2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS) | 978-1-6654-0610-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSITSS54238.2021.9683131

Analysis of Machine Learning Algorithms by Developing a Phishing Email and Website Detection Model

Nimisha Dey
Dept. of CSE
RV College of Engineering
Bangalore, India
email: nimishadey.cs20@rvce.edu.in

Samhitha S
Dept. of CSE
RV College of Engineering
Bangalore, India
email: samhithas.cs20@rvce.edu.in

Malavika Hariprasad
Dept. of CSE
RV College of Engineering
Bangalore, India
email: malavikah.cs20@rvce.edu.in

Anagha Anand
Dept. of CSE
RV College of Engineering
Bangalore, India
email: anaghaanand.cs20@rvce.edu.in

Veena Gadad
Dept. of CSE
RV College of Engineering
Bangalore, India
email:veenagadad@rvce.edu.in

***Abstract*— Machine Learning is a key branch of Artificial Intelligence that concentrates on the development of computational algorithms by creating models. It has caught major attention in the technological domain due to its various applications in speech recognition, recommendation engines, computer vision, automated stock trading etc. The model's performance is dependent on the dataset provided and its accuracy can easily be enhanced by expanding the training dataset. Post Covid-19, it has been observed that phishing websites are appallingly on the rise, especially the phishing attacks. These attacks are caused by cybercriminals using PDF's, Microsoft office documents and other attachments via emails. This paper focusses on discussion and comparison of different machine learning algorithms that are capable of detecting phishing emails and websites. The experiments have shown that that MultinomialNB attains the highest efficiency of 98.06% for phishing email detection and Decision Tree Classifier offers the maximum efficiency of 95.41% for phishing website detection.**

Keywords— Machine Learning, Artificial Intelligence, Classification, Naive Bayes, Decision Tree, Logistic Regression, Random Forest, Phishing detection, Sentiment Analysis

I. INTRODUCTION

In current era, with applications ranging from speech recognition, automated customer service or recommendation engines, to computer vision or automated stock trading, Machine learning (ML) is a rapidly growing topic of interest. Artificial intelligence (AI) is a stream of computer science and technology with the purpose of developing methods and algorithms, with the intent of simulating human intelligence [1]. Much like neurons in our brain, which form synapses or connections which are the reason for our understanding of the world, these algorithms make connections based on the data fed to it. As more data is fed, the accuracy of the system improves. Through the use of statistical methods, algorithms are trained to make classifications, predictions or uncover key insights.

Phishing is a malicious attack in online theft to steal the user's private information. It is a kind of scam in which a cybercriminal tries to gain private data and thus users fall into such traps [2]. These scams have been in prominence since 2014. Fig. 1 shows the number of malware sites vs the number of spam sites. Around 2007, malware sites were extensively prevalent. In recent times, phishing sites have turned out to be nearly 75 times more frequent than malware sites [3].

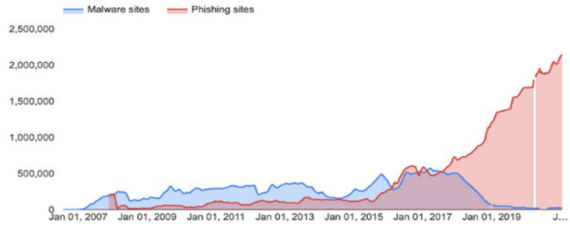

Fig. 1: Trends of increase in phishing websites

Fig. 2 depicts the strategy behind a phishing attack in the form of a flowchart. The expansion of phishing attack techniques is aided by global digitization, especially in the times of COVID-19, where work is becoming remote. This calls for rapid upgrading of anti-phishing systems which is possible by using ML. Being a problem of increasing importance, the need for phishing detection can be understood. This paper discusses how to develop a phishing website and email detection model using different algorithms in ML to compare their efficiencies and discuss the results.

A vast range of algorithms are available in literature. This paper focuses on 4 of the most popular ML classification algorithms – Random Forest, Decision Tree, Logistic Regression and Naïve Bayes Classifier.

II. CONTRIBUTION OF THE PAPER

Plethora of ML algorithms are available in the literature. In this paper, an attempt is made to analyse these algorithms by developing models. The contributions of the paper are:

1. To develop a robust machine learning model that has the capability of detecting phishing websites and emails.

2. To train the model using different algorithms and provide a comparison between their accuracies and training duration.

978-1-6654-0610-9/21/$31.00 ©2021 IEEE

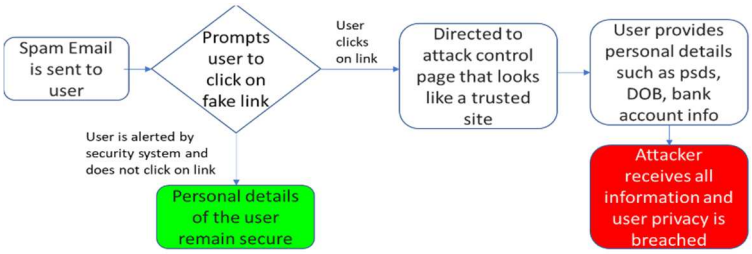

Fig. 2: Mechanism of phishing attack

III. Machine Learning Algorithms

In this section, an in-depth mathematical interpretation of the principles behind various ML algorithms is proposed.

A. Logistic Regression

Logistic regression is a classification algorithm used to allot observations to a number of unique categories. It would be inappropriate for the prediction of data which is continuous, like weight, height, age etc. It incorporates the concepts of probability. The 3 main types of Logistic Regression are:

Binary Logistic Regression: In this model, the outcome value can have two possible types.

Multinomial Logistic Regression: In this type of regression, the outcome value can have three or more possible types that are unordered.

Ordinal Logistic Regression: In this type of regression, the outcome value can have three or more possible types that are ordered.

a) Sigmoid Function

Linear functions cannot be used to represent the hypothesis because the function can take values beyond the 0 to 1 range, which defies the probability concept. So, the sigmoid function is used, which takes any real input value and outputs another value between 0 and 1, to map predictions to probabilities.

Hypothesis representation:

$$F(z) = \frac{1}{1-e^{-z}} \quad \ldots\ldots\ldots\ldots\ldots (1)$$

$$z = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n \quad \ldots\ldots\ldots (2)$$

In equation 2, w_0, w_1 …, w_n, represent the regression of the co-efficients of the model, obtained through another algorithm called Maximum Likelihood Estimation and x_0, x_1, …, x_n, represent the features or the independent variables.

Finally, F(z) (the hypothesis) calculates the outcome probability (binary) and based on the probabilities, the feature is classified into one of the two categories.

$$0 \leq h_\theta(x) \leq 1 \quad \ldots\ldots\ldots\ldots (3)$$

Hypothesis range (ɵ is the parameter i.e., w)

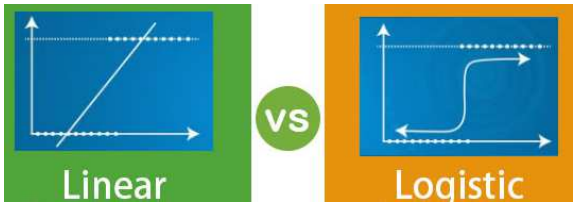

Fig. 3: Linear vs Logistic Regression

In Fig. 3, it is observed that the range for linear function can exceed the 0-1 range, which is not permissible for an outcome of logistic regression (as it is based on probability). Therefore, the sigmoid function is used as the hypothesis, thus, the outcome is confined within 0 and 1.

b) Decision Boundary

Classifier, built upon the concept of probability, gives an outcome in the following way. A new observation is passed to the hypothesis and a probability score between 0 and 1 is given as the output. For example, consider 2 classes, tiger and wolf (1- tiger, 0-wolf). In a simpler sense, a threshold value is decided above which values are classified into Class 1 and if the value goes below the threshold, they are classified into Class 2 [4].

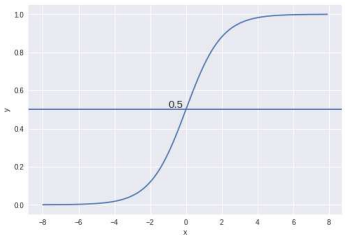

Fig. 4: Threshold value is 0.5

As shown in Fig. 4, the threshold value chosen is 0.5, so if a value of 0.7 is returned by the hypothesis, then this observation would be classified as Class 1 (tiger). If the prediction returns a value of 0.2 then, the observation would be classified as Class 2 (wolf).

c) Cost Function

Cost function represents optimization objective. It is created and minimized so that an efficient model with high accuracy can be developed [5]. For logistic regression, the cost function is defined in Equation 4.

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases} \ldots\ldots\ldots (4)$$

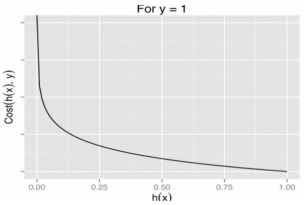

Fig. 5: Cost Function for y=1

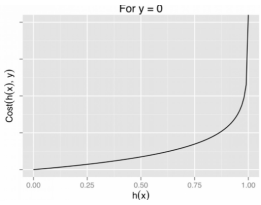

Fig. 6: Cost Function for y=0

In Fig. 5, it is observed that for an actual outcome of 1, if the hypothesis predicts a value 0, the cost function tends to infinity, indicating the model is not accurate. Similarly, in Fig. 6, if hypothesis predicts a value 1 for actual outcome 0, hypothesis tends to infinity. Hence, there is a need to minimize the cost function for an accurate model, which is done by another algorithm, gradient descent.

The function obtained from the combination of the two function is given by: $J(\theta) = -\frac{1}{m}\sum[y^{(i)} \log\left(h\theta(x(i))\right) + (1 - y^{(i)}) \log\left(1 - h\theta(x(i))\right)]$..... (5)

d) Gradient Descent

Gradient descent is an algorithm to achieve the minimization of cost function. This is accomplished by running the gradient descent function on each parameter (θ) as shown in Equation 6.

$\theta j := \theta j - \alpha \frac{\partial}{\partial \theta j} J(\theta)$ ..... (6) Where, α is the learning rate

Substituting the value of J(θ) and simplifying:

Want $min_\theta\, J(\theta)$:

Repeat {

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \text{..... (7)}$$

(Simultaneously update all θ_j)}

B. Naive Bayes Classifier

Naïve Bayes classifier is a linear classification algorithm commonly used for text and document classification. It works on the principle of Bayes formula in probability stated in Equation 8.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad \text{..... (8)}$$

where, P(A|B) = posterior probability; P(B|A) =likelihood; P(A) =prior probability; P(B) =marginalization

It is called Naïve because it makes the assumption that all its features are independent from each other. This assumption may cause a few errors in the prediction; however, it has been found to have very high efficiency regardless. To understand the algorithm in better detail, take the example of a system of spam and ham messages shown in the Fig. 7 [6].

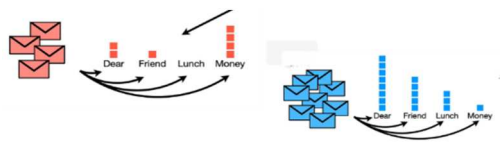

Fig. 7: Spam and Ham messages

Here, the probability of the term "Dear Friend" occurring in the spam and ham messages is to be found (posterior probability). The algorithm determines both these values and the one with greater probability is the group into which it is classified. For this, 2 tables are created: the Frequency table and the Likelihood table as shown in Table 1 and Table 2. The probability of all the words occurring in the spam and ham messages (the different likelihoods represented by P(B|A) in the formula) are also noted down as shown in Fig. 8.

TABLE 1: FREQUENCY TABLE

Word	*Spam*	*Ham*
Dear	2	8
Friend	1	5
Lunch	0	3
Money	4	1
Total	7	17

TABLE 2: LIKELIHOOD TABLE

Word	*Spam*	*Ham*	*P(word)*	*P(word)*
Dear	2	8	10/24	0.4167
Friend	1	5	6/24	0.25
Lunch	0	3	3/24	0.125
Money	4	1	5/24	0.2083
Total	7	17	24	
	7/24	17/24		
	0.29167	0.7083		

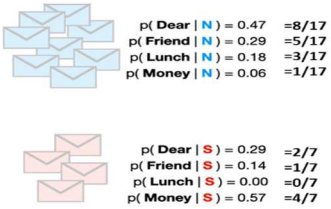

Fig. 8: Representation of Likelihoods

Probability of "Dear Friend" occurring in spam and ham is found using the formula:

$$P(N|Friend) = \frac{P(N) \cdot P(Dear|N) \cdot P(Friend|N)}{P(Dear) \cdot P(Friend)} = 0.92$$

$$P(S|Friend) = \frac{P(S) \cdot P(Dear|S) \cdot P(Friend|S)}{P(Dear) \cdot P(Friend)} = 0.11$$

Here, it is seen that the probability of "Dear Friend" occurring in the normal message is higher, hence the algorithm classifies it as ham. In the actual algorithm, a step called Laplace smoothing takes place to avoid occurrences of zero. It involves addition of 1 to both the denominator and numerator of the function while finding the prior probability. While testing the algorithm, the log of each conditional probability is found and added to give the total probability. Based on the highest value of total probability, classification is done.

- Types Of Naïve Bayes:

Bernoulli NB: It is used for classification of binary data. Involves use of Bernoulli distribution. In this paper, it has been used for the phishing website detection model.

Multinomial NB: It is used for classification with discrete features. In this paper, it has been used for classification of the phishing emails.

Gaussian NB: Specifically used when the features have continuous values.

C. Decision Tree Classifier

Decision Tree Algorithm is one among the most prevalent Supervised ML Algorithms. It involves the creation of a tree

containing nodes with the purpose of classification. It is considered to be one of the most frequently used algorithms because it mimics human thinking. The tree forms an integral part in classification since it incorporates various factors and conditions provided by the dataset. The decision tree involves the following 3 parts (shown in Fig. 9): -

- Root Node: The top of the tree forms the most important node called the root node and the branching begins from here.
- Decision Nodes: These are the internal nodes of the tree. They have arrows pointing towards as well as away from them.
- Leaf Nodes/Leaves: Leaf nodes mark the end of the path of the decision tree and have only arrows pointing to them. When the leaf node is reached, it implies that the classification is complete.

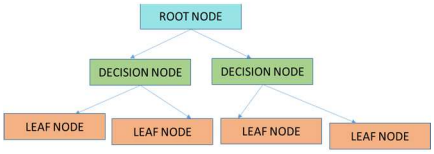

Fig. 9: Parts of a Decision Tree

The process of dividing a node into two or more sub-nodes is called splitting. Initially, the entire population of the dataset sample will be present at the root node. When the model is run, the respective entries get divided up based on their answers to the decision nodes. Finally, at the end, the programmer obtains homogeneous sets in the leaf nodes.

The main challenge faced in this model is the formation of the decision tree. After data pre-processing, many attributes of the dataset will be highlighted. For example, in diagnosing a person's illness, cough, throat pain, fever, body ache form some of the attributes. Now, which attribute must form the root node is to be ascertained. Since the root node is of utmost importance, the attribute whose contribution is highest must be placed there. To further explain this, a simple model that checks for heart diseases by choosing the attributes as chest pain, good blood circulation and blocked arteries is considered. To know which attribute must be at the top, each attribute must be compared with the result. It must determine which attribute alone predicts the best out of the three. The attributes do separate out the patients but not perfectly. Since all these measurements are impure, a method of comparing the impurities is adopted. The attribute with the lowest impurity forms the node. The most popular method of computing the impurity is called "Gini" and the impurity is hence called "Gini Impurity". The formula for calculation of GI is given by the Equations 9 and 10.

$$GI = 1 - \sum_{i=1}^{n}(p)^2 \quad \ldots\ldots\ldots\ldots (9)$$

$$GI = 1 - [(P_{(+)})^2 + (P_{(-)})^2] \quad \ldots\ldots\ldots (10)$$

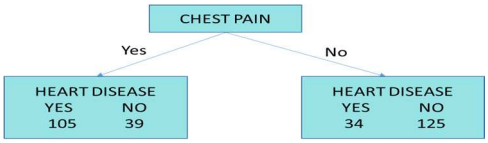

Fig. 10: An example of Decision Tree

The tree in Fig. 10 shows that when the answer for chest pain is YES, 105 people actually had heart diseases while 39 of them did not. When the answer for it is NO, 125 of them were not diagnosed of any heart diseases while 34 of them did actually have it [6]. The Gini impurity in this case is calculated by,

$$GI = 1 - \left(\frac{105}{105+39}\right)^2 - \left(\frac{39}{105+39}\right)^2 = 0.395 \quad \ldots\ldots\ldots\ldots (11)$$

$$GI' = 1 - \left(\frac{34}{34+125}\right)^2 - \left(\frac{125}{34+125}\right)^2 = 0.336 \quad \ldots\ldots\ldots\ldots (12)$$

So, the mean impurity GI=weighted average of the two Gini impurities

$$GI\ for\ Chest\ Pain = \left(\frac{144}{144+159}\right)0.395 + \left(\frac{159}{144+159}\right)0.336 = 0.364 \quad \ldots\ldots\ldots (13)$$

In the similar fashion, it is calculated for the other attributes as well. The factor whose Gini Impurity value is least will form the root node. The same principle is applied to the decision nodes as well. The final decision tree is shown in Fig. 11.

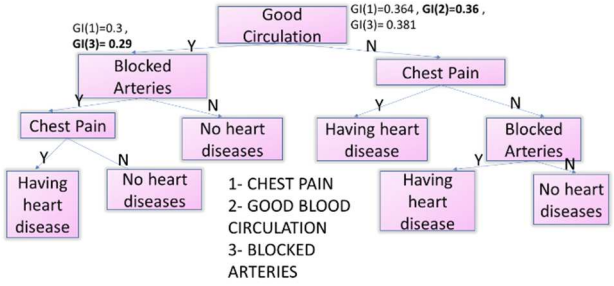

Fig. 11: Final Decision Tree obtained after applying Gini Impurity

D. *Random Forest Classifier*

Random forests are inclusive of decision tree. Decision Trees are less complicated as compared to Random Forest Classifier. Then the question that arises is what is the need of Random Forest classifier. The decision tree algorithm is quite straightforward i.e., easy to understand and interpret. But many a times, a single tree is not enough to produce accurate results. Therefore, the Random Forest algorithm comes into play. Decision Trees have only one aspect, hence they become imprecise when it comes to categorizing new samples. Decision Trees also tend to overfit. For a large amount of data, Random forests generates highly precise predictions. It can effectively estimate missing data maintaining a high accuracy. It also reduces the risk of overfitting by the use of multiple trees. Random Forest Classifier is an ensemble supervised ML algorithm which is quite flexible as they can be applied to both classification and regression problems. It randomly creates decision trees as base classifiers and then adopts majority voting to merge the results of all randomly generated decision trees. The randomization is done in 2 major steps i.e., using bootstrapped dataset and by taking into consideration a random subset of variables at each step [7].

Step 1: The 4 samples of the original dataset, as shown in Fig.12, is the entire dataset which will form the tree [8]. To create a bootstrapped dataset, samples are randomly selected from the original dataset as shown in Fig. 12. The important detail is that the same samples can be picked more than once.

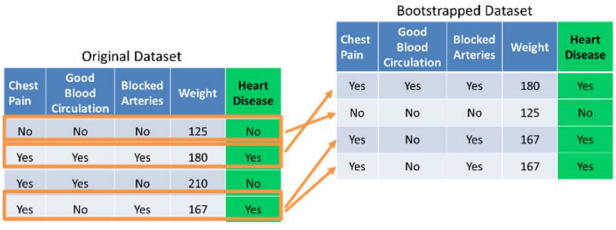

Fig. 12: Bootstrapped Dataset from Original Dataset

Step 2: A decision tree using Bootstrapped dataset is created, but only a random subset of variables is selected at every step. In this example only 2 variables are considered at each step. Thus, instead of considering all 4 variables to figure out how to split the root node, two variables are randomly selected. In this case Good Blood Circulation and Blocked arteries are randomly selected as candidates for root node.

Just for the sake of example assume that Good Blood Circulation did the best job separating the samples as shown in Fig. 13. Similarly, variables are randomly selected at each node and a decision tree is formed.

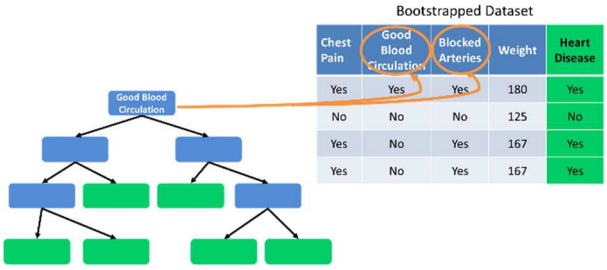

Fig. 13: Random subset of variables at each step

Now repeat the steps from step 1. Make a new bootstrapped dataset and construct a tree considering a random subset of variables at every step. In this way, a wide variety of trees are formed as shown in Fig. 14. The variety is what makes random forest more effective than individual decision tree.

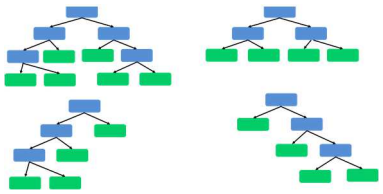

Fig. 14: Wide variety of trees

Now, the data is taken and run over the first tree obtained. Assume that the first tree says "YES" and keep track of it. And then repeat for all the trees obtained. After running the data through all the trees in the random forest, the option receiving more votes is observed. Based on the number of votes, the final result is concluded.

IV. Methodology

This section outlines the step-by-step procedure adopted in the creation of ML model to detect phishing websites and emails. Figure 17 shows the procedure. It involves the selection of appropriate legitimate datasets and functions from Python Libraries to improve efficiency. Further, training and evaluation of the model is conducted which gives a clear insight about how the model would function in real life applications.

- Data Collection: This step involves obtaining formerly collected data in the form of datasets from Kaggle, UCI, etc. The quantitative and qualitative aspects of the raw data procured dictates the accuracy of the model. In this paper, datasets have been chosen from a website called Kaggle. Phishing website dataset consists of 11,055 URLs with 6157 phishing sites and 4898 legitimate sites. Phishing Emails dataset contains approximately 5171 emails generated by employees of the Enron Corporation.

- Data Preparation: This comes under data-pre-processing wherein the data is converted into a table format displaying the desired features. Errors, duplicated sets, missing values are dealt with. Tokenizer from nltk (Natural Language Tool Kit) library is used to remove punctuations and make a list of tokens out of the text. Vectorization of the text is achieved using CountVectorizer() [9]. This creates a list of unique words and transforms the text into a vector on the basis of frequency of occurrence of each word.

- Selection of Model: Selection of ML algorithms from various libraries must be done depending on the operations that are to be performed. Choosing between supervised and unsupervised learning also comes into picture. Both the datasets in this model are labelled (contain a column indicating whether it is spam or ham) and therefore, the ML model comes under supervised learning. The model is required to classify a given input as either spam or ham, hence it is a classification problem. The classification algorithms chosen are: Random forest Classifier, Logistic Regression, Naïve Bayes Classifier and Decision Tree Classifier [7].

- Train the Model: The training dataset is utilized in teaching the model to attain the desired goal. Different instances from the training dataset are iterated through the model. This process is continued until the preferred accuracy is obtained. The model learns by trial and error and the difference is minimized gradually.

- Evaluate the Model: The model's proficiency is assessed by using the testing dataset. The model is then able to encounter situations that are not a part of its training. Hyperparameters can be optimized after evaluation to offer better accuracy.

- Make Predictions: This step gives a better approximation of how the model will perform in the real world by making predictions. The model becomes autonomous and can handle new data samples.

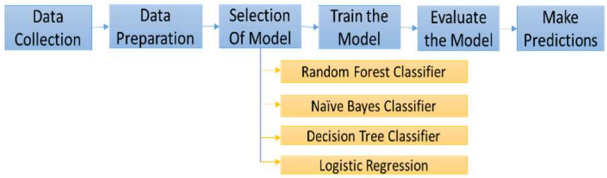

Fig. 17: Flowchart depicting methodology

V. Results and Discussions

After successfully creating the model and augmenting its efficiency, the comparison between Random Forest, Decision Tree, Logistic Regression and BernoulliNB ML algorithms is drawn with respect to accuracy and training time. These outputs have been visualized from Fig. 18 to Fig. 24 in the form of bar graphs and tables. Fig. 24 showcase few examples that display the real and predicted values of the test dataset.

- BernoulliNB has the least efficiency of 90.35% in phishing website detection and Decision Tree Classifier has the highest efficiency of 95.41%.

- Random Forest Classifier has the least efficiency of 78.16% in email detection and MultinomialNB has the highest efficiency of 98.06%.

- Random Forest takes the maximum amount of time and Naïve Bayers takes the least amount of time to train the model.

- Pre-processing of data greatly improved the efficiency of ML algorithms of the models.

- Tokenizer from nltk (Natural Language Tool Kit) library is used to remove punctuations and make a list of tokens out of the text.

- Vectorization of the text is done using CountVectorizer(). This creates a list of unique words and transforms the text into a vector on the basis of frequency of each word that occurs.

- It is noticed that Lemmatization of the text reduces the number of unique words by removing inflectional endings and returning the root forms, but also decreases the accuracies of the models by around 1%.

- Explicit removal of unimportant words like "subject" by the creation of user-defined functions improves the accuracies of the models by 0.6%.

	Model	Accuracy	Training time (sec)
0	DecisionTreeClassifier	0.954175	0.46
1	RandomForestClassifier	0.950558	0.62
2	LogisticRegression	0.925837	0.09
3	BernoulliNB	0.903527	0.02

Fig. 18: Accuracy and Training time of Phishing Website Detection Model

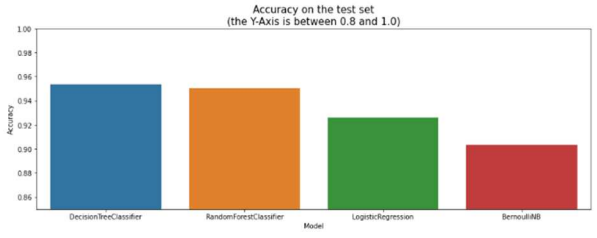

Fig. 19: Visualization of accuracy of 4 different algorithms

Fig. 20: Visualization of training time of the algorithms

	Model	Accuracy	Training time (sec)
0	MultinomialNB	0.980676	0.02
1	LogisticRegression	0.976812	0.46
2	DecisionTreeClassifier	0.946860	0.84
3	RandomForestClassifier	0.781643	0.99

Fig. 21: Accuracy and Training time of Phishing Email Detection model

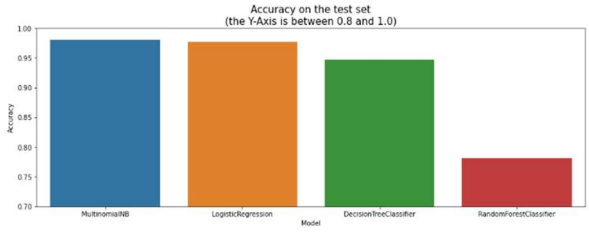

Fig. 22: Visualization of accuracy of the algorithms

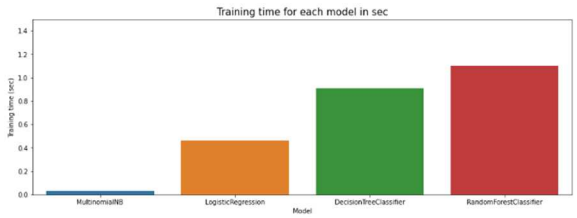

Fig. 23: Visualization of training time of the algorithms

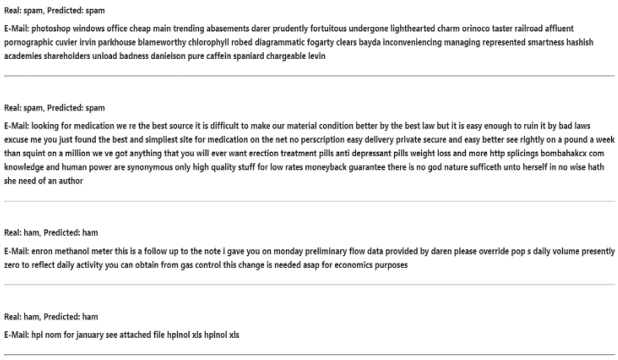

Fig. 24: Sample output

VI. Conclusion

This paper presents a thorough analysis of various classification algorithms used in ML models. It provides emphasis on the advantages of data analysis using ML. With the rising threat posed by phishing attacks, creation of more advanced detection systems is of utmost importance. Hence, this paper focuses on the creation of a phishing detection model as an application of ML and the results have been highlighted in pictorial form. MultinomialNB and Decision Tree offer the best accuracy. The training time is least for Naïve Bayes and highest for Random Forest classifier. A brief synopsis of the procedure to be followed is also discussed. The accuracy of the model can be improved further by the incorporation of deep learning concepts like semantic analysis and word embedding [10]. The amalgamation of different ML algorithms, which forms the general idea behind ensemble learning, can be used to attain highest competency.

References

[1] Artificial Intelligence in the Rising Wave of Deep Learning: The Historical Path and Future Outlook [Perspectives]Published in: IEEE Signal Processing Magazine (Volume: 35, Issue: 1, Jan. 2018) Page(s): 180 - 177 Date of Publication: 10 January 2018 ISSN Information: Print ISSN: 1053-5888 Electronic ISSN: 1558-0792 INSPEC Accession Number: 17454613 DOI: 10.1109/MSP.2017.2762725 Publisher: IEEE

[2] Mohammad Nazmul Alam, Dhiman Sarma,Farzana Firoz Lima,Ishita Saha, Rubaiath-E- Ulfath,Sohrab Hossain. Phishing Attacks Detection using Machine Learning Approach. Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020) IEEE Xplore Part Number: CFP20P17-ART; ISBN: 978-1-7281-5821-1

[3] https://www.tessian.com/blog/phishing-statistics-2020/

[4] Charu Singh, Smt.Meenu. Phishing Website Detection Based on Machine Learning: A Survey. Date of Conference: 6-7 March 2020 INSPEC Accession Number: 19557120 DOI: 10.1109/ICACCS48705.2020.9074400 Publisher: IEEE Conference Location: Coimbatore, India

[5] https://youtu.be/-la3q9d7AKQ

[6] https://youtube.com/c/joshstarmer

[7] Ammar Odeh, Ismail Keshta, Eman Abdelfattah. Machine Learning Techniques for Detection of Website Phishing: A Review for Promises and Challenges. Publisher: IEEE Published in: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) DOI: 10.1109/CCWC51732.2021.9375997

[8] Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, Dr.Aram Alsedrani. Detecting Phishing Websites Using Machine Learning. Publisher: IEEE Published in: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) Date of Conference: 1-3 May 2019 DOI: 10.1109/CAIS.2019.8769571

[9] https://www.kaggle.com/databeru/spam-classifier-model-comparison-accuracy-97

[10] Sikha Bagui, Debarghya Nandi, Subhash Bagui, Robert Jamie White. Classifying Phishing Email Using Machine Learning and Deep Learning. 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security) Date of Conference: 3-4 June 2019 INSPEC Accession : 19112529 DOI: 10.1109/CyberSecPODS.2019.8885143 Publisher: IEEE Conference Location: Oxford, UK