A Comparison of Machine Learning Algorithms on Detection of Phishing Websites

Kamala Ramesh
School of Graduate Studies
University of Central Missouri
Lee's Summit, United States of America
kxr54510@ucmo.edu

Madhuri Vedulla
School of Graduate Studies
University of Central Missouri
Lee's Summit, United States of America
mxv16250@ucmo.edu

Priyanka Bojja
School of Graduate Studies
University of Central Missouri
Lee's Summit, United States of Americ
pxb95281@ucmo.edu

Sai Priyanka Narra
School of Graduate Studies
University of Central Missouri
Lee's Summit, United States of America
sxn16130@ucmo.edu

***Abstract*—The Phishing of Websites has recently become the one of the most major cybersecurity threats. They include pretending to be trustworthy websites in order to steal sensitive data including login credentials, credit card information, and personal information. The information provided on these websites is then used for identity theft, financial fraud or other illegal purposes. In the worst case people who are not aware of these attacks might lose their hard-earned savings. Hence it is important to have a system that might be able to help them distinguish between malicious and genuine websites. With all these advances in our day-to-day sciences, Machine Learning proves to be a great way to make that happen. With a collection of websites that we already have knowledge of we can train our system to identify a phishing website. Even if the attackers have found new ways to trick users, we can simultaneously train the model with the latest information. Thus, detection of phishing websites is important for protecting personal information and preventing financial losses. Recently Machine learning techniques proved better results in separating the phishing websites from the genuine ones. In this paper we have collected URLs from different sources and combined them into a single data set with their corresponding target variables namely benign, defacement, phishing and malware URLs. To train the algorithms, we need numerical inputs hence we extract the possible features from the URL based on the Lexical Structure of the URL and other resources and from a new data set with numerical features. These features have been visualized and filtered before training the models. The machine learning algorithms implemented in this paper are, Logistic Regression, K-Nearest Neighbor classifier, Naive Bayes Classifier, Random Forest classifier, Decision tree classifier and Support Vector Machine. The implementation includes determining the accuracy, precision, recall, and F1 score for the trained models using the testing set. With the help of these metrics, the performance of the above-mentioned algorithms is evaluated.**

***Index Terms*—Machine Learning, lexical structure of URL, phishing, Security, Visualization, Results**

I. Introduction

Phishing is the practice of developing fake web pages that look authentic in order to deceive users [1] into providing sensitive information, like login credentials and credit card numbers. These fake internet sites are made to look like the actual thing; they frequently use the same domain names, logos, and designs. Phishing attacks can be carried out through links to fake websites in emails, social media posts, and messaging apps. The Phishing attack involves, sending the URLs in emails or messages to the targets, and once when the links are accessed, the attackers might be able to fetch users information or inject viruses that might corrupt the systems and if any personal details entered by the user might be fetched and misused by the attackers. Individuals risk financial loss and identity theft if thieves utilize the information they enter, for fraudulent reasons. Phishing attacks must be avoided using both technology and user awareness.

A. Types of Phishing Attacks

Over the years, phishing attacks have evolved extensively, and cybercriminals are constantly coming up with new ways to trick people and businesses. Spear phishing, clone phishing, and whaling are a few examples of common phishing attacks.

- Spear phishing - A targeted phishing assault known as spear phishing is made specifically for one person or group of people. Cybercriminals use information about the target's relationships, interests, and preferences to craft a personalized and compelling message in spear phishing. The message may contain a link or an attachment that, when clicked, might result in the installation of malware or the theft of important information, even if it may appear to come from a reliable source, such as a colleague or a supplier. Spear phishing attempts can cause major financial losses and data leaks and are frequently challenging to spot.
- Clone phishing - It is a method for stealing login details and other sensitive data by fabricating a duplicate of an authentic email or website. Clone phishing involves cybercriminals making a fake version of an authentic email or website and sending it to people or organizations. They frequently employ social engineering techniques to get the recipients to click the link or enter their

GitHub Repository Link: https://github.com/KXR54510/MachineLearning_Project

information. Although the cloned website may have the same design as the original website, its URL will be changed, enticing users to enter their login information or other sensitive data. Attacks using clone phishing software can be hard to spot and result in identity theft and financial damages.

- Whale Phishing - Another kind of phishing assault called Whaling targets prominent people like CEOs and other executives. Cybercriminals utilize social engineering techniques in whaling attacks to convince victims to pay money or divulge critical information by feigning urgency or authority. Attacks known as "whaling" can be quite successful since the targets frequently have access to sensitive information and have the power to approve financial transactions. An effective combination of user knowledge and technology measures is needed to stop whaling assaults.

Phishing attacks continue to pose a significant threat to individuals and organizations, and cybercriminals continue to develop new techniques to deceive their victims. Spear phishing, clone phishing, and whaling are some of the common types of phishing attacks.

B. Flow of Phishing Attacks

The process of a phish attack usually involves a number of steps, beginning with the attacker selecting a victim and finishing with the attacker exploiting the data they have obtained for their own immoral purposes as illustrated in Figure 1.

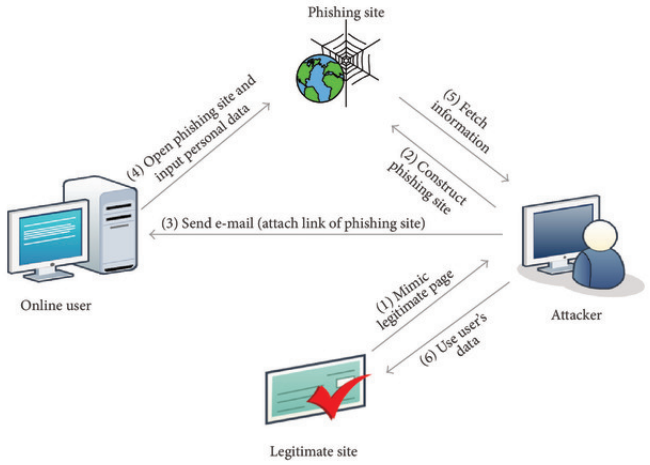

Fig. 1. Flow of Phishing Attack

The attacker selects a target as the initial stage in a phishing assault. Attackers frequently target people who are employed by a certain company since doing so gives them access to valuable data like usernames and passwords. A person who is known to be susceptible to phishing assaults, such as someone who lacks technological sophistication or is readily misled by social engineering techniques, may also be the target of the attacker's attention. The attacker will begin planning their attack once they have decided on their target. Making a convincing email or message is the second step in a phishing assault. Making a false login page, replicating an authentic website, or even pretending to be someone the victim knows may be involved in this. As an example, the message can assert that the victim's account has been compromised or that they need to update their personal information in order to instill a sense of urgency or anxiety in the recipient. The victim will also be instructed to click on a link or attachment in the message. The use of the information obtained through phishing for malevolent purposes is the last stage of the attack. It might entail breaking into the victim's accounts, assuming their identity, or even utilizing their credit card information to make unauthorized purchases. In some circumstances, the attacker may sell the stolen data on the dark web or employ it in additional cyberattacks. It is crucial to be watchful against phishing attempts and to be wary of any unsolicited messages or requests for personal information because the victim might not even be aware that they have been targeted until it is too late.

C. Role of Machine Learning Algorithms

Preventing these attacks requires a combination of user awareness and technical controls. Machine learning algorithms have shown promising results in detecting phishing attacks [2] by analyzing the characteristics of phishing emails and websites. The task of identifying phishing threats is crucial for cybersecurity because these assaults can lead to large monetary losses and data breaches. Although several approaches have been suggested to identify phishing assaults, machine learning algorithms are gaining popularity because of their efficiency and scalability.

Machine learning algorithms examine enormous volumes of data using statistical models to find trends and anomalies that can point to a possible phishing assault. These algorithms can learn to recognize new attacks based on the traits of previous attacks by being trained on historical phishing attacks [3]. Machine learning algorithms can offer a thorough defense against phishing assaults by continuously learning and adjusting to new threats. They can offer an improved and scalable solution for spotting phishing threats than conventional rule-based methods. To identify possible phishing attacks, rule-based techniques employ a set of established criteria or heuristics, however they might not be able to spot fresh or sophisticated attempts. Machine learning algorithms, on the other hand, can examine huge data sets and spot small patterns that can point to a phishing attempt, even if the attack is recent and hasn't been detected before.

In comparison to conventional rule-based methods, machine learning algorithms offer a more efficient and adaptable alternative for detecting phishing attacks. Machine learning algorithms can offer a thorough defense against phishing assaults, lowering the risk of financial losses and data breaches by continuously learning and adjusting to new threats. In order to provide a thorough defense against cyber threats, it is crucial

to keep in mind that machine learning algorithms are not a panacea and must be used in conjunction with other strategies, like human awareness and technical controls.

II. Motivation

Numerous advantages have resulted from the growth of the internet, but it has also raised the risk of cybercrime, such as phishing attempts. Phishing is a fraudulent tactic that can lead to considerable financial loss, identity theft, and other types of cybercrime. Several research have reported good accuracy rates for machine learning algorithms' ability to identify phishing websites. On which method performs best, there is no unambiguous agreement, as different algorithms may have varying strengths and weaknesses based on the dataset and the features employed. Finding the best machine learning algorithm for phishing website detection is one of the key goals. This can aid in the development of more accurate and reliable anti-phishing systems, hence lowering the frequency of phishing assaults and their negative effects on people and businesses. Additionally, the study can shed light on how various feature sets and data pretreatment methods affect how well machine learning algorithms work. Cybersecurity is a field that is continually changing, and new dangers and attack methods are frequently discovered. Therefore, it is crucial to continuously enhance and improve the methods and technologies used to identify and stop cybercrime. The paper can offer insightful information and guide future research and development in this field by examining how machine learning algorithms perform in the detection of phishing websites. This can aid in the creation of improved anti-phishing technologies and provide insight for future cybersecurity research. This study can shed light on the advantages and disadvantages of various feature sets, data preparation methods, and machine learning algorithms, which can be helpful in other fields where machine learning is applied.

III. Objectives and Milestones

The objectives and the milestones reached in each stage of this paper includes the following,

- Literature Study and Analysis - Gathering and assessing relevant material, examining the benefits and drawbacks of prior study, and identifying research gaps were all part of the review and analysis of the body of literature and studies that already existed.
- Dataset Collection - This involved the collection and preparation of a suitable dataset for training and testing the machine learning models [5][6]. This involved selecting a diverse range of phishing websites and non-phishing websites, cleaning and preprocessing the data, and creating appropriate feature sets for the models[7].
- Model Selection - This involved choosing the appropriate algorithms for our implementation that are to be trained and tested. The algorithms selected are Random Forest, Support Vector Machine, K Nearest Neighbor, Gaussian Naive Bayes, Decision Tree Classifier and Logistic Regression[4][14][15].
- Implementation - This involved splitting the dataset into training and testing sets, evaluating the performance of the models using metrics such as accuracy, precision, and recall, and optimizing the hyperparameters of the models for better performance.
- Model Comparison and Analysis - Comparing the accuracy, precision, recall, and F1-score of the models, determining the advantages and disadvantages of each model, and determining which model is best for spotting phishing websites.
- Results Visualisation and Presentation - Presenting the results of the project in a clear and visually appealing manner. This includes creating graphs, charts, and tables to visualize the performance of the machine learning models, and presenting the results in a comprehensive manner.

IV. Related Work

This section provides a thorough and pertinent overview of the relevant literature to support the research questions and potential answers. Over the past few years, there have been an increase in various types of phishing, during which attackers discover new strategies by examining people and modernizing themselves with cutting-edge technology to make the pages appear stronger and more comfortable than before [8]. This chapter also covers the associated suggested methods and provides an overview of phishing. Aaron Blum et. al., [9] explored the possibility of utilizing confidence weighted classification combined with content-based phishing URL detection to produce a dynamic and extensible system for detection of present and emerging types of phishing domains, and authors further claims the system can detect emerging threats and can provide an increased protection against zero-hour threats, unlike traditional blacklisting techniques which function reactively. Many security researchers today rely on machine learning techniques to address shortcomings in current methodologies [10]. This method uses a wide variety of algorithms because it simply needs past data to interpret or forecast future data. During supervised learning, machine learning creates analytical models by utilizing complicated intervention [11]. This method is appropriate for phishing website detection because it may be used to classify complex situations like these. On the basis of the categorization of existing websites, machine learning techniques were employed to advance measures to recognize phishing operations, and these copies may now be included hooked on the browser [12]. Instantaneously identifying a valid site, machine learning models advance production to the user on the other end. The primary accomplishments are the website structures in the input data set and the accessibility of suitable sites for the development of machine learning models for automated anti-phishing identification [13] The study highlights the importance of selecting the appropriate machine learning algorithm based on the dataset and provides insights into the performance of various algorithms in detecting phishing websites. Jitendra Kumar et. al., [14] have discussed the randomization of the

dataset, feature engineering, feature extraction using lexical analysis host-based features and statistical analysis. They compared different machine learning techniques for the phishing URL classification task and achieved the highest accuracy for Naïve Bayes Classifier. M. H. Alkawaz et. al., [15] built a model to protect users from phishing attacks in their research. Decision Tree, Linear Model, Random Forest and Neural Network algorithms were used on a phishing dataset in their research.

Numerous techniques have been put out to detect intrusions [16] [17] [18] using machine learning, with a focus on logs from various sources. This illustrates how machine learning techniques can be used in many different areas of computer and network security to detect anomalies. The paper [16] provides a detailed description of the proposed framework, including the pre-processing steps for the data, the feature selection and extraction process, the clustering algorithm used for anomaly detection, and the evaluation metrics used to assess the performance of the framework. The authors also compare their framework with other existing approaches and demonstrate the effectiveness of their proposed method in detecting various types of attacks, including DoS, port scanning, and botnet activity. It presents a comprehensive and detailed framework for unsupervised anomaly-based intrusion detection, which can help improve the security of computer networks and prevent potential cyber attacks. According to a strategy put forth by Asif-Iqbal et al. [17], relevant events from many sources can be clustered together. This technique enables security analysts to concentrate on a smaller number of events that are more likely to represent signs of an attack. It provides a thorough explanation of the suggested method, detailing the data preprocessing procedures, the feature selection and extraction procedure, the clustering algorithm used to classify the events, and the evaluation metrics used to gauge the method's effectiveness. The authors show how their proposed methodology works well at weeding out pointless events and lightening the strain on security analysts by comparing it to other methodologies currently in use. In order to improve the accuracy and dependability of the identification of anomalies in computer networks, the paper[18] suggests a more sophisticated filter that can recognize and eliminate noise and outliers in the input data. The suggested enhanced filter is thoroughly described in the paper, along with the feature selection and extraction procedure, the data normalization and standardization methods applied, and the evaluation metrics used to gauge the filter's effectiveness. Additionally, the authors contrast their improved filter with the original UHAD framework to show how effective it is in spotting other kinds of network assaults, such as DoS, port scanning, and botnet activities. Overall, the research offers a useful and efficient way for enhancing the UHAD framework's performance, which can enhance the precision and effectiveness of anomaly detection in computer networks. Researchers and professionals working in the field of cybersecurity who are interested in creating more reliable and effective ways for intrusion detection and prevention may find the proposed enhanced filter to be helpful.

To model and detect malicious URLs, Moitrayee Chatterjee et. al., [19] created a model based on deep reinforcement learning. The suggested model can adjust to the changing behavior of phishing websites and so learn the characteristics of phishing website detection. The suggested model can adjust to the changing behavior of phishing websites and so learn the characteristics of phishing website detection. Black-list-based, Lexical-based, content-based, security and identity-based methods were investigated by e. Youness Mourtaji et al., [20], who built a model with machine learning classifiers for phishing website identification. Akihito Nakamura [21] et. al., emphasized compared phishing mitigation techniques, such as blacklist, heuristics, visual similarity, and machine learning and concluded that these techniques have limitations in dealing with zero-hour attacks and proactive detection of phishing websites. The authors proposed suspicious domain names generation and to predicts likely phishing web sites from the given legitimate brand domain name and scores and judges suspects by calculating various indexes to detect phishing websites.

V. Proposed Framework

The machine learning algorithms implemented for detecting phishing websites are Random Forest classifier, Naive Bayes Classifiers, K-Nearest Neighbor classifier, Logistic Regression, Decision tree classifier and Support Vector Machine. The implementation includes determining the accuracy, precision, recall, and F1 score for the trained models using the testing set. With the help of these metrics, we can evaluate the performance of the above-mentioned algorithms. In order train these model data preparation is required. The framework as illustrated in figure 2, involves data collection, feature extraction and filtering the required ones, followed by splitting the train and test data and finally model implementation and evaluation.

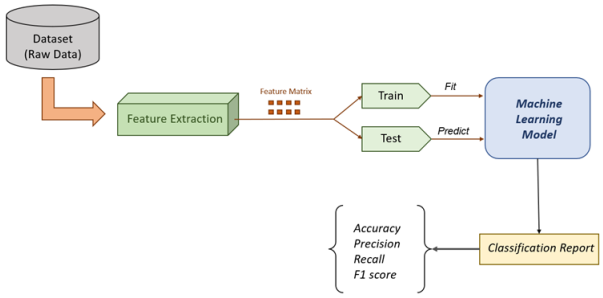

Fig. 2. Process Flow

A. Data Collection

The dataset contains URLs belonging to four classes namely, Benign, Defacement, Phishing and Malware. We have collected the URLs from different sources such as Phish tank Dataset [5], Phish storm Dataset [6], URL Dataset (ISCX-URL-2016) and Faizan repo. All these URLs are combined

into a single dataset and labeled with their corresponding classes. Let us discuss the classes of these URLs, Benign, Defacement, Phishing, and Malware.

Benign URLs: These types of URLs are safe to access. They are not considered harmful and are legitimate to access. Some of the examples of Benign URLs are,

- https://www.google.com/
- https://www.psgtech.edu/hme.php?var=ECE
- https://www.amazon.com/

Defacement URLs: Defacement URLs are created with the intention of hacking the web site and hosting the hacker's own version on the original website. Government websites, Bank website, religious websites are susceptible to these kinds of attacks. Example,

- http://www.vnic.co/khach-hang.html
- http://www.myenrg.com/southwest/9-texas
- http://www.sidecocarrelli.it/x.txt

Phishing URLs: Phishing URLs mainly focus on stealing the personal or financial information, login and password credentials, Internet Banking details, credit card numbers etc. The message often asks the user to input their username, password, or other sensitive information and offers a link to a false login page or a website that mimics the real site. Some of the examples are,

- rcarpe95.beget.tech
- facebook.jolims.tk
- http://pastehtml.com/view/bfza8l1z0.html

Malicious URLs: Once these types of URLs are visited, they try to inject viruses into our system. They might contain malware in the form of viruses, trojans, spyware, and ransomware, which can harm a user's device severely when they infect it. Some of the examples are,

- ru-net.cv.ua/wp-content/config/file.php
- christinelebeck.com
- thaitp47.com/libraries/cms/review/file.php

B. Feature Extraction

A legitimate URL connects us to a real website or online service and is utilized for legal functions. It is used to link users to the appropriate web page or resource and is a legitimate address that has been registered and owned by the owner of the website in good standing.A legitimate URL is made up of three basic components namely,

- Protocol: An identifier that decides what protocol to use, example: http, https, etc.
- Hostname: It contains the domain name or IP address.
- Path: It specifies the path to the location of the resource

According to the figure 3, wisdomml.in.edu is the domain name. The top-level domain is one of the components of the domain name that defines the nature of the website like, organization(.org), commercial(.com), educational(.edu), etc.

There are many other minor lexical features that help us in training our Machine Learning models. So in feature extraction we will be extracting these features from the URLs and store

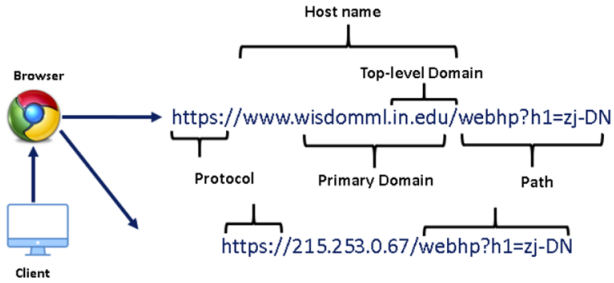

Fig. 3. Lexical Structure of URL

it in a numerical forms in separate columns. The other lexical features might include,

- Use of IP addresses (IPv4, IPv6) in the URLs
- Verifying the identity of URL using hostname
- Check if the URL is indexed in google search console.
- Count of (.) dot in a URL – Each domain is separated by a dot. Malware or
- Phishing websites use more than two subdomains. So, if there are more than three dots in the URL then it increases the probability of the URL not being safe.
- Count of www – In general all URLs have one www in it.
- Count of @ - When there is an @ symbol in the URL, it ignores everything before it. In most cases, @ symbol is used to separate username and password from the rest of the URL.
- Count of dir – The probability of suspicious URL increases if the URL has multiple directories.
- Count of // - The presence of // in the URL helps us to verify the number of embedded domains in a URL, which can be helpful in detecting phishing URLs.
- Short URL - To check whether a URL has been shortened using a service, such as bit.ly, goo.gl, go2l.ink, etc.
- Count of https - Malicious URLs normally avoid using HTTPS protocols because they generally demand user credentials and guarantee that the website is secure for transactions. Hence, whether HTTPS is present or not is a key component of the URL.
- Count of http - Phishing or dangerous websites have many HTTPs in their URLs, whereas secure websites have just one HTTP.
- Count of % - We know that spaces are forbidden in URLs. Normal URL encoding substitutes the symbol (%) for spaces. Secure websites typically have less space in their URLs than dangerous websites, which means that they have more spaces.
- Count of ? - The query string, which contains the information to be provided to the server, is indicated by the symbol (?) in the URL. A URL that has more ?s is undoubtedly suspected.
- Count of – : In order to make the URL appear legitimate, phishers generally append dashes (-) to the brand name's

prefix or suffix. Example: Using the URL as your-bank.com instead of yourbank.com might make the users believe it is a legitimate URL.
- URL Length - Attackers try to hide the domain name by using lengthy URLs. The average length of safe URL is found to be 74 characters. The length of the hostname also plays a part in detection.
- Count of digits - In general, URLs with numbers in them are suspicious URLs. Counting the number of digits in a URL is a helps us in identifying malicious URLs because safe URLs typically do not have digits.
- Count of letters – The letter count of the URLs plays an important role in recognizing malicious URL. The phishers might try to increase the digits and letters in the URLs to conceal the domain name.

When combined all these features we get a new dataset with features extracted, that is going to be used for training and testing the model.

C. Feature Analysis

The analysis of the features Use of IP address, Abnormal URL, Short URL, Google Index and Suspicious URL using seaborn countplot and we found that google index has only one value hence it can be filtered out from the training data set. The other features such as count of values is analyzed using seaborn catplot which proved helpful. Thus, we proceeded with training our model with all the features extracted except google index.

D. Implementation and Evaluation

With the updated dataset after feature extraction and analysis, we perform the train test split and obtain the data for training and predicting the models. We have split the data with a ratio of 80:20 for the train and test respectively. The Machine learning models, Random Forest Classifier, Gaussian Naive Bayes, K Nearest Neighbour, Logistic Regression, Decision Tree Classifier and Support Vector Machines are trained with the training dataset and tested with the test set.

a) Random Forest Classifier: Random Forest combines multiple decision trees to make a more accurate and reliable prediction. A group of decision trees are trained in a random forest classifier using a random subset of the input features and data samples. The final prediction is created by combining the decision trees, with each tree serving as a ”vote” for the anticipated class. The class that obtains the most votes is the final predicted class. Overfitting, which can happen when a decision tree is trained on a dataset that is too particular, is prevented by the random subset of input features and data samples that are utilized to train each tree. The random forest classifier can increase the model's precision and generalizability by mixing the predictions from various decision trees.

b) Naive Bayes: It is based on the Bayes theorem. It makes the assumption that, given the class, the input features are independent of one another. Calculating each class's probability for a certain set of input features is how the algorithm operates. The likelihood of each attribute for each class is modeled using a probability distribution known as the Gaussian distribution or normal distribution.

c) K-Nearest Neighbours: Commonly known as KNN, this algorithm operates by measuring the separation between the input feature vector and each training set data point. Based on how close they are to the input feature vector, it then chooses the k-nearest data points. The majority class label among the k-nearest neighbors is then used to determine the input's class label. The features of the dataset and the particular issue at hand can both influence the value of k. A lower number of k could result in overfitting, whereas a higher value could result in underfitting.

d) Logistic Regression: It is a kind of generalized linear model that simulates the likelihood that the input will fall into a specific class. The algorithm operates by fitting the input features to a logistic function, which converts the input features to a probability between 0 and 1. By minimizing a loss function, such as cross-entropy loss, the algorithm learns the ideal weight vector values during the training phase. Usually, optimization algorithms like gradient descent are used for this process. The logistic regression model can be used to categorize fresh input data after learning the weight vector by computing the probabilities of each class and choosing the class with the highest probability.

e) Decision Trees: It operates by recursively splitting the input data into subsets according to the values of the input features, and then giving each subset a label or a value. The algorithm constructs a structure that resembles a tree, where each node corresponds to a feature and each edge to a decision based on the value of that feature. Finding the feature and the choice that divides the data into two or more subsets that are as homogeneous as feasible with regard to the target variable is the objective.

f) Support Vector Machines: It operates by identifying the hyperplane that most effectively divides the data into two or more classes. In order for the algorithm to function, the input data are represented as points in a high-dimensional space, and the hyperplane that maximizes the margin between the two classes is found. The margin is the separation between each class's closest points, or support vectors, and the hyperplane. By reducing a cost function like hinge loss or squared hinge loss during the training phase, the algorithm discovers the ideal hyperplane. Common optimization algorithms used in this process include quadratic programming and gradient descent.

To compare the performance the actual results of the test set and predicted result were used. The performance of these models is evaluated using the accuracy score, precision, recall and F1-score. A classification report is obtained using the predicted and actual results. For visualization on the performance, we have plotted the seaborn heatmap of the confusion matrix, between actual and predicted values.

VI. Results & Analysis

For training and testing the Machine Learning Algorithms we have used a dataset of 165198 URLs, out of which 101612 are benign, 23709 are defacement, 22942 are phishing, 16935 are malware URLs. Based on the lexical features of URL discussed we have created a new set of features based on the URL that we used to train the models.

For training the model, first we begin with the test-train split. The training set and the test set are separated at random from the dataset in the test train split process. The test set is used to assess the model's performance, while the training set is used to train the model. Particularly in the case of unbalanced datasets, the split is typically performed in a way that maintains the proportion of examples in each class. The split ratio that we have used here is 80-20, where 80% of data belongs to training set and 20% of data belongs to testing set. The model is trained on the training set once the dataset has been divided, and its performance is assessed on the test set. Accuracy, precision, recall, and F1 score are some of the measures used to assess the model's performance. Few points have been highlighted, during implementation as the target variable belongs to four different classes, in Logistic regression, the multi_class argument is specified as "multinomail" and for Support Vector Machine, One Vs One SVC Classifier is used to overcome the multi class functionality.

The quantity of true positives, true negatives, false positives, and false negatives for a particular classification task is broken down in great detail by a confusion matrix. The number of instances that are correctly classified as positive is shown by the true positives (TP), whereas the number of instances that are correctly classed as negative is indicated by the true negatives (TN). False negatives (FN) are instances that are mistakenly classed as negative, while false positives (FP) are cases that are mistakenly labeled as positive. The total number of instances for each real class is determined by adding the counts in each row, and the total number of instances for each predicted class is determined by adding the counts in each column. A confusion matrix provides details on the model's accuracy, precision, recall, and F1 score, which aids in assessing the effectiveness of a classification method.

a) Accuracy:: The percentage of cases that are correctly classified out of all instances is called accuracy. In other words, it assesses the frequency with which the model predicts correctly given all predictions.

$$Accuracy = \frac{TP + TN}{P + N} \tag{1}$$

where P represents total number of instances in Positive class and N represents the total number of instances in Negative Class, which together give the total instances in the data set.

b) Precision:: Precision is the percentage of cases correctly categorized as positive (true positives) among all instances the model classified as positive. In other words, it assesses the frequency with which the model predicts positively given all positively predicted cases.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

c) Recall:: The fraction of actual positive incidents that are true positives is known as recall. It assesses the model's ability to recognize positive cases given all actual positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

d) F1 score:: The harmonic mean of recall and precision is the F1 score. It offers a single statistic and strikes a compromise between the trade-off between recall and precision.

$$F1score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4}$$

Table 1 tabulates the results obtained for the algorithms implemented containing the accuracy, precision, recall and f1 score values calculated.

TABLE I
Evaluation Metrics of the Classifiers Implemented

Machine Learning Classifiers	Evaluation Metrics			
	Accuracy(%)	*Precision*	*Recall*	*F1 score*
Random Forest	98.5	0.985	0.985	0.985
Gaussian Naive Bayes	75.2	0.751	0.752	0.720
K-Nearest Neighbor	96.1	0.961	0.961	0.961
Logistic Regression	90.7	0.907	0.907	0.906
Decision Tree	97.8	0.978	0.978	0.978
Support Vector Machine	92.0	0.922	0.920	0.918

The visualization for the comparison of evaluation metrics of the classifiers implemented has been illustrated in Figure 4.

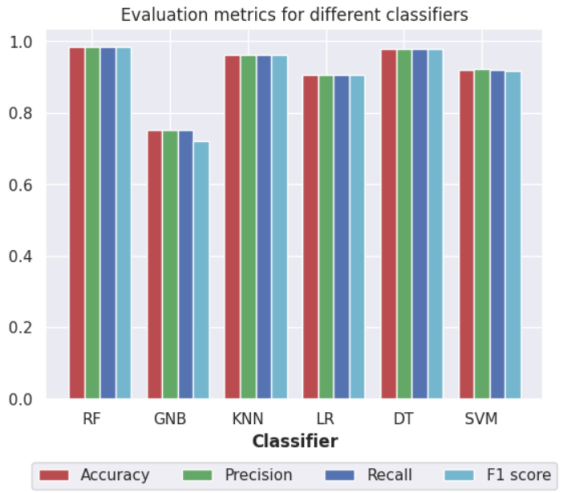

Fig. 4. Comparison of Evaluation Metrics

From the Table 1 and Figure 4 we can come to a conclusion that, the Random Forest Classifier beats the other models in

terms of accuracy, precision, recall, and F1 score, according to our testing and evaluation of several models. We also noticed that when dataset is increased, the classifier's accuracy increased. We used straightforward regular expressions to take a simple approach to extracting the features from the URLs. Additional aspects might be tested, which could result in a further improvement in the system's accuracy. In addition, we discovered that the feature selection procedure is quite important in deciding how well the machine learning models perform. We were able to greatly enhance the models' performance by choosing the most important features and removing the irrelevant ones. The overall findings emphasize the significance of using the right machine learning algorithm and streamlining the feature selection procedure to get the best results in identifying phishing websites. The knowledge collected from this effort will help create more precise and effective anti-phishing technologies in future that will shield users from online fraud and scams.

REFERENCES

[1] Narendra. M. Shekokar, C. S. (2015). An Ideal Approach for Detection and Prevention of Phishing Attacks. Proceedings of 4th International Conference on Advances in Computing, Communication and Control, Elsevier,, Vol.49.

[2] Yang, Y. (2019). Effective Phishing Detection usingMachine Learning Approach.

[3] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-6.

[4] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman and M. S. A. N., "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2020, pp. 391-394.

[5] PhishTank-Friends of PhishTank," PhishTank. [Online].

[6] Marchal, S. (Creator) (2014). PhishStorm - phishing / legitimate URL dataset. Aalto University. urlset(v.zip).

[7] S. Zaman, S. M. Uddin Deep, Z. Kawsar, M. Ashaduzzaman and A. I. Pritom, "Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques," 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 2019, pp. 1-6.

[8] Abbasi, A., Dobolyi, D. G., Vance, A., & Zahedi, F. M. (2021). The Phishing Funnel Model: A Design Artifact to Predict User Susceptibility to Phishing Websites. Information Systems Research.

[9] Aaron Blum,Brad Wardman,Thamar Solorio,Gary Warner, "Lexical feature based phishing URL detection using online learning," 3rd ACM workshop on Artificial intelligence and security, Chicago, Illinois, USA, pp. 54-60, August 2010

[10] Routhu Srinivasa Rao, Tatti Vaishnavi, Alwyn Roshan Pais, "CatchPhish.(2019) Detection of Phishing Websites by Inspecting URLs", Journal of Ambient Intelligence and Humanized Computing, Springer, Vol. 10

[11] Routhu Srinivasa Rao, Alwyn Roshan Pais.(2018) "Detection of Phishing Websites Using an Efficient Feature-Based Machine Learning Framework", Neural Computing and Applications, Springer

[12] Alkawaz, M. H., Steven, S. J., & Hajamydeen, A. I. (2020). Detecting Phishing Website Using Machine Learning. In 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 111-114). IEEE.

[13] Gandotra, E., & Gupta, D. (2021). An Efficient Approach for Phishing Detection using Machine Learning. In Multimedia Security (pp. 239-253). Springer, Singapore

[14] M. Korkmaz, O. K. Sahingoz and B. Diri, "Detection of Phishing Websites by Using Machine Learning-Based URL Analysis," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225561.

[15] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen and R. Ramli, "A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 82-87.

[16] Hajamydeen, A. I., & Udzir, N. I. (2019). A detailed description on unsupervised heterogeneous anomaly based intrusion detection framework. Scalable Computing: Practice and Experience, 20(1), 113-160.

[17] Asif-Iqbal, H., Udzir, N. I., Mahmod, R., & Ghani, A. A. A. (2011). Filtering events using clustering in heterogeneous security logs. Information Technology Journal, 10(4), 798-806.

[18] Hajamydeen, A. I., & Udzir, N. I. (2016). A refined filter for UHAD to improve anomaly detection. Security and Communication Networks, 9(14), 2434-2447.

[19] M. Chatterjee and A. -S. Namin, "Detecting Phishing Websites through Deep Reinforcement Learning," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 2019, pp. 227-232

[20] Youness Mourtaji,Mohammed Bouhorma,Alghazzawi, "Perception of a new framework for detecting phishing web pages," Mediterranean Symposium on Smart City Application Article No. 11, Tangier, Morocco, October 2017

[21] Akihito Nakamura,Fuma Dobashi, "Proactive Phishing Sites Detection," WI '19 IEEE/WIC/ACM International Conference on Web Intelligence), pp. 443-448, October 2019