ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

IDENTIFICATION OF FRAUDSTERS INVOLVED IN PHISHING BY DIFFERENT MACHINE LEARNING MODELS

By

MD. FAIYED BIN KARIM (170021030)
NUSHERA TAZREEN (170021047)
SAMIHA TARANNUM (170021095)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical and Electronic Engineering

Academic Year: 2021-2022

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
A Subsidiary Organ of OIC
Gazipur, Dhaka, Bangladesh
May 2022

Declaration of Authorship

We, Faiyed Bin Karim (170021030), Nushera Tazreen (170021047) and Samiha Tarannum (170021095) declare this thesis titled ‘An investigation of phishing techniques to detect phishing websites by means of complexity reduction’ as our work. We have done the above-mentioned thesis and all works related to it as done for the partial fulfillment of our Bachelor of Science in Electrical and Electronic Engineering degree. We stand testimony to the fact that this work has not been submitted anywhere else for obtaining any degree.

Submitted By:

_____

Faiyed Bin Karim, 170021030

_____

Nushera Tazreen, 170021047

_____

Samiha Tarannum, 170021095

IDENTIFICATION OF FRAUDSTERS INVOLVED IN PHISHING BY DIFFERENT MACHINE LEARNING MODELS

Approved By:

Safayat Bin Hakim

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology

Acknowledgements

First and foremost, we are most grateful to the Almighty for enabling us to work in sound health even in the face of a global pandemic.

We would than like to thank our honorable supervisor sir for guiding us through this journey with utmost patience and consistency. Even in the face of various uncertainties and roadblocks, he was steadfast in his guidance and supervision. We had earned multifarious knowledge of this topic in different fields for his insightful discussions.

We would than like to thank all the faculty members of EEE, IUT for their help and support.

Last but not the least, we would like to give our hearty thanks to all our well-wishers specifically our parents and those friends who had helped us with their immense knowledge wholeheartedly whenever we faced any problems. No matter how small the help was, be it indirect or direct, in the form of inspiration or acts of service we are utterly grateful.

Abstract

With digitization of the current age, number of fraudsters in the digital realm has increased manifolds. Although the internet can be used for much good of the general population, the increase in number of unscrupulous people in online is a grave danger to the general public. Among many of the vices in the internet, one of the common one is phishing. To tackle phishing many approaches has been taken, of them ML based approach is one of the leading approaches. In our research work, we compared and contrasted many ML models to find out which one is most suitable for phishing detection. Our research is unique in regards that we have integrated data preprocessing and reduced the number of features for complexity reduction. Among these models XGBoost brought the highest accuracy after the hyperparameter tuning which was 97.0455%.

Contents

List of Figures

List of Tables

Abbreviations

SVM	Support Vector Machine
SVC	Support Vector Classifier
LGBM	Light Gradient Boosting Model
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
QDA	Quadratic Discriminant Analysis
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Chapter 1

Introduction and Background

Phishing is a type of cybersecurity attack that aims to steal private personal and collaborative information from users by delivering fake emails that appear to be from a trustworthy source. Manipulating links, evading filters, forging websites, covert redirection, and social engineering are all strategies used in phishing attempts. Phishers' standard strategy is to create a spoofing website that is frequently a perfect replica of the genuine website. According to the United States' crime statistics, these were the most serious complaints, according to the FBI's Internet Crime Complaint Center. According to FBI figures [2] online crimes such as exploitation, theft, and fraud cost the United States $2.7 billion in 2018.

The IC3 suffered a loss of more than $1.2 billion in that year. It got 20,373 corporate email compromise (BEC) and email account compromise reports (EAC). From 2019 to 2020, the number of phishing events nearly doubled, from 114,702 to 241,324. Phishing was the most popular kind of cybercrime in 2020. Phishing reports surged 11-fold from 2016 to 2020. According to Verizon's Data Breach Investigations Report (DBIR) for 2021 [3] 43% of breaches of data were caused only by phishing or pretexting. Every industry has a different frequency of assaults. In the year 2020, 75 percent of firms throughout the world suffered phishing assaults, 35 percent spear phishing attacks, and 65 percent BEC attacks. However, there is a distinction to be made between an attempted and successful attack.

By country, the United States has 30 percent more successful phishers than the worldwide average. According to ESET's Threat Report [4], malicious email detections increased by 9% in 2020 between Q2 and Q3. Phishing assaults are carried out in 96 percent of cases using email, 3 percent by rogue websites, and 1 percent by phone. The rise in phishing assaults suggests that email communication networks are increasingly

infested with malware. Email is frequently the most widely utilized form of official communication, and as a result, it frequently contains the most critical information. According to Symantec research, one out of every 4,200 emails sent in 2020 will be a phishing email.

According to the survey, the number of sophisticated assaults has increased in recent years. Phishing offences have increased in recent years, according to the Anti-Phishing Working Group (APWG) [1]. Phishers can target network-level security, authentication, client-side tools, user education, server-side filters, and classifiers, among other things. Every phishing attempt has its own characteristics, but when seen together, they all follow a similar pattern. Our study will focus on a comparative examination of the best machine learning model to detect phishing trends, since machine learning methods have shown to be a great tool for recognizing patterns in data. Ada-Boost, Support Vector Machine (SVC, NuSVC, and linear SVC), Gradient Boosting, LGBM, and XGBoost are the machine learning algorithms used.

Chapter 2

Literature Review and Motivation

In the context of modern age machine learning has already become important in how modern services and organizations function. Models of machine learning are employed in a variety of fields, including social networking platforms, healthcare, and finance. Depending on the work at hand and available data, the techniques for training and deploying a model varies.

Machine learning model techniques are divided into supervised and unsupervised learning. They differ in terms of how models are created and the data that must be given for training. Both types of learning has its own benefits and drawbacks. So both type of techniques tackle different challenges [27].

Supervised machine learning requires data labeled as input and output data. Before being used to train and test the model, the data is labeled input or output by the data scientist. After learning relationship between input and output data, the model may be used to categorize and predict outcomes for new and unfamiliar datasets [27].

Because at least some of this technique involves human control, it's termed supervised machine learning. The vast bulk of data is unlabeled and unprocessed. In most cases, human engagement is essential to appropriately classify data that is ready for supervised learning. Naturally, as enormous arrays of precisely labeled training materials are required, this can be a resource-intensive procedure.

In unsupervised machine learning, in spite of configuration of model hyperparameters like number of cluster points by human, the model analyzes large amount of data without human intervention [27]. Thus, it is very efficient. So, unsupervised machine learning is better suited to answering queries concerning patterns and relationships inside data that were previously unknown [27].

For detecting phishing websites, supervised machine learning models have shown to be effective. Several approaches for identifying phishing websites from legal websites have been presented in recent years. A previous study by Shahrivari et al. [6] categorized a dataset identical to ours by training and evaluating multiple classification, regression, and artificial neural network models. 90:10 train test sets were created from the dataset. The SVM model was investigated using a variety of kernels, including linear, RBF, sigmoid, and polynomial. For a better outcome, these kernel routines optimize some particular parameters. Tree classifiers, boosting methods, and a logistic regression model were also trained. The KNN model was trained with several K values, with the K value with the greatest accuracy being 95.27%. In addition, the ANN model was applied and trained on a variety of hidden layer numbers, yielding a 96.9879 percent accuracy. The XGBoost method had the highest accuracy of all the models, with a score of 98.3235 percent at a train-test ratio of 90:10.

The degree of relevance of each feature varies with relation to each other, hence feature selection in the dataset might be crucial while training any machine learning models. Displays a detection strategy by utilizing feature selection and ensemble learning, as proposed by Ubing et al. [9]. This approach uses a dataset with 5126 entries and 30 characteristics. The features in this strategy were chosen using a random forest regression model and then trained using an ensemble earning model. The model's highest level of accuracy was 95.4 percent.Subasi et al. [7] preferred Random Forest Classifier above other classification models in their methodology. ANN, k-NN, SVM, C4.5, and Rotation Forest were the remaining classification models. On the supervised classification models, the research used the same dataset. Several performance measures, including ROC area, F-measure, and accuracy, were assessed after the models were trained. Precision and recall levels were used to calculate the F-measure. Random Forest Classifier exhibited the greatest accuracy and F-measures among the models, with 97.36 percent and 97.4 percent, respectively. Despite the precision and robustness of the Random Forest Classifier's output, the dataset features were not pre-processed. As a result of maintaining the characteristics the same, the training model can maintain its complexity.

In addition to traditional models, hybrid models have been proposed, which may be a more reliable method. Tahir et al. [8] used such resilient strategies in their suggested research, where two separate supervised learning algorithms were integrated to generate a hybrid model and boost performance output. J48, IBk, Random Forest, Nave Bayes, Bayes Net, SMO, and FURIA were some of the individual classification models used. Two of these models were chosen at the same time for the hybrid model. Individual accuracy was best in the RF and IBk, both of which were above 95%. Following the training and testing of the hybrid models, it was discovered that all of the hybrid models had a test data accuracy of greater than 97 percent.

The maximum accuracy was 97.75 percent, which was discovered to be the result of two hybrid models, which are appropriately BN, IBk models and J48, IBk models. It was clear from the observations that hybrid models outperform separate categorization models.

The types and number of characteristics retrieved from the dataset determine the likelihood of effectively detecting a website. Hong J. et al. [12] conducted a comprehensive investigation in which characteristics were retrieved from website URLs and 19 features were followed throughout the procedure. The emphasis was on the 18 lexical characteristics. After that, fivefold cross validation was used to divide the dataset into an 80:20 train-test ratio. Following the lexical patterns, the classification models were trained and evaluated.

The 1DConv+LSTM model had the best accuracy of 90.2 percent in the prediction result, but it was unable to recognize many websites owing to its poor recall and precision values, whereas the Random Forest classifier had superior precision and recall percentages of 84 percent and 85 percent, respectively. According to the findings of the investigation, the highlighted engineering-based model outperformed deep learning algorithms.

The research efforts have attracted attention as supervised machine learning models because of their excellent accuracy and exact output. However, the main disadvantage of these models is the large dataset with numerous characteristics and instances, which might make training and testing the models more difficult. Our research model is motivated by a desire to simplify things. Reduce the average train and test time of the model by lowering the feature while maintaining the performance correct. As a result, the detection procedure might become more reliable and consistent. In our suggested model, we included a preprocessing phase that will remove some characteristics based on their relevance, resulting in a more compact dataset. Furthermore, hyperparameter tuning was used, which enhanced performance metrics and resulted in better outcomes. Additionally, the model is trained and evaluated using a larger test set and a smaller training set, allowing it to detect more phishing websites.

Chapter 3

Types of Phishing Techniques

3.1.1 Link manipulation: A malicious link might be made to appear on a website as a hyperlink with a name. URLs that are misspelled but written similarly can also be used to send people to malicious websites. Typosquatting is described as the registration of a domain name strikingly similar to that of a well-known brand or corporation (for example, www.paypal.com and www.paypa1.com) [1. IDN Spoofing is a more deceitful type of typosquatting. Phishers utilize a non-English character that looks identical to an English character. For example, instead of English "c" or "a," use a Cyrillic "c" or "a." [23].

3.1.2 Filter Evasion: Phishers employ photos to illustrate the contents of their website or utilize Adobe Photoshop to circumvent filter detection, leaving some phishing detection systems worthless. The easiest way to avoid this sort of assault is to employ recognition by optical characters, which is the electrical or mechanical translation of typed, handwritten, or printed text into machine encoded data [24-25].

3.1.3 Website Forgery: Some cross-site scripting attacks are carried out by modifying the Javascript code, which are difficult to detect due to the victim utilizing a legitimate website [25].

3.1.4 Covert Redirect: When attempting to offer access to a genuine website, users grant a token to a malicious service [25].

3.1.5 Social Engineering: Phishers utilize scamming methods on unsuspecting people in this way. Phishers look into users' weak personalities and communicate with them in order to get security information [25]. Baiting, scareware, pretexting, and spear phishing are examples of social engineering phishing techniques [25].

3.1.6 Techniques based on the address bar: If the URL includes an alternate domain name, such as an IP address in hexadecimal form, there is a good chance it is a phishing site. To mask the suspicious component of the URL, a lengthy URL is frequently employed. Phishers have been observed taking it a step further to avoid detection by recognizing lengthy URLs utilizing URL shortening services such as small URL [25].

Using @ in the web address is another method of sending consumers to malicious websites. All characters before @ are ignored. The '//' sign [25] can be used in phishing to redirect to a phishing site.

To make consumers believe they are dealing with a legitimate company, phishers append prefixes or suffixes to the domain name, separated by (-). A genuine web page http://www.Confirme-amazon.com/ [25] is an example.

Multiple subdomains are frequently an indication of a phishing website. If there is only one dot left after removing the www. and ccTLD, the link is regarded valid. The link is suspect if there are two dots. In case of more than two dots, the URL is probably a phishing site [25].

3.1.7 HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer): Make sure the site the user is visiting has been validated by reputable certificate issuers like 'GoDaddy' and 'Geo Trust,' among others.

3.1.8 Domain Registration Length: Phishing websites are known to have a limited lifespan. The majority of reputable websites have a one-year or longer existence [25].

3.1.9 Favicon: A favicon is a little graphic that is connected to a website. If the favicon comes from a different domain, the website might be a phishing site. The website is valid [25] if the Favicon is loaded from the same domain as the URL bar.

3.1.10 Using a Non-Standard Port: The port function is used to check whether a specific service on a specific server is up or down. It is a valid website if a port is in a chosen state (open or closed). It's a phishing website if it doesn't have a favored status. [25]

3.1.11 The Domain's "HTTPS" Token URL part: In order to deceive consumers, phishers frequently utilize https in the domain part. [25]

3.2 Abnormal based features:

3.2.1 Request of URL: The website is authentic if the URL request is less than 22%. If the percentage is between 22% and 61%, the website is questionable and maybe a phishing site. It's a phishing website if it's more than 61 percent.
Anchor's URL is The anchor is a tag-defined element. A website is authentic if the proportion of the URL of the anchor is less than 31%. It's suspicious if it's between 31% and 61%. It's a phishing website if it's longer than that.

3.2.2 Links in <Meta>, <Script> and <Link> tag: If the proportion of links in the "Meta>," "Script>," and "Link>" tags is less than 17%, the website is authentic; if it is between 17% and 81 percent, the website is suspect; and if it is more than that, the website is a phishing website.

3.2.3 URL of Anchor: An anchor is an element specified by the a> tag. If the proportion of the anchor's URL is less than, it is suspicious; else, it's safe.

3.2.4 Submitting Information to Email: If “mail ()" or "mailto:" function is used to Submit User Information, it is a phishing website. Otherwise, the website is legitimate [25].
3.2.5 Abnormal URL: If host name is not included in URL, it is a phishing website. Otherwise, it is a legitimate website [25].

3.3 HTML and JavaScript based Features:

3.3.1 Website Forwarding: If the number of redirect pages is less than or equal to one it is a legitimate website, if it is within 2 to 4, the website is suspicious. If there are more than 4 redirect pages the page it is surely a phishing website [25].

3.3.2 Status Bar Customization: If onMouseOver Changes Status Bar, it is a phishing website. If it does not change status bar it is a legitimate website [25].

3.3.3 Disabling Right Click: If right click is disabled, there is high probability that it is a phishing website. If right click is not disabled then it is a legitimate website [25].

3.3.4 Using Pop-up Window: If pop-up window uses text fields, there are high chances of the website being a phishing website. If there are no text files in pop-up windows the site is legitimate [25].

3.3.5 IFrame Redirection: IFrame is an HTML tag used to display an additional webpage to the currently shown one. Phishers make use of this tag and make it invisible. If there is an absence of this tag then the website can be deemed legitimate.

3.4 Some Domain based features:

3.4.1 Age of domain: If age of domain is 6 months or more then the website is safe [25].

3.4.2 DNS Record: If there is a DNS record of the website then the website is safe [25].

3.4.3 Website Rank: If website rank by popularity is less than 100,000 then it is a legitimate website. If it is greater than 100,000 then it is suspicious. Any more than that and the website is considered a phishing website.

3.4.4 Page rank: If page rank is more than 0.2, it is legitimate. Otherwise it is a phishing website.

Chapter 4

Methodology

4.1 Dataset Description: The major goal of our study is to collect a collection of traits that may be used to describe the properties of any website available online. The dataset was developed and supplied by Mohammad et al. [21] from the UCI machine learning repository [23]. The PhishTank website's web data is included in the collection. There are 11055 cases and 30 characteristics in total in the dataset [22]. The occurrences were separated into two groups depending on whether the website was phishing or not. The dataset is summarized in Table I.

TABLE I. Website Data Distribution

Split Set	Phishing	Legitimate	Total
Train Set	3428	4310	7738
Test Set	1470	1847	3317
Total	4898	6157	11055

4.2 Data Preprocessing: Data preprocessing is preparing raw data for use in a machine learning model. It's the initial step and very crucial in creating a machine learning model.

Before completing any data-related action, the data must be cleaned and formatted. As a result, we employ a data preprocessing procedure

Real world data may have noise. Additionally, it may also have also have missing values or be in a format not appropriate. Thus, directly using machine learning models becomes

hardly possible. In cleaning and preparation of data for any machine learning model, data preprocessing is a very important step. It is indispensable in enhancing the models accuracy and well as efficiency.

Machine Learning Data Preprocessing Steps:

1. Obtain the data set
2. Add all of the necessary libraries to your system.
3. Adding the data to the spreadsheet
4. To recognize and dealing with missing values
5. Categorical data encoding
6. Dataset segmentation
7. Scaling of features

Data-preprocessing may be broken down into two groups. The first one goes through how to deal missing data. In this stage of the data collection, it can be selected to ignore the missing values (called a tuple). For data that is noisy, the second data cleaning approach is used. If you want the entire process to function well, you must get rid of worthless data that can't be read by the systems.

After dealing with the problems, data preparation moves on to the transformation step. It is utilized to analyze data by converting it into meaningful conformations. Normalization, attribute selection, discretization, and Concept Hierarchy Generation are some of the methods that may be applied. Sifting through enormous datasets may take a long time, even with automated approaches. That is why the data reduction step is so important: it shrinks data sets by focusing them on the most critical information, enhancing storage efficiency while decreasing the financial and time costs of interacting with them.

The dataset is preprocessed to make it easier for the models to be trained more efficiently and reliably. Data preprocessing is a phase that takes a raw dataset and transforms it into a compact and reduced dataset that can be used to train models in the same way as the raw

dataset. Because our dataset comprises 11055 instances and 30 characteristics, we use the idea of correlation to classify the features based on their closeness on a certain parameter. Two common forms of co-relation coefficients are Pearson's product moment correlation coefficient and Spearman's rank correlation coefficient. Pearson's correlation formula has been used.

$$R_{x,y} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Where,

n = amount of feature data

x_i = ith data of feature x

y_i = ith data of feature y

$R_{x,y}$ = correlation coefficient output of feature x & y

Throughout the whole process each two from the whole features are compared and the similarities among the features are evaluated. It is found that, 9 features had more than 70% similarities compatibilities and are terminated. Furthermore, the remaining 21 features are ranked according to their importance by applying the Random Forest Classifier [14] algorithm and from that top 15 features were selected by dropping off 6 least important features. Visual representations of correlation heatmap and importance graph of the remaining 15 features are illustrated respectively in Fig. 1 and Fig. 2.

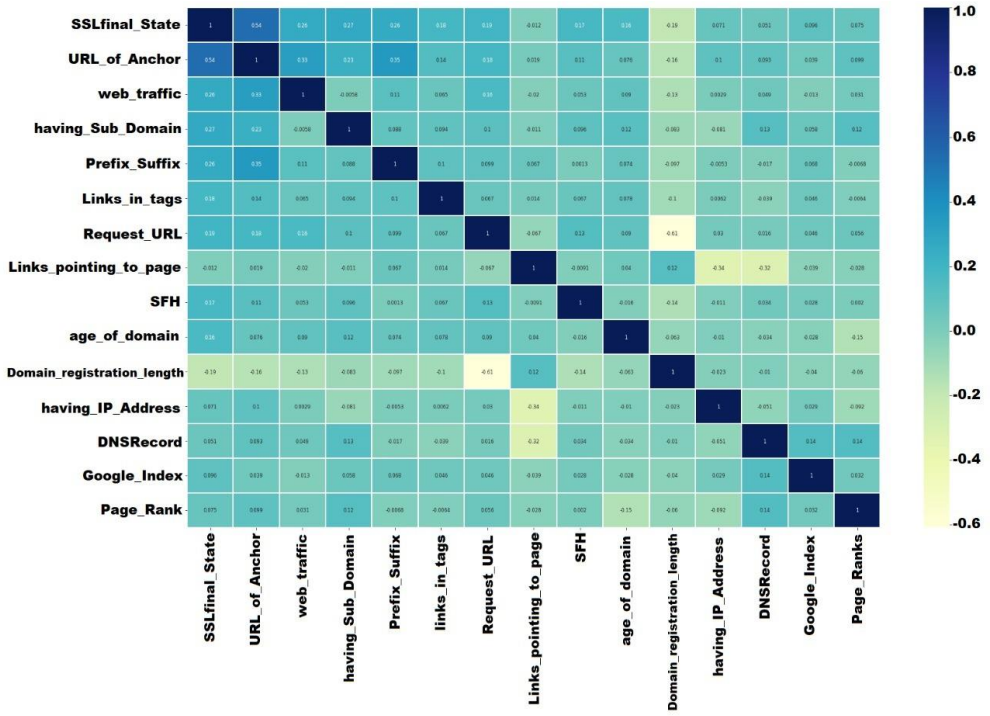

Fig. 1. Correlation heatmap of top 15 features

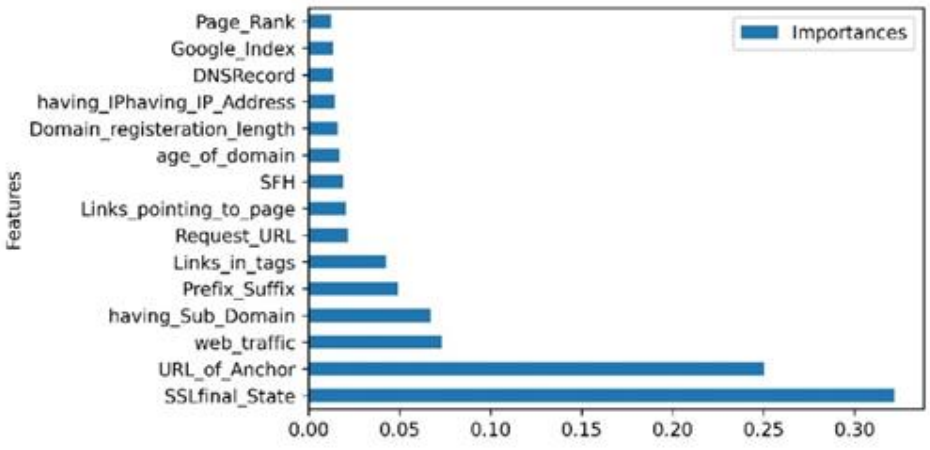

Fig. 2. Top 15 features from Random Forest Classifier

SSLfinal State has the greatest relevance score among the 15 attributes, as seen in the graph. The

values of the features are then scaled between 0 and 1 using the MinMax scaler. The dataset is then partitioned into a 70:30 train test ratio. This is accomplished through the use of a Stratified Train-Test Divided, which separates the dataset according to their suitable

instance ratio, with attributes split appropriately according to their availability. Hyperparameter adjustment was done while the model was being trained. Grid search cross validation (GridSearchCV) was used to make the hyperparameter tuning process automated, easing the complexity of human tuning and reducing time consumption.

SSLfinal State has the greatest relevance score among the 15 attributes, as seen in the graph. The values of the features are then scaled between 0 and 1 using the MinMax scaler. The dataset is then partitioned into a 70:30 train test ratio. This is accomplished through the use of a Stratified Train-Test Divided, which separates the dataset according to their suitable instance ratio, with attributes split appropriately according to their availability. Hyperparameter adjustment was done while the model was being trained. Grid search cross validation (GridSearchCV) was used to make the hyperparameter tuning process automated, easing the complexity of human tuning and reducing time consumption.

Chapter 5

Study of Algorithms

5.1 Support Vector Machine (SVM): Support vector machines [16] are classifiers that extract the nearest point between two classes based on the biggest distance in between them, with the margin being the shortest gap between any two classes. The maximal margin classifier, which is produced in the (rare) circumstance when two classes are linearly separable, is particularly sensitive to outliers in training data. To make the threshold less vulnerable to outliers, we allowed misclassifications.

We set a threshold highly sensitive to training data (low bias), but underperformed when introduced to new data (having high variance). As a consequence, we selected a threshold that was less sensitive to training data while still allowing for a modest number of misclassifications (higher bias). When misclassifications are permitted, the difference between them grows.

$$(w*, b*) = min(||w||/2) + ci \sum_{i=1}^{n} \xi i$$

Where c is the maximum number of errors that can be made. We acquire it via tweaking hyperparameters, and c stands for regularization. The value of error is indicated by the letter i. Under generalized Kernel functions, also known as Radial Basis Functions, SVM employs

polynomial Kernel and Radial Kernel (RBF). It is carried out in order to locate Support Vector Classifiers in greater dimensions. The Kernel function looks like this:

$$K(\bar{x}) = \begin{matrix} 1 & if \ \| \bar{x} \| \leq 1 \\ 0 & otherwise \end{matrix}$$

By adjusting the degree of polynomial, represented as d, the polynomial Kernel progressively increases the dimensions. The Support Vector Classifier is determined by the relationships between each observation pair. Radial Kernel locates Support Vector Classifier in an unlimited number of directions, essentially determining the connection between two locations. It acts in the same way as the Weighted Nearest Neighbor model. Kernel functions calculate relationships between every pair of points considering them to be in higher dimensions; nevertheless, the transformation is not performed. The 'Kernel Trick' computes high-dimensional relationships. But it does not actually transform them. SVC and nuSVC are theoretically comparable in scikit. The approaches are both based on the libsvm library. The main difference is that SVC uses the parameter c. The c parameter tells the SVM how much misclassification we may accept for each training example, while nuSVC utilizes the nu parameter. The nu parameter serves as a lower constraint on the number of support vector samples. It also serves as an upper limit for the number of samples that fall outside the hyperplane.

5.2 Ada Boost: AdaBoost is abbreviated from Adaptive Boosting. It is an ensemble method in machine learning.

During training data, Ada Boosting creates a certain number of decision trees. In the initial decision tree/model, the improperly categorized record is given priority. Only the misclassified data stated above are used as input for the second model. The procedure continues until the desired number of basic learners is determined. All boosting strategies allows repetition.

Unlike Random Forests, which uses n trees with appropriate roots and leaves, Ada Boost uses just stumps, which means the method creates nodes with two leaves.

The steps to ada boosting are summarized below:

1. The first stump is created by the algorithm using the first characteristic. It will generate the same number of stumps as features, and decision trees will be generated from the features. The algorithm will choose one model based on Giny and Entropy from among the models developed from each attribute. First base learner will be chosen from the stump with the lowest value.

2. The error is added to all of the sample weights errors in the categorized record.

3. Determining the stump's performance The Stump's Performance Formula is as follows:

$$Stump\ Performance = \frac{1}{2}\ln\frac{[1-TE]}{TE}$$

where, ln is natural log and TE is Total Error.

4. Updating Weights:

For records classified incorrectly,

$$New\ Sample\ Weight\ = Sample\ Weight \times e^{Performance}$$

For correctly classified records,

$$New\ Sample\ Weight\ = Sample\ Weight \times e^{-Performance}$$

5. The new dataset created using and considering the normalized weights will have more incorrectly classified records than correct ones. It will probably select wrong records and that will be the send $stump\ /decision\ tree$.

To sum up in a single formula:

$$F_T\,(x) = \sum_{t=1}^{T} f_t\,(x)$$

where,

F_T is a weak feature.

Error function E_t is calculated as,

$$Et = \sum_{i} E[F_{t-1}(x_i) + \alpha h(x_i)]$$

5.3 Gradient Boosting: Gradient Boosting is similar to Ada Boosting, except instead of producing stumps from earlier rounds, Gradient Boosting creates trees. A set learning rate is used to achieve a balance between bias and variation. The Ada Boost model finds the flaws by using h data points having high weight. Gradient Boosting, on the other hand, employs the same techniques but utilizing gradients in the loss function [28]. The loss function is a standard for how well the coefficients of the models fit the underlying data [28].

The steps involved in gradient boosting are as follows:

1. *F0(x)*, the initialization function for the boosting algorithm, is defined as

$$F_0(x) = armin_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$$

2. The gradient of the loss function is calculated iteratively by the following:

$$r_{im} = -\alpha[\frac{\delta(L(y_i\ ,\ F(x_i))}{\delta F(x_i)}]_{F(x)=F_{m-1}(x)}$$

Where,

α = learning rate

3. At each step gradient gains a fit hm(x).

4. Afterwards, the multiplicative factor γm for each terminal node is derived. The boosted model Fm(x) is defined as follows [29]

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

5.4 XGBoosting: Extreme Gradient Boosting [26] is what XGBoosting stands for. Each tree improves traits that have caused previous trees to be misclassified. It's the most effective of all the boosting algorithms. It prevents overfitting by regularizing its boosting. It can automatically detect missing data and enable early quitting by determining the appropriate number of repetitions. It makes the best use of all of the computer's cores and runs many threads in parallel. Another of the XGBoost algorithm's numerous advantages is that it can prune its trees backwards, resulting in deeper and more optimized trees.

Booster (gbtree or gblinear), goal (multi:softmax or multi:softprob), Max depth (which fixes tree depth), and min child weight are some of the XGBoost hyperparameters that need to be modified (it controls overfitting but setting it too high will cause it to underfit). The generalized objective function at the t^{th} iteration is:

$$L^{(t)} = \sum_{i=1}^{n} l(y_{i,y_i^{i(t-1)}}) + f_t(x_i)) + \Omega(f_t)$$

here,

ft = minimization variable of objective function
ŷ_i^t = prediction on t^{th} iteration at t^{th} instance
Ω = model complexity penalizing function
L = loss function

5.5 LGBM: Light is represented by the letter L. Light GBM is made to work quickly. Gradient boosting is what it is. It separates the tree leaf-by-leaf, instead of separating it level-by-level. By specifying the depth of splitting, overfitting may be avoided. It outperforms existing boosting methods in terms of speed of training and how efficient it is, reducing use of memory, improving accuracy, supporting parallel, distributed, and GPU earning, and handling enormous amounts of data. The following is the gain equation:

$$V_{(j|O)}(d) = \frac{1}{n_o}\left(\frac{\left(\sum_{\{x_{i\epsilon O||x_{ij\leq d}\}} g_i\right)^2}{n^j_{l|o}(d)} + \frac{\left(\sum_{\{x_{i\epsilon O||x_{ij\leq d}\}} g_i\right)^2}{n^j_{r|o}(d)}\right)$$

where,

j = feature number

V(d) = function of the largest gain in feature

5.6 LDA: For supervised classification problems, a dimensionality reduction method goes by different names. It is known as Linear Discriminant Analysis, Normal Discriminant Analysis, or Discriminant Function Analysis is frequently used. It's used to show group distinctions like splitting two or more classes. It's a technique for projecting higher-dimensional properties onto a lower-dimensional area.

Classes can have a number of qualities in order to efficiently split two classes. As seen in the picture below, using only one feature to categorize them may result in some overlap. As a result, the number of attributes must be raised or appropriate classification is necessary.

It's somewhat similar to PCA (Principal Component Analysis). Principal Component Analysis is an unsupervised learning method [26]. It reduces dimensionality in machine learning. It transforms observations of correlated qualities to a collection of linear data that is uncorrelated via orthogonal transformation. It takes a statistical approach in doing so. The newly adjusted qualities are the Principal Components. It is one of the most popular exploratory data analysis and predictive modeling packages. It's a technique for reducing variations and extracting significant patterns from a dataset.

PCA seeks the lowest-dimensional surface on which the high-dimensional data is to be projected.

Both PCA and LDA rank new axes in order of importance. PC1 (first new axis created by PCA) accounts for most variation of data. LD1 (first new axis created by LDA) also

accounts for most variation of data. Both lets us dig in and see which features are driving the new axis.

It maximizes separability between two groups and makes best decisions. It creates new axis to maximize separation of two categories. Data is projected to the new axis.

The new axis is created by maximizing distance between the two means. Variety within each category is minimized. LDA calls variation scatter and is represented by s^2. If distance between means and scatter is optimized there is a nice separation. If there are more than two dimensions, axis is created that maximizes distance between two means for two categories while minimizing scatter.

5.7 QDA: QDA is a development of LDA. It stands for Quadratic Discriminant Analysis. Each class calculates its own variance estimate. It also calculates covariance when there are multiple input variables).

In QDA, a Gaussian distribution is assumed for each class. The class-specific prior is the proportion of data points that belong to the class. The class-specific mean vector is made up of the average of the class-specific input variables [30]. The class-specific covariance matrix is made up of the covariance of the vectors in the class [30].

It provides for more flexibility in the covariance matrix. So, QDA is better at fitting data than LDA, but it has more parameters to estimate. Parameters increase considerably with QDA as each covariance matrix will be different.

Chapter 6

Results and Analysis

6.1 Results and Analysis: Four types of boosting methods (GBC, AdaBoost, LGBM, XGBoost), three types of Support Vector Machine classifiers (SVC, NuSVC, and linear SVC), LDA and QDA were trained using the gathered dataset in our study, and performance measurements were generated from which performance parameters were assessed. Python was used to do all of the necessary simulations and assessments. The performance of the ML models has improved dramatically thanks to hyperparameter optimization. Tables II to XV provide the visual representations of the simulated confusion matrices.

A confusion matrix is another name for error matrix. It is a table pattern that allows to see how well an algorithm performs. Important metrics in prediction such as recall, specificity, accuracy, and precision are shown by confusion matrices. Confusion matrices are valuable because they enable us to compare values such as True Positives, False Positives, True Negatives, and False Negatives directly [31].

TABLE II. Confusion Matrix (XGBoost-Tuned)

XGBoost (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1410	60
	Positive	38	1809

TABLE III. Confusion Matrix (XGBoost-Without Tuning)

XGBoost (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1400	70
	Positive	35	1812

TABLE IV. Confusion Matrix (AdaBoost-Without Tuning)

AdaBoost (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1328	142
	Positive	80	1767

TABLE V. Confusion Matrix (AdaBoost-Tuned)

AdaBoost (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1331	139
	Positive	75	1772

TABLE VI. Confusion Matrix (Gradient Boosting-Without Tuning)

Gradient Boosting (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1373	97
	Positive	71	1776

TABLE VII. Confusion Matrix (Gradient Boosting-Tuned)

Gradient Boosting (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1402	68
	Positive	48	1799

TABLE VIII. Confusion Matrix (LGBM-Without Tuning)

LGBM (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1391	79
	Positive	46	1801

TABLE IX. Confusion Matrix (LGBM-Tuned)

LGBM (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1402	68
	Positive	33	1814

TABLE X. Confusion Matrix (SVC-Without Tuning)

SVC (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1355	115
	Positive	61	1786

TABLE XI. Confusion Matrix (SVC-Tuned)

SVC (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1382	88
	Positive	55	1792

TABLE XII. Confusion Matrix (NuSVC-Without Tuning)

NuSVC (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1306	164
	Positive	91	1756

TABLE XIII. Confusion Matrix (NuSVC-Tuned)

NuSVC (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1382	88
	Positive	55	1792

TABLEX IV. Confusion Matrix (Linear SVC-Without Tuning)

Linear SVC (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1312	158
	Positive	108	1739

TABLE XV. Confusion Matrix (Linear SVC -Tuned)

Linear SVC (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1330	140
	Positive	110	1737

TABLE XVI. Confusion Matrix (LDA-Without Tuning)

LDA (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1313	157
	Positive	94	1753

TABLE XVII. Confusion Matrix (LDA-Tuned)

LDA (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1317	153
	Positive	93	1754

TABLE XVIII. Confusion Matrix (QDA-Without Tuning)

QDA (Without Tuning)		Predicted	
		Negative	Positive
Actual	Negative	1469	1
	Positive	1391	456

TABLE XVI. Confusion Matrix (QDA-Tuned)

QDA (Tuned)		Predicted	
		Negative	Positive
Actual	Negative	1353	117
	Positive	113	1734

Performance measures have been calculated from the confusion matrices which included accuracy, precision, recall, F1_score, specificity, and error rate. Performance parameters can be represented in the following manner:

TABLE XVII Performance Metrics (Accuracy, Precision and Recall) of Machine Learning Models

Algorithm		Accuracy	Precision	Recall
XGBoost	**Without Tuning**	96.8345%	0.962806	0.98105
	Tuning	97.0455%	0.967897	0.979426
AdaBoost	**Without Tuning**	93.3072%	0.925616	0.956687
	Tuning	93.5484%	0.927263	0.959394
GBC	**Without Tuning**	94.9352%	0.948211	0.961559
	Tuning	96.5029%	0.963578	0.974012
LGBM	**Without Tuning**	96.2315%	0.957979	0.975095
	Tuning	96.9551%	0.963868	0.982133
SVC	**Without Tuning**	94.694%	0.939506	0.966973
	Tuning	95.69%	0.9532	0.9702
NuSVC	**Without Tuning**	92.3123%	0.914583	0.950731
	Tuning	95.69%	0.9532	0.9702
Linear SVC	**Without Tuning**	91.98%	0.9167	0.9415
	Tuning	92.46%	0.9254	0.9404

TABLE XVIII Performance Metrics (F1-score, Specificity, Error Rate) of Machine Learning Models

Algorithm		F1-score	Specificity	Error rate
XGBoost	**Without Tuning**	0.971842	0.952381	0.031655
	Tuning	0.973628	0.959184	0.029545
AdaBoost	**Without Tuning**	0.940895	0.903401	0.066928
	Tuning	0.943055	0.905442	0.064516
GBC	**Without Tuning**	0.954839	0.934014	0.050648
	Tuning	0.968767	0.953741	0.034971
LGBM	**Without Tuning**	0.966461	0.946259	0.037685
	Tuning	0.972915	0.953741	0.030449
SVC	**Without Tuning**	0.953042	0.921769	0.05306
	Tuning	0.9616	0.9401	0.0431
NuSVC	**Without Tuning**	0.932307	0.888435	0.076877
	Tuning	0.9616	0.9401	0.0431
Linear SVC	**Without Tuning**	0.9290	0.8925	0.0802
	Tuning	0.9329	0.9048	0.0754

Thus, the performance metrics were assessed and represented in Table XVI and Table XVII where Table XVI represents precision, accuracy and recall and Table XVII represents F1-score, specificity and error rate. Both tuned and without tuned cases have been considered for the evaluation.

Table XVI shows that XGBoost with hyperparameter adjustment has the greatest accuracy of all the trained machine learning algorithms, with a score of 97.04 percent. XGBoost also has the greatest accuracy among the ML models without hyperparameter tweaking, at 96.83 percent. Again, when it comes to precision, XGboost on hyperparameter optimization has the highest ratio of 0.967897. However, with a ratio of 0.98105, the greatest recall for XGBoost without hyperparameter adjustment has been observed. It is clear from the results that the XGBoost algorithm outperforms competing models in both tuning and non-tuning scenarios. Furthermore, both SVC and NuSVC performed better than Linear SVC among the three SVMs, and the accuracies of all the models in the study were higher.

Above that, the accuracies of all the models are more than 91 percent in both circumstances (tuned and untuned), demonstrating the durability of the machine learning techniques as well as the preprocessing phase.

Table XVII further shows that the F1-score and specificity for all machine learning models are both good, with tweaked XBoost providing the greatest performance with an F-score of 0.973628 and specificity of 0.959184. Furthermore, the error rates are less than 0.09, which is extremely low in all models. The features of the dataset were decreased due to dataset preparation, while the feature effect was preserved, and following hyperparameter adjustment, the models' performance was improved by a significant margin.

6.2 Comparison of Accuracy: The number of true positives and true negatives divided by the total number of true positives, true negatives, false positives, and false negatives is called accuracy [32]. A data point that the algorithm properly identified as true or false is referred to as a true positive or true negative. A false positive or false negative is a data point wrongly categorized by the algorithm. It would be a false positive if the algorithm identified an erroneous data point as true.

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ True\ Values + All\ False\ Values}$$

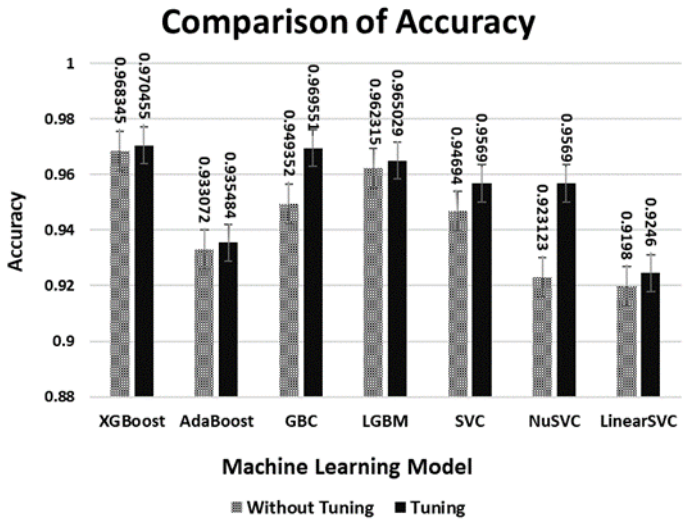

Fig. 3. Accuracy Diagram of Trained Models

6.3 Comparison of Precision: Precision is defined as the number of true positives divided by the total number of positive predictions [33].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

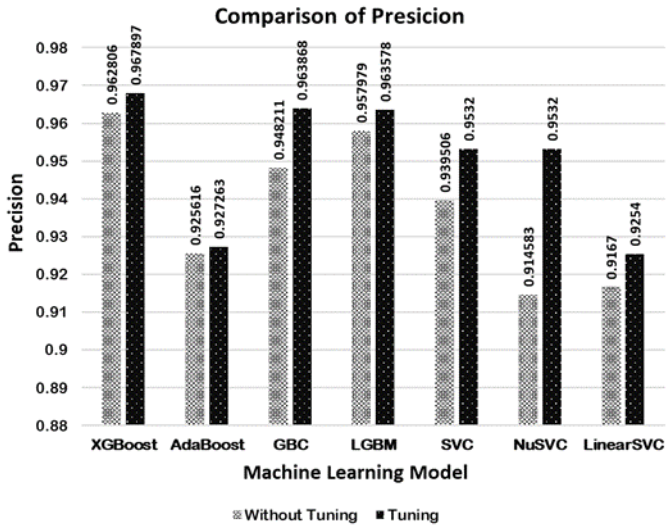

Fig. 4. Precision Diagram of Trained Models

6.4 Comparison of Recall: The recall is calculated by multiplying the total number of Positive samples with the number of Positive samples that were correctly classified as Positive [34]. The recall measures the model's ability to recognize Positive samples. Higher recall implies more positive samples are discovered [34].

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

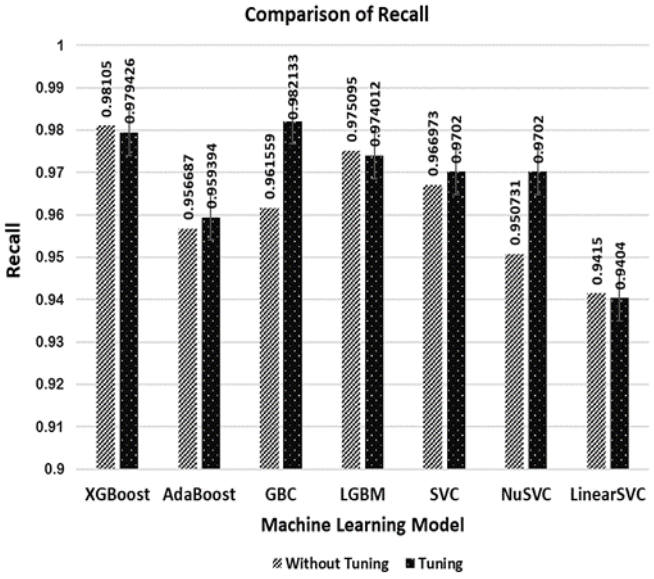

Fig. 5. Recall Diagram of Trained Models

6.5 Comparison of F1-score: The F Score or F Measure is the other name for F1-score. To put it in a different light, the F1 score represents the balance of accuracy and recall.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

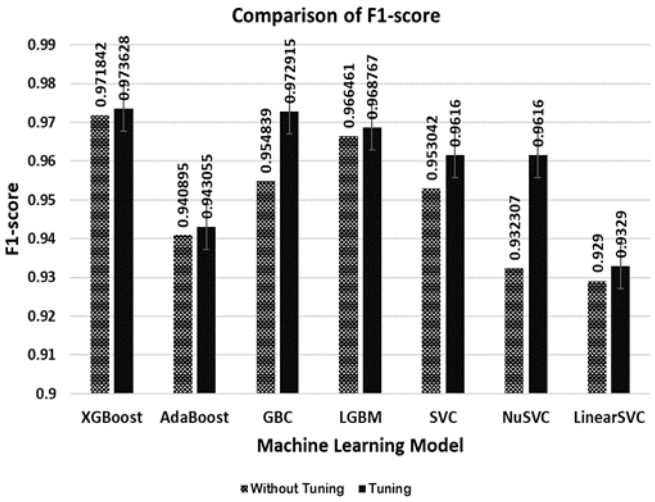

Fig. 6. F-1 score Diagram of Trained Models

6.6 Comparison of Specificity: The ratio of true negative and sum of true negative and true positive is called Specificity.

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive}$$

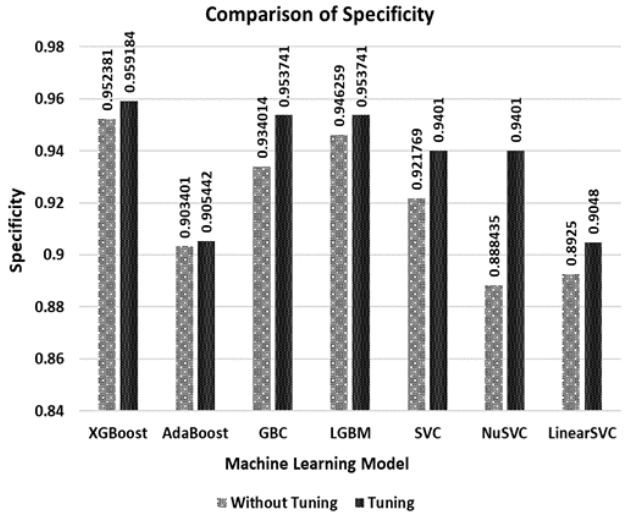

Fig. 7. Specificity Diagram of Trained Models

6.7 Comparison of Error Rate: Error rate expresses the proportion of cases where prediction is wrong.

$$Error\ Rate = \frac{False\ Postive + False\ Negative}{All\ True\ Values + All\ False\ Values}$$

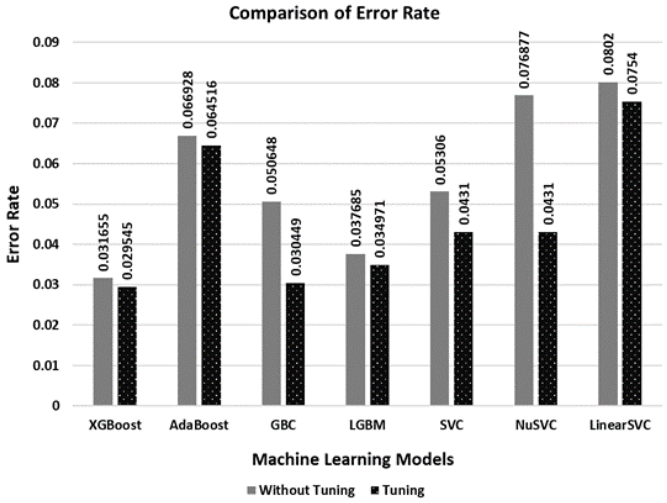

Fig. 8. Error Rate Diagram of Trained Models

6.8 Comparative Analysis with Other Works:

TABLE XIX Comparative Analysis with Other Works

Paper	Train-Test Ratio	Model	Accuracy
Shahrivari et al. [16]	90:10	XGBoost	98.3235%
		Random Forest	97.2682%
Ubing et al. [17]	Ratio not mentioned	Feature selection and ensemble learning	95.4%
Hong J. et al. [20]	80:20	1DConv+LSTM	90.2
Our models	70:30	XGBoost (tuned)	97.0455%
		LGBM (tuned)	96.955%
		XGBoost	96.8345%
		SVC	94.694%

Chapter 7

Conclusion and Future Works

7.1 Conclusion: The primary focus for our research was to establish a robust classification structure following the machine learning model that will not only give an efficient output but also sustain the efficiency and consistency of the classification efficiency while making the training process more compact by reducing the dataset feature through preprocessing. As technology has advanced and access to internet has broadened, it has become easier to create illegitimate websites which are becoming more vulnerable for the users' end. Various approaches have been made over the years to detect these phishing websites which certainly has been able to detect with much accuracy but due to the large number of websites being created very often, it has become perplexing to process such huge dataset. Our approached model can reduce the website features through preprocessing, yet keeping the feature characteristic importance unharmed. The dataset has also been split at 70:30 train test ratio which is considered to be an ideal ratio due to which higher number of websites can be classified. Also, hyperparameter optimization has been performed on the machine learning models to enhance the classification performance. Furthermore, a comparison among various boosting and SVM model over the phishing website detection has been visualized among which it was comprehensible that, XGBoost has outperformed other machine learning models which has achieved the optimum accuracy of 97.0455%. The approached model can possibly make significant impact in field of these website detection process and are set to perform consistently on vast websites with larger mass of features.

7.2 Future Works: In the future, we plan on integrating deep learning models in phishing detection. Deep learning falls under Machine Learning. While Machine learning tries to

automate every process with less human intervention deep learning tries to create models that are self-acting and work like the human brain. We plan to work with a larger dataset in the future unlike the small dataset we have used here.

References:

[1] V. Bhavsar, A. Kadlak, and S. Sharma, "Study on phishing attacks," Int.J. Comput. Appl, vol. 182, pp. 27–29, 2018.

[2] F. N. P. Office, "Internet crime complaint center2018 internet crime report," 2019. [Online]. Available:https://www.fbi.gov/news/pressrel/press-releases/fbi-releases-theinternet-crime-complaint-center-2018-internet-crime-report

[3] Verizon, "2021 data breach investigations report," 2021. [Online].Available: https://www.verizon.com/business/resources/reports/dbir/

[4] ESET, "From crisis response to transformation," 2020. [Online].Available: https://www.eset.com

[5] M. Rosenthal, "Must-know phishing statistics," 2022. [Online].Available: https://www.tessian.com/blog/phishing-statistics-2020/

[6] SonicWall, "2020 sonicwall cyber threat report: Threat actors pivot toward more targeted attacks, evasive exploits," 2020.[Online]. Available: https://www.sonicwall.com/news/2020-sonicwallcyber-threat-report/

[7] APWG, "Phishing activity trends reports." [Online]. Available: https://apwg.org/trendsreports/

[8] A. K. Dutta, “Detecting phishing websites using machine learning technique,” PloS one, vol. 16, no. 10, p. e0258361, 2021.

[9] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, “Urlnet: Learning a url representation with deep learning for malicious url detection,” arXiv preprint arXiv:1802.03162, 2018.

[10] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu,G. Mureddu, D. Ariu, and F. Roli, “Deltaphish: Detecting phishing webpages in compromised websites,” in European Symposium on Researchin Computer Security. Springer, 2017, pp. 370–388.

[11] M. M. Nishat, F. Faisal, T. Hasan, M. F. B. Karim, Z. Islam, andM. R. K. Shagor, “An investigative approach to employ support vectorclassifier as a potential detector of brain cancer from mri dataset,”in 2021 International Conference on Electronics, Communications andInformation Technology (ICECIT). IEEE, 2021, pp. 1–4.

[12] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, “Machine learning and deep learning methods for cybersecurity,”Ieee access, vol. 6, pp. 35 365–35 381, 2018.

[13] M. Ahsan, R. Gomes, and A. Denton, “Smote implementation on phishing data to enhance cybersecurity,” in 2018 IEEE International Conference on Electro/Information Technology (EIT). IEEE, 2018, pp.
0531–0536.

[14] W. Ali, “Phishing website detection based on supervised machine learning with wrapper features selection,” International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, pp. 72–78, 2017.

[15] V. S. Lakshmi and M. Vijaya, “Efficient prediction of phishing websites using supervised learning algorithms,” Procedia Engineering, vol. 30, pp. 798–805, 2012.

[16] V. Shahrivari, M. M. Darabi, and M. Izadi, “Phishing detection using machine learning techniques,” arXiv preprint arXiv:2009.11116, 2020.

[17] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Jhanjhi, and M. Supramaniam,“Phishing website detection: an improved accuracy through feature selection and ensemble learning,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 1, pp. 252–257, 2019.

[18] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, “Intelligent phishing website detection using random forest classifier,” in 2017 International conference on electrical and computing technologies and applications (ICECTA). IEEE, 2017, pp. 1–5.

[19] 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). IEEE, 2016, pp. 1126–113

[20] J. Hong, T. Kim, J. Liu, N. Park, and S.-W. Kim, “Phishing url detection with lexical features and blacklisted domains,” in Adaptive Autonomous Secure Cyber Systems. Springer, 2020, pp. 253–2

[21] A. Moubayed, M. Injadat, A. Shami and H. Lutfiyya, "DNS Typo-Squatting Domain Detection: A Data Analytics & Machine Learning Based Approach," *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-7, doi: 10.1109/GLOCOM.2018.8647679.

[22] V. B. et al, “study on phishing attacks,” International Journal of Computer Applications, 2018

[23] I.-F. Lam, W.-C. Xiao, S.-C. Wang, and K.-T. Chen, “Counteracting phishing page polymorphism: An image layout analysis approach,” in International Conference on Information Security and Assurance,pp. 270–279, Springer, 2009.

[24] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, “Advanced social engineering attacks,” Journal of Information Security and applications,vol. 22, pp. 113–122, 2015.

[25] Phishing Websites Data Set. Available at: https://archive.ics.uci.edu/ml/datasets/phishing +websites (Accessed on: 13 May 2022)

[26] Hura, & Vyas. (2021). Advances in communication and computational technology. Springer Singapore.

[27] https://www.guru99.com/supervised-vs-unsupervised-learning.html (Accessed on: 13 May 2022)

[28] Mathanker, S. K., Weckler, P. R., Bowser, T. J., Wang, N., & Maness, N. O. (2011). AdaBoost classifiers for pecan defect classification. Computers and electronics in agriculture, 77(1), 60-68.

[29] https://medium.com/almabetter/xgboost-dd38f73233fa (Accessed on: 13 May 2022)

[30] https://towardsdatascience.com/quadratic-discriminant-analysis-ae55d8a8148a?gi=a210488bc789 (Accessed on: 13 May 2022)

[31] https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e (Accessed on: 13 May 2022)

[32] https://www.nottingham.ac.uk/nmp/sonet/rlos/ebp/sensitivity_specificity/page_four.html (Accessed on: 13 May 2022)

[33] https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/#:~:text=Precision%20is%20a%20metric%20that,positive%20examples%20that%20were%20predicted. (Accessed on: 13 May 2022)

[34] https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/ (Accessed on: 13 May 2022)