Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7764722

MACHINE LEARNING MODEL FOR IDENTIFYING PHISHING WEBSITES

Uday Bhaskar Penta
PG Research Scholar, Department of Computer Science Engineering, Raghu Engineering College, Dakamarri, Visakhapatnam – 531162, Andhra Pradesh, India.
Udaynaidu238@gmail.com

Dr Panda B S
Professor, Department of Computer Science Engineering, Raghu Engineering College, Dakamarri, Visakhapatnam – 531162, Andhra Pradesh, India.
Bspanda.cse@gmail.com

Dr Sasanko Sekhar Gantayat
Professor, Department of Computer Science Engineering, Bharat Institute of Engineering and Technology, Mangalpally, Hyderabad – 501510, Telangana, India.
drssgantayat@ieee.org

Abstract— Advances in cloud and internet technology have resulted in a major expansion in electronic trade, wherein consumers conduct online shopping and payments, in recent times. Unauthorized users can gain access to private data and cause financial harm to businesses as a result of this expansion. Phishing is a well-known attack that misleads people into accessing dangerous content and discovering their personal information. Many phishing sites are indistinguishable from legitimate ones, both visually and in terms of their universal resource location (URL). Many methods, including blacklists, heuristics, and others, have been proposed for identifying phishing sites. Yet, the number of victims is growing exponentially because of ineffective security systems. Studies done so far have revealed that the effectiveness of anti-phishing technology is poor. Customers need an effective method to safeguard themselves from cybercriminals. In this research, we use machine learning methods like K Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naive Bayes (NB) to identify phishing websites on their own. Data for the study comes from PhishTank, and essential attributes are extracted via Feature Extraction (FE) methods. FE makes use of two methods: URL-based and hyperlink-based approaches. The outcome of both FE approaches is given to the ML model and validated using the metrics. The outcome of the metrics helps to identify the best combination of FE and ML models for phishing website detection.

Keywords— Websites, Features, Phishing, URL, Support Vector Machine, Accuracy, Bar graph.

Introduction

A web service is a protocol and collection of instructions that facilitates data exchange between computers in a networked environment. One way in which computers can talk to one another is through the use of "web services," which are "extensions" to the core infrastructure of the

World Wide Web [1]. TCP/IP, Hypertext Transfer Protocol, Java, HTML, and XML are all open protocols that provide the infrastructure of the internet. When considering the effects of computers on society, web services stand out as particularly significant. The public's top concerns have always been Internet fraud and web security [2], due to the rapid growth of the Internet and the increasing use of electronic payment for web services. Web phishing is a form of social engineering used to trick internet users into giving away sensitive information through various forms of electronic communication. Account information, payment tokens, credit card numbers, and other sensitive data are stolen when victims enter them on a false website. Attempts to steal information via phishing first appeared on AOL (America Online) in the early 1990s [3]. One who specializes in phishing targeted AOL subscribers and successfully acquired their details. It raises the possibility of an attack on the online payment system, which could lead to the fraudulent use of credit card data.

In the context of cybercrime, "phishing" refers to the fraudulent acquisition of sensitive information using electronic and software engineering. Spoof emails sent from what look to be legitimate social network domains are frequently used to fool customers into entering malicious websites and revealing sensitive data [4]. Once a dangerous virus has been installed on a user's computer, hackers frequently employ automated methods to steal the user's login credentials for their online accounts. The phisher may use a variety of methods, including email, links, IM, forum posts, phone calls, and SMS messaging. Phishing content mimics the structure of legitimate content to deceive readers into disclosing personal information. Phishing is mostly used to steal people's identities or private information for monetary gain. Financial systems around the world feel the effects of phishing attempts. The bulk of phishing attempts, according to the most recent Phishing pattern research conducted by the Anti-Phishing Working Group (APWG), are directed at financial/payment institutions and webmail.

In addition, it encourages the theft of logos, trademarks, and other firm identities upon which consumers rely for authentication purposes [5], which in turn greatly aids the Internet's explosive growth as a means of communication. Spammers can send to several recipients at once using "spooled" emails. Links in these emails often take recipients to fake versions of legitimate websites. Sensitive information about consumers is easily exploitable. As a result of all these factors, the issue of phishing has become extremely pressing and important in modern society. Recent anti-phishing studies have focused on countermeasures that take domain information into account, including precise URLs, content, and even the website's code and a snapshot of the homepage [6]. Users of a business might benefit from anti-phishing solutions that quickly spot fraudulent URLs. A major threat to the safety of the website and the personal information of its users is malicious code, which can lead to things like the theft of user data and the installation of viruses. Analysis using ML allows for quick and simple detection of malicious URLs on the web. The standard method of URL detection employs the usage of a blacklist (collection of harmful URLs compiled from user reports and human judgment). The updated blacklist can be used to verify a URL's legitimacy. The number of malicious URLs that are not blocked is rising, despite a decrease in the number of blacklisted harmful URLs. Domain Generation Algorithms are a method that fraudsters can use to generate new malicious URLs and avoid getting caught by the blacklist (DGA). Therefore, it is essential to have a comprehensive blacklist of all potentially malicious URLs. URLs that may cause problems are notoriously difficult to pinpoint. Using conventional FE tools, we provide an ML-based

technique to identify phishing websites. Some of the most recent articles used as resources for this study are discussed below.

In a publication [7], we proposed a method for identifying phishing attacks using ML. Over 4,000 phishing emails were intercepted and analysed, all of which were directed towards the University of North Dakota. Using a vast dataset and a selection of 10 key traits, we constructed a model of these attacks. All ML methods were taught, checked, and tested on this dataset. Probabilities of detection, miss-detection, false alarm and accuracy have all been utilized as performance metrics. The tests demonstrate that simulating a neural network can improve detection quality.

The research [8] aims to investigate the several ML strategies deployed in this space, as well as the datasets and URL properties that have been employed to train ML models. Many different ML algorithms and strategies for enhancing their accuracy are compared and contrasted. The research community can use the results of this poll as a reference to learn about recent developments and offer advice on how to improve phishing detection systems.

This problem can be solved by employing ML technology to identify phishing websites, as suggested by the author [9]. The HTML code structure of the websites' linkages was analysed by the algorithm. Two different ML approaches, Random Forest and SVM, are evaluated and contrasted for their ability to spot phishing websites. Each algorithm's efficacy was measured with its unique criterion. The purpose of this research was to develop a more efficient algorithm for detecting phishing websites using the Random Forest classification method.

The authors [10] propose an ensemble model that uses URL attributes to identify potential phishing sites. The "Phishing website Detector - phishing website dataset" from Kaggle was used. Following this, the models were constructed utilizing several well-known ensemble techniques. Finally, several tools were employed to evaluate the models' performance. The improved performance of ensemble models has led to their exclusive use. As a hybrid of Random Forest and K-Nearest Neighbors, the XGBoost model is recommended for its superior performance. A trustworthy way of identifying phishing websites is the best option going forward. Furthermore, it helps secure users' personal information whenever they visit a phishing website by integrating it into an application or browser extension.

The study's [11] overarching goal is to apply ML models to identify and categorize the programming language employed by phishing websites. Over 29,000 phishing websites' HTML content was extracted via web scraping. This list of phishing sites was produced using PhishTank, a freely available database. Using a dataset of over 36,000 valid websites, we compared the HTML coding styles and syntax of phishing websites to that of actual websites. Sites that didn't have the bare minimum of content were taken down. The source codes of 10,800 websites (5,400 in each category) were processed, and 11 features were retrieved from the content of each website using the cleaned datasets of phishing and genuine websites. When it comes to identifying phishing sites, our Random Forest algorithm much exceeded the competition.

In the journal [12], the authors suggest a layered ensemble learning method in which each successive layer of estimators is informed by the predictions of the layer below it. When tested across multiple datasets, the suggested model showed high levels of accuracy (between 96% and 98%) as shown by the results. The proposed approach is tested on data collected by UCI and Mendeley between 2018 and 2020. On datasets where there was a large gap between the

proposed model and the baseline models in terms of accuracy and F score, the new model also performed better.

To determine which technique delivers the highest accuracy rate and the most efficient use of time, the authors of the study [13] compare and contrast several ML algorithms for recognizing phishing URLs using a hybrid stacking model. Ensemble methods like Adaboost and Gradient boost are explored in the proposed work to improve the performance of Logistic Regression and other ML algorithms. The research shows that the proposed Stacking Classifier achieves a high rate of accuracy. With today's available classifiers, we can make more accurate phishing prediction predictions. However, our findings suggest that the hybrid approach provides more reliable predictions of phishing websites.

Methodology

The identification of phishing URLs is very much important in the wide usage of social networks in society. The research focuses on the automatic identification of the phishing URL using the ML model. The steps involved in the research are given below.

Step 1-Data Collection: The Phishing, as well as legitimate URLs, are collected from a trustable website. The greater number of data leads to enhancing the ML model efficiency. Hence, around 20,000 data were gathered.

Step 2-Feature Extraction: The raw data holds the elements in the form of strings and special characters. It is not possible to use those data directly in the ML model. The hidden pattern in the URL is retrieved using URL and Hyperlink based approaches.

Step 3-ML model: The 80% of extracted feature samples of both approaches are given to the ML model. The three ML models are employed such as KNN, SVM, and NB.

Step- ML model validation: The outcome of three models for two extracted features is validated using the standard metrics. This helps to identify the best combination of FE and ML models for phishing website detection.

Data Collection and Processing

The data of both phishing and legitimate URL is important for training the ML model. Due to the limited lifespan of phishing sites, we only collect the website address if they are active. We developed a phish crawler to collect phishing URLs from the PhishTank website [14]. Using the "Phish Search" option on the website we find the currently active phishing URLs. We use BeautifulSoup to retrieve the page source code. Using 'IDs' and requests to check the page source, we might potentially extract the valid phishing URL we desire. As of right now, we have successfully crawled 10,000 phishing URLs and 10,000 non-phishing URLs from the dataset. Then 8000 data from each URL are employed for training and 2000 for testing. The consolidated report of data used in the research is given in table 1. Figure 1 illustrates the data distribution using a pie chart.

Table 1. Phishing and Legitimate URL data

URL	Actual Data	Used Data	Train	Test

Benign URL	17058	10000	8000	2000
Phishing URL	19653	10000	8000	2000

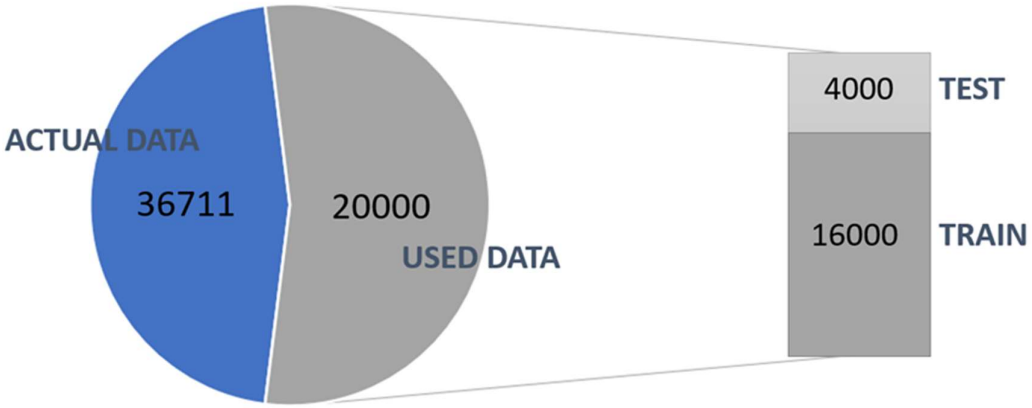

Fig. 1. Data used for phishing URL identification

Feature Extraction

In this section, the FE techniques are used to convert the raw URL into a useful form. Totally 25 features are extracted by employing two approaches (URL features and hyperlink features). The features' titles and descriptions are presented below.

URL based features

The Uniform Resource Locator (URL) is the standard method for locating digital media, web pages, and other online resources. The URL format is separated into sections. The first component is the protocol prefix. Accessing web resources necessitates the use of a protocol prefix such as HTTPS, HTTP, FTP, and so on. Hypertext Transfer Protocol Secure (HTTPS) is the most secure and frequently used protocol today. The second component is the IP address of the server that hosts the resource. The hostname is composed of three distinct parts. There are three types of domains: subdomains, primary domains, and top-level domains (TLD). The TLD is again separated into generic and country code. The third component is a route that points to a specific resource inside the domain to which the visitor has requested access. A single '/' slash separates the domain section and the route. The path structure has two fillable spaces. The initial form of inquiry is always a question mark ('?'). The other is a statement that has been cut off and is preceded by a hash sign (#).

Domain Identifier in URL

Apart from the "www." section of the URL, we record the entire domain name as a convenience. This functionality has little practical purpose. It will be removed from the model's features during the training phase.

Calculate the number of subdomains in a URL.

This function computes the total dots in the URL hostname. Except for 'www.' a legitimate URL usually contain two dots. Attackers use many dots in URLs and add more subdomains,

including the original website's domain name, to deceive consumers. When the feature value is 0.5 and the hostname contains three dots, the URL is regarded as "phishing." When the total dots in the URL exceeds three and the feature value is one, the URL is deemed phishing since it is likely to include malicious content across multiple subdomains.

Internet Protocol Address Identification in Domain Name

An attacker can sometimes spoof the domain section of the URL by utilizing the IP address instead of a legitimate domain name. When an IP address (V4 or V6) appears in the URL domain, it signifies a data theft attempt.

Urls with a "@"

The entire URL is checked for the existence of a "@" character. Usually, phishers add "@" as an end character in the legal website's domain to fool the users. Clicking on this link will download the address following the "@" sign; browsers will ignore the content preceding it. If the URL contains the "@" sign, the feature value is 1, otherwise, it is 0.

URL Length

Attackers create customized phishing URLs to rob data to mask dangerous elements in the URL. The presence of more characters in the URL enhances the risk of a website being phished. There is no pre-set length to distinguish phishing from legitimate URLs. From research, we learned that the maximum valid URL length is 75. If the length is between 75 to 100, the URL is classified as phishing and given a feature score of 0.5.

URL depth

This function calculates the number of levels of depth contained in a website's URL. Subpages are denoted in the URL path section by a single forward slash ("/"). The official website's file directory is easy to navigate. However, attackers keep their bogus websites in the deepest tiers of servers. As a result, their website's file path is usually more convoluted than it is. The URL measures the value of the feature.

URL redirection with "/"

The "//" symbol can be used to forward a user to another website. Attackers exploit "//" in the URL to fool people into accessing a fraudulent site. We check for the instance of "//" in the URL. In practice, this character appears in the sixth or seventh position (next to http: or https:). As a result, if the index is more than 7, we can deduce that "//" appears in the URL in a nonconforming manner. This property has only two possible values: 1 (phishing) or 0. (legitimate).

URL scheme including "http" or "https"

Phishers will sometimes insert a "https" or "http" token into the URL domain section to fool victims. We check the "domain" element of a URL in this function to see if it contains "http/https." If the URL domain component begins with "http/https," then the feature score is 1 (phishing), else it will be 0 (not phishing).

Https in the Scheme

This utility guarantees that the URL protocol is proper. If the URL is "https," the feature value is set to 0 (legitimate), otherwise, it is set to 1 (phishing). When creating a URL, the protocol must be taken into account. The vast majority of trustable sites employ the "https" protocol to enable a safe connection in delivering sensitive data. Modern phishers, on the other hand, construct "https" connections to fool victims. So far, this capability has been ineffective in discriminating between dangerous and legitimate websites.

Shortened link service

Uniform Resource Locators (URLs) can be abbreviated using a service such as bit.ly to conserve characters while still referring to the right page. A redirect is used to send the user to a website with an extremely long URL to accomplish this.

A domain prefixed or suffixed with "-"

This function looks for the "-" character in the URL's domain part. Dashes are rarely used in legitimate domain names. Adding a prefix or suffix to a domain name separated by a "-" is a frequent method employed by hackers to deceive their victims. Users are duped when they attempt to navigate this phishing website because it appears legitimate to them.

Sensitive Word Exists

Tokens or words such as "login," "update," "verify," "activate," "secure," and so on are regularly used in phishing URLs. Cybercriminals use these terms in a URL to deceive consumers into visiting a phishing site and stealing their personal information. We define 18 of these phrases as phrases that are widely used in phishing attempts. If the submitted URL holds any of the phishing words, the feature's score will be 1 (phishing), else it will be 0 (not phishing).

Presence of a popular brand

Fraudulent websites usually look like legitimate ones from well-known firms. As a result, the URLs of phishing sites frequently feature well-known company names. When a visitor visits a phishing website and notices the brand name in the URL, they confuse it with the legitimate site. The top 19 brands commonly targeted by phishers are shown below.

The use of upper case

Lowercase characters are preferred in legitimate URLs. Unfortunately, phishing URLs frequently use all capital letters in an attempt to fool consumers. Every capital letter in a URL indicates that it is suspicious, making it simple to identify a phishing link.

URL dot count

The URL of a legitimate website will have no more than two dots after the "www." This method can thus be used to calculate the total dots in a URL. If the total is larger than 2, the URL is most likely a phishing attack.

Hyperlink based features

Here, we examine the characteristics of the website's hyperlinks as determined by parsing the site's source code. Anchor, src, link, and form tags are heavily examined for this FE.

No Hyperlink

A lot of pages are a sign of a legitimate website. Phishing websites, on the other hand, typically only feature a few pages. To add insult to injury, phishing sites typically don't provide any clickable links, especially if the attackers are employing a hidden hyperlink tactic. If this website is legitimate, we should be able to locate at least one active link within its HTML source code. Href, link, src, and tags are counted as part of the overall number of hyperlinks. This feature's value becomes one (phishing) if the sum of all hyperlink values is zero; otherwise, it remains at zero (legitimate).

Internal Hyperlink

When talking about websites, links that go to the local/base domain are considered internal links. To obtain critical information, hackers frequently develop phishing websites that look

legitimate. That's why they just replicate the legitimate website's code to make a phishing site and steal all your details. When they do this, the copied code could have several references to the hacked site, as that is what they are copying. The hyperlink of the legitimate and phishing site shares the same base domain. When implementing this function, we count the number of internal links and compare them to the overall number of connections in the code. The website accepts the ratio as legitimate if it is more than or equal to 0.5 else the website is considered as legitimate.

External Hyperlink

External links go to websites hosted on other domains or in foreign countries. For the most part, external links make up the bulk of the website that contains phishing scams. However, official sites often only connect to external resources occasionally. Because of this, we count the number of external links and compare them with the entire links in the website's source code. The external hyperlink rate of an official website is typically low because there are so few external connections on those sites. This parameter's value is zero if the ratio is less than fifty percent and one otherwise.

Null Hyperlink

The only HTML element we're looking for here is the anchor tag, or a>. This function determines the ratio of a website's Null links to its total anchor links. The goal of the assault is to keep the users of the targeted website on the same page until they provide sensitive information. Users are redirected to the login page if they click any of the login page's links.

Internal/External Cascading Style Sheets (CSS)

CSS is a markup language that affects the presentation of web pages written in markup languages. To trick their victims, phishing websites look and feel like legitimate ones. Creating a phishing site with the intent of collecting sensitive information is generally a lazy attempt at hacking. That's why they try to use the official site's CSS file rather than making their own. Two distinct flavors of CSS exist: in-house and server-side. The link> tag is used to insert external CSS file links. To locate external links of a CSS file, we look for the link> tag with the characteristics rel ='stylesheet' and href = 'URL. The HTML code of webpages contains internal CSS. To impersonate official websites and steal critical information, attackers frequently exploit the official websites' external CSS files as a starting point for their phishing attacks. Thus, to determine a feature's worth, we examine the website's source code to see if any external CSS files exist. The feature value is 1 if the linked CSS file is a remote resource, and 0 otherwise.

Suspicious Form Link Action

Login or sign-up forms are commonplace on phishing websites and are used to steal victims' sensitive information. If a person submits this form on a malicious website thinking it is legitimate, the attackers will receive all of the data they supplied. Standard practice dictates that a legitimate website's current URL be placed in the action field of a <form> tag. False login forms, however, often have external URLs or PHP files in their "action" fields. The action field may occasionally be empty, including the hash symbol (#), or be set to a function that returns nothing (javascript: void()). Thus, to ensure the login form is legitimate, we examine the action field score within the <form> tag. This attribute's value is a binary one.

External/ Internal Favicon

Simply put, a favicon is a little icon that represents a certain website. Using the <link> tag, a favicon can be included on a webpage. Websites that use an externally-hosted favicon in the browser's address bar are suspected of being malicious phishing sites. It is common practice for the attacker to utilize a favicon that looks identical to the official sites. A duplicate favicon shown in the address bar might fool many users into thinking they are visiting the correct website. So, we look at the favicon's link> tag and see if it points to the same domain. The feature score is 0 (true) if the favicon is used internally, and 1 (phishing) otherwise.

Common Page Detection ratio

As a result, attackers may swiftly put-up malicious clones of popular websites with minimal effort. To make the website appear legitimate, they add several anchor connections. The problem is that they don't create very many pages for anchor links. The phishers may change any or all of the links to lead to the same malicious website. This situation typically results in the rapid identification of commonly used phishing pages.

Common Page in Footer section

It draws attention to the more frequently used page detection found in the footer.

Server Form Handler (SFH)

An empty or blank SFH may indicate an attempt at phishing for personal information. Due to the necessity of following through on the form's next steps based on user input. Furthermore, if the SFH domain is external, this is also a red flag. Accordingly, we give a value of 0.5 to a feature on a potentially malicious website, a value of 0 to a legitimate one, and a value of 1 to a phishing one.

Machine Learning Model

The ML model used to classify the legitimate and phishing website from the extracted features are detailed in the below section.

KNN

To classify data, the advanced K-nearest neighbor approach searches the training documents for K items that are most similar to the test value. When attempting to classify an unlabelled item, we first calculate the distance between it and the identified item and then locate its K nearest neighbors. When compared to the nearest neighbor classifier, the accuracy of this technique is strongly dependent on the value of K [15]. For large data sets, a higher value of K may be utilized to reduce the impact of the inaccuracy. Experimentation is a possible approach for identifying the ideal value for K because it permits the categorization of a subset of the training set using the remaining training patterns. The ideal value of K will be found to minimize the classification error. If several of the K-nearest neighbors share the same class, the likelihood score of that class with relation to the test document is determined as the weighted sum of that class's per-neighbor weights. A ranking for the sample document is established by arranging the scores of potential classes. The KNN decision rule, indicated by Score(d_j, c_j), can be written as follows:

$$\sum_{d_j \epsilon KNN(d)} Sim(d, d_j)\, \delta(d_j, c_j)$$

Where d is the test data, c_j is a list of classes that the system will use to find K-nearest neighbors in the training data, $KNN(d)$ is the list of data in the training set that is closest to d,

$(d_j, c_j)$is the classification of data d_j concerning class c_j. If d_j is a member of $\delta(d_j, c_j)$, the value is 1, otherwise, it is 0. The category with the largest weighted total should be used to classify test data d.

SVM

The SVM is another powerful ML algorithm. Using the SVM technique, each data point is represented in n-dimensional space, and the program then constructs a hyperplane to divide the data into two groups. An SVM's purpose is to find a group of points known as support vectors and then draw a line connecting them. The SVM then constructs separating lines that are perpendicular to the connecting line and bisect them at right angles [16]. For the optimum data classification, the margin should be maximized. In this situation, the margin is the distance between the hyperplane and the pillars of support. To deal with the problem that separating complicated and non-linear data is hard in practice, SVM uses a kernel approach that transfers data from lower dimensions space to higher dimensional space. In the case of linearly separable data, the categorization function $f(X)$ is linear if the hyperplane that passes through the plot center separates the 2 groups. After learning this function, classifying a new data sample X_n is as simple as using the function $f(X_n)$ to determine whether or not X_n has a positive or negative value: If $f(X_n) > 0$, then X_n is a member of the phishing website. The adaptation error of a classifier decreases as the gap between classes increases. It's effective for high-dimensional feature sets and could use the kernel approach to convert non-linear to linearly separable data.

NB

To address the issue of continually fluctuating Keyword patterns, the article [17] presented an ML method for building a filter that reads both spam and legitimate communications received in the past and uses this information to automatically block incoming spam messages. Email communications must be represented as feature vectors for Bayesian Classification techniques to be directly applicable in the context of text classification. The Naive Bayesian classifier treats each article as if it were a vector y. Call the set of numbers from y_1 to y_n. In this scenario, the vector space model's values of the qualities $Y_1, \ldots \ldots Y_n$ are represented by $y_1, \ldots \ldots y_n$. Use binary characteristics, with $Y_1 = 1$ if the website is phishing and $Y_1 = 0$, otherwise. Furthermore, we employ the MI attribute to filter down the list of probable traits. $MI\ (Y; C)$ formula

$$\sum_{x\in(0,1),c\epsilon(phishing,true)} P(Y = y, C = c).log\frac{P(Y = y, C = c)}{P(Y = y).P(C = c)}$$

Consider the case when Y is a categorical variable representing some other characteristic C. It takes time to choose attributes with high Mutual Information (MI) levels. By comparing the frequencies of events, we compute the probabilities P(Y|C),P(C), and P(y). The likelihood that a given document, given its x-coordinates (y_1,…….y_n), belongs to a certain category c is calculated using the theorem of Bayes and total probability. Due to its ease of use and reliability, the Naive Bayes model has widespread popularity.

Results and Discussion

The outcome of ML-based identification of phishing websites is detailed in this section. The data of legitimate as well as active phishing websites is collected from PhishTank. Then the features are extracted using the URL and hyperlink-based approaches. First, the extracted features from the URL method are given to the ML model for training and testing. The predicted outcome of the ML model is compared with the actual outcome. Then the results are

validated using the metrics like accuracy, specificity, recall, precision, and F1-score. Table 2 illustrates the metrics value of all three models. The highest accuracy score attained by SVM and the value is 98.05%, and the lowest accuracy is 95.67% by KNN. For specificity, the maximum value of 98.24% was attained by SVM, and the minimum value of 94.40% by KNN. The SVM and KNN give the greatest and least recall scores of 97.86% and 96.99%. Next, the precision values are compared and the good result given by SVM and its value is 98.25%. Finally, the F1-score is analysed. The SVM gives 98.05% which is the highest value when compared with the other two models KNN and NB which produce the F1-score of 95.65% and 96.66%. Then all the metrics value of ML in table 2 is converted into a bar chart for visual comparison and it is given in figure 2.

Table 2. Performance metrics of ML model using URL-based FE data

MODEL	KNN	SVM	NB
Accuracy	95.675	98.05	96.675
Specificity	94.4063	98.2403	96.2339
Recall	96.9929	97.8618	97.1241
Precision	94.348	98.2526	96.2019
F1-Score	95.6522	98.0568	96.6608

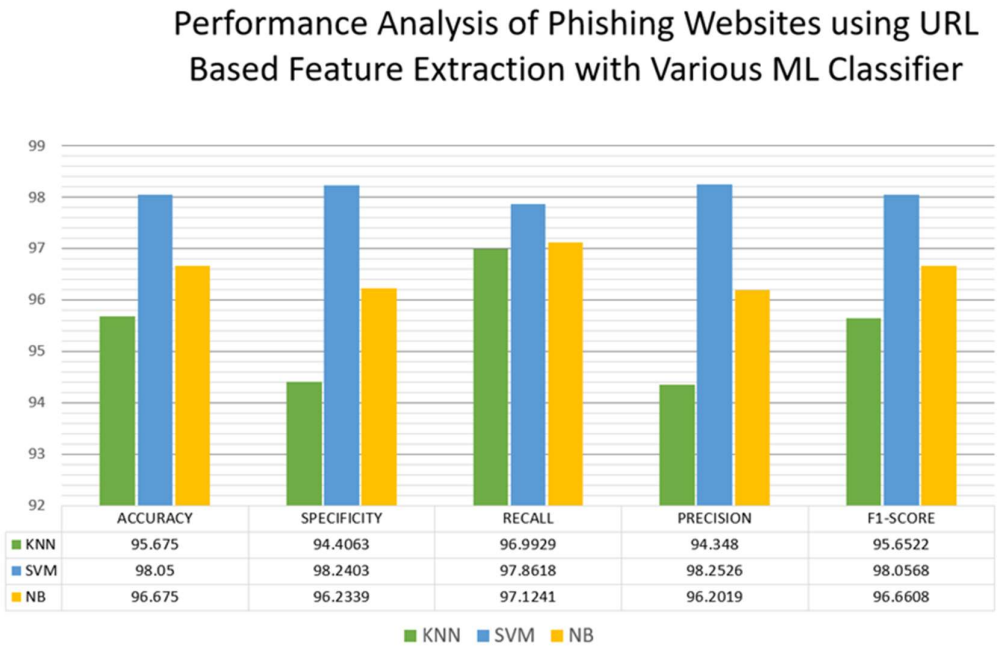

Fig. 2. Performance comparison of URL-based FE with various ML model

The Hyperlink method's retrieved features are then used in the ML model for the training and testing process. Results from the ML model's predictions are compared to the actual results. The aforementioned metrics are then used to verify the results. Table 3 shows the comparative metrics value of the three models. KNN achieves the lowest accuracy at 89.87%, while SVM achieves the greatest at 94.55%. Maximum specificity was achieved by SVM (93.76%) and lowest by KNN (90.87%). SVM and KNN respectively yield the highest and lowest recall scores of 95.36 and 88.87%. The precision results are then compared, and SVM's value, which

is 93.64%, is found to be satisfactory. The F1-score is analysed at the end. When compared to the other two models (KNN and NB), which yield an F1-score of 89.72% and 91.41%, respectively, the SVM gives the greatest value, at 94.49%. Figure 3 provides a bar chart representation of all of the ML metric values from Table 3.

Table 3. Performance metrics of ML model using Hyperlink based FE data

MODEL	KNN	SVM	NB
ACCURACY	89.875	94.55	91.6
SPECIFICITY	90.8772	93.7653	92.1829
RECALL	88.8778	95.3642	90.9969
PRECISION	90.7332	93.6468	91.8378
F1-SCORE	89.7959	94.4977	91.4154

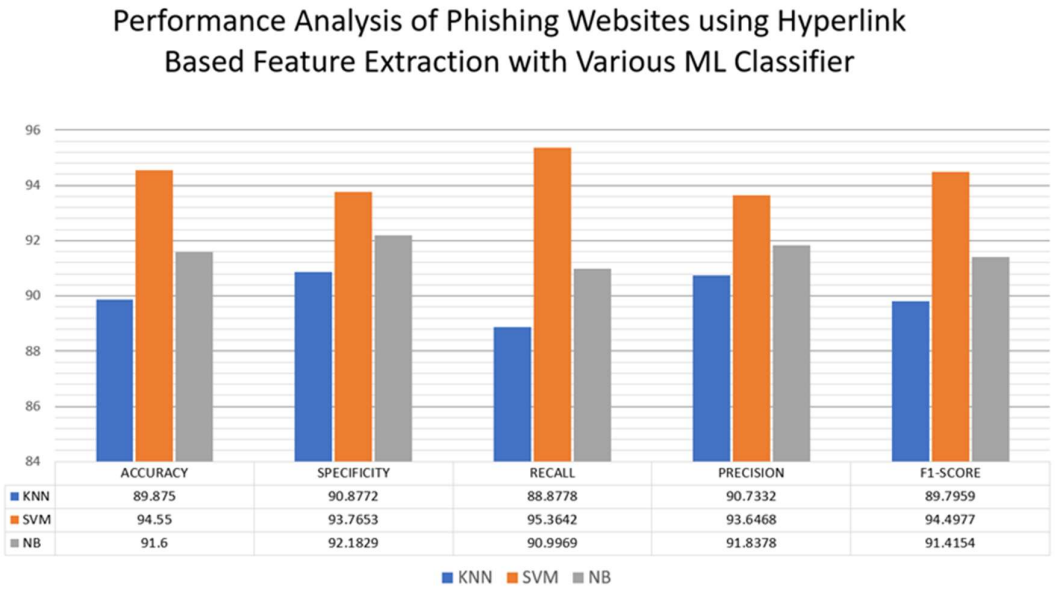

Fig. 3. Performance comparison of Hyperlink-based FE with various ML model

Finally, by analysing table 2 and table 3 outcomes, the SVM model with URL-based feature is excellent when compared to other combinations of FE and ML models.

Conclusion

Cybercriminals engage in phishing when they create a fake website in an attempt to steal sensitive information from unsuspecting visitors, such as login credentials, banking information, social security numbers, and so on. In terms of frequency of use, phishing scams may currently rank as the most common form of cybercrime. Imitating legitimate websites in both appearance and function is a common tactic used by phishers. There has been a dramatic rise in the intelligence of phishing attacks due to technological developments in recent years. Anti-phishing technology is needed to detect phishing attacks and keep users safe. In this study, we developed an ML model by extracting features using URL and Hyperlink-based methods. The data gathered for the study was made up of string and special characters. Those raw data cannot be fed into the ML model without some transformation. The outcome of the ML model

using two FE techniques is compared and validated using the metrics. The experimental results demonstrate that combining URL-based FE with SVM is effective for detecting phishing sites. The maximum accuracy (98.05%), specificity (98.24%), recall (97.86%), precision (94.34%), and F1-score (95.65%) are all achieved by the URL-based FE with SVM.

REFERENCES

Adam, Omer, Young Choon Lee, and Albert Y. Zomaya. "Stochastic resource provisioning for containerized multi-tier web services in clouds." IEEE Transactions on Parallel and Distributed Systems 28, no. 7 (2016): 2060-2073.

Huang, Hsiu-Chuan, Zhi-Kai Zhang, Hao-Wen Cheng, and Shiuhpyng Winston Shieh. "Web application security: threats, countermeasures, and pitfalls." Computer 50, no. 6 (2017): 81-85.

Yi, Ping, Yuxiang Guan, Futai Zou, Yao Yao, Wei Wang, and Ting Zhu. "Web phishing detection using a deep learning framework." Wireless Communications and Mobile Computing 2018 (2018).

Dutta, Ashit Kumar. "Detecting phishing websites using machine learning technique." PloS one 16, no. 10 (2021): e0258361.

Hong, Jiwon, Taeri Kim, Jing Liu, Noseong Park, and Sang-Wook Kim. "Phishing url detection with lexical features and blacklisted domains." Adaptive autonomous secure cyber systems (2020): 253-267.

AlEroud, Ahmed, and George Karabatis. "Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks." In Proceedings of the sixth international workshop on security and privacy analytics, pp. 53-60. 2020.

Salahdine, Fatima, Zakaria El Mrabet, and Naima Kaabouch. "Phishing Attacks Detection A Machine Learning-Based Approach." In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0250-0255. IEEE, 2021.

Anusree, A., Blessy Jose, Karthika Anilkumar, and Ojus Thomas Lee. "Phishing Detection using Extra Trees Classifier." In 2021 5th International Conference on Information Systems and Computer Networks (ISCON), pp. 1-6. IEEE, 2021.

Noh, Norzaidah Binti Md, and M. Nazmi Bin M. Basri. "Phishing Website Detection Using Random Forest and Support Vector Machine: A Comparison." In 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AIDAS), pp. 1-5. IEEE, 2021.

Gu, Jiaqi, and Hui Xu. "An ensemble method for phishing websites detection based on XGBoost." In 2022 14th international conference on computer research and development (ICCRD), pp. 214-219. IEEE, 2022.

Almousa, May, Ruben Furst, and Mohd Anwar. "Characterizing Coding Style of Phishing Websites Using Machine Learning Techniques." In 2022 Fourth International Conference on Transdisciplinary AI (TransAI), pp. 101-105. IEEE, 2022.

Kalabarige, Lakshmana Rao, Routhu Srinivasa Rao, Ajith Abraham, and Lubna Abdelkareim Gabralla. "Multilayer stacked ensemble learning model to detect phishing websites." IEEE Access 10 (2022): 79543-79552.

Nadar, Vinitha Kumaresan, Bhavesh Patel, Vidyullata Devmane, and Uday Bhave. "Detection of Phishing Websites Using Machine Learning Approach." In 2021 2nd Global Conference for Advancement in Technology (GCAT), pp. 1-8. IEEE, 2021.

Phishtank opensourcd platform. http://phishtank.org/. Accessed 2 Oct 2020

Zhang, Zhongheng. "Introduction to machine learning: k-nearest neighbors." Annals of translational medicine 4, no. 11 (2016).

Durgesh, K. SRIVASTAVA, and B. Lekha. "Data classification using support vector machine." Journal of theoretical and applied information technology 12, no. 1 (2010): 1-7.

Sahami, Mehran, Susan Dumais, David Heckerman, and Eric Horvitz. "A Bayesian approach to filtering junk e-mail." In Learning for Text Categorization: Papers from the 1998 workshop, vol. 62, pp. 98-105. 1998.