limitations on certain social media websites and mobile devices where space is always limited.

Using short URLs presents a range of potential risks. One of the main problems is the lack of transparency associated with these short URLs. Short URLs hide the original web address, creating a situation where users are left in the dark as to the actual destination they are about to reach, creating an element of confusion. Attackers exploit the ease of short URLs to redirect victims to phishing websites or start downloading malware onto their devices. Violation of privacy is another major concern. The information contained in these links, including statistics about the site owners, such as the number of visitors who have clicked on the link, may be accessed by unauthorized parties. The lifespan of short URLs also poses a unique challenge. Short URLs are temporary, and their validity depends on the continued existence of the service provider or the original link owner. When these terms change, short URLs can become outdated, rendering them ineffective for their intended purposes. Therefore, relying on short URLs carries a risk associated with their instability and dependence on external factors. According to the study conducted by [12], the results show that the Twitter URL shortener is not very good at filtering phishing and malware URLs, leaving users on Twitter exposed to these online threats. Short URLs have been used in phishing attacks, spam, scams, and malware distribution [13]. Cybercriminals use short URLs to take advantage of Twitter's limited text space and hide the destination of URLs [14, 15].

The life cycle of phishing using short URLs begins by designing a phishing website so that it appears to be a legitimate website. On the one hand, attackers fake the URL of the legitimate website, particularly the domain name and network resource directory, using spelling errors, similar alphabetic characters, and other methods. The link "https://twitter-cc.xyz/xyz" for example, imitates “https://www.twitter.com”, this is followed by shortening the link and then sending it via e-mail or posting it on social networks. Although the browser on the computer can see the URL address by moving the mouse to the clickable link, it is difficult for the ordinary user to identify these URLs as impersonating legitimate URLs with the naked eye and memory. On the other hand, imitation of web content is an important consideration. Scripts are commonly used by attackers to obtain logos, web layouts, and text from legitimate web pages. Cybercriminals frequently impersonate form submission pages that require sensitive information from the user, such as the login page, payment page, and find password page [16]. Fig. 2 shows the life cycle of phishing short URLs.

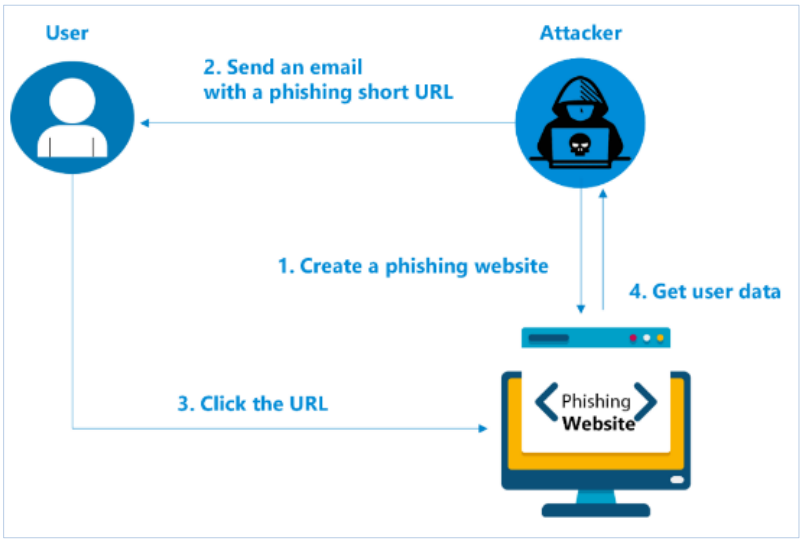

Fig. 2. Phishing short URL life cycle.

1.3 Statement of the Problem

The problem is the lack of an effective system that can detect and prevent phishing attempts using short URLs. Traditional URL analysis approaches are often insufficient for identifying phishing URLs, as attackers constantly develop their ways to avoid detection. As a result, there is an increasing demand for an intelligent solution that uses machine-learning techniques to analyze and classify short URLs as legitimate or phishing.

The main danger of short URLs is their potential to redirect users to malicious websites. By hiding the true destination of a link, according to the Global Phishing Survey [17], phishers continue to use URL shortening services to hide phishing URLs. Phishing websites often use short URLs to trick users into clicking on malicious links. These websites typically create short URLs that appear to be legitimate, such as a popular social media or banking website, but redirect users to a fake website designed to steal their personal information. This can lead to serious security breaches and financial losses for both individuals and businesses. Another issue with short URLs is their lack of transparency. Because they hide the original URL, it can be difficult for users to know what website they are visiting this can lead to confusion. Thus, there is an urgent need to address the risks associated with URL-shortening services and develop effective measures to protect users from potential threats.

1.4 Research Contribution

Despite the considerable steps made in the domain of cybersecurity, the detection and prevention of phishing attacks, particularly those utilizing short URLs, continue to impose considerable challenges. Existing research emphasizes the need for more strong and sophisticated approaches to counter the constantly evolving strategies employed by cybercriminals. By explaining this research gap, our aim of this investigation is to design and implement a Phishing Short URL Detection and Prevention System (PSUDPS) that employs blacklists and machine learning algorithms. The system's objective is to support transparency protect user privacy, and increase the precision and efficiency of identifying phishing attacks hidden behind Short URLs, thereby granting users real-time protection against phishing attacks. This solution advances knowledge in the domain of cybersecurity and provides valuable insights for developing future phishing prevention mechanisms.

2. Related Work

The use of short URLs in phishing attempts presents a unique challenge to typical security measures. This section examines methods used to detect and prevent phishing attacks that use short URLs and emphasizes the importance of specialized detection and protection measures.

The authors [1] discuss the widespread use of social media networks and how they have become an important platform for people to share and obtain information. it also highlights that social networks are vulnerable to online fraud, and phishing is a common tactic used by hackers. The paper proposes a method for detecting phishing short URLs on social networks by using a hierarchical hidden Markov model. The method includes a training phase and an identification phase, and the effectiveness of the method is validated through an experiment based on real datasets from Weibo. The experimental results show that the method can effectively detect the phishing short URL, and HHMM is better than HMM in describing the link-jumping process, the accuracy of the HHMM model was 96%.

The paper [12] investigates the effectiveness of Twitter's URL shortening service (t.co) in protecting users from phishing and malware attacks. According to the study, a significant number of blacklisted phishing and malware URLs were posted on Twitter, resulting in a large number of clicks from users and potential exposure to cyber-attacks. While the number of blacklisted URLs posted on Twitter has decreased over time, the analysis shows that Twitter's URL shortener is not particularly effective at filtering these harmful URLs. Only a small percentage of blacklisted URLs are blocked, indicating that Twitter users are still vulnerable to phishing and malware attacks.

Pattewar and others [9] performed a survey Short URLs and the mechanism for creating short URLs that are easier to remember and use. It also proposes the use of tabu search and gradient descent search for optimizing weights in creating short URLs. The traditional way of detecting malicious short URLs through blacklists is insufficient, thus, the paper suggests a proposed system that not only detects but also analyzes malicious short URLs using blacklists and host-based features. Overall, the paper emphasizes the need for a more effective system to detect and prevent malicious actions that can be performed during the redirection of short URLs.

Le Page and others [11] propose the use of URL shortener click analytics to compare the life cycle of phishing and malware attacks. The study collected over 7,000 malicious short URLs categorized as phishing or malware for the 2-year period covering 2016 and 2017. The analysis found that phishing attacks are most active 4 hours before the reported date, while malware attacks are most active 4 days before the reported date. The study also showed that phishing attacks have a higher click-through rate with a shorter time span, while malware attacks have a lower click-through rate with a longer time span. Based on the observation that 50% of malware attacks have been active for several years, while less than 50% of phishing attacks have been active past 3 months, the study concludes that the efforts against phishing attacks are stronger than the efforts against malware attacks.

The popularity of social networks has attracted cyber-criminals to spread spam and malware through fake profiles and stolen legitimate accounts, often employing short URLs to redirect users to malicious websites. The authors Venkatesh, Rout, and Jena [18] suggested an algorithm for controlling the spread of spam and malware, a trust score was calculated for each user to determine their reliability. Using this trust score, the proposed algorithm achieved 92.6% accuracy and an F-measure of 81% in detecting malicious activities. The trust score was calculated based on trending topics followed by users and successfully detected malicious users.

The study [19] proposes a solution to the vulnerability of short URL services that are commonly used to transmit information via SNS and SMS. Attackers can use short URLs to distribute malicious code through Phishing, Smishing, and drive-by-download attacks. The problem with short URLs is that one cannot determine the target URL until one click on it, making it difficult to know whether it is a web document or a file that could download malicious code. The proposed solution is to write destination information when generating a short URL so that the user can verify whether it leads to a web document or a file. Short URL service providers can also monitor the risk of the target URL page and decide whether to provide the service. By measuring and evaluating the risk of the webpage and blocking the short URL according to a threshold, it can prevent attacks such as drive-by downloads through the short URL. The study suggests applying verification-based technology to the existing system that generates short links to resolve the vulnerability of the short URL.

The purpose of paper [13] is to investigate the detection of malicious shortened URLs on Twitter, a popular Online Social Network. URL shortening services are popular due to Twitter's character limit, but they pose security risks because users cannot determine the destination of shortened URLs. The study proposes a novel approach to identifying malicious short URLs that rely solely on visible features from tweets and user profiles. Using random forest for classification, the approach achieves up to 97% accuracy when tested against four machine learning algorithms. The findings demonstrate the viability and efficacy of using visible attributes from social networks to detect malicious URLs.

In paper [20], they examine the hazards associated with ad-based URL-shortening services, which provide compact URL aliases and display ads to link-clicking users while paying commissions to link-shortening users. These services face additional risks compared to traditional URL shortening services due to their monetary incentives and the presence of third-party advertising networks. The paper analyzes these services, their advertisers, and users and identifies issues that are actively exploited by malicious advertisers, putting users at risk. The paper suggests defense mechanisms that services and users can use to protect themselves from these hazards.

In study [21], the introduction highlights the widespread of spam URLs on email and social media platforms, as well as the function of URL shorteners like Bitly in disguising dangerous material. The research examines a series of suspicious Bitly short URLs, demonstrating flaws in Bitly's spam detection capabilities. The researchers suggest a method for determining if Bitly URLs are malicious or benign, with an accuracy of 86.41%. The conclusion emphasizes issues such as spammers abusing Bitly's policies, the ineffectiveness of detection services, and the creation of a technique to detect bad links on Bitly utilizing certain features.

The study [15] investigates the use of URL shortening services by cybercriminals to hide malicious URLs and the impact of such abuses on web users. The authors found that existing countermeasures by popular shortening services were ineffective, and they collected a large dataset of 24,953,881 distinct short URLs to study the abuse of short URLs. They discovered that users are seldom exposed to threats spread via short URLs and that in-the-browser defense tools such as blacklists can alert users before visiting malicious URLs, regardless of whether they are short or long URLs. Additionally, users exhibit different usage patterns depending on the type of content behind short URLs.

The paper [22] discusses the misuse of URL shorteners by spammers to camouflage and improve click-through rates. By analyzing click traffic data from Bitly, the authors investigate the characteristics of spam and non-spam short URLs, determine the top click sources for each, and develop a classification algorithm. The Random Tree algorithm achieved the best performance with an accuracy of 90.81% and an F-measure value of 0.913.

Klien and Strohmaier [23] discusses the usage of URL shorteners in social media and the problem of spam. The authors analyze a URL shortener service operated by their group and find that 80% of the shortened URLs contained spam. They also reveal that this problem has an international scale and that there are imbalances between creating and resolving short URLs. The authors suggest that sophisticated algorithms are needed to identify URL spam and call for future research into understanding spam behavior in this new domain.

The study [14] examines the use of URL shorteners by phishers in online social media to steal personal information from users. The authors found that phishers use URL shorteners not only to reduce space but also to hide their identity and that social media websites like Facebook, Habbo, and Orkut are among the top brands targeted by phishers. They also discovered that a majority of references to phishing tweets from Twitter are from inorganic accounts that use attractive words and multiple hashtags to spread their message. According to the researchers, this was the first study to correlate blacklisted phishing URLs from PhishTank, URL stats from bit.ly, and signals from Twitter to track the impact of phishing in online social media.

Neumann, Barnickel, and Meyer [24] conducts an empirical investigation of the security and privacy risks associated with the use of URL-shortening services. It identifies and investigates the most prominent URL shortening services on Twitter for harmful activity, user tracking, and URL leakage to search engines. A new attack scenario is also included, allowing SSL-only circumvention through SSLStrip and truncated URLs. The paper also empirically evaluates the use of URL shortening services in over 7 million spam emails gathered over the last seven years, evaluating the performance of the most prominent services in terms of spam detection. The results show that, while none of the prominent URL-shortening services exhibit malevolent behavior, many are well-prepared for user tracking. The investigation demonstrates the possibility of sensitive information being discovered through the enumeration of URL shortening services and the leakage of submitted URLs to search engines.

PhishTank Usage and Limitations

2.1 Discussion

Given the above studies, short URLs represent a significant security risk for users, as they can be used to hide malicious links, redirect users to phishing websites, and spread malware. The studies reviewed in the literature looked into various approaches to phishing. Machine learning, behavioral analysis, and user education were all included. The significant progress made in understanding the characteristics and risks associated with phishing attacks is one notable aspect. While these studies have taken commendable steps in the field, several critical observations need to be made. Some of the reviewed studies were limited in scope, additionally, most of the studies focus on the use of blacklists to

stop short URLs threats or use some URL-specific features to detect phishing websites. This suggests that a more specialized approach may be required to effectively combat phishing short URL attacks. One notable gap in the existing literature is the lack of comprehensive studies specifically targeting phishing short URLs. Despite the growing use of short URLs in phishing campaigns, most studies either mention them briefly or completely ignore them. This represents an important gap in research and emphasizes the need for a more customized approach. Our research stands out by focusing on the challenges associated with phishing short URLs. In this investigation, we introduce a system to detect phishing short URLs using a proposed approach that combines blacklists and machine learning algorithms by selecting a variety of features to identify and then prevent phishing websites. In Table 1, we show systems developed to protect users from the dangers of phishing short URLs.

Table 1. Phishing short URL detection and prevention systems.

System	Show Destination URL	Detect phishing short URL	Detect method	Prevent phishing short URL
Xie, Li, & Na [1]	No	Yes	Hierarchical Hidden Markov Model	No
Bell & Komisarczuk [12]	No	Yes	Blacklists	Yes
Nepali & Wang [13]	No	Yes	Machine learning Algorithms (NB, RF, SVM, LR)	No
Gupta, Aggarwal, & Kumaraguru [21]	No	Yes	Machine learning algorithms (NB, DT, RF)	No
Chhabra, Aggarwal, Benevenuto, & Kumaraguru [14]	No	Yes	Blacklist	No
PSUDPS (Proposed system)	Yes	Yes	Blacklist and Machine Learning algorithms (GB, RF, DT, KNN, SVM, NB, LR, AdaBoost)	Yes

3. Proposed System

The proposed system assists in avoiding the risks of phishing URLs by verifying the safety of the short URL with a PhishTank database and machine-learning detectors. Fig. 3 shows the proposed system (PSUDPS) flowchart which consists of two phases: detection and prevention.

3.1 Detection Phase

In this phase, we show how the proposed system detects phishing attacks through six steps as follows: the first step is to enter a short URL. The second step is to determine whether the URL is valid. The proposed system checks the validity by checking whether the short URL starts with Hypertext Transfer Protocol Secure (HTTPS) and its length is between 20 and 25 characters. In the third step, we unshorten the URL and return it to its original form so that it shows the user the real destination they are going to. In the fourth step, we used PhishTank databases to determine whether the URL leads to a phishing website. PhishTank is an online service that collects and verifies user and expert reports of phishing websites. A phishing website pretends to be a legitimate website to trick into revealing sensitive information such as passwords, credit card numbers, or bank accounts. If the URL leads to a phishing website, the proposed system blocks it. In the fifth step, we check the URL safe by using a machine learning detector. A machine learning detector is a program that analyzes the characteristics of a URL using machine learning algorithms and classifies it as phishing or legitimate. In our research, we used eight machine learning algorithms, and then we adopted the best algorithm in terms of accuracy, which is the GB algorithm. Based on the results of the algorithm, if the URL is classified as a phishing URL, the proposed system will block the website. The sixth and final step is to allow the user to continue to the website if the URL passes all previous checks.

The primary difference between PhishTank and a machine learning detector is that the former is a service that relies on human reports and verification, whereas the latter is a program that relies on artificial intelligence and data analysis. PhishTank has the advantage of speed in detecting phishing sites based on user feedback and the presence of the URL in the database but is less accurate than a machine learning detector since the file might not contain the latest or most comprehensive phishing data. The advantage of using a machine learning detector is that it can detect new or unknown phishing URLs based on their features and patterns, but it also has the disadvantage of being unable to handle dynamic and complex phishing websites that change frequently. Therefore, using both PhishTank and machine learning detectors can provide a more comprehensive and effective way of checking if a short URL is phishing or not.

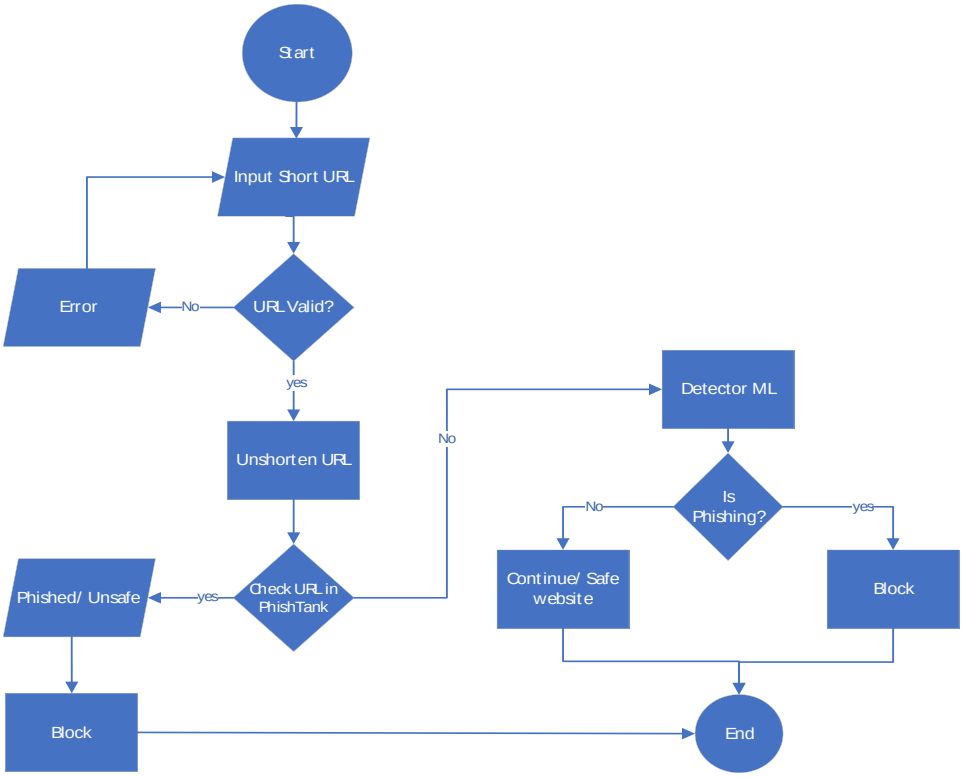

Fig. 3. Flowchart of PSUDPS.

3.2 Prevention Phase

Preventing users from accessing phishing websites is critical in limiting the risks associated with these attacks. This research provides a mechanism for proactively restricting access to phishing websites using the Windows host file. After detecting a phishing website in the previous phase, the proposed system blocks the phishing website either using blacklists or a machine learning algorithm. The host file in Windows is a critical system file that maps hostnames to IP addresses. It is normally located at "C:\Windows\System32\drivers\etc\hosts". Fig. 4 shows the path to the host file in the Windows system.

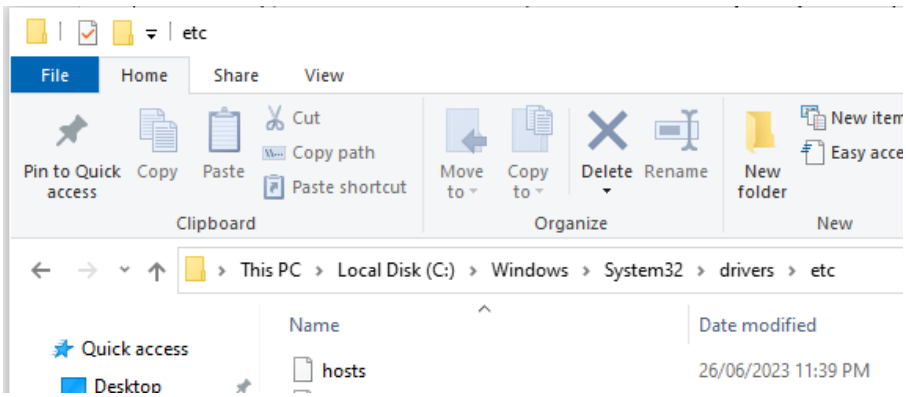

Fig. 4. Host file path.

The process of preventing phishing websites using the Windows host file is an effective approach to prevent access to phishing websites. When a phishing website is identified, the system modifies the host file to create an entry to connect the website's domain with the localhost IP address, 127.0.0.1. This redirection sends any requests for the phishing website back to the user's own computer rather than reaching the phishing server. As a result, users are effectively prevented from accessing the phishing website. Through this technique, the system provides a proactive measure for protecting users without requiring additional software. This technique leverages the functionality of the host file and ensures that access to identified phishing websites is blocked at the operating system level, offering a user-transparent layer of defense. Fig. 5 shows an example of phishing websites we blocked.

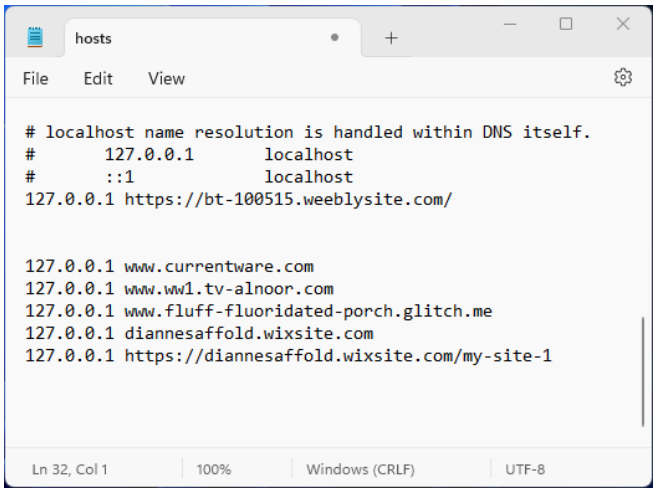

Fig. 5. Blocked websites.

Python we used to automate the process of adding host file entries. To modify files and conduct system-level activities, the 'os' and 'sys' libraries are utilized. The Python script reads the host file, checks for existing entries, and appends new entries if the phishing website is identified. Error handling is also included in the code to ensure the host file's integrity.

The proposed system uses the Windows host file to stop short phishing URLs. However, this strategy has certain limitations, such as the need for administrative access to modify the file, potential conflicts with other security software, and the chance that attackers may circumvent local host file changes.

4. Experimental and Methodology

The experimental section of this research aims to design and develop a PSUDPS based on machine learning using the GB algorithm and PhishTank blacklist techniques. The goal is to evaluate the system's performance in detecting and preventing phishing attacks using short URLs. In this section, we discuss five topics: The techniques used (PhishTank and Algorithms), Dataset used, Feature extraction and selection, and Training dataset.

4.1 PhishTank Blacklist

A blacklist is a collection of elements that must be blocked; it is an access control list. Our research looks at phishing blacklists that are used to prevent access to malicious URLs. This study focuses on the PhishTank blacklist. Blacklisting approaches are a common and traditional technique for detecting malicious URLs, and they frequently keep a list of known malicious URLs. A database lookup is performed whenever a new URL is visited. If the URL is on the blacklist, it is considered malicious and a warning is generated; otherwise, it is assumed to be benign. Because new URLs can be easily generated on a daily basis, blacklisting suffers from the inability to maintain an exhaustive list of all possible malicious URLs, making it impossible to detect new threats. This is especially concerning when the attackers generate new URLs and thus bypass all blacklists [25].

PhishTank is a community-based phishing website reporting and verification system that was launched in October 2006. Users of the website can submit URLs of suspected phishing websites, and the Phish Tank community votes on whether or not these URLs are phishing [26] (PhishTank, 2023a). In order to classify a URL as a phish, PhishTank requires four votes from users. Once confirmed, the phish is added to the central blacklist. The blacklist is sometimes downloaded to local computers.

4.2 Algorithm Used

In this research, we compared the performance of eight classifiers utilized as machine-learning methods for the suggested system: Gradient Boosting (GB), Random Forest (RF), Decision Tree (DT), K Nearest Neighbor (KNN), Support Vector Machine (SVM), Na ïve Bayes (NB), Logistic regression (LR), and AdaBoost.

Gradient Boosting (GB): GB is a common boosting algorithm in machine learning that is used for classification and regression applications. Boosting is a type of ensemble Learning method that trains the model consecutively, with each new model attempting to correct the preceding model. It combines numerous weak learners into one powerful learner. Gradient boosting combines groups of relatively weak prediction models to create a better prediction model. This algorithm is an effective method for developing prediction models [27]. This algorithm's application varies widely across numerous sectors, including data management systems. Gradient boosting has demonstrated successful practical applications in a variety of machine learning and data mining difficulties, including cryptocurrency theft, power grids, neurorobotics, and so on. Gradient boosting is a common machine-learning approach [28].

Gradient Boosting detected short phishing URLs with the highest accuracy (97.1%), making it the most successful algorithm in this research. However, GB, like any machine learning model, has limits. One major worry is its potential for overfitting, particularly when trained on static datasets that do not respond to changing phishing strategies. Furthermore, GB demands significant processing resources, which may limit its real-time use in large-scale applications. In dynamic contexts where phishing methods are always evolving, periodic retraining with updated datasets and the incorporation of adaptive learning processes are required to maintain high detection accuracy.

Random Forest (RF): 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 [29].

Decision Tree (DT): 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 [30].

K Nearest Neighbor (KNN): One of the nonparametric supervised predictive modeling techniques is KNN classifier. It is, in particular, one of the simplest yet most practical methods of data classification [31]. 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 [32]. The Euclidean distance between two points with coordinates (x, y) and (a, b) is calculated as seen in Eq (1).

$$Dist\ ((x,y),(a,b) = \sqrt{(x-a)^2 + (y-b)^2)} \tag{1}$$

Support Vector Machine (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 [33].

Naïve Bayes (NB): The NB 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 [34]. Eq (2) is used to calculate the later probability of class c for a dataset with feature vector X = (x1, x2, ... xn), given predictor attribute x. 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.

$$P\ (c|x) = \frac{P\ (x|c)P(c)}{P(c)} \tag{2}$$

Logistic regression (LR): LR is a Machine Learning classification algorithm that predicts the likelihood of specific classes based on some dependent variables. The logistic regression model computes a sum of the input features (usually with a bias term) and the logistic of the result. The logistic regression output is always between 0 and 1, which is appropriate for a binary classification task. The greater the value, the more likely the present sample is categorized as class=1, and vice versa [35].

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 [36].

4.3 Dataset

Data collection involved utilizing a dataset sourced from the UCI Machine Learning Repository [37] to classify the URLs as legitimate or phishing websites. UCI Machine Learning Repository has a plethora of diverse datasets that address many elements of supervised and unsupervised machine learning. We used the UCI dataset for experimentation, which contained 11,055 websites categorized into those employed for phishing activities and others deemed legitimate. Each website in the dataset is assigned 30 features, which include characteristics such as domain-based, anomalous, HTML and JavaScript-based, and URL-based features. The dataset contains only categorical features, and the categories are assigned symmetrical values. The "Result" class label contains 6157 instances of '1' representing phishing websites and 4898 instances of '-1' representing legitimate websites. Fig. 6 shows a visual depiction of the UCI dataset distribution.

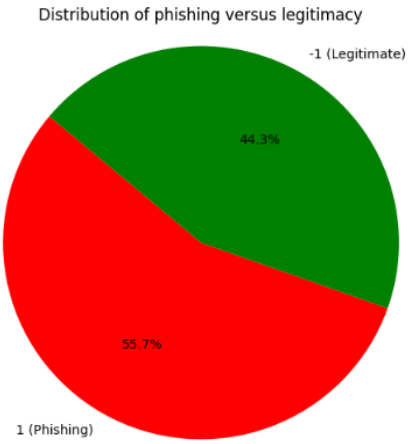

Fig. 6. Distribution of data in UCI dataset.

The Fig. 6 of the chart employs red for the "Phishing" category and green for the "Legitimate" category, making it easy to distinguish between them. The chart depicts the relative prevalence of each category and provides a clear summary of the dataset, approximately 55.7% of the cases are classed as phishing, whereas around 44.3% are classified as legitimate. As a result, this is a reasonably balanced dataset.

4.4 Feature Extraction and Selection

These features were chosen based on their relevance in identifying phishing attempts, including domain-based features, anomalous features, HTML and JavaScript features, and features based on URLs [37]. For instance, URL length, presence of special characters, and domain age are crucial factors in distinguishing phishing from legitimate sites.

These features are used to examine the behavior of the website and detect any malicious activity or purpose that might be dangerous. The chosen features played a crucial role in reviewing website behavior, enabling the detection of potentially malicious activities or purposes that pose a threat.

4.5 Training the Dataset

A training dataset serves as the foundation for training machine learning models. An effective approach to assess algorithm performance on a comparable problem involves creating a train and test split of the dataset. The training dataset is utilized for model preparation and training. Classification algorithms, namely GB, RF, DT, KNN, SVM, NB, LR, and Adaboost, were trained using the provided training data in this study. It is worth noting that 80% of the data was set aside for training, while the remaining 20% was set aside for testing and evaluating the model. This split guarantees that both processes are carried out independently using separate datasets to produce the most trustworthy estimates of the models' performance. This method was used in the study to avoid overfitting, which occurs when a model thrives on training data but performs badly on test data.

5. Implementation

The proposed system for PSUDPS involves a comprehensive implementation process. In this section, we will show the proposed system interfaces. The following figures (7,8,9,10) show the user interface for the PSUDPS as appearing to the user with a description of every interface:

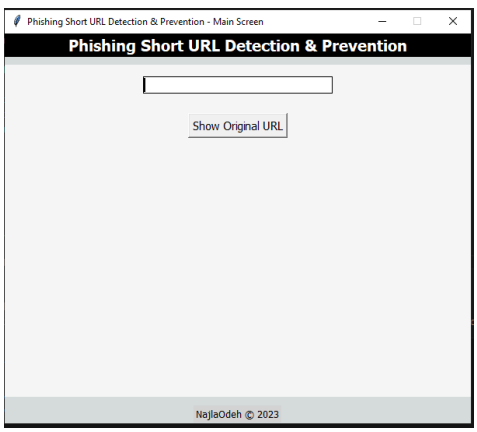

Fig. 7. Main interface.

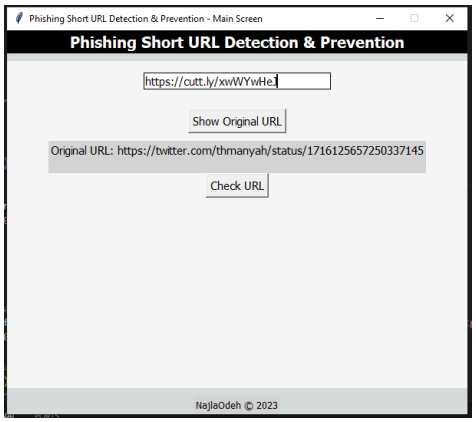

Fig. 8. Show original URL.

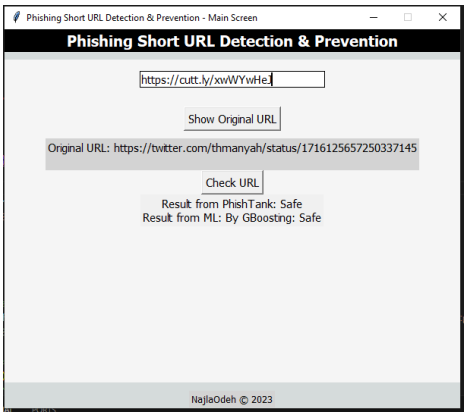

Fig. 9. Check result by PhishTank and machine learning (safe).

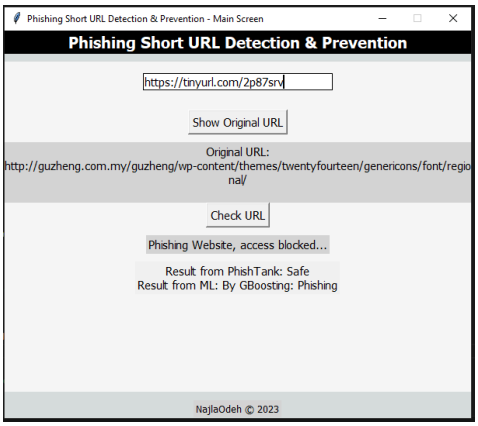

Fig. 10. Check result by PhishTank blacklist and machine Learning (phishing).

6. Evaluation

The experimental evaluation section of this research aims to evaluate the system's performance in detecting and preventing phishing attacks using short URLs.

6.1 Algorithm Evaluation Metrics

Machine learning models can be used to detect phishing websites, and the accuracy of these models is dependent on the datasets used for training and testing, the features extracted from websites, and the algorithms used. In this research, we compare the performance of eight machine learning algorithms and their application to the proposed system: GB, RF, DT, KNN, SVM, NB, LR, and Adaboost. Using evaluation metrics including the Precision, Recall, F1-score, Accuracy, Matthews Correlation Coefficient (MCC), and Receiver Operating Characteristic (ROC) Curve, we compared the different algorithms. 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 2.

Table 2. Confusion Matrix.

Class	Phishing	Legitimate
Phishing	TP	FN
Legitimate	FP	TN

Description of the fields of the confusion matrix:

- True Positive (TP): Number of phishing websites classified correctly as phishing
- True Negative (TN): Number of legitimate websites classified correctly as legitimate
- False Positive (FP): Number of legitimate websites classified incorrectly as phishing.
- False Negative (FN): Number of phishing websites incorrectly classified as legitimate.

6.2 Evaluation Parameters Used

Common parameter metrics were used to determine the efficiency of the proposed system. Table 3 shows these performance metrics and their effects on model performance.

Table 3. Evaluation Parameters Used.

Parameters	Meaning	Formula
Precision	The proportion of true positive predictions among all positive predictions. The best result is high precision, meaning that when the model predicts positive.	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall	The model's recall score indicates the model's ability to correctly predict positives from real positives. The best result is high recall.	$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}}$
F1- Score	It is the precision and recalls harmonic mean. It gives a quick way to compare classifiers and is between 0 and 1. The best result is a high F1-Score, indicating a good balance between precision and recall.	$\text{F} - \text{Measure} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FN} + \text{FP}}$
Accuracy	It is the percentage of both legitimate and phishing websites that have been accurately detected. The best result is a high accuracy, meaning that the model makes correct predictions for both positive and negative instances.	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} * 100$
Matthews Correlation Coefficient (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.	$\text{MCC} = \frac{\text{TP X TN} - \text{FB X FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$
Receiver Operating Characteristic (ROC) Curve	The ROC curve is a graphical representation that shows the trade-off between the true and false positive rates at different decision points. The AUC value ranges from 0 to 1, with 1 indicating perfect performance and 0.5 indicating random guessing.	$\text{AUC} = \frac{1}{2}(\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$

6.3 Experimental Results

This study extracted results after conducting training and testing the same dataset on several algorithms to classify URL addresses into phishing websites and legitimate websites. Results show that the GB algorithm achieved an accuracy rate of 97.1%. Followed by the KNN algorithm which achieved an accuracy rate of 94%. Table 4 shows the performance results of all algorithms.

Table 4. Performance of algorithms.

Evaluation	Accuracy	Precision	Recall	F1 Score	MCC	AUC
GB	97.1%	96%	98%	97%	0.94	0.9959
RF	93%	92%	96%	94%	0.86	0.9959
DT	92%	92%	93%	93%	0.84	0.9732
KNN	94%	94%	95%	94%	0.88	0.9820
SVM	93%	92%	94%	93%	0.86	0. 9862
NB	60%	99%	26%	42%	0.36	0.9655
LR	92%	92%	93%	93%	0.84	0.9770
AdaBoost	93%	93%	95%	94%	0.86	0.9858

Accuracy it is the percentage of both legitimate and phishing websites that have been accurately detected. Fig. 11 shows the accuracy achieved by various machine learning algorithms in the task of classifying websites into legitimate and phishing categories. The GB algorithm achieves an exceptional accuracy rate of 97.1%, placing it first among the algorithms evaluated. The KNN algorithm comes in second with an accuracy level of 94%, highlighting its strong suitability for this classification task. The RF, SVM, and AdaBoost algorithms achieved 93% accuracy, The LR and DT algorithms achieved an accuracy of 92%, and the NB algorithm achieved an accuracy of 60%, which, although lower, provides useful insights into its performance.

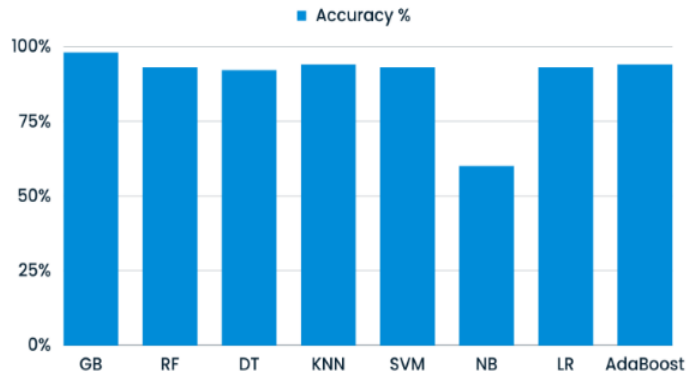

Fig. 11. Accuracy results.

Precision is a critical performance metric that evaluates the accuracy of positive predictions made by machine learning algorithms, as shown in Fig. 12. It expresses the rate of true positive predictions (positive instances that were correctly classified) in relation to all positive predictions generated by the model. The Precision values for the various algorithms in the context of our study are as follows: GB has a Precision rate of 96%, RF, DT, SVM and LR have a Precision rate of 92%, KNN has a Precision rate of 94%, and NB has a Precision rate of 99%, while AdaBoost has a Precision rate of 93%. These Precision results provide useful information about the algorithms' ability to accurately identify positive instances, which is especially important in applications where false positives can have serious consequences. The NB algorithm's notably high Precision indicates its proficiency in minimizing false positives, making it an appealing choice for precise positive classifications.

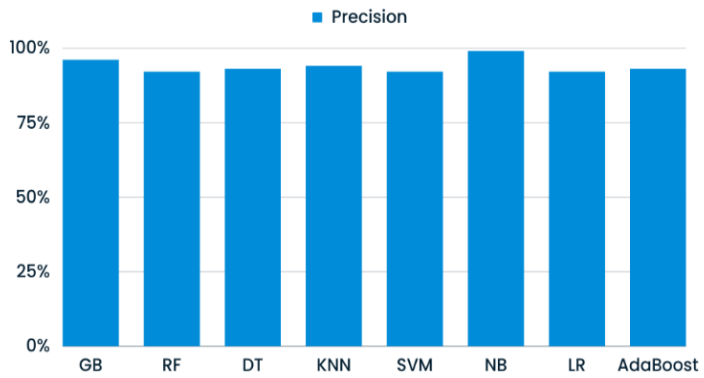

Fig. 12. Precision results.

The recall values for each algorithm are depicted in Fig. 13. Recall is a fundamental performance metric in machine learning that measures an algorithm's ability to correctly identify all positive instances. The recall values for the various algorithms in our study are as follows: GB has a recall rate of 98%, demonstrating its ability to capture the majority of actual positive instances. RF has a strong recall of 96%, demonstrating its ability to effectively retrieve positive cases. The DT has a recall of 93%, demonstrating its accuracy in identifying positive instances. Recall rates of 95% are achieved by KNN, SVM, and LR, emphasizing their ability to capture true positive instances. Additionally, the recall of NB is 26%, indicating potential limitations in recognizing positive cases. The AdaBoost algorithm has a recall rate of 95%, demonstrating its ability to effectively recall true positive instances.

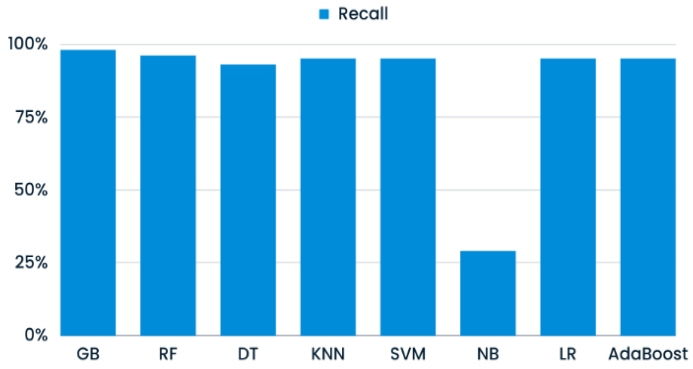

Fig. 13. Recall results.

The F1 Score is a critical performance metric that acts as a harmonic mean between precision and recall, allowing for a balanced evaluation of classification models. The F1 Score values for the various algorithms in our study as in Fig. 14: GB achieves an F1 Score of 97%, indicating a harmonious combination of high precision and recall. RF, KNN, and AdaBoost maintain an F1 Score of 94%, indicating a well-balanced precision and recall performance. The F1 Score of DT, SVM, and LR is 93%, indicating its ability to strike a balance between precision and recall. NB produces an F1 Score of 42%, indicating potential difficulties in achieving a balance between precision and recall.

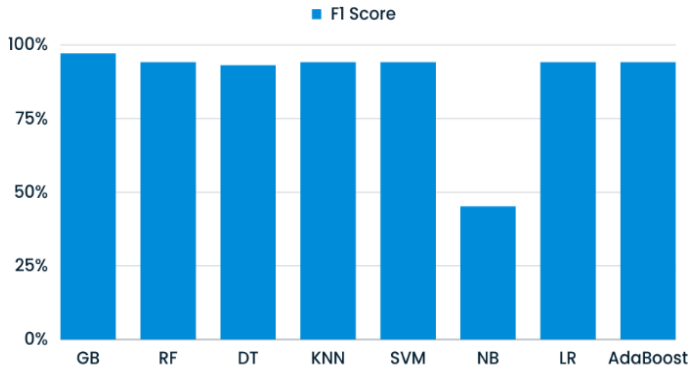

Fig. 14. F1 score results.

The MCC is a reliable performance metric that assesses the accuracy of binary classification models by taking into account both true and false positives and negatives, its value ranges from -1 to 1. Inverse predictions are represented by a coefficient of -1, whereas perfect predictions are represented by a value of +1. The MCC scores are in Fig. 15. range from 0.39 to 0.94. The MCC values for the various algorithms in our study are as follows: GB achieves a significant MCC score of 0.94, indicating a high degree of agreement between predicted and actual classifications. RF, SVM, and AdaBoost have a notable MCC score of 0.86, highlighting their ability to provide reliable classifications. The MCC score of the DT and LR algorithms is 0.84, indicating its ability to achieve good classification agreement. MCC scores of 0.86 are recorded by KNN demonstrating their consistency in delivering accurate classifications. However, NB has a lower MCC score of 0.36, indicating some limitations in its classification agreement.

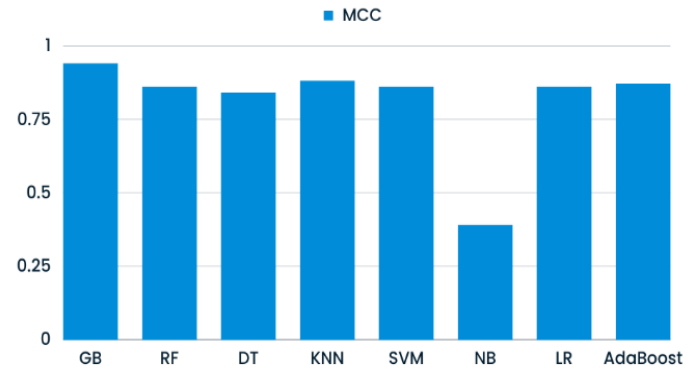

Fig. 15. MCC results.

The following Fig. 16 shows a comparison of the Accuracy, Precision, Recall, F1 score, and MCC results for the eight algorithms used in the search. It shows that the GB algorithm is the best.

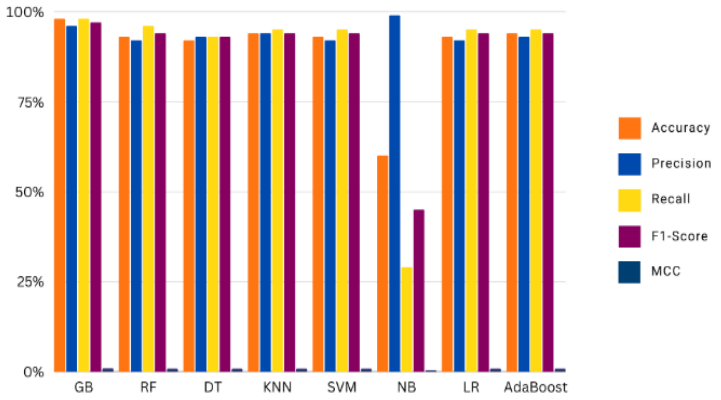

Fig. 16. Comparison of algorithms.

The ROC curve is a graphical tool for evaluating a classification model's performance. It depicts the relationship between the FPR and the TPR at various threshold values for the model. Fig. 17 depicts the ROC curve generated by the GB algorithm on the test dataset. The GB algorithm has an AUC of 0.9959. This value indicates the superiority of the algorithm in classification tasks, with a high ability to distinguish between positive and negative classes. The RF algorithm has an AUC of 0.9959 as well. This score, which is comparable to GB, represents the model's ability to effectively separate classes. The KNN algorithm has an AUC of 0.9819. This score is lower than the previous values, indicating that this model may be less accurate in class separation than the other algorithms. The Adaboost algorithm has an AUC of 0.9858. This result indicates that the model is capable of distinguishing between classes, however, it is slightly lower than the AUC values for GB and RF. Also, the DT and LR algorithms have an AUC of 0.97. As for the NB algorithm, it obtained an AUC of 0.96.

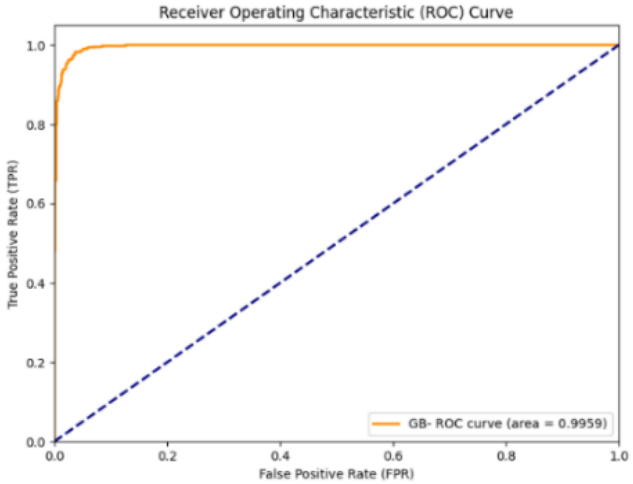

Fig. 17. ROC AUC curve for GB.

7. Conclusion and Future Work

With the rise of digital threats, phishing attacks using short URLs have become a major concern. These links often conceal malicious destinations, making detection challenging for users and organizations. Our research highlights this vulnerability and emphasizes the need for advanced real-time security technologies. Existing studies reinforce the importance of innovative approaches to enhance protection against these evolving cyber threats.

To combat phishing via short URLs, we developed the Phishing Short URL Detection and Prevention System (PSUDPS), combining blacklisting methods with machine learning to enhance accuracy and user security. This system improves transparency by revealing the true destination of shortened URLs, reducing the risk of phishing attacks.

Future research will focus on expanding the dataset developing user-friendly mobile applications to facilitate broader adoption, and integrating deep learning models like CNNs and RNNs to improve phishing short URL detection accuracy. Expanding datasets to cover diverse attack methods and developing real-time detection mechanisms, such as browser-based security features, will strengthen cybersecurity defenses and improve accessibility.

References

[1] B. Xie, Q. Li, and W. Na, "Phishing short URL detection based on link jumping on social networks," in ITM Web of Conferences, 2022, Vol. 47: EDP Sciences. [Online]. Available: https://doi.org/10.1051/itmconf/20224701009.

[2] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," Computers & Security, vol. 68, pp. 160-196, 2017. [Online].

[3] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," Expert Systems with Applications, vol. 106, pp. 1-20, 2018. [Online].

[4] Verizon, "2018 Data Breach Investigations Report," 2018. [Online]. Available: https://www.verizon.com/business/resources/reports/DBIR_2018_Report.pdf.

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

[6] Statista, "Number of unique phishing sites detected worldwide from 3rd quarter 2013 to 34th quarter 2022". Available at: https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/, accessed on 2024-01-13.

[7] P. Rajivan and C. Gonzalez, "Creative persuasion: a study on adversarial behaviors and strategies in phishing attacks," Frontiers in psychology, vol. 9, p. 135, 2018. [Online]. Available: https://doi.org/10.3389/fpsyg.2018.00135.

[8] T. Stojnic, D. Vatsalan, and N. A. Arachchilage, "Phishing email strategies: understanding cybercriminals' strategies of crafting phishing emails," Security privacy, vol. 4, no. 5, p. e165, 2021. [Online]. Available: https://doi.org/ 10.1002/spy2.165.

[9] T. Pattewar, C. Mali, S. Kshire, M. Sadarao, J. Salunkhe, and M. A. Shah, "Malicious Short URLs Detection: A Survey," International Research Journal of Engineering and Technology (IRJET), Vol. 6, No. 11, 2019. [Online].

[10] MetaFilter. "We want 'em shorter". Available at: https://www.metafilter.com/8916/We-want-em-shorter, accessed on 2024-01-03.

[11] S. Le Page, G.-V. Jourdan, G. V. Bochmann, J. Flood, and I.-V. Onut, "Using url shorteners to compare phishing and malware attacks," in 2018 APWG Symposium on Electronic Crime Research (eCrime), 2018, pp. 1-13: IEEE. [Online]. Available: https://doi.org/10.1109/ECRIME.2018.8376215.

[12] S. Bell and P. Komisarczuk, "Measuring the effectiveness of twitter's url shortener (t. co) at protecting users from phishing and malware attacks," in Proceedings of the Australasian Computer Science Week Multiconference, 2020, pp. 1-11. [Online].

[13] R. K. Nepali and Y. Wang, "You look suspicious!!: Leveraging visible attributes to classify malicious short urls on twitter," in 2016 49th Hawaii International Conference on System Sciences (HICSS), 2016, pp. 2648-2655: IEEE. [Online]. Available: https://doi.org/10.1109/HICSS.2016.332.

[14] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi. sh/$ ocial: the phishing landscape through short urls," in Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, 2011, pp. 92-101. [Online].

[15] F. Maggi et al., "Two Years of Short URLs Internet Measurement," 2013. [Online]. Available: https://doi.org/10.1145/2488388.248846.

[16] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," Machine Learning and Knowledge Extraction, vol. 3, no. 3, pp. 672-694, 2021. [Online]. Available: https://doi.org/10.3390/make3030034.

[17] G. Aaron, R. Rasmussen, and A. Routt, "Global Phishing Survey: Trends and Domain Name Use in 1H2014," in Anti-Phishing Working Group, 2015, [Online]. Available: https://docs.apwg.org//reports/APWG_GlobalPhishingSurvey_1H2014.pdf.

[18] R. Venkatesh, J. K. Rout, and S. Jena, "Malicious account detection based on short URLs in Twitter," in Proceedings of the International Conference on Signal, Networks, Computing, and Systems: ICSNCS 2016, Volume 1, 2017, pp. 243-251: Springer. [Online]. Available: https://doi.org/10.1007/978-81-322-3592-7_24.

[19] H.-J. Mun and Y. Li, "Secure short url generation method that recognizes risk of target url," Wireless Personal Communications, vol. 93, pp. 269-283, 2017. [Online]. Available: https://doi.org/10.1007/s11277-016-3866-8.

[20] N. Nikiforakis et al., "Stranger danger: exploring the ecosystem of ad-based url shortening services," in Proceedings of the 23rd international conference on World wide web, 2014, pp. 51-62. [Online].

[21] N. Gupta, A. Aggarwal, and P. Kumaraguru, "bit. ly/malicious: Deep dive into short url based e-crime detection," in 2014 APWG Symposium on Electronic Crime Research (eCrime), 2014, pp. 14-24: IEEE. [Online].

[22] D. Wang, S. B. Navathe, L. Liu, D. Irani, A. Tamersoy, and C. Pu, "Click traffic analysis of short url spam on twitter," in 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013, pp. 250-259: IEEE. [Online].

[23] F. Klien and M. Strohmaier, "Short links under attack: geographical analysis of spam in a URL shortener network," in Proceedings of the 23rd ACM conference on Hypertext and social media, 2012, pp. 83-88. [Online].

[24] A. Neumann, J. Barnickel, and U. Meyer, "Security and privacy implications of url shortening services," in Proceedings of the Workshop on Web 2.0 Security and Privacy, 2010. [Online].

[25] Y.Alshboul, R.Nepali, & Y.Wang, "Detecting malicious short URLs on Twitter," in Americas Conference on Information Systems, Puerto Rico. 2015.

[26] PhishTank, "PhishTank | Join the fight against phishing". Available at: https://www.phishtank.com/.

[27] A. A. A. Ahmed, H. Paruchuri, S. Vadlamudi, and A. Ganapathy, "Cryptography in Financial Markets: potential channels for future financial stability," Academy of Accounting Financial Studies Journal, vol. 25, no. 4, pp. 1-9, 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4774829.

[28] A. Ganapathy, "Cascading Cache Layer in Content Management System," Asian Business Review, Vol. 8, No. 3, pp. 177-182, 2018. [Online]. Available: https://doi.org/10.18034/abr.v8i3.542.

[29] E. R. Sruthi, "Understand Random Forest Algorithms With Examples (Updated 2023) ". Available at: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ accessed on 2024-02-01.

[30] Javatpoint, "Decision Tree Algorithm in Machine Learning". Available at: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm, accessed on 2023-12-01.

[31] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, No. 1, pp. 21-27, 1967. [Online]. Available: https://doi.org/10.1109/TIT.1967.1053964.
[32] M. Bansal, A. Goyal, and A. Choudhary, "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, p. 100071, 2022. [Online]. Available: https://doi.org/10.1016/j.dajour.2022.100071.
[33] Geeksforgeeks, "Support Vector Machine SVM Algorithm" Available at: https://www.geeksforgeeks.org/support-vector-machine-algorithm/, accessed on 2023-12-13.
[34] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," Soft Computing, vol. 25, No. 3, pp. 2277-2293, 2021/02/01 2021. [Online]. Available: https://doi.org/10.1007/s00500-020-05297-6.
[35] S. Jessica, "How Does Logistic Regression Work? ", Available at: https://www.kdnuggets.com/2022/07/logistic-regression-work.html, accessed on 2023-12-13.
[36] A. Saini, "AdaBoost Algorithm: Understand, Implement and Master AdaBoost" Available at: https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/, accessed on 2023-12-13.
[37] R. Mohammad and L. McCluskey, "Phishing Websites. UCI Machine Learning Repository," ed, 2015. [Online]. Available: https://archive.ics.uci.edu/dataset/327/phishing+websites

Authors' Profiles

Najla Odeh, Palestine Technical University Kadoorie, Computer Science Department, Faculty of Information Technology, Tulkarm, Palestine. She obtained a bachelor's degree in Computer Information Systems from Al-Quds Open University, Tulkarm, Palestine, in 2011. Subsequently, she received a master's degree in Computer Science from Palestine Technical University Kadoorie (PTUK) in 2024. She worked as a programmer and web designer at IDEX Company, Tulkarm, Palestine, from 2011 to 2015. Currently, she is employed in the role of technical support at PTUK. Her areas of research interest primarily focus on machine learning, network technologies, information systems, and network.

Sherin Hijazi received her B.S. degree in Management Information System from An-Najah National University, Nablus, Palestine, in 2005. She finished her M.S. degree in Computer Information System from Al-Yarmouk University, Irbid, Jordan, in 2012. She was granted a Scholarship from Palestine Technical University Kadoorie (PTUK) to pursue her PhD in Computer Science at the University of Jordan, Amman, in 2015. She finished her Ph.D. degree in Computer Science from the University of Jordan, Amman, Jordan, in 2020. From 2005 to 2007, she was an Administrative Assistant with the IT Department in Palestine Securities Exchange (PSE). From 2007 to 2011, she was a Programmer with PTUK, Tulkarm, Palestine. From 2011 to 2013, she was the Head of programming with the Computer Center, PTUK. From 2013 to 2020, she was a Lecturer with the Department of Applied Computing, PTUK. From 2020 to 2022, she has been a Professor Assistant with the Department of Applied Computing, PTUK. From 2022 until now, she has been a head of Computer Science Department with the Information Technology Faculty, PTUK. From 2022 until now, she has been a head of Information System Department with the Information Technology Faculty, PTUK. She has eight publications in various fields of computer science. Her research interests are in artificial intelligent, knowledge representation, network security, software engineering, database systems, and information system. Dr. Hijazi received the IEEE Systems Journal Best Paper Award in 2020. Dr. Hijazi received the Reviewer Certificate of Security and Privacy in 2020 and 2021. Dr. Hijazi received the Reviewer Certificate from IEEE Access in 2020-2022.

How to cite this paper: Najla Odeh, Sherin Hijazi, "Detection and Prevention of Phishing Short URLs Using Machine Learning and Blacklist Approaches", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.15, No.3, pp. 37-53, 2025. DOI:10.5815/ijwmt.2025.03.03