Detecting Phishing Websites using Hybrid Model Approach in Machine Learning

Hyder Reza Telegraphy
Computer Science Department,
Illinois Institute of Technology,
htelegraphy@hawk.iit.edu,
A20527634

Juveriya Fatima
Computer Science Department,
Illinois Institute of Technology,
jfatima1@hawk.iit.edu,
A20528182

Shadaan Arzeen
Computer Science Department,
Illinois Institute of Technology,
sarzeen@hawk.iit.edu,
A20528043

ABSTRACT

Phishing is a form of online crime where perpetrators deceive victims into visiting malicious websites that mimic legitimate ones, in order to obtain their sensitive information. This practice poses a serious threat to many sectors, including e-commerce and online banking. Detecting phishing sites is a complex and dynamic issue that involves multiple elements and criteria that are constantly changing. We utilized various machine learning classification algorithms such as Naïve Bayes, Logistic Regression, Multilayer Perceptron, Support Vector Machine, Random Forest, Decision Tree, XGBoost classifier, and Autoencoder Neural Network to classify data. Finally we assessed the accuracy of each classifier to compare their performance.

KEYWORDS

phishing website, random forest, naive bayes, XGBoost, Feature extraction

ACM Reference Format:
Hyder Reza Telegraphy, Juveriya Fatima, and Shadaan Arzeen. 2018. Detecting Phishing Websites using Hybrid Model Approach in Machine Learning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Detecting Phishing Websites using Hybrid Model).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Phishing attacks have become increasingly common in recent times, with cyber criminals using social engineering techniques to breach security protocols and gain access to sensitive information. Since the beginning of the internet, hackers have used a number of strategies to deceive unwary users into disclosing their identification credentials. Phishing attacks have gained notoriety for their high success rate in luring users into entering their personal information on fake websites that appear to be legitimate. [9] The primary aim of these attacks is to construct a fake website that is visually indistinguishable from the real one and then trick users into providing their login credentials,which can later be exploited.

Hackers frequently use a range of techniques to mask their bogus websites, including constructing URLs that closely resemble legitimate ones. Despite the existence of a few giveaways that can reveal the true nature of these websites, many users are still unable to detect them. Some indicators for phishing websites includes the absence of an "https" tag or the presence of unusual symbols in the URL.[10] Additionally there are more sophisticated techniques that are constantly being improved to evade detection. In order to reduce the risks posed by these offenses, users are urged to use browser extensions that can alert them of a phishing website in advance.

These phishing attacks are not limited to fraudulent websites, they can also be carried out through social media platforms where spoofing links are sent via anonymous messages or from seemingly legitimate organizations that users follow. The primary objective of these attacks is to steal personal information that can be used to confirm a person's identity, such as usernames, passwords, and account numbers. Unfortunately, a considerable users fall victim to these assaults as they they lack sufficient on the subject.[5] This highlights the importance of educating users on how to discern between genuine and counterfeit web pages and raising awareness about the hazards posed by phishing attacks.

The attacks are designed to exploit human error by employing social engineering techniques. Attackers usually use fake web pages which closely replicate authentic ones. Cyber criminals place these deceptive pages on popular websites or send them via email to unsuspecting users. By using domain names that are similar to the legitimate ones, the attackers are able to direct the victim from the real web server to the bogus server. The use of social media to carry out these offenses has become more prevalent, with over 70% of successful phishing attacks being conducted through social media.[11] Lack of education and unawareness on the subject has made many users vulnerable to these attacks. Therefore, it is crucial to educate users on the risks affiliated with phishing attacks and to increase awareness of how to detect and evade them.

1.1 Problem Description

The growing prevalence of phishing attacks lately has accentuated the need for effective methods to identify and block such attacks. Phishing attacks use social engineering methods to appeal unsuspecting users into entering their personal information on fraud websites that appear to be authentic. Although there has been constant efforts to educate users on how to identify these fake websites, many users are still vulnerable to these attacks. Therefore, there is a need for a constructive technique to detect and prevent phishing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Detecting Phishing Websites using Hybrid Model, ,
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

attacks. This project focuses on the need for a hybrid model approach in machine learning to successfully detect phishing websites. The hybrid model approach combines multiple machine learning algorithms to achieve high accuracy in identifying and preventing phishing attacks. The proposed solution aims to provide users with a reliable and effective method for detecting and avoiding phishing websites, ultimately enhancing cyber-security and protecting user data.

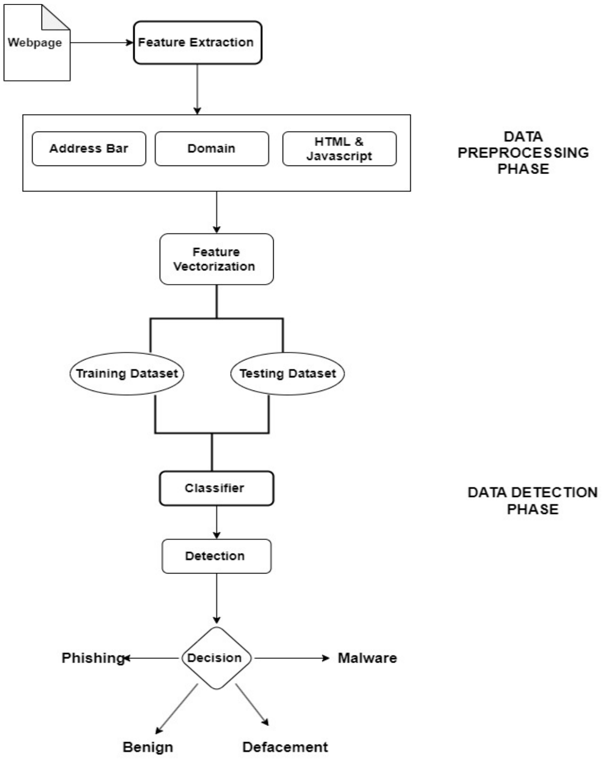


Figure 1: Flowchart Diagram of Phishing Website Detection

1.2 Objectives

The objective is to develop a method that can accurately detect phishing website features. By utilizing a hybrid approach that incorporates various techniques, we can narrow down the strategies used by attackers to trick victims into believing that the website is legitimate. This can help us identify patterns used to disguise the URL.

1. Developing a hybrid approach to detect phishing features of a website: A hybrid approach can be developed by combining different techniques to identify suspicious URLs and detect patterns used by attackers to disguise the URL. The method can include content-based, heuristic-based, and list-based approaches to carefully analyze numerous URL attributes. This can help limit different approaches used by attackers to obfuscate the URL and deceive users into thinking that the website is genuine.
2. Stacking multiple classification algorithms: To further improve the accuracy and effectiveness of the phishing detection approach, multiple machine learning classifiers can be stacked together. By comparing various classifiers, the most capable one suitable for the feature set can be chosen. Combining these classifiers can help create a more robust and accurate model for detecting phishing websites.
3. Improving past approaches and standards: It is crucial to constantly improve past approaches to stay updated and to keep up with the evolving techniques used by attackers. This can be done by updating and developing new algorithms, and combining them with existing ones. Doing so can help ensure that the phishing detection approach remains useful and relevant in the current field.

By developing a hybrid approach and stacking multiple classification algorithms, the accuracy and effectiveness of detecting phishing websites can be improved. This can help to discern and hinder cyber attacks, which have become more widespread presently. It is also important to keep improving the current models and standards to keep up with the evolving nature of cyber attacks. By doing so, we can stay ahead of attackers and protect sensitive information from being compromised.

2 LITERATURE REVIEW

Phishing attacks use various methods to create deceptive websites, but many of these websites share similar features in their design to trick users. Hence researchers conducted extensive research to combat phishing by examining different characteristics present in phishing websites.

Alam et al[4] proposed a hybrid technique to detect phishing websites.Their approach involved using three different algorithms, logistic regression, decision tree, and artificial neural network, to extract features from the URLs and website content. But their approach had few drawbacks, such as the study did not evaluate the performance on different types of phishing attacks. Also the data used for the study was from 2016 and this does not represent the current state of phishing attacks.

Tahir et al [18] conducted a similar study to detect phishing websites by using supervised learning algorithms. They used several supervised learning algorithms and combined the results of these algorithms using weighted ensemble methods. The limitations in their study is that they used limited set of features. Also they failed to provide a comprehensive analysis of performance by every individual algorithm. Due to this it was hard to distinguish which algorithm effectively detected phishing websites.

The journal "Expert Systems with Applications" [10] published a paper which detected phishing webpages by identifying common characterstics which led to these attacks. One of the major drawback of the study was that it was conducted on one specific dataset and it may not generalize well to real world scenarios. Furthermore, the paper lacks details on the efficacy of the proposed framework in detecting advanced phishing attacks that utilize sophisticated methods to trick users.

3 DATA DESCRIPTION

In this case study,a Malicious URLs dataset[17] of 6,51,191 URLs is used, out of which 4,28,103 are benign or safe URLs, 96,457 are defacement URLs, 94,111 are phishing URLs, and 32,520 are malware URLs.

The ISCX-URL-2016 [15] dataset is used to collect URLs classified as benign, phishing, malware, and defacement. After gathering the URLs from various sources, they were separated into distinct dataframes and combined to preserve the URLs and their corresponding class type.

1. Benign URLs: These are safe to browse URLs. Some examples of benign URLs are:
   - mp3raid.com/music/krizz_kaliko.html
   - infinitysw.com
   - google.co.in
   - myspace.com
2. Malware URLs: These types of URLs inject malware into the victim's system once he/she visits such URLs. Some examples of malware URLs are:
   - proplast.co.nz
   - http://103.112.226.142:36308/Mozi.m
   - microencapsulation.readmyweather.com
   - xo3fhvm5lcvzy92q.download
3. Defacement URLs: Using methods like code injection, cross-site scripting, etc., hackers typically generate defacement URLs with the goal of infiltrating a web server and replacing the hosted website with one of their own. Religious websites, official websites, bank websites, and corporate websites are frequent targets of defacement URLs. Defacement URL examples include the following:
   - http://www.vnic.co/khach-hang.html
   - http://www.raci.it/component/user/reset.html
   - http://www.approvi.com.br/ck.htm
   - http://www.juventudelirica.com.br/index.html
4. Phishing URLs: Hackers attempt to obtain sensitive personal or financial information, such as login information, credit card numbers, online banking information, etc. by constructing phishing URLs. Below are a few instances of phishing URLs:
   - roverslands.net
   - corporacionrossenditotours.com
   - http://drive-google-com.fanalav.com/6a7ec96d6a
   - citiprepaid-salarysea-at.tk

4 METHODOLOGY

4.1 Data Preparation

In this study, the dataset is transformed by extracting features from the URL and performing train-test splitting to build a resilient model.

In the first step of the data preparation process, features are extracted from the URLs using various feature extraction functions and vectorized them. The various features that have been extracted will be discussed in the following section.

In the second step, the dataset is partitioned by performing a train-test split to create two subsets: a training dataset and a testing dataset. The training dataset is utilized to construct the model, while the testing dataset is used to evaluate its performance. To ensure that the data samples are distributed randomly across the subsets, the dataset is shuffled. Moreover, we tackled the issue of imbalanced datasets, which occurs when one class has significantly fewer instances than the other classes. This problem can lead to a biased model, where the algorithm prioritizes predicting the majority class at the expense of the minority class.

Before proceeding with the overall data preparation process, the type labels - malicious, benign, defacement, and phishing - were encoded into numerical values to facilitate the machine learning training process. Specifically, these labels were mapped to 0, 1, 2, and 3, respectively. This encoding ensures that the labels can be understood by machine learning algorithms and is a standard practice in such applications.

4.2 Feature Extraction

The lexical features listed below are taken from the raw URLs since they will be utilized as input features when the machine learning model is trained. The following features were extracted:

- **having_ip_address**:The IP address is typically used by online criminals to conceal the identity of a website instead of the domain name. This function determines whether or not the URL contains an IP address.
- **abnormal_url**: The WHOIS database can be used to extract this feature. For a legitimate website, identity is typically part of its URL.
- **google_index**: In this feature, URL is checked to see whether it is indexed in Google Search Console or not.
- **Count(.)**: Phishing or malware websites generally use more than two sub-domains in the URL. Each domain is separated by a dot (.). If any URL contains more than three dots(.), then it increases the probability of a malicious site.
- **Count(www)**:Most secure websites often just have one "www" in their URL. If the URL contains no "www" or several "www"s, this feature aids in the identification of phony websites.
- **count(@)**: The presence of the "@" symbol in the URL ignores everything before it.
- **Count(dir)**: The presence of multiple directories in the URL generally indicates suspicious websites.
- **Count(embed_domain)**: It can be useful to count the embedded domains when looking for dangerous URLs. It is done by looking for the character "//" in the URL.
- **Suspicious words in URL**: Malicious URLs generally contain suspicious words in the URL such as PayPal, login, sign-in, bank, account, update, bonus, service, ebayisapi, token, etc. We have found the presence of such frequently occurring suspicious words in the URL as a binary variable i.e., whether such words present in the URL or not.
- **Short_url**:The purpose of this feature is to show whether a URL has been shortened using a service like bit.ly, goo.gl, go2l.ink, etc.
- **Count(https)**: Generally, malicious URLs do not use HTTPS protocols as it generally requires user credentials and ensures that the website is safe for transactions. So, the presence

or absence of HTTPS protocol in the URL is an important feature.

- **Count(http)**: Most of the time, phishing or malicious websites have more than one HTTP in their URL whereas safe sites have only one HTTP.
- **Count(%)**: URLs cannot contain spaces. URL encoding normally replaces spaces with the symbol "%". Safe sites generally contain fewer spaces whereas malicious websites generally contain more spaces in their URL hence more number of "%".
- **Count(?)**: A query string that comprises the information to be given to the server is indicated by the symbol "?" in the URL. A URL is considered suspicious if it has a greater number of "?".
- **Count(-)**: Phishers or cybercriminals generally add dashes (-) in the prefix or suffix of the brand name so that it looks like a genuine URL. For example. www.flipkart-india.com.
- **Count(=)**:A variable value is passed from one page to another when the URL contains the equals sign ("=") and a value. The URL is regarded as being riskier because anyone may change the values and alter the page.
- **URL length:** Attackers generally use long URLs to hide the domain name. This feature calculates the length of the URL and compares it with the average length of safe URLs (which is 74) to determine if it is a suspicious URL.
- **Hostname length:** Another crucial aspect for identifying fraudulent URLs is the length of the host name. This function determines whether a URL is suspicious by calculating the length of the host name and comparing it to the typical length of safe URLs.
- **First directory length:** This feature helps in determining the length of the first directory in the URL. It counts the number of characters between the domain name and the first forward slash (/) in the URL. The length of the first directory is then compared with the average length of safe URLs to determine if it is a suspicious URL.
- **Length of top-level domains:**One of the domains at the top of the Internet's hierarchical Domain Name System is a top-level domain (TLD). For instance, the top-level domain is com in the domain name www.example.com. Thus, recognizing hostile URLs also depends on the length of the TLD. This function determines whether a URL is suspicious by calculating the TLD's length and comparing it to the typical length of safe TLDs.
- **Count digits:** The presence of digits in the URL generally indicates suspicious URLs. Safe URLs generally do not have digits, so counting the number of digits in the URL is an important feature for detecting malicious URLs.
- **Count letters:** When recognizing fraudulent URLs, the URL's letter count is also a key factor. Attackers typically do this by adding more letters and numbers to the URL in an effort to lengthen it and conceal the domain name. This feature calculates the likelihood that a URL is malicious by counting the number of letters in the URL.

4.3 Exploratory Data Analysis

4.3.1 URL Wordcloud. The word cloud is a powerful tool in natural language processing that helps to analyze the distribution pattern of words in different categories. For instance, in benign URLs, words such as html, com, org, and wiki are more frequent, while phishing URLs have more frequent tokens such as tools, ietf, www, index, and battle. In comparison, the word cloud of malware URLs contains higher frequency tokens of exe, E7, BB, and MOZI, as these URLs attempt to install trojans in the form of executable files on users' systems. Defacement URLs intend to modify the original website's code, and their word cloud is dominated by common development terms such as index, php, itemid, https, and option. Overall, the word cloud provides an intuitive way to understand the key features and patterns of different types of URLs.

4.3.2 Data Visualization. Figure 8 illustrates the distribution of class labels for a subset of features in the dataset. However, it should be noted that the figure does not represent the entire population of features, as there are many more features in the dataset.

4.4 Model Training and Evaluation

4.4.1 ***Defining Metrics****.* The following metrics are used for evaluating the performance of the models after the training phase

(1) *Accuracy:* Accuracy is the number of correctly classified phishing websites and legitimate websites divided by the total number of websites in the dataset. It is a good metric when the classes are balanced, i.e., there are roughly equal numbers of phishing and legitimate websites in the dataset.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

(2) *Precision:* It is the number of correctly classified phishing websites divided by the total number of websites classified as phishing websites. Precision measures the ability of the classifier to not label legitimate websites as phishing websites.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

(3) *Recall:* This is the number of correctly classified phishing websites divided by the total number of actual phishing websites in the dataset. Recall measures the ability of the classifier to detect all phishing websites.

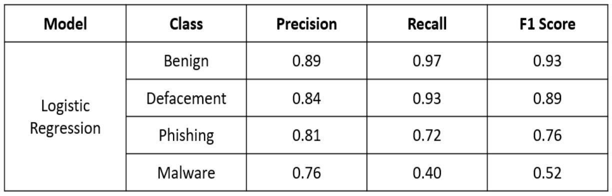
$$Recall = \frac{TP}{TP + FN}$$

(4) *F1 Score:* The harmonic mean of precision and recall, with equal weights. It combines both metrics into a single score.

$$F1\,Score = 2 * \frac{Precision * Recall}{(Precision + Recall)}$$

4.4.2 ***Model Evaluation***. The following machine learning models were trained for the task:

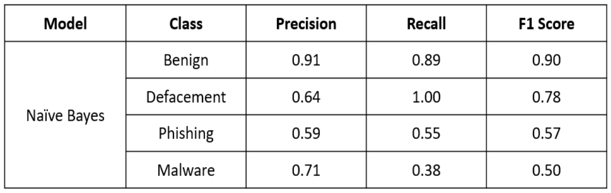
(1) **Logistic Regression:**
The task at hand deals with a multi-classification problem, the aim is to predict the likelihood of an outcome based on a set of input features which in this case is the features extracted from the URL. [12] By estimating probabilities with the use of a logistic function, the algorithm simulates the relationship between the dependent variable and predictors (features), generating an accuracy of about 86.90%.



Model	Class	Precision	Recall	F1 Score
Logistic Regression	Benign	0.89	0.97	0.93
	Defacement	0.84	0.93	0.89
	Phishing	0.81	0.72	0.76
	Malware	0.76	0.40	0.52

Figure 2: Metric for Logistic Regression

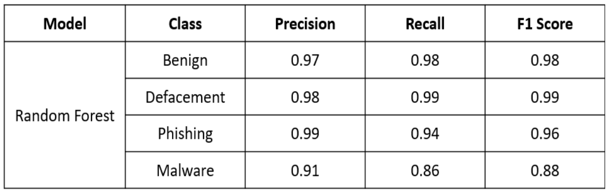
(2) **Naive Bayes:** Utilizing Naive Bayes classifier with the availability of 22 selected features, the model was trained on a labeled dataset. The results indicate that the Naive Bayes classifier with the selected features was effective in detecting phishing websites with a high degree of accuracy of about 81.50%.



Model	Class	Precision	Recall	F1 Score
Naïve Bayes	Benign	0.91	0.89	0.90
	Defacement	0.64	1.00	0.78
	Phishing	0.59	0.55	0.57
	Malware	0.71	0.38	0.50

Figure 3: Metric for Naive Bayes Classifier

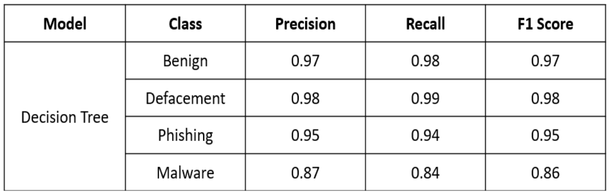
(3) **Random Forest:** The algorithm constructs a number of decision trees using a collection of randomly selected training data subsets, and then combines the predictions from each tree to produce the final result. Each node is divided randomly using a subset of features, which helps to lessen overfitting and increase the generalization of the model. The results generated with the selected features was effective in detecting phishing websites with a high degree of accuracy of about 96.6%.



Model	Class	Precision	Recall	F1 Score
Random Forest	Benign	0.97	0.98	0.98
	Defacement	0.98	0.99	0.99
	Phishing	0.99	0.94	0.96
	Malware	0.91	0.86	0.88

Figure 4: Metric for Random Forest Classifier

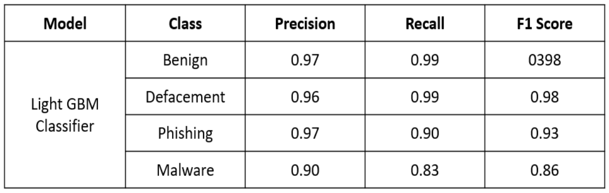
(4) **Decision Trees** This model constructs a single decision tree by recursively partitioning the data into subsets based on the most discriminative features. The algorithm splits each node by selecting the feature that best separates the classes, with the goal of maximizing the information gain at each step. This generates a set of decision rules that can be easily interpreted and applied to new instances.[12] The model's performance was evaluated using various metrics, and the results indicate that the Decision Tree with the selected features was successful in detecting phishing websites with a high degree of accuracy of about 95.80%.



Model	Class	Precision	Recall	F1 Score
Decision Tree	Benign	0.97	0.98	0.97
	Defacement	0.98	0.99	0.98
	Phishing	0.95	0.94	0.95
	Malware	0.87	0.84	0.86

Figure 5: Metric for Decision Tree

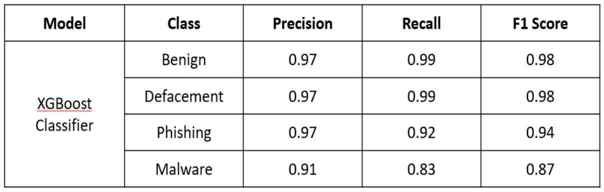
(5) **Light GBM** This algorithm is a gradient boosting framework that uses tree-based learning algorithms. It has been designed to be scalable and efficient than conventional gradient boosting algorithms, with quicker training times and better accuracy.[14] It uses a special technique called Gradient-based One-Side Sampling (GOSS) which reduces the number of samples used in training and improves the speed and efficiency of the algorithm. This model helped generate an accuracy of about 95.90%.



Model	Class	Precision	Recall	F1 Score
Light GBM Classifier	Benign	0.97	0.99	0398
	Defacement	0.96	0.99	0.98
	Phishing	0.97	0.90	0.93
	Malware	0.90	0.83	0.86

Figure 6: Metric for Light GBM

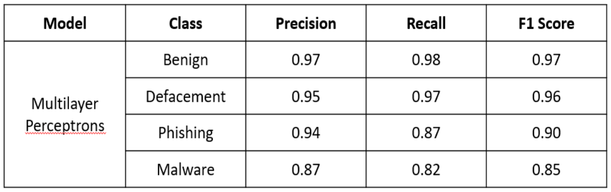
(6) **XGBoost** Gradient Boosting was utilized, an ensemble algorithm that constructs multiple decision trees iteratively to create a more accurate model. [14] The model was trained on a labeled dataset with four target labels. A selection of 22 features was made based on their relevance and importance in identifying phishing websites. The results show that Gradient Boosting with the selected features was successful in detecting phishing websites with a high degree of accuracy of about 96.20%.



Model	Class	Precision	Recall	F1 Score
XGBoost Classifier	Benign	0.97	0.99	0.98
	Defacement	0.97	0.99	0.98
	Phishing	0.97	0.92	0.94
	Malware	0.91	0.83	0.87

Figure 7: Metric for XGBoost classifier

(7) **Multilayer Perceptron** Multilayer Perceptron is a type of artificial neural network commonly used for classification and regression tasks. It consists of multiple layers of interconnected nodes, where each node in a layer is connected to every node in the next layer. The algorithm learns to classify input data by adjusting the weights of the connections between nodes during the training process. Multilayer Perceptron can learn complex non-linear relationships between input features and the target variable.



Model	Class	Precision	Recall	F1 Score
Multilayer Perceptrons	Benign	0.97	0.98	0.97
	Defacement	0.95	0.97	0.96
	Phishing	0.94	0.87	0.90
	Malware	0.87	0.82	0.85

Figure 8: Metric for Multilayer-perceptron

(8) **Auto-encoder** Autoencoder Neural Network is an unsupervised deep learning algorithm that learns to encode and decode input data. A decoder network reconstructs the original data from the compressed representation after an encoder network converts the input data into a compressed representation. Feature extraction and dimensionality reduction activities can be accomplished with autoencoders.

5 RESULTS

By combining various approaches and carefully filtering features that holds relative importance in each of the domains, the paper aims to improve the detection of phishing websites by preparing a hybrid model.

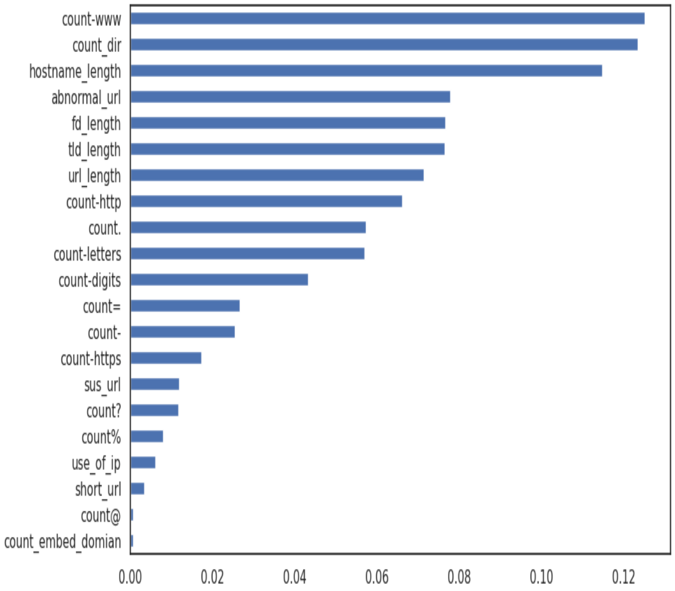


Figure 9: Feature Importance

After training a total of 8 models, each of the models displayed different importance on features. Out of which, Random forest classifier generated a promising accuracy of about 96.60% emphasizing the count of www as the dominant feature in detecting phishing website. By analyzing the metrics, it also displayed low false positive rate giving better performance over other models.

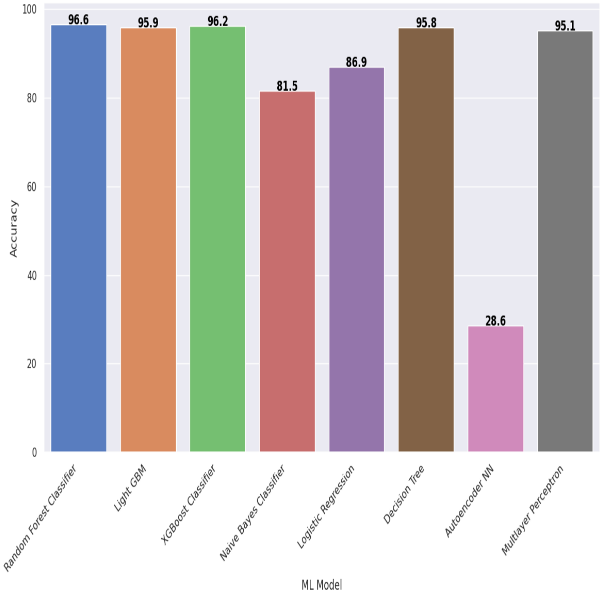


Figure 10: Model Accuracies

6 LIMITATIONS

Although using a hybrid model in machine learning can be a useful strategy for identifying phishing websites, there are several constraints to take into account in this particular undertaking.

1. Limited availability of training data: The success of any machine learning algorithm depends on the availability and quality of training data. However, in the case of detecting phishing websites, the availability of a large and diverse data set can be limited, as many phishing websites are quickly taken down by authorities or web hosts.
2. Constant evolution of phishing techniques: Phishing attacks are constantly evolving and becoming more sophisticated.This poses a challenge in developing a model that can detect all variations of phishing websites effectively. To overcome this challenge, continuous updates and retraining of the hybrid model will be necessary to ensure its effectiveness in detecting new types of phishing attacks.
3. Limited Features: Phishing websites can exhibit a wide range of features, we have extracted only 22 features. The limited number of features used in the model could affect its accuracy because it may not include all the important characteristics. Consequently, the model may not be able to effectively identify certain types of phishing websites, potentially limiting its practical application.

7 CONCLUSION AND FUTURE WORK

One approach to enhancing the applicability of such models in the real world is to deploy them as applications for real-time detection of phishing websites. By doing so, users can receive immediate warnings if they visit a phishing website, thereby reducing the risk of falling prey to online scams. This approach not only improves the user's experience but also enables a timely response to emerging phishing attacks. Furthermore, integrating machine learning models with other security tools and services can provide more comprehensive protection against cyber threats.

Future work for this project can explore deep learning techniques, such as CNN and RNN, to extract complex features from URLs and website content, which can be difficult for traditional machine learning algorithms to distinguish. Another approach to improve the performance of the model is to improve feature selection. Techniques such as the principal component analysis and mutual information can help identify the most important features. Testing the proposed approach on a larger scale, such as with a high-risk corporation or organization, can identify any limitations and provide insights on how to improve it. Lastly, incorporating human factors, like user behavior and perception, into the phishing detection approach can significantly impact its effectiveness.

REFERENCES

[1] N. Abdelhamid, A. Ayesh, and F. Thabtah. Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959, 2014.

[2] S. Abu-nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69, 2007.

[3] A.A. Akinyelu and A.O. Adewumi. Classification of phishing email using random forest machine learning technique. *Journal of Applied Mathematics*, 2014:1–8, 2014.

[4] Md Mahmudul Alam, Md Rafiqul Islam, Ahmad Almogren, Osama Alfarraj, Mohammad Alfraih, and Mansour Alsulaiman. Hybrid model for detecting phishing websites using logistic regression, decision tree, and artificial neural network. *Sensors*, 19(19):4258, 2019.

[5] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3:563060, Mar 2021.

[6] Raouf Amin, Rafiqul Islam, G P Biswas, A A Khan, and Samir Kumar Basak. Phishing websites features and detection approaches: A survey. *Journal of Network and Computer Applications*, 36(2):623–634, 2013.

[7] M. Aydin and N. Baykal. Feature extraction and classification phishing websites based on url. In *Communications and Network Security (CNS) 2015 IEEE Conference on*, pages 769–770, 2015.

[8] Mohammad Hammoud, Imad Elhajj, Rawad Elhajj, Ayman Kayssi, and Ali Chehab. Phishing detection using deep learning: A comprehensive review. *Journal of Information Security and Applications*, 62:102924, 2021.

[9] Hawanna, Varsharani Ramdas, V. Y. Kulkarni, and R. A. Rane. A novel algorithm to detect phishing urls. In *Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on*, pages 548–552, 2016.

[10] M. He, S.J. Horng, P. Fan, M.K. Khan, R.S. Run, J.L. Lai, and et al. An efficient phishing webpage detector. *Expert Systems with Applications*, 38(10):12018–12027, 2011.

[11] Jain, Ankit Kumar, and B. B. Gupta. Comparative analysis of features based machine learning approaches for phishing detection. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, pages 2125–2130, 2016.

[12] Mustafa F Mohammed, Mohammed AAM Al-Qaness, Othman Abdulsalam, Ahmed Al-Ani, Ola I Khalaf, and Aboul Ella Hassanien. A review of machine learning algorithms for covid-19 detection from chest x-ray images. *SN Computer Science*, 2(3):1–25, 2021.

[13] G. A. Montazer and S. Yarmohammadi. Detection of phishing attacks in iranian e-banking using a fuzzy-rough hybrid system. *Applied Soft Computing*, 35:482–492, 2015.

[14] Tuan Anh Nguyen, Anh Tuan Nguyen, Van Hoang Nguyen, Dai Minh Tran, and Dinh Thai Nguyen. Survey on machine learning techniques for intrusion detection system. *arXiv preprint arXiv:1912.11856*, 2019.

[15] University of New Brunswick. CIC-2016: The CIC Dataset - URL collection. https://www.unb.ca/cic/datasets/url-2016.html, 2016.

[16] T. Patil, V. Rodrigues, S. Jaswal, S. Shirke, and N. Shetty. Dynamically heuristic anti-fraudulence system. 2015.

[17] sid321axn. Malicious urls dataset. https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

[18] M. A. U. H. Tahir, S. Asghar, A. Zafar, and S. Gillani. A hybrid model to detect phishing-sites using supervised learning algorithms. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1126–1133, 2016.