National College of Ireland

Improving the classification rate for detecting Malicious URL using Ensemble Learning Methods

MSc Research Project
Cyber Security

Santosh Raj Bingi
Student ID: x20123035

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Santosh Raj Bingi
Student ID:	x20123035
Programme:	Cyber Security
Year:	2021
Module:	MSc Research Project
Supervisor:	Michael Pantridge
Submission Due Date:	16/08/2021
Project Title:	Improving the classification rate for detecting Malicious URL using Ensemble Learning Methods
Word Count:	XXX
Page Count:	25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Improving the classification rate for detecting Malicious URL using Ensemble Learning Methods

Santosh Raj Bingi
x20123035

Abstract

The surge in the use of the internet has created the biggest hurdle for the security of the digital world. Malicious URLs are the main source of performing phishing activities, transmission of viruses such as trojans, worms etc. Various malicious URLs try to retrieve user information by releasing distinct malicious software. A legitimate user who cannot detect and remove malicious URLs by end-users can leave them vulnerable. Malicious URLs also allow attackers to gain unauthorized access to user data. Therefore, it is essential step to identify the countermeasures for stopping such activities with the help of new and advance technologies. In order to correctly identify the URL as Malicious or benign, machine learning based methods has been considered as one of the efficient approach. However, using the machine learning approach the number of false positive and false negative outcomes are found to be more. Hence, there is still a scope of improvement for identifying correctly the URL as Malicious or Benign. In this research, a extended version of machine learning methods has been proposed where the properties of two or more models are combined, can be referred as ensemble learning methods. Using ensemble learning methods, we were able to achieve more accurate and better results.

1 Introduction

With the exponential growth of the internet, various activities have entered from physical means to digital means. Multiple new developments in information technology are driving e-commerce applications and meanwhile, also creating the new opportunities for attackers which lead to various malpractices. Today, a large number of these websites are available on the internet that are commonly referred to as malicious websites. Moreover, it is noted that due to advancements in technology, several technologies are used to scam or attack the user such as social network spam SMS, fraudulent prize-winning, phishing, online gambling, financial fraud and fake TV shopping. These methods are widely used these days for performing malicious practices. As the detection of malicious URLs is not yet fully resolved and causes huge losses each year. Based on the recent surveys in 2019, around 200k phishing URLs were detected. Various practices are implemented to safeguard the digital world, but these practices are not capable enough to perform efficiently. The security mechanism is not enhanced enough with technological advancement which causes problems in dealing with the malicious contents. For this purpose, researchers have conducted various practical implementations to gather the information and to provide effective solutions in identifying malicious URLs.

One of the oldest technique is the blacklisting approach, which has been used by the various antivirus companies. In the process of blacklisting method a large list of malicious URLs are maintained by the antivirus companies. If any link clicked by the user, matches with the current list of URL then it notifies the user about it. However, blacklisting has certain disadvantages such as new or unknown sites which not available in the database can not be detected, also searching the URL from large list of database is computationally exhaustive. To improve the security components, innovative applications based on machine learning and artificial intelligence have been developed to address these challenges from the past few decades. Researchers have come to favour machine learning and artificial intelligence predictions rather than blacklisting or signature-based methods for detecting malicious URLs. The working of this machine learning architecture in such a way that the model is feed with a set of URLs as training data to recognize the weights and features through predictive function for classifying the URL whether it is malicious or not. In this work, a combined analysis of machine learning based approach and ensemble learning based methods will be performed. This research comprises of total 7 algorithms where we will perform the comparative analysis between the each algorithm. We are categorizing the algorithms as machine learning method and ensemble learning method. For machine learning method we have deployed 3 different models named as logistic regression, naive bayes and decision tree classifier. On the other end, for ensemble learning approach 4 different algorithms have been deployed named as Random forest classifier, AdaBoost classifier, LightGBM classifier and XGBoost classifier. The main objective of this research is to identify the optimal algorithm for classification of Malicious URLs.

1.1 Research Question

- How efficiently Ensemble learning algorithms can reduce the false positive and false negative prediction rate and improves the model performance for URL Classification?
- Which algorithm efficiently classifies the Malicious and Benign URL from the large set of URL data?

2 Literature Review

Detecting malicious URLs is creating a big problem in academia. For solving these queries, the most important attention goes to machine learning algorithms that would be used to rectify these problems easily. In the field of machine learning and cybersecurity branches, more logistic and growing literary studies should be analyzed to explore the significant aspects. Machine learning is mostly geared towards empirical research that investigates the effectiveness of different techniques in experiments. However, hardly any realistic research has been carried out to try and develop a practical strategy that can deal with the huge amount of statistical analysis Gawale and Patil (2015). In addition, there is still much doubt in the potential of machine learning techniques that solely examine the stable characteristic of URL sequences to detect dangerous URLs. However, to provide an efficient solution to solve the particular challenges, various sub-questions are necessary to consider. According to the view of the study, this literature review is to solve these respective queries:

- What are the significant properties that help distinguish the malicious URLs?

- What are the major existing problems that are created due to malicious URLs?
- What are the techniques of machine learning to battle malicious URLs?
- What are the important lexical-based features that can be used for searching malicious URLs?
- Which are the major components or layers that should be added in machine learning architecture for designing a system that helps in detecting the malicious URLs?

These questions allow us to approach the research objectives by giving a framed theoretical concept on the literature. The main objective of this literary chapter is to resolve all the questions which could find primary and secondary sources. Furthermore, this section should be categorized into four sections for achieving the desired goal. This section includes Malicious URL categorization, attacks, and existing technology of machine learning for malicious URL detection.

2.1 Malicious URL Categorization

The term 'malicious URL' is the arbitrariness in the definition that mostly are an arguable weakness of the studies that lead to problems in the existing network. Malevolence is considered ambiguous, yet the amount of malice has to be lowered to make the danger clearer Janet et al. (2021). Thus, it could be more efficient to have categorized URLs that would give us a better knowledge of the features of the present malicious URLs. These framed URLs assist to establish a comprehensive master learning classification in the studied period and function as the crucial step. Interest in the scientific community in the last decade investigating the identification of malicious URLs. Mostly they did not define the main term malicious URL". In the experiment, phishing and spamming URLs are explained to provide information on the various attacks that are also marked as malicious in a single label. Contrary, the author asserted that malicious activities such as spamming and phishing have unlike properties and their identification is also varied.

Nagaonkar and Kulkarni (2016) had proposed the first experimental demonstration of categorization and separate detection of malicious URLs. above in this methodological study are conducted in which malicious URLs are observed in two stages in that the first one is that the benign and malicious should be divided by a machine learning binary classifier and the malicious URLs are allocated three different labels which are phishing, malware and spamming. The researchers also found that a URL in several categories should be available simultaneously. Tan et al. (2018), Kumar et al. (2017) has adopted a similar approach and claimed that harmful URLs must be separated from the object of the website visitors favour. They utilized the three types of malicious URLs namely malware, spamming, and phishing. Thus, these three types of malicious URLs are decided to accept which were commonly described by different scholars and it is considered as the basic attacks that are checked while evaluating the model.

2.2 Attacks

Various attacks such as phishing attacks and URLs obfuscation techniques are highlighted by Manyumwa et al. (2020)and Zhang et al. (2013). (2008) pointed out two common ways

to contract phishing URLs that further help in seeing the trends in phishing attacks. However, few authors have led a systematic study of phishing techniques. According to the same recent research that Manyumwa et al. (2020) performed, the success of the phishing attack depends on several factors. These elements include assignments, obfuscation techniques, time, and user devices. However, according to the goal of this treatise, it is required to go to the phishing URL for vocabulary attributes. Therefore, this subsection focuses on various phishing URL obfuscation techniques. In terms of methods, these systemic perspectives lead to the development of anti-phishing techniques with a more effective and holistic approach to solving phishing problems. In event of an attack, an intruder is likely to rely on a grammatical error, while the user unintentionally presses the adjacency command or empty letters to mistype the web link. Incorrectly entered website address directs the user to the phishing website. However, not all phishing attacks depend on it. It is similar to a regular Web resource URL. The interactive pictures rather than words are used as an additional URL geometric distortion strategy. Usually, this can be used for emails that include a JPEG picture. It is like a lawful e-mail from a bank or business that generally holds the official logo. In a similar context, Manyumwa et al. (2020) provided a detailed explanation about the advanced URL obfuscation methods actively used for phishing attacks. For example, using an alternate encoding is another obfuscation technique that makes URLs unrecognizable

2.3 Machine Learning Approach

Three different forms of machine learning techniques may be used for harmful techniques of URL identification: map learning, okara learning and ring learning. Various malicious URL methods rely on the technique for machine learning have been studied. SVM, logistic regression, naive bays, decision trees, ensembles, online learning and many more are part of these machine learning techniques. Two methods, SVM and RF are used in this study. The reliability and efficiency are displayed in the observational data for both methods with varied simulation results. URLs are components and structures in two principal clusters: static and dynamic. The researchers provided in the paper an approach for determining and extracting URLs with native support, server, and popularity. Interactive learning techniques and SVMs are the machine learning algorithms employed in this research. Malicious URL identification is illustrated using variable URL action Yang et al. (2019). This paper analyzes both static and dynamic URL characteristics. Certain attribute groups, including character and semantic groups, are examined. A special group of websites and groups based on hosts and correlation groups. The methodology uses static vocabulary highlighting extracted from URLs and has the question that these highlighting are significantly unusual for malicious or benign URLs. Leveraging these static highlights is generally safer and faster because it does not include many dormant URLs or boycott queries when making decisions. The purpose of the property is to achieve high impact, for example, to recognize many harmful URLs that can be reasonably expected Yang et al. (2019). URL strings are generally unstructured and noisy. Since then, storage calculations have generally produced a large number of students preparing for different parts of the preparation information, reducing the differences in these methods, and a solid business match has been found. Sequence models were tried on five different test sets and found to have a typical False Negative Rate of 0.1 percent and a general accuracy of 87 percent. The results obtained show significant evidence that the methodology of absolute vocabulary can be used to specify persuasive URLs.

2.4 Malicious URLs detection

Various authors reported that they applied a machine-running approach to the problem of detecting malicious URLs with promising results Yang et al. (2019). This article also explores related approaches that apply machine learning as well as other alternative technologies. After analyzing the published literature, the approach to detect fraudulent URLs by Machine Learning Approach includes Blacklist Approach and Heuristic Approach. It can be divided into two categories based on the number of papers published in the last decade, the Academy seems to be increasingly looking for solutions to this problem with its machine learning approach. It emphasized it, despite the many proposed solutions. However, few solutions are practically applicable in the current industry. In reality, there is a trade-off between the calculated price of the solutions and their performance, accuracy and speed. Emphasized the data collection and problems which are major obstacles to the machine learning approach, as they cannot be applied globally. However, in some cases, it shows that vocabulary-based features have not helped detect malicious URLs Femi (2013).

As previously reported, these high-weight features are more likely to detect malicious URLs. In this regard, the latest machine learning approaches to obtain the next whole image includes sources of data, applied features and applied machines. It is considered three-dimensionally in the learning algorithm. Previous studies on data collection have also suggested several options for collecting data that can be transformed into functions. Despite the high precision which can guarantee content-based functionality, still there are two main drawbacks to consider as well. The first concern is security, as extracting these features requires a full download of the web page Jagielski et al. (2018). This increases the likelihood that malicious code will be executed before the classifier labels it as malicious. The next is the use of resources, which requires significant calculation power and working time for all the aforementioned functions.

Consequently, it is debatable if it is beneficial to develop a content-based characteristic classification. The functionality that may be extracted at this stage is termed lexical URL and host-based characteristics. Content-based functionalities are not yet accessible. This information is usually obtained from the DNS server for host-based services. You may get the name, location, IP address, registration deadline and DNS Kumaraguru et al. (2008) update date information of the domain owner. In a previous chapter, we said that malicious URLs tend to change location from time to time and not only persist for a short period. The host-based algorithm thus has significant importance for an exact ranking. There are additional important host functions, such as connecting speed and IP address, that may be directly retrieved from the web host. It pointed out that changing the IP address for each new attack is difficult. Therefore, the information for the IP address helps the classifier's accuracy. However, that host-based functions have clear disadvantages, including implementations. For data gathering and forecasting, DNS servers cannot be used. The training data thus includes irrelevant features which might impair the sorter's performance Xu et al. (2013). Moreover, both the DNS server and the webserver might slow down regularly, affecting the anticipated speed too. However, provided the signal strength is fast enough, in a matter of seconds you may acquire some information about the host, too much in actual circumstances. Because of these characteristics, the host's functioning in a real setting is realistic. The function of the vocabulary base of the URL

is obtained by the name string of the URL. In other words, the classifier learns to distinguish malicious URLs and benign URLs based on their shape and text structure Lin et al. (2013).

In general URL length, length of the domain name, number of special characters, etc. measured features are extracted as important features. Binary characteristics are also retrieved as characteristics, such as the existence of some letters or words on the supplied URL. These features also have significant drawbacks. For example, you can think of it as an extension of the classifier blacklist approach, which is built solely on URL vocabulary-based features. One of the downsides of URL-based features is that they can algorithmically generate new URL names that can prevent classifiers. However, many studies analyze the alphanumeric distribution claiming that algorithms generated patterns can be recognized. The choice of algorithm is also important to step through the machine learning method. However, this section does not provide an in-depth assessment of the fundamentals of mathematics based on the opinions of previous scholars and experts in the field of machine learning. Algorithms can also be classified into batch mode and online mode according to the course mode. The procedure may not seem like an easy one, as no method can be implemented to all processes uniformly. A balancing of numerous transactions is therefore needed Kiruthiga and Akila (2019). According to an analysis of previously performed experiments, more specific map algorithms such as Naive Bayes (NB), Supported vector machine (SVM) and Logistic Regression (LG) are most liked by the scholars. About 30 studies that applied three algorithms were commonly investigated. Online learning algorithms as well. The summary of the literature review shows that we can achieve five of the seven research goals.

2.5 Literature Review Summary

As per our findings and analysis, it has been observed that lot of work has been carried out in this domain. Various authors and researcher has proposed different methods for identification of malicious URLs. The method of comprises of different data preprocessing technique, feature extraction methods, feature selection method, developing the new kind of models, usage of online algorithms and many more. However, in most of the studies we have found that experiments has been carried out over the small subset of URL dataset. Other than that using various machine learning methods, high number of false positive and false negative values are predicted. In this research, we have identified the scope of improvement using ensemble learning approach. In upcoming chapters, we will discuss about the proposed approach, our analysis, finding and will also discuss in depth about the evaluation results.

3 Methodology

The current digital scenario of today's era has seen a surge in internet usage. With the growing advantage of the internet, the disadvantage has also increased in proportionate terms. Each day, there is a formation of various URLs and due to which it is a complicated task for an internet user to identify a Malicious URL. Even though there are various approaches to identify malevolent behaviour but the approaches are not accurate, efficient and feasible. In this research we have proposed a framework for identifying the URL

as malicious or benign. Our research paper considers the various steps such as data acquisition, pre-processing, visualization, feature extraction and many other. In terms of model training, we are utilizing the both machine learning and ensemble learning approaches to identify and segregate the URL as benign or malicious. In the sub-sections below, we will discuss more on dataset acquisition, pre-processing of dataset, visualization of data, feature engineering, data modelling and model evaluation.

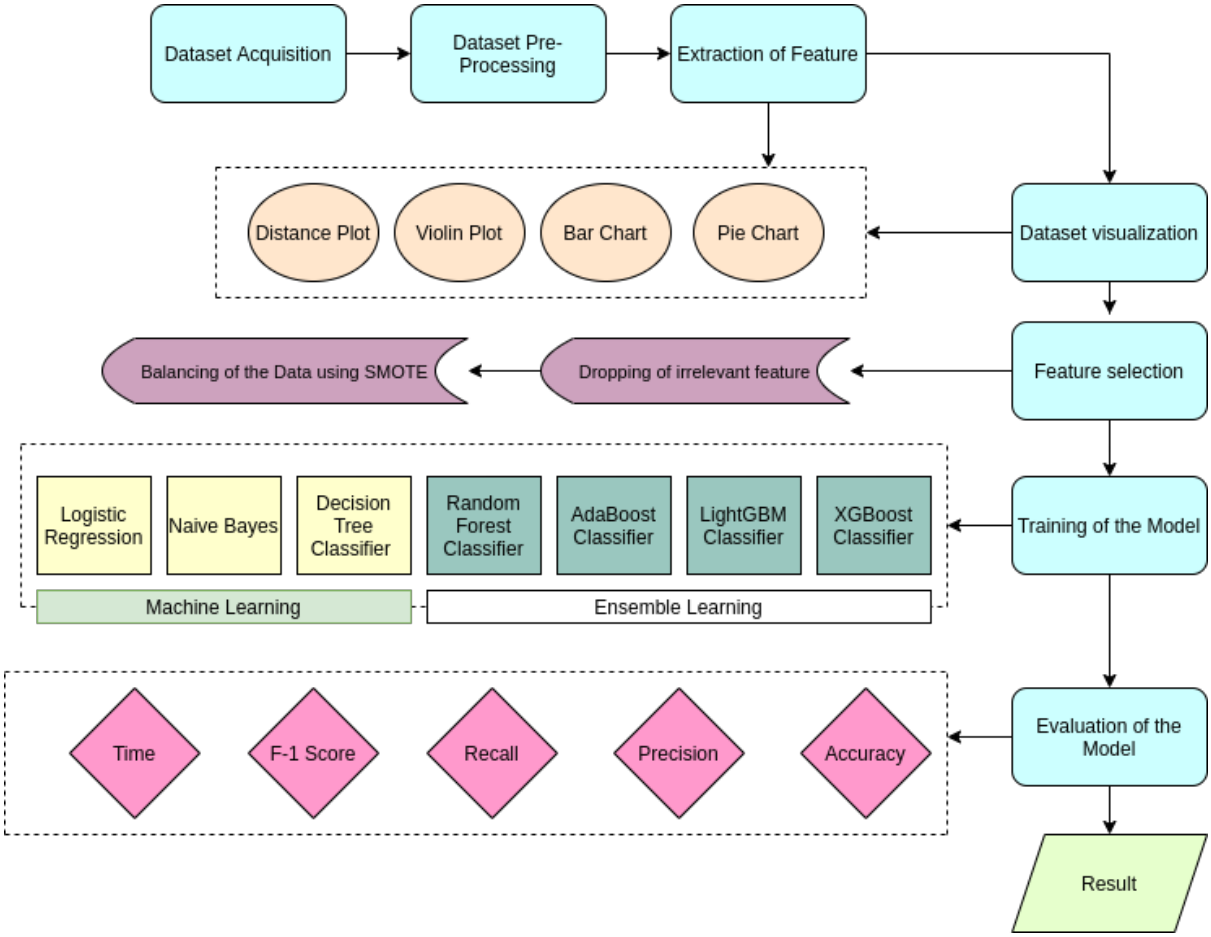

Figure 1: Proposed Framework for Malicious URL Detection

3.1 Dataset Acquisition

To detect the malicious URL, we need to acquire a large set of URL data for training machine learning algorithms. For this, we have acquired a pre-collected dataset of various URLs from Kaggle *Malicious And Benign URLs* (n.d.). The dataset contains 450,176 URL which was labelled to be either benign (safe) or malicious (un-safe). It was further shown in the numbered form where 0 meant benign and 1 meant malicious. The dataset contains only the URL and its associated label, it does not contains any other feature information. The dataset is found to be imbalanced in nature, the number of benign URLs are more in number as compared to the malicious URLs.

3.2 Dataset Pre-processing

The dataset pre-propcessing is the key step where the certain operations on the dataset is performed. Data pre-processing steps includes the identification of null and missing values, data balancing, data generalization and many others. After cleaning we have found reduction in the number of URL samples. Many of the URL links were not labelled in the dataset also the dataset was imbalanced in nature. In our dataset, we have used the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset by increasing the number of cases from the existing samples. Before balancing the dataset, it contained 70,004 URLs. The remaining steps, is explained in feature engineering steps in much more detail.

3.3 Extraction of the Feature

As the obtained URL dataset does not contains any feature information, we need to extract the certain features from the dataset. We will extract mainly two types of features from the URL dataset that are Lexical features and Host-based features. In the lexical features, we will extract the lexical properties of the URL such as count of TLD, count of different symbols, directory length, length of path, number of digits, argURLratio, pathURLratio, domainURLratio and many others. In host-based features we will extract the Header information, response of the URL, content-type, header length and loadtime of the URL. Combinely after extracting all these features, we have obtained 29 different features including the URL and target variable. Most of the extracted features contains the integer or float values. A list of all the extracted features is shown in Figure 2.

```
Data columns (total 29 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   url                     70004 non-null  object
 1   label                   70004 non-null  object
 2   result                  70004 non-null  int64
 3   status                  70004 non-null  float64
 4   loadtime                70004 non-null  float64
 5   content-length          70004 non-null  float64
 6   top_level_domain_count  70004 non-null  int64
 7   count_of_/              70004 non-null  int64
 8   directory_length        70004 non-null  int64
 9   has_ip_in_domain        70004 non-null  int64
 10  shorten_url             70004 non-null  int64
 11  lenght_of_path          70004 non-null  int64
 12  lenght_of_host          70004 non-null  int64
 13  length_of _url          70004 non-null  int64
 14  len_-                   70004 non-null  int64
 15  len_@                   70004 non-null  int64
 16  len_?                   70004 non-null  int64
 17  len_%                   70004 non-null  int64
 18  len_dot                 70004 non-null  int64
 19  len_=                   70004 non-null  int64
 20  count_of_http           70004 non-null  int64
 21  count_of_https          70004 non-null  int64
 22  count_of_www            70004 non-null  int64
 23  digit_count             70004 non-null  int64
 24  alpha_count             70004 non-null  int64
 25  pathUrlRatio            70004 non-null  float64
 26  argUrlRatio             70004 non-null  float64
 27  argDomainRatio          70004 non-null  float64
 28  domainUrlRatio          70004 non-null  float64
```

Figure 2: Feature Information

3.4 Dataset Visualization

After extracting the various features from the URL data, the information about the feature needs to be visualized in order to gain the better insight about the dataset. We have visualized various features of the URL dataset in the form of bar graph, Pie-chart, stack bar graph, violin graph and many others. Some of the information about the each feature of the dataset will be extracted, which will help to perform the better modelling.

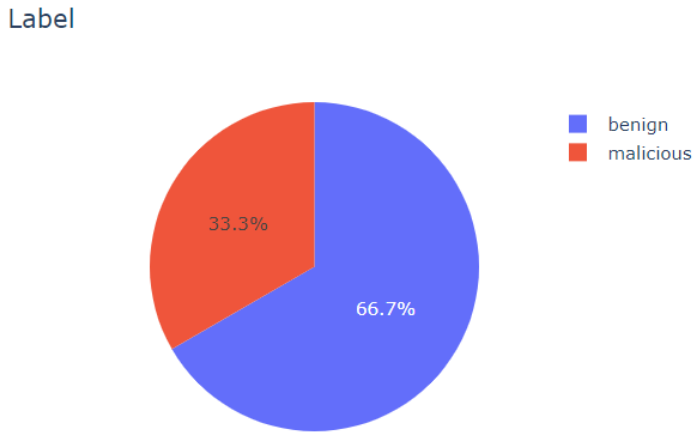

Figure 3: Label Distribution of URL Dataset

In the dataset insight visualization of pie chart in figure 3, it was found in the dataset that 33.3% were malicious URLs and the remaining 66.7% were Benign URLs.This represented the label distribution of the pre-collected information of the dataset.

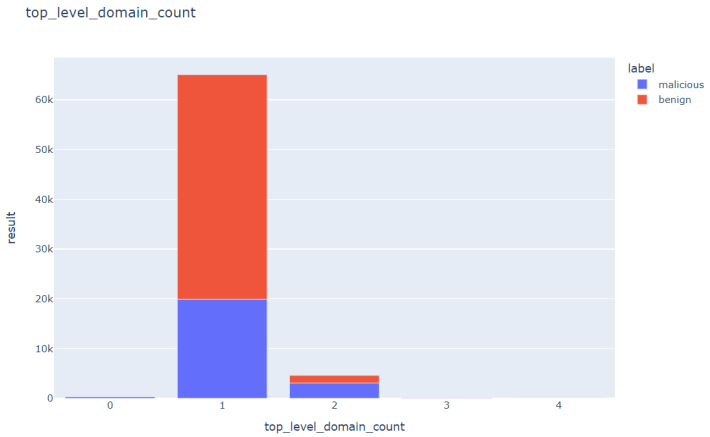

Figure 4: Top Level Domain Count

In another plot figure 4, the URL either without any extensions or with two extensions was majorly represented as malicious and on the other hand URL with one extension was majorly safe. As per the graph, we should be conscious with the zero top level domain count.

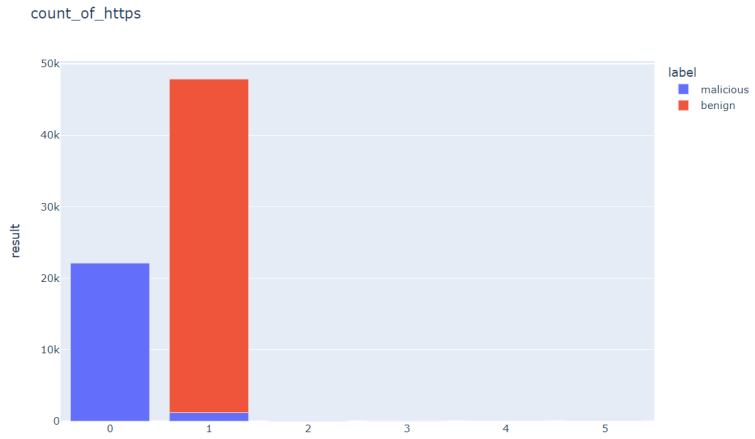

Figure 5: Count of HTTPS

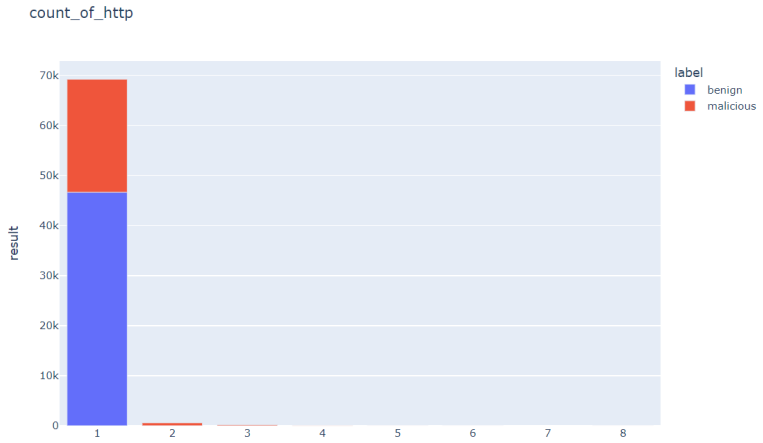

Figure 6: Count of HTTP

With concern to the SSL certification, URLs with HTTPS were very much safe and had very less chances of being malicious in figure 6. But, the one without SSL is all considered to be malicious in the figure 7. Therefore, it is always suggested to look the SSL certificate of the webpage before entering the critical information.

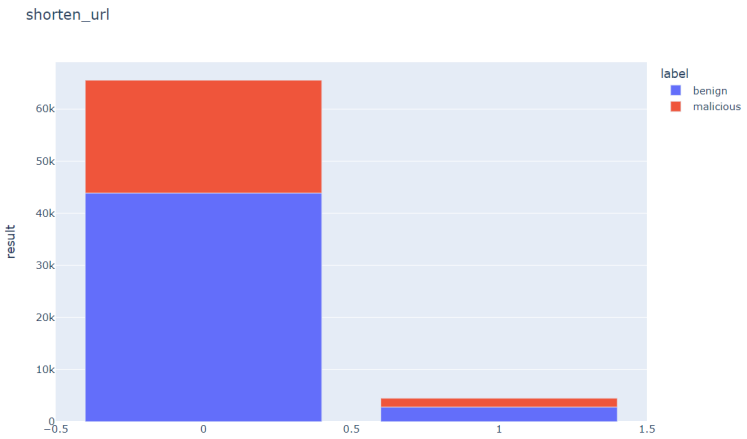

Figure 7: Shortened URL

Similarly in the figure 7, it was also found that the number of digits in the malicious URL was certainly longer than usual. As the same, we did the data visualization for all the 29 features of our dataset.

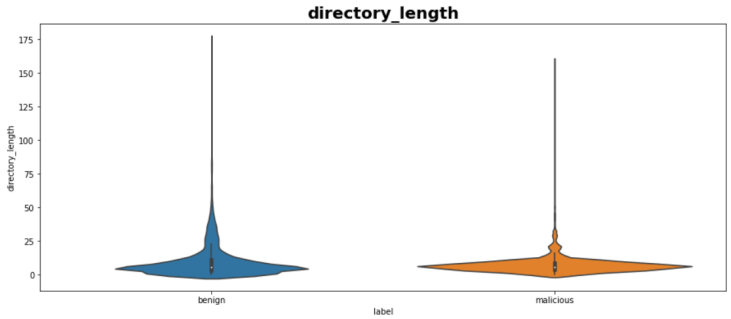

Figure 8: Directory Length

Also, bar charts for the count of lexical tokens were provided to different the URL to be benign or malicious. The violin plot in the figure 8 for the directory length represented that the malicious URL has a shorter length than the benign one.

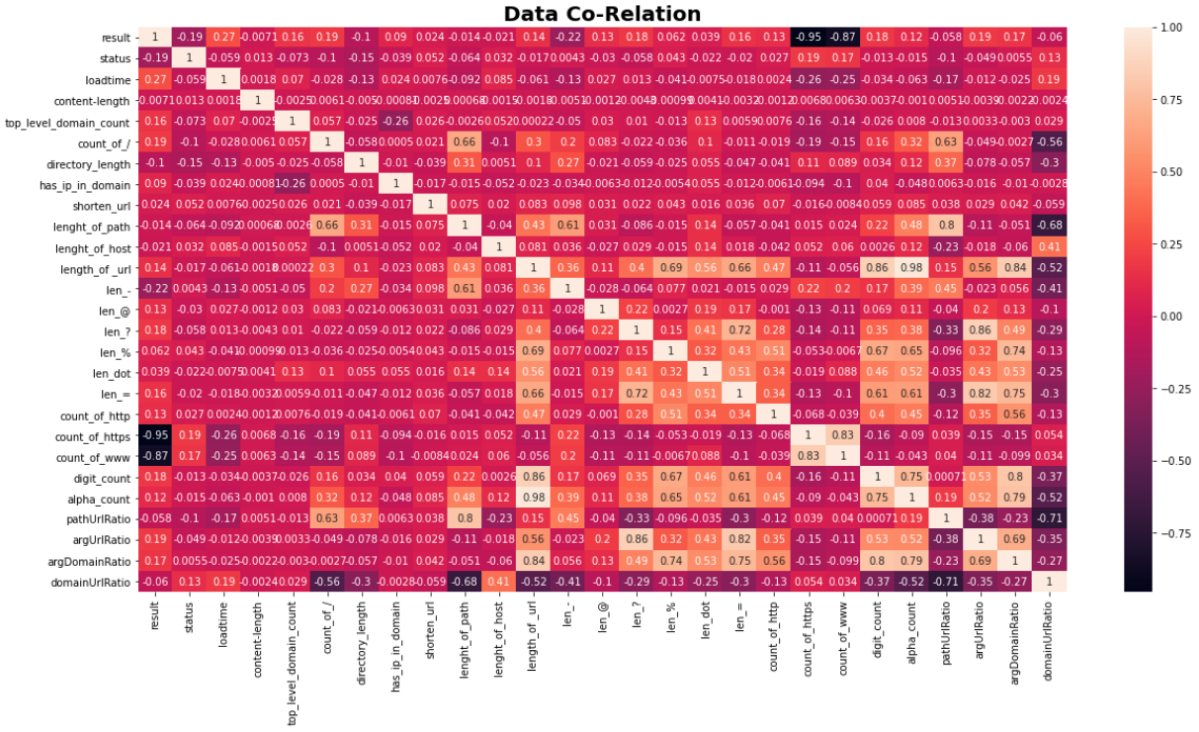

Figure 9: Correlation Matrix of URL features

The co-relation among the features is represented with the help co-relational metrix. By analysing the correlation matrix, we identify the highly correlated features. the highly correlated features can be removed as they generated noise in the dataset feature. Correlation matrix acts as feature selection techniques where highly correlated features can be dropped. By analyzing the correlation matrix shown in Figure 9 it has been observed that

alphabet count and the length of the URL are found to be highly-correlated. Therefore, any one feature can be removed from the data for achieving the better outcomes.

3.5 Model Training

Before training the model, the dataset needs to be balanced in order to avoid the biased results. Therefore, in this work we are using the SMOTE function for balancing the dataset. The dataset consist of approximately 70,000 number of samples. Among them 46665 are the benign URLs. On the other hand, the remaining 23,339 URLs are the malicious URL. After applying the SMOTE function the dataset is balanced where the number of samples for malicious and benign URLs are 46665. Collectively the balanced dataset contains 93,000 URL samples in total. Labels after balancing the dataset is shown in Figure 10.

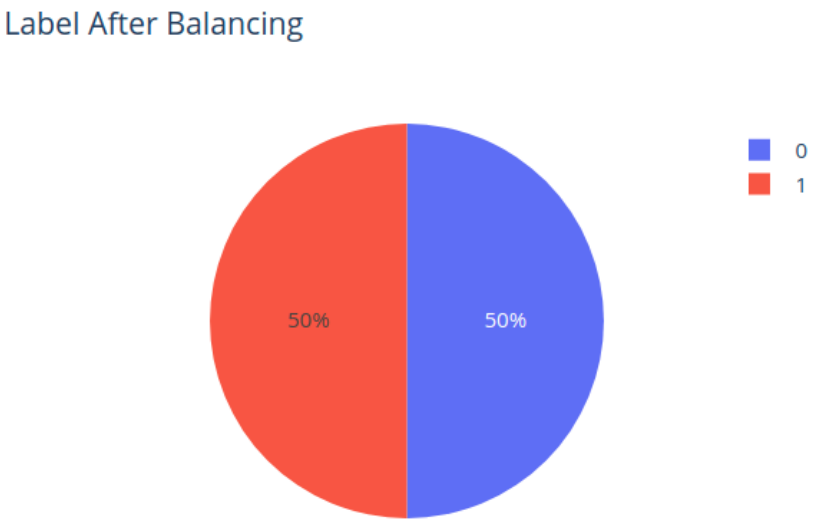

Figure 10: Balanced URL dataset for Model training

After the process of feature engineering, we have split the dataset into the training set and test set. For this, we have used the train_test_split module from the sci-kit library. We have split the dataset in the ratio of 80:20. The training set was considered to be 80% and the test set was considered to be 20%. Our ultimate purpose is to identify an efficient machine learning approach through our study. We have used seven machine learning approaches for our model out of which some are traditional approaches such as Logistic Regression, Naïve Bayes, and Decision Tree Classifier and some ensemble learning approaches such as Random Forest Classifier, Ada Boost Classifier, LightGBM Classifier, and XGBoost Classifier. Once we have implemented these seven algorithms, then we have to evaluate the models and conclude with the most preferable learning model.

3.6 Evaluation of the Model

As we have implemented the seven machine learning algorithms which are Logistic Regression, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, Ada Boost Classifier, LightGBM Classifier, and XGBoost Classifier. For each model, we have evaluated the accuracy matrix stating it's True Positive, True Negative, False Positive and

False Negative. These are the accuracy metrics that help in understanding the accuracy of the model. Other important metrics are also used such as Precision, Recall and F-1 score. Here, precision is the ratio of the true positive with the predicted positive whereas recall is the ratio of true positive with the accumulative positives. On the other hand, the F-1 score is the ratio of precision and recall. In the other sections below, we have discussed the model specification of our algorithms.

4 Model Approach

To detect the malicious URL and hinder cybercrime, an efficient algorithm must be taken into the account. With most of the approaches, taken into the account it is found that Machine Learning is the most suitable approach. Also, for the more precise output ensemble learning approach is considered. Therefore, in our paper, we compared both the traditional approach and ensemble to conclude the best result. Below, we will discuss a brief overview of each method we will be implementing.

4.1 Logistic Regression

Logistic regression is among the most common techniques of machine learning, covered under the supervised learning method. The categorized regression analysis is predicted using a certain set of individual variables. The output of a variable depending on a category is predicted. The result must thus be unequivocal or unambiguous. It may be yes or no, 0 or 1, true or false, etc, but it provides probabilistic amounts between 0 and 1, rather than giving the precise value of 0 and 1, etc. In an advanced term, the implementation of sigmoid to the simple linear regression (SLR) results in Logistic Regression.

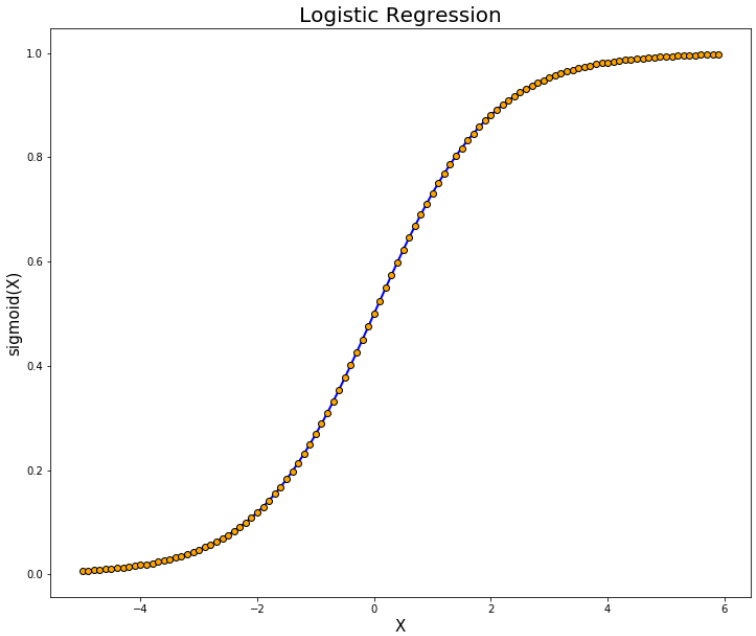

Figure 11: Logistic Regression

4.2 Naïve Bayes

Naïve Bayes is a supervised learning method based on the theorem of Bayes used to solve issues in classification. It is typically used with elevated training data set for text categorization. Bayesian Classifier is among the most efficient and basic classification algorithms that help to create rapid prediction models for machines. It is a probabilistic classifier that implies that the probabilities of an item are predicted.

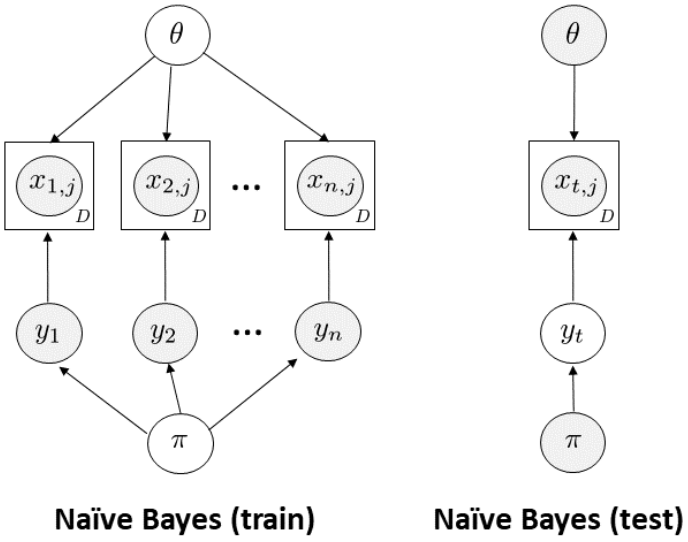

Figure 12: Naive Bayes

4.3 Decision Tree Classifier

Decision Tree is a supervised learning approach; however, it is generally preferable for resolving classification challenges for both regression and classification applications. It is a tree classification where core nodes reflect the characteristics of a data set, branches represent the rules of choice and every leaf node is the result. There seem to be two nodes in a decision tree, namely the Decision node and the Leaf node. Decision nodes are used for taking any decision and have numerous branches, whereas leaf nodes are the result and have no branching.

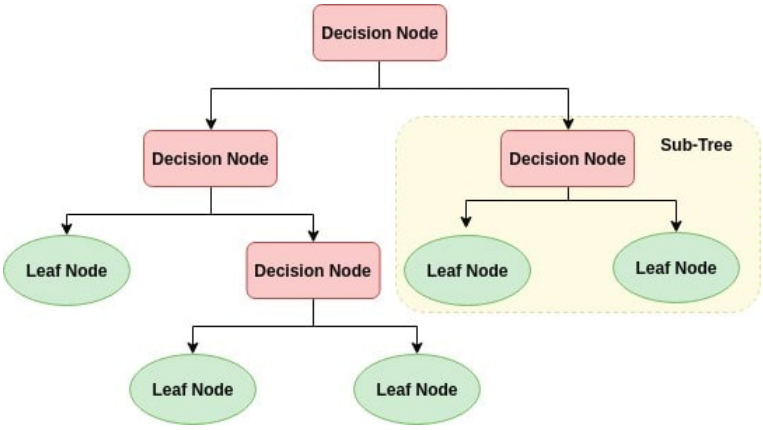

Figure 13: Decision Tree Classifier

4.4 Random Forest Classifier

Random Forest is an important method for ML algorithms that are included in the supervised learning process. It can be implemented both in classification and regression. It is based on the notion of ensemble learning that combines several classifiers to resolve a complicated problem and increase the model's performance. Random Forest is a classification that comprises a series of decision tree models on different subsets of the data set and takes an average to enhance the prediction accuracy of the data set. The random forest pulls from each tree a forecast based on majority vote projections instead of relies on a single decision tree and forecasts the ultimate result.

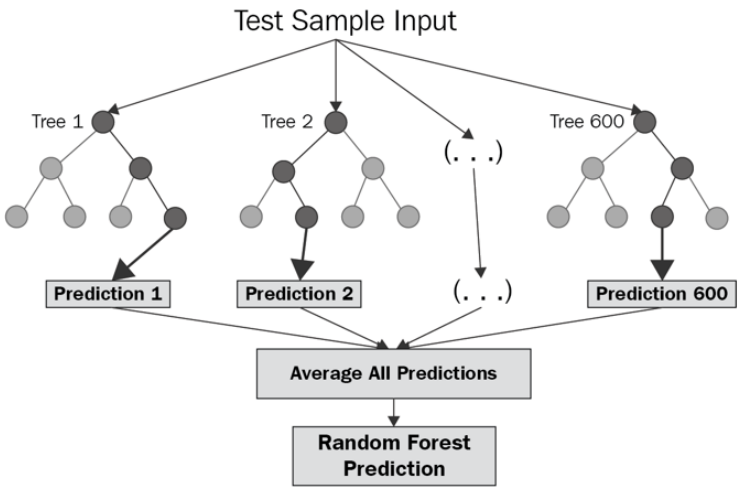

Figure 14: Random Forest Classifier

4.5 Adaptive Boost Classifier

Adaptive Boosting which is also known as the AdaBoost method is an Ensemble Method for Machine Learning Boosting approach. Adaptive boosting is termed as the masses are reassigned to each case, and larger weights are applied to erroneously categorized cases. Boosting is used to minimize bias and variation in supervised education. It operates on

the student's premise which is sequentially increasing. Every successive learner is produced by previously grown students except for the first. Weak learners are transformed into strong ones with simple phrases. The AdaBoost algorithm operates with a little variation based upon the same idea.

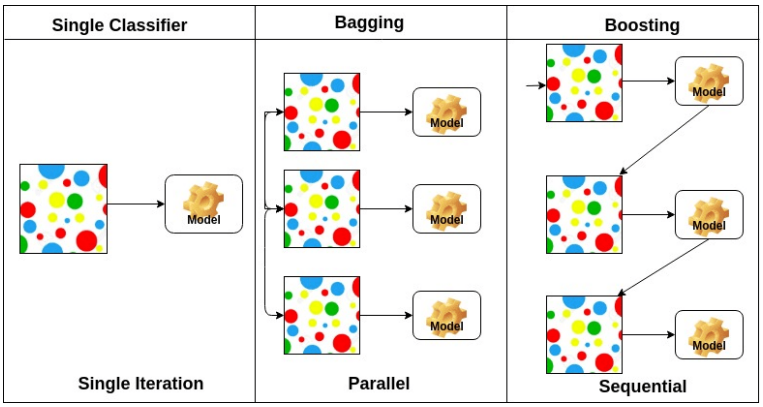

Figure 15: Adaptive Boost Classifier

4.6 LightGBM Classifier

LightGBM Classifier is an ML method used for grading, categorization and many other typological problems, which is a rapid, spread, elevated gradients boosting architecture based on decision tree techniques. The dataset size is rising exponentially. For conventional algorithms, accurate findings are becoming increasingly challenging. Because of its fast speed, Light GBM is classified as Light. Light GBM is capable of handling the huge data size and needs lesser storage. It focuses on outcomes accuracy and enables GPU learning. LGBM on tiny data sets is likewise not advisable. Light GBM is overfitting sensitive and may overfit smaller data easily.

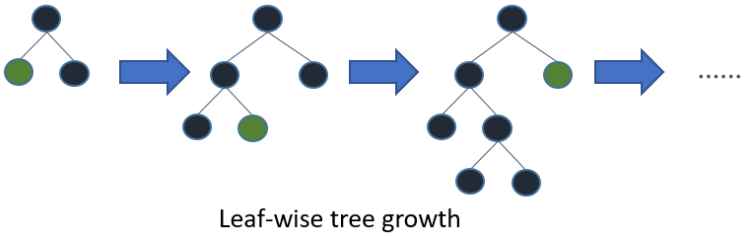

Figure 16: LightGBM Classifier

4.7 XGBoost Classifier

XGBoost is an ensemble learning method based on the decision tree that employs the framework for gradient boosting. ANN algorithms tend to exceed traditional algorithms or systems in forecasting issues involving unorganized data (pictures, language, etc.). When it comes to organized small to medium data, however, decision tree-based techniques are now regarded as the quickest.

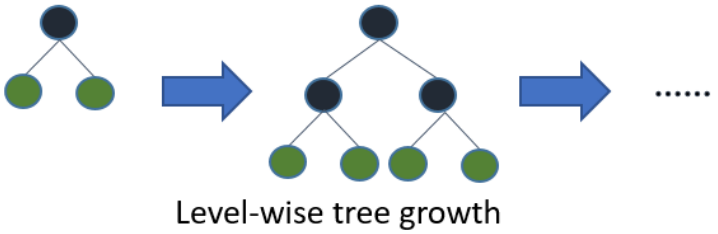

Figure 17: XGBoost Classifier

5 Implementations

In this paper, our preliminary goal is to explore the machine learning and ensemble learning based algorithm for classification of URL with better accuracy. With new URLs, each day on the internet, the acquisition of data with different patterns is also getting overwhelmed. Therefore, it is necessary to refine the data acquired efficiently before implementing it in the ML model. For this purpose, we have used various libraries such as sklearn, imbalanced-learn, matplotlib, pandas, seaborn, requests, re, urllib, data editor, warnings and NumPy. We have used Synthetic Minority Oversampling Technique (SMOTE) from the imbalanced-learn library to balance the minor samples. We have dropped irrelevant features and only considered the features very much relevant concerning the co-relation matrix. The matplotlib is used for data visualization such as in bar graphs, pie charts, violin graphs and more. We also used the seaborn library to integrate the pandas' library with the matplotlib library for making statistical graphics. Also, we used the request library to exchange the HTTP web data of the URL. The purpose of re library in our model is for word processing. We have also used the warning library to alert some parts of the program. Here, the urllib module for handling the URL. Once, we implement all the algorithms we will do a through comparison between machine learning approach and ensemble learning approach. As our models were implemented in a single machine, below is the specification for our proposed framework.

- RAM: 8GB
- Hard Disk: 100GB
- Operating System: Ubuntu 20.04
- Programming Language: Python3
- Libraries: Pandas, NumPy, Matplotlib, Sklearn, Imbalanced-learn, Seaborn, Urllib, Data Editor, Warnings, Lightgbm, Xgboost

6 Evaluation

In our research paper, our foremost aim is to find out an efficient algorithm to identify the malicious URL. We utilized different approaches to implement in our models such as Logistic Regression, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier,

AdaBoost Classifier, LightGBM Classifier and XGBoost Classifier. To find the best-resulting model following its efficiency and performance, we used various metrics such as accuracy, precision, recall, f-1 score, and training time. With these evaluation metrics, we can consider an efficient algorithm to detect the malicious URL. Here, we will discuss the performance of each model based on its metrics.

6.1 Experiment 1 / Evaluation Based on Accuracy

As the number of correct predictions increases in comparison with the total prediction of the output from the trained model, the accuracy of the model also increases herewith. It is a prominent metric to sort from the efficient models. In our research, we implemented seven algorithms combined from both the machine learning approach and ensemble learning approach. As shown in the figure 18, logistic regression and naïve Bayes shows the least accuracy out of all other models. Logistic Regression shows an accuracy of 66.91 whereas Naïve Bayes shows an accuracy of 49.79 which is the lowest of all. It is also found that the Decision Tree Classifier and Adaptive Boost Classifier shows the same accuracy of 99.65. When compared Random Forest classifier and LightGBM Classifier, there differ by the accuracy level of 0.01. Random forest classifier and LightGBM classifier show an accuracy level of 99.79 and 99.8, respectively. Finally, XGBoost Classifier shows the highest accuracy level out of all the models. It shows an accuracy level of 99.81.

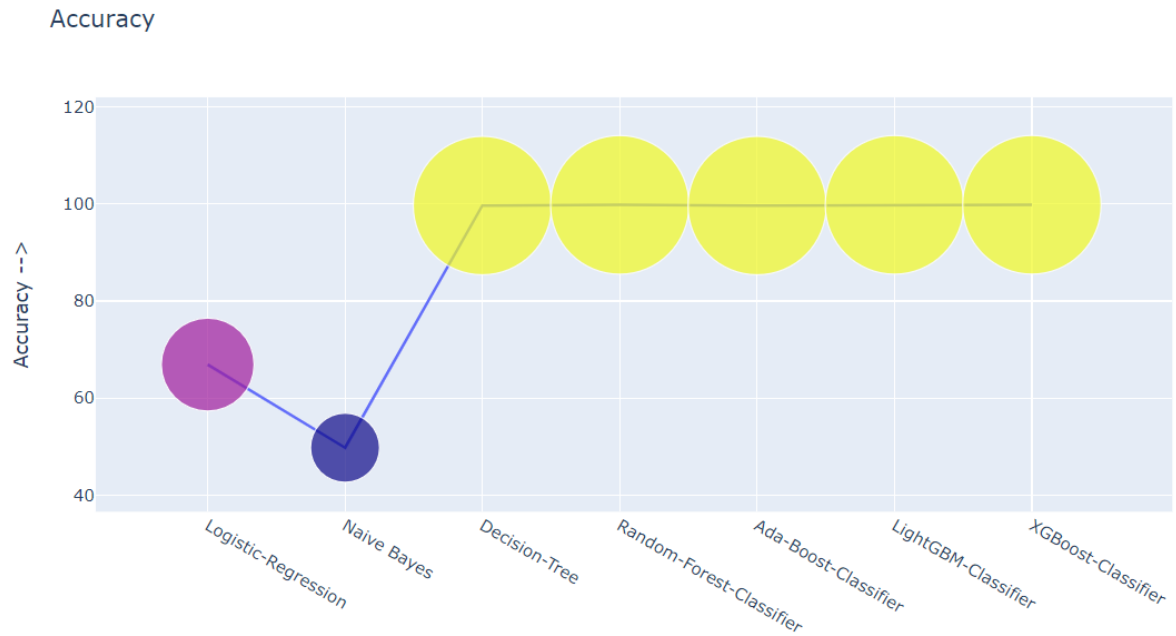

Figure 18: Evaluation of Accuracy for all algorithms

6.2 Experiment 2 / Evaluation Based on PRF Score

To assess the efficiency of the model, the PRF score is calculated in order gain the indepth analysis about the model performance. In evaluation metrics, we assess the model through the metrics such as precision, recall and f-1 score. Precision is the ratio of the

true positive with the predicted positive whereas Recall is the ratio of true positive with the accumulative positives.

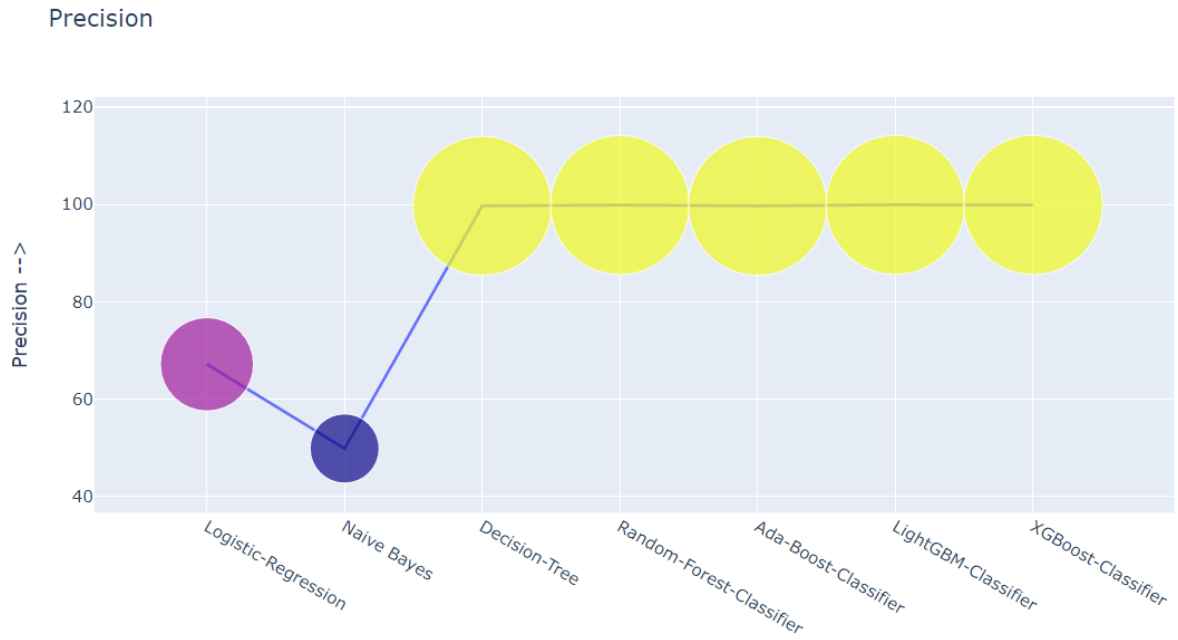

Figure 19: Evaluation of Precision Score for all algorithms

Here, the F-1 score is the ratio of Precision and Recall. As per these graphs, we found that all the three metrics of Logistic Regression is found to be the lowest of all. In logistic regression, the precision (figure 19), recall (figure 20), and the f-1 score (figure 21) are found to be 67.14, 65.82 and 66.47, respectively. The precision and f-1 score for naïve Bayes is less whereas the recall score for naive bayes algorithm is higher. The obtained PRF score of naive bayes algorithm are 49.81, 66.47 and 99.89, respectively.

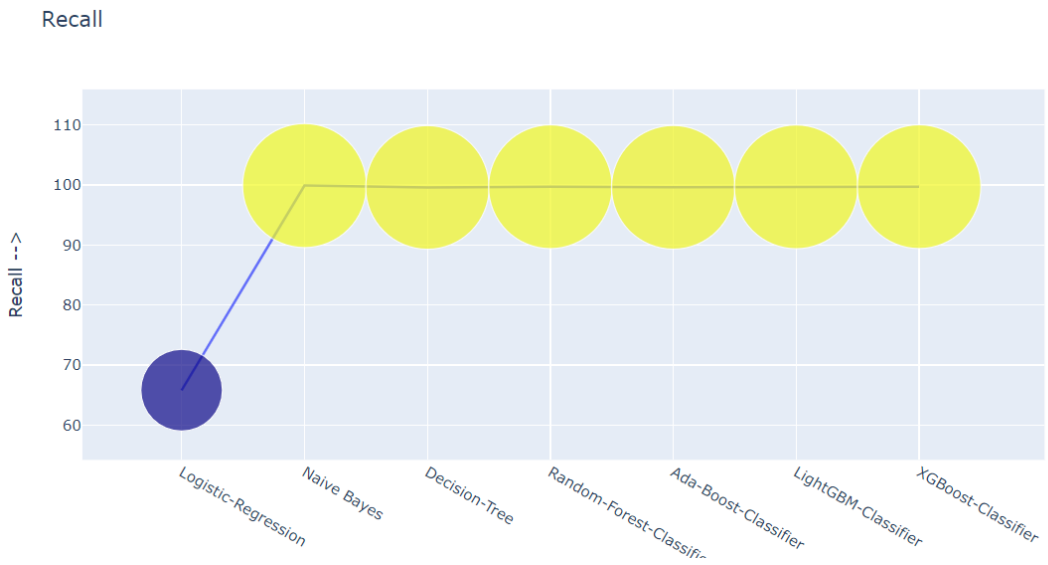

Figure 20: Evaluation of Recall Score for all algorithms

For the Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, LightGBM Classifier and XGBoost Classifier, the precision, recall and f-1 score is mostly similar with some minor difference of decimal values. However, the F1-Score obtained from logistic regression and naive bayes is found to be comparatively low.

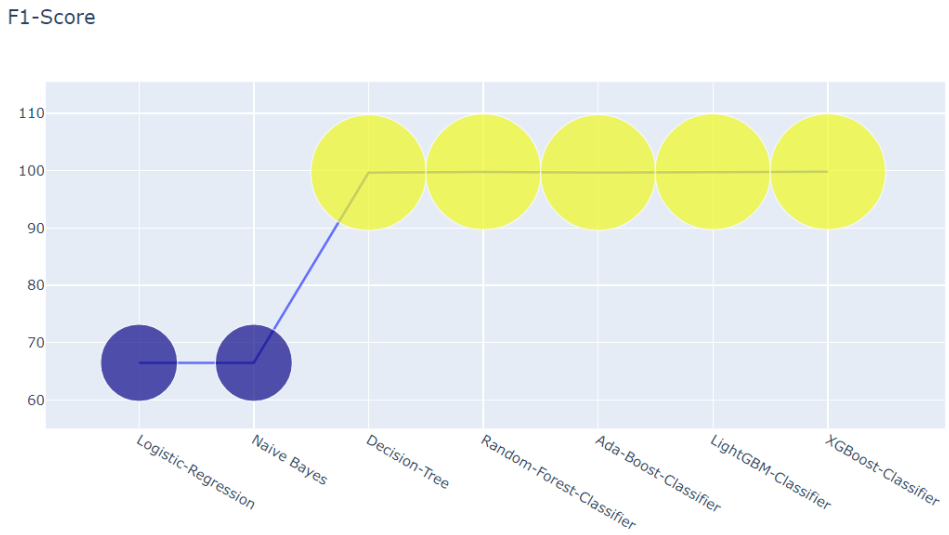

Figure 21: F-1 Score

6.3 Experiment 3 / Evaluation Based on the Training Time

The model training time is the third most prominent metric to assess the efficiency and performance of the model. The time factor adversely impacts the performance of the model. Therefore, in our study, we assessed the training time of each model. As shown in

the figure 22, Naïve Bayes requires the minimum training time for model training whereas Random Forest Classifier takes the highest time for training the models. As per our research, logistic regression requires 2.23 seconds for model training whereas Naïve Bayes takes only 0.08 seconds which is the lowest. Here, the decision tree classifier takes 1.09 seconds whereas the Random Forest classifier takes 14.81 seconds which is the highest. Also, Adaptive Classifier, LightGBM Classifier and XGBoost Classifier take 7.08 seconds, 0.72 seconds and 4.38 seconds, respectively. From overall analysis it can be concluded that with respect to machine learning based methods the naive bayes algrirhtm requires the minimum training time and for ensemble learning methods, the light-GBM training time is minimum.

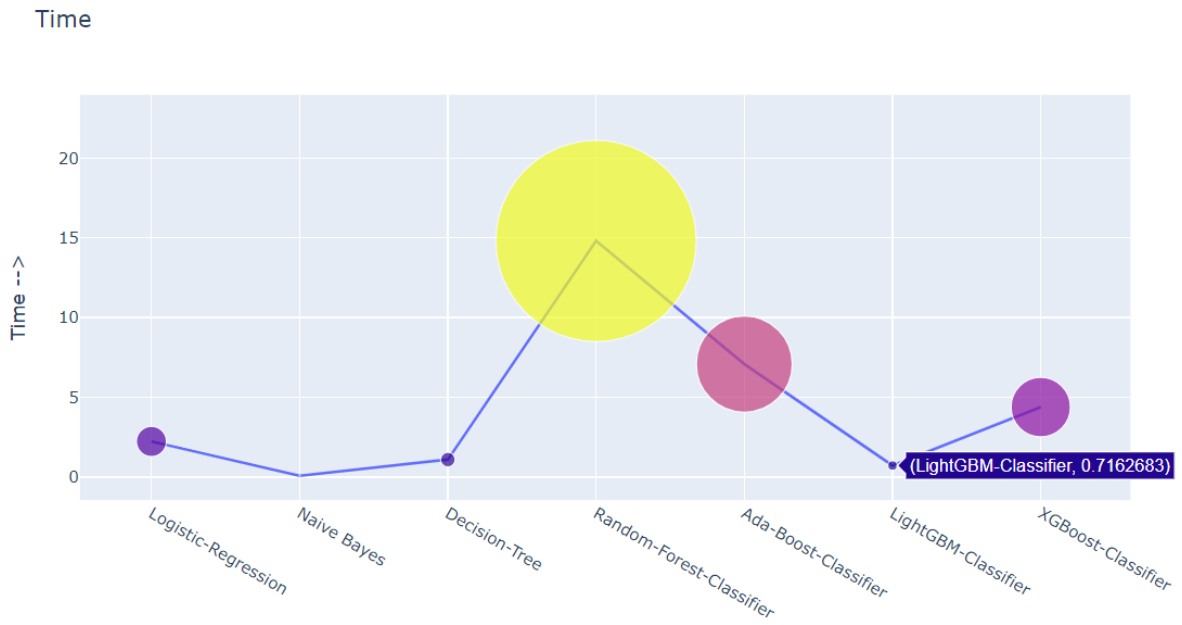

Figure 22: Training time Comparison

6.4 Discussion

In the overall assessment of the metrics for the efficiency and the performance of the model, we have found that the accuracy, precision, recall, f-1 score and computational time vary across the different algorithms as per the metric table in figure 23. We have found that Naïve Bayes having the highest Recall and training time, but falls at other metrics as it is lower than anticipated. Here, the Random Forest classifier takes more training time than anticipated. Within the observation, it is also seen that the accuracy, precision, recall, and f-1 score are much more similar for the LightGBM Classifier and XGBoost Classifier, but XGBoost Classifier takes a longer computational time than LightGBM Classifier.

	Time	Accuracy	Recall	Precision	F1-Score
Logistic-Regression	2.227809	66.913104	65.820879	67.134554	66.471227
Naive Bayes	0.079391	49.791064	99.892485	49.809682	66.473492
Decision-Tree	1.089405	99.646416	99.580690	99.709334	99.644970
Random-Forest-Classifier	14.807247	99.791064	99.698957	99.881517	99.790153
Ada-Boost-Classifier	7.082148	99.646416	99.602193	99.687937	99.645047
LightGBM-Classifier	0.716268	99.796421	99.677454	99.913784	99.795479
XGBoost-Classifier	4.384250	99.807136	99.709709	99.903049	99.806285

Figure 23: Metric Table

As our primary goal is to classify the URL as benign or malicious using an appropriate algorithm. Therefore, our metric which depends on various metrics has baseline metrics called True Positive, True Negative, False positive and False-negative. Through this metric, we will assess the accuracy, precision, recall and f-1 score. True positive (TP) states if malicious URL that is properly indicated as malicious URL whereas True negative (TN) states if benign URL that is properly indicated as benign URL. On the other hand, False positive (FP) states if benign URL that is improperly indicated as malicious URL and False negative (FN) states if malicious URL that is improperly indicated as benign URL. These metrics for the models can be judged easily through the confusion matrix. As shown in figure 24 and figure 25, shows the confusion matrix for LightGBM Classifier and XGBoost Classifier, respectively.

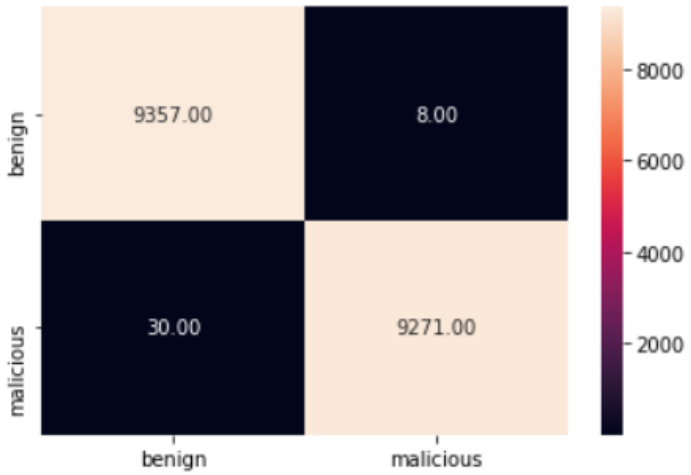

Figure 24: LightGBM Classifier Confusion Matrix

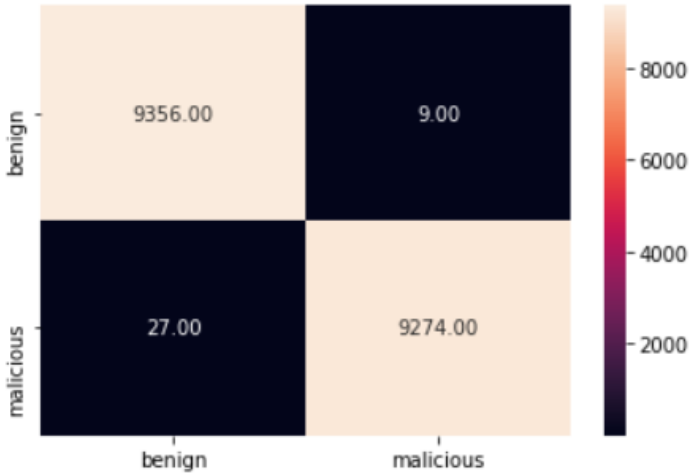

Figure 25: XGBoost Classifier Confusion Matrix

The lightGBM and XGBoost are the best performing model, where both algorithms are ensemble learning based. Accuracy and PRF score of LightGBM and XGBoost are very similar. Although there is major difference in the training time has been observed. In terms of training time LightGBM outperforms than XGBoost algorithm and in terms of model performance XGBoost overtake the LightGBM algorithm.

7 Conclusion

Increasing internet usage has directed an exponential surge in the cyber threat. Nowadays as people prefer the online means for most of their tasks such as emails, banking, communication, etc then their pieces of information such as passwords, name, address become easily available into the database. These critical pieces of information are the target by the attackers. With the massive generation of new and unknown URLs, it has become a complicated task to identify the malicious part. Every new day, attackers come with a novel pattern of malicious URLs. The most common web pages are targeted by attackers and infiltrate to collect critical pieces of information. Therefore, it is necessary to have an efficient practice to hinder the threat and identify the upcoming threat as well. A common approach of blacklisting was used, but it had multiple disadvantages. With the new pattern of malicious URLs now and then, blacklisting was not that efficient. Therefore, a machine learning approach was considered to solve this issue. But there are various other factors to consider such as the model's accuracy and performance. For this purpose, we have implemented and compared two approaches namely, machine learning and ensemble learning. In this, we considered seven models called Logistic Regression, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, Adaptive Boost Classifier, LightGBM Classifier and XGBoost Classifier. Here, we implemented these models and compared their accuracy and performance based on the metrics. We have considered metrics such as accuracy, precision, recall, f-1 score and time. With an overall comparison, we found that the ensemble learning approach is the best performing than the machine learning approach. The accuracy level was much more similar in most of them. But in our research, we found that LightGBM Classifier and XGBoost Classifier have the best performance and were at par. XGBoost Classifier took a longer computational time but had a bit

higher performance than LightGBM Classifier. But on the other hand, LightGBM Classifier took very minimal computational time. In our future work, we can explore the big data technologies for handling the larger set of URL data can convert the URL data into streaming data using kafka and spark streaming. As the new data enters the digital world each day and with the upcoming technological advances, we look to propose an advanced novel approach than the existing one in the field of streaming technologies.

References

Femi, A. (2013). Perception of performance appraisal and workers' performance in wema bank headquarters, lagos, *Global Journal of Arts, Humanities and Social Sciences* **1**(4): 89–101.

Gawale, N. S. and Patil, N. N. (2015). Implementation of a system to detect malicious urls for twitter users, *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, pp. 1–5.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C. and Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 19–35.

Janet, B., Kumar, R. J. A. et al. (2021). Malicious url detection: A comparative study, *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, IEEE, pp. 1147–1151.

Kiruthiga, R. and Akila, D. (2019). Phishing websites detection using machine learning, *Int. J. Recent Technol. Eng.(IJRTE)* **8**.

Kumar, R., Zhang, X., Tariq, H. A. and Khan, R. U. (2017). Malicious url detection using multi-layer filtering model, *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, IEEE, pp. 97–100.

Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F. and Hong, J. (2008). Lessons from a real world evaluation of anti-phishing training, *2008 eCrime Researchers Summit*, IEEE, pp. 1–12.

Lin, M.-S., Chiu, C.-Y., Lee, Y.-J. and Pao, H.-K. (2013). Malicious url filtering—a big data application, *2013 IEEE international conference on big data*, IEEE, pp. 589–596.

Malicious And Benign URLs (n.d.).
URL: *https://kaggle.com/siddharthkumar25/malicious-and-benign-urls*

Manyumwa, T., Chapita, P. F., Wu, H. and Ji, S. (2020). Towards fighting cybercrime: Malicious url attack type detection using multiclass classification, *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 1813–1822.

Nagaonkar, A. R. and Kulkarni, U. L. (2016). Finding the malicious urls using search engines, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp. 3692–3694.

Tan, G., Zhang, P., Liu, Q., Liu, X., Zhu, C. and Dou, F. (2018). Adaptive malicious url detection: Learning in the presence of concept drifts, *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, pp. 737–743.

Xu, L., Zhan, Z., Xu, S. and Ye, K. (2013). Cross-layer detection of malicious websites, *Proceedings of the third ACM conference on Data and application security and privacy*, pp. 141–152.

Yang, W., Zuo, W. and Cui, B. (2019). Detecting malicious urls via a keyword-based convolutional gated-recurrent-unit neural network, *IEEE Access* **7**: 29891–29900.

Zhang, T., Zhang, H. and Gao, F. (2013). A malicious advertising detection scheme based on the depth of url strategy, *2013 Sixth International Symposium on Computational Intelligence and Design*, Vol. 2, IEEE, pp. 57–60.