EasyChair Preprint

№ 10478

# Robust Machine Learning Technique for Detection and Classification of Spam Mails

B Aruna Kumari and C Nagaraju



June 30, 2023

# ROBUST MACHINE LEARNING TECHNIQUE FOR DETECTION AND CLASSIFICATION OF SPAM MAILS

**B. ARUNA KUMARI[1], Dr.C. NAGARAJU[2]**
[1]Research Scholar, YSR ENGINEERING COLLEGE OF YVU, Department of CSE, Proddatur, India
[2]Professor, YSR ENGINEERING COLLEGE OF YVU, Department of CSE, Proddatur, India
E-mail: [1]arunakumarib1421@gmail.com, [2]cnrcse@yahoo.com

**ABSTRACT**

A massive number of spam mails have become difficulty for Internet users. Spammers can collect data by creating fake URLs, fake websites and fake chat rooms. Spam mails may lead to harassing, bullying or social traumatizing situations. To overcome this drawback in email architecture, it is essential to boost up the existing technology and lay a stone for new outcomes. Spam mails can also be filtered using URLs, but this will lead to error prone. To solve these problems, several models have been existed and tested but none of those models achieved high accuracy. In this research work, a new method is proposed with the support of NLP with Machine learning and achieved 98.5% of accuracy on SMS Spam Collection dataset.

**Keywords:** XGboost, Random Forest, Logistic regression, Decision tree, Stacking classifier.

## 1. INTRODUCTION

The Internet has become an integral part of our lives. More than two-thirds of the global population use it for different activities, including connecting people, communicating and sharing information. E-mail is the mostly used method of communication on the web and a great tool to exchange data. Unfortunately, this also makes users more vulnerable to spam attacks. Spam is unsolicited material, such as fake or malicious content and URLs, that has been sent to multiple recipients. The goal of spammers is usually to access personal information in order to gain financially. Distinguishing between spam and legitimate emails is not always easy due to constantly changing email content. To tackle this problem, anti-spam tools such as corporate email systems, mail filtering gateways or contracted antispam services have been created - although with limited effectiveness. Various models of ML like XGBoost, Random Forest, KNeighbors Classifier and Light GBM can also help classify emails efficiently.

XGBoost - "Extreme Gradient Boosting", an efficient distributed gradient boosting library. It is mainly used for efficient training of machine learning models. It has proven to be very successful in handling large datasets and its state-of-the-art performance makes it a popular choice for maching learning algorithms [1]. XGBoost effectively handles missing values, which allows for more accurate predictions in real-world situations. Additionally, XGBoost supports parallel processing, so it can train models on large datasets in a short amount of time. XGBoost algorithm is widely used in various applications such as Kaggle competitions, recommendation systems and click-through rate prediction among others.

Random Forest (RF) is the supervised ensemble machine learning algorithm, utilized for both regression and classification problems. As represented in figure 1, it involves a number of decision trees (DT) on diverse subgroups of the dataset and considers the average to increase the dataset's predictive accuracy. Rather than based on single tree, this random forest (RF) algorithm considers each tree's prediction and look at majority of predictions' votes and finalize the last result. The most trees in the forest may result in high accuracy and prevent overfitting [2]. Random Forest algorithm predicts output with high accuracy for the large datasets and also involves less training time. Although a large proportion of data was missed, this algorithm maintains accuracy. Random Forest

algorithm may use in several applications such as Banking, Medical applications, land use and also in marketing [14].

LightGBM [3]- "Light Gradient Boosting Machine". It depends on decision trees to enhance the efficiency of the model and decreases memory consumption. LightGBM is also an ensemble maching learning algorithm utilized for regression or classification predictive modeling issues. By employing a gradient descent optimization algorithm and an arbitrary differentiable loss function models can fit, when the model is fit, the loss gradient is minimized. Two novel strategies have been utilized by LightGBM such as Exclusive feature bundling and one-side gradient sampling. To make the model work well, these two models work together. Decision trees can be created by lightGBM that grow leaf wise i.e., based on the gain only a single leaf is split. Sometimes the leaf wise trees can overfit with smaller datasets.

Gradient Boosting is a powerful boosting algorithm. The Gradient Boosting algorithm assembles multiple weak learners into strong learners. It utilizes a large number of base learners like decision trees and linear models. Based on the gradients it updates the weights so the gradient boosting algorithm is morestrong. Based on the previous model using gradient descent, every single new model is prepared to limit the loss function i.e., mean squared error or cross-entropy. The new model predictions are then added to the ensemble, and the procedure is sustained unless a preventing criterion met [4].

Logistic regression is a supervised machine learning method. The classification problems are solved using logistic regression [5].

KNeighborsClassifier [6] will depends on the k nearest neighbors of a sample that has to be classified, where 'k' is an integer that should specified by the user.

Decision tree is a supervised machine learning algorithm. It is utilized for solving the problems of classification. A decision tree (DT) builds a tree structure looking like a flowchart. Decision rules are served by the links, while dataset characteristics are served by internal nodes in the tree. Two nodes are there in any decision tree such as Decision node and leaf node. Decisions are made using the characteristics of the dataset in the decision algorithm [7]. For predicting the given dataset, the algorithm begins from the root node. The values of the root attribute and the actual dataset attribute will be compared by the algorithm, and based on the comparison, the algorithm moves on to the next node. The value of the attribute is once more compared to that of other subnodes by the next node, and the process continues until it reaches the tree's leaf node [13].

## 2. LITERATURE SURVEY

In [8], Various approaches and open research problems in email spam filtering are discussed. The objective of this research is to identify and filter spam mails and applied several methods such as clustering technique, neural networks, naïve bayes classifier, firefly algorithm, support vector machine classifiers, rough set classifier, decision trees, random forest and ensemble classifiers. However, the authors focused on feature-free methods. With feature-free techniques, email classification causes high computational costs. Most of the spam filters classify only text spam messages, so there is a need of developing more efficient image spam filters.

In [9], the research employs neural networks, decision trees, random forest and naïve bayes methods to identify and filter spam mail. There are few limitations in this research. For example, labeled data in spam detection is the significant issue and also the false positive rate is greater than required, it should be diminished. The majority of spam filters available today are unable to update their feature sets.

In [10], the objective of the research is to find and remove spam mails. The Bayesian classifier was used in this case. However, in order to increase accuracy, the content-based spam detection system and fake URL detection must be combined.

In [11], a survey on Spam email detection was performed to detect spam mails. The use of naïve bayes, support vector machines, decision trees and random forest was made.

However, the authors only considered recall, accuracy, and precision, while the model's time complexity should also be taken into account. Although the header, email's subject line and body of the message are all considered features for classifying spam mails, they are not sufficient to produce accurate results. Manual features need to be considered.

In [12], a comprehensive survey on intelligent spam email detection was done. Several artificial intelligence and machine learning techniques were utilized. There is a need to develop antispam software to work against multiple attacks with a single installation.

## 3. PROPOSED METHOD

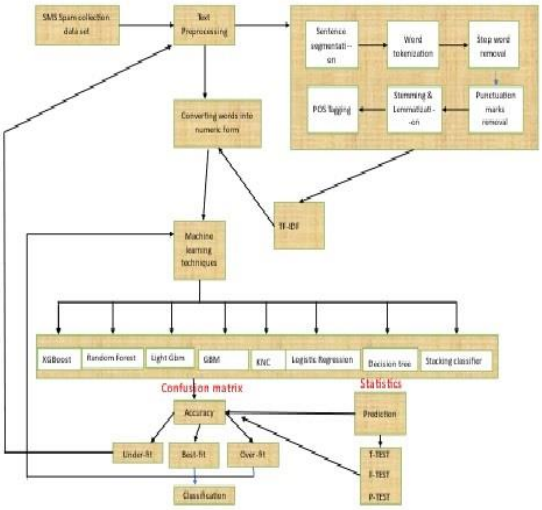
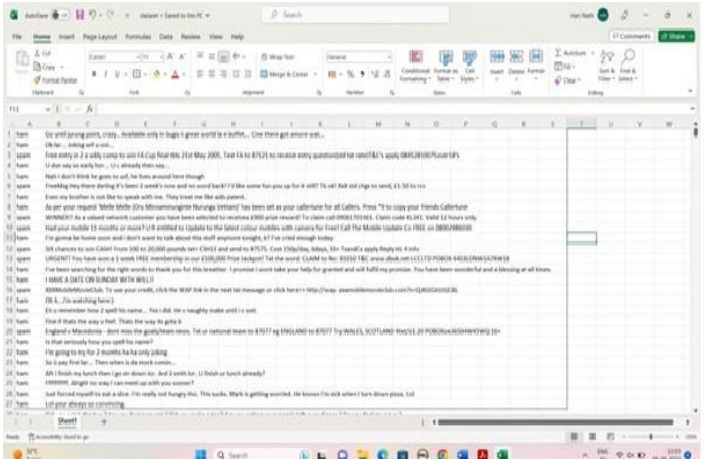



*Figure 1: Block diagram of proposed method*

### 3.1. Dataset



For this research, the SMS Spam Collection dataset was gathered from online source Kaggle, which will be utilized to train the models of machine learning. There are a total of 5572 records. The first attribute in the dataset is titled as "type of mail", and it was used to distinguish emails between spam and ham. The email's text, which included a variety of email content, is the second attribute.

### 3.2. Text Preprocessing

In Natural language processing (NLP), text processing is the process of cleaning and transforming unstructured text data, so that it can be analyzed. We use text preprocessing to get the text ready for the model building. Sentence segmentation, word tokenization, stemming, lemmatization, stop-word elimination and part-of-speech tagging are all included.

During the sentence segmentation phase, a paragraph is taken as an input and divide that paragraph into meaningful sentences corresponding to it. Each and every sentence of the paragraph will be included, it is shown in below figure 4.

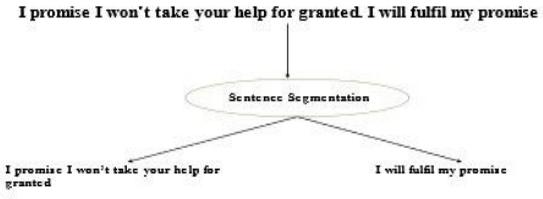



*Figure 2: Sentence segmentaion*

Word Tokenization includes dividing a sentence into a group of words, known as tokens. Tokens are the fundamental building blocks upon which analysis and various methods are built. Consider an example sentence ***Fulfil my promise***, it is tokenized as 'Fulfil', 'my', 'promise' as shown in figure 2.

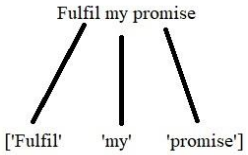



*Figure 3: Word Tokenization*

There are a lot of words that are used a lot in English but don't really add much meaning to sentences, like "is", "and", "the" are. Natural language processing pipelines will mark these words as stop words. In the above sentence ***Fulfil my promise***, "my" is the stop word and have no meaning. The stop words might be removed before any statistical analysis is carried out.

Removal of punctuation marks is the next phase in the text processing. In general, there are 14 punctuation marks in English such as question mark, comma, colon, semicolon, parenthesis, braces, brackets, hyphen, dash.

Stemming phase involves the operation of converting all words into their root form which is known as stem. In most cases, the word and its stem are determined using a lookup table. This stemming procedure is used for document retrieval based on user queries by the majority of search engines. Stemming also used at the preprocessing stage for applications. An example of stemming is shown in figure 4.

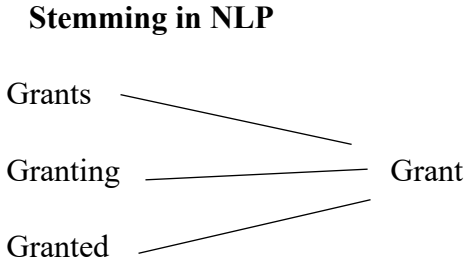



*Figure 4: Stemming*

The advanced form of stemming known as "lemmatization" involves changing each word to its corresponding root form which is known as "lemma". Stemming does not make use of words in their context or parts of speech. Additionally, a lemmatizer standardizes various word forms. Compared to a stemmer, lemmatizer works with multiple rules and contextual information. Lemmatizer searches a dictionary for matching words. So that this lemmatization process takes more processing time than stemmer to produce output.

The purpose of dependency parsing is to specify the relationship between each word of the sentence. Based on the assumption that each sentence has a relation to the others, a sentence is divided into many sections. Such hyperlinks are known as dependencies.

parts of speech (POS), which includes Noun, verb, adverb, and adjective are recognized. It shows how a word works together with its meaning and grammatically in the sentences. Depending on the context in which it is used, a word can have one or more parts of speech.

### 3.3. TF – IDF for creating features from text

TF- IDF is a metric of scoring used in summarization and acquiring information. The TF-IDF measure indicates how relevant a term in a given document. When a word appears more than once in a document, it is given more weight than words that appear less frequently. It is referred as Term Frequency (TF).

It is referred to as "Inverse Document Frequency", When a selective word appears more than one time in a document but is also available more than one time in other documents. This indicates that the word may be repeated but we should not give a lot of importance to it.

TF-IDF gives higher values for less frequent words, and smaller values for high frequent words. If both TF and IDF values are high then that word is available rarely in all the documents but frequent in a single document.

Calculating Term Frequency (TF) using a formula,

$$TF = \frac{(Frequency\ of\ the\ word\ in\ the\ sentence)}{(Total\ no,of\ words\ in\ the\ sentence)} \qquad (1)$$

Calculating IDF values from the formula,

$$\text{IDF} = \frac{Total\ number\ of\ sentences}{Number\ of\ sentences\ containing\ that\ word} \quad (2)$$

### 3.4. Stacking Classifier

Stacking classifier is used to predict multiple nodes for a new model and boost model performance. We can train multiple models to solve lookalike problems with stacking, which then generates a new model with improved performance based on their combined results. To achieve better output prediction model, the input of several weak learner's predictions and also meta learners' combination can be applied. Stacking is also called as stacked generalization, in which a new model will created by producing the equivalent participation of all sub-models based on their performance weights.

Stacking is one kind of ensemble learning technique in which the predictions of multiple classifiers are used to train a meta classifier. The following figure shows how three different classifiers get trained. In the below figure 8, C1, C2, C3 are the level 1 classifiers and P1, P2 and P3 are the level 1 predictions.

A rule like "level one predictions should come from a subset of the training data that was not used to train the level one classifier" is used by the target stacking classifier.

A basic method for accomplishing this is to divide the dataset in half. Utilize the primary half of information to teach the level one classifier. Later utilize this prepared level one classifier to produce expectations on the final part of the preparation information. Now the meta-classifier is trained by using these predictions.


*Figure 5: K-fold cross validation used with a stacking classifier framework*

The level one prediction can be made by using K-fold cross validation. The training data can be partitioned into 'K' number of folds in this way. The first K-1 folds are utilized to train level one classifiers. In order to produce a subset of level one predictions, validation fold is used. Each distinct group will go through this procedure.

### 3.5. Accuracy Parameters

To evaluate the proposed method's effectiveness, SMS Spam Collection dataset is taken. Applying confusion matrix, with "yes" or "no" predicted clauses based on TN(True Negative), TP(True Positive), FP(False Positive), and FN(False Negative).

TP means actually true and predicted as true. TN means actually true but predicted as false. FP means actually false but predicted as true. FN means actually false and predicted as false.

Based on these probabilities, accuracy is computed with the following formula.

**Accuracy**

$$\text{Accuracy} = \frac{truepositive+truenegative}{truepositive+falsepositive+truenegative+falsenegative} \quad (3)$$

**T-TEST**

$$\text{T-TEST} = \frac{difference\ between\ sample\ mean}{estimated\ sample\ error\ of\ differences\ between\ mean} \quad (4)$$

**F-TEST**

$$\text{F-TEST} = \frac{variance\ between\ sample\ mean}{variance\ expected\ by\ error} \quad (5)$$

**P-TEST**

It is a statistical measure that helps to determine whether the null hypothesis is accepted or rejected by the models.

## 4. EXPERIMENTAL RESULTS

The proposed stacking classifier is implemented and tested on SMS Spam Collection dataset which is having two fields of label and text field. This dataset contains 5572 records. The confusion matrix, P – test, F – test, and t – test is applied on proposed method. The results of this method is compared with XGBoost, Light GBM, Gradient Boosting Machine, Decision tree, KNeighbors Classifier, Random Forest, Logistic regression and the results are represented in the form of tables and graphs.

Table 1 represents the test condition values of P – test, F – test, and t – test for various samples with stacking classifier. At sample size 35% of the values are minimum. So that it represents at 35% of the sample size the accuracy rate is high comparatively with remaining sample sizes.

Table 2 represents accuracy rates of existing methods and proposed method at different sample sizes. The proposed method produces better accuracy than all other existed methods on most of the sample sizes taken. These accuracies are represented graphically with bar chart. This bar chart clearly represents the proposed method with high accuracy rate comparatively with existing methods in most of the samples.

*Table 1: Test_size Vs Test_values*

| Test_size | t-test | P-test | F-test |
|---|---|---|---|
| 0.05 | 0.36 | 0.71 | 0.78 |
| 0.1 | 0.13 | 0.89 | 0.72 |
| 0.15 | 0.08 | 0.93 | 0.78 |
| 0.2 | 0.26 | 0.79 | 0.78 |
| 0.25 | 0.03 | 0.97 | 0.79 |
| 0.3 | 0.21 | 0.82 | 0.78 |
| 0.35 | 0.59 | 0.55 | 0.78 |
| 0.4 | 0.91 | 0.35 | 0.77 |
| 0.45 | 1.17 | 0.24 | 0.78 |
| 0.5 | 0.99 | 0.32 | 0.78 |

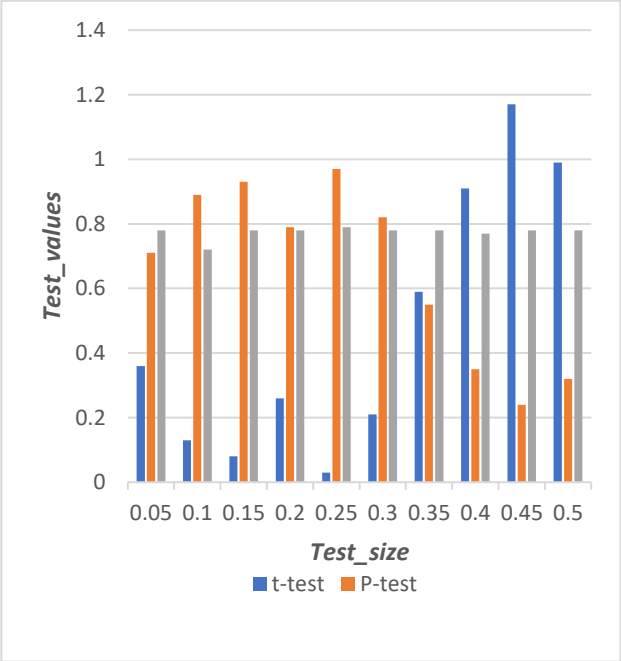



*Figure 6: Shows the relationship between test_size and test_values*

*Table 2: Test_size Vs Percentage of accuracy*

| Test size | Xgboost | Random forest | Light GBM | GBM | Kneighbors Classifi-er | Logistic Regression | Decision tree | Stacking Classifier |
|---|---|---|---|---|---|---|---|---|
| 0.05 | 97.1 | 97.1 | 97.1 | 96.4 | 97.4 | 97.4 | 96.7 | 96.7 |
| 0.1 | 98.2 | 98 | 98 | 96.7 | 96.9 | 97.8 | 96.9 | 98.2 |
| 0.15 | 98.4 | 98 | 98.4 | 97.2 | 97.1 | 98.3 | 96 | 98.5 |
| 0.2 | 98.2 | 98 | 98.1 | 97.3 | 97.1 | 98.3 | 96.5 | 98.2 |
| 0.25 | 98.1 | 98.1 | 98 | 97.4 | 96.9 | 98.2 | 96 | 98.3 |
| 0.3 | 98 | 97.9 | 98.1 | 97.4 | 96.6 | 97.8 | 96.4 | 98.3 |
| 0.35 | 98.1 | 98 | 98.1 | 97.4 | 96.6 | 98.1 | 95.6 | 98.5 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0.4 | 97.8 | 98.2 | 98.3 | 97.4 | 96.5 | 98 | 96.5 | 98.4 |
| 0.45 | 97.7 | 98 | 98.1 | 97.4 | 96.4 | 97.8 | 95.6 | 98.4 |
| 0.5 | 97.8 | 98 | 98 | 97.5 | 95.7 | 97.4 | 95.3 | 98.3 |

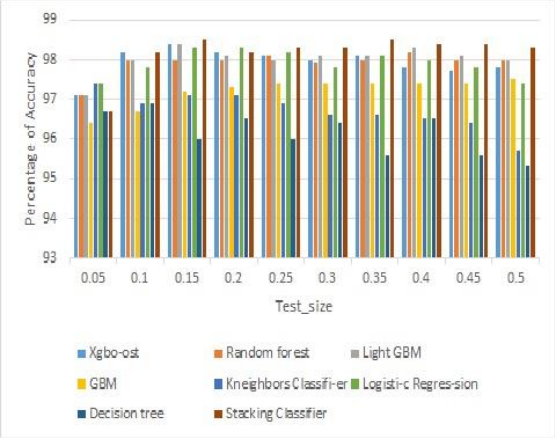



*Figure 7: Illustrates the Relationship Between Test_size and percentage of accuracy.*

## 5. CONCLUSIONS

Spam email detection plays a vital role in the development of secured mail systems. The text pre-processing phase involves improving the quality of text and that text is converted to numeric form using TF-IDF. The proposed stacking classifier which ensemble different kinds of classifiers and predictors and applied on SMS Spam Collection dataset. It achieved 98.5% accuracy. However, it fails to achieve 100% of accuracy rate because stacking classifier assumes data is independent so that it increases added complexity for implementation and it is much harder to explain with the no correlated or low correlated based models to improve the efficiency of stacking models.

## 6. REFERENCES

[1] p.U. Anitha, Dr. C.V. Guru Rao, Dr. D. Suresh babu, Email spam filtering using machine learning based Xgboost classifier method, Turkish Journal of Computer and Mathematics Education, 2021, 12(11): 2182-2190,202.

[2] Kothapally Nithesh Reddy, Dr. Vijayalakshmi Kakulapati, Classification of Spam messages using Random Forest algorithm, Journal of Xidian University, 2021, 15(8): 495-505, 2021.

[3] Elena-Adriana MINASTIREANU, Gabriela MESNITA, Light GBM Machine learning algorithm to online click fraud detection, Journal of Information Assurance & Cybersecurity, 2019.

[4] Fahima Hossain, Mohammed Nasir Uddin, Rajib kumar Halder, Analysis of optimized machine learning and deep learning techniques for spam detection, IEMTRONICS- 2021, 552-558.

[5] Manoj Sethi, Sumesha Chandra, Vinayak Chaudhary, Yash, Email spam detection using machine learning and neural networks, IRJET, 2021, 8(4): 349-355.

[6] Mangena Venu Madhavan, Sagar Pande, Pooja Umekar, Tushar Mahore, Dhiraj Kalyankar, Comparative analysis of Detection of email spam with aid of machine learning approaches, ICCRDA, IOP Conf. Series: Materials Science and Engineering,2020.

[7] Deepika Mallampati, An efficient spam filtering using supervised machine learning techniques, IJSRCSE, 2018, 6(2):33-37.

[8] Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa. Machine learning for email spam filtering: review, approaches and open research problems, Heliyon 5, 2019.1-23.

[9] Naeem Ahmed, Rashid Amin, Hamza Aldabbas, Deepika Koundal, Bader Alouffi, and Tariq Shah , Machine Learning Techniques for Spam Detection in Email and IoT Platforms analysis and research Challenges, Hindawi - Security and Communication Networks, 2022,1-19.

[10] Sunil B. Rathod, Tareek M. Pattewar, Content Based Spam Detection in Email using Bayesian Classifier, IEEE ICCSP conference, 2015, 1257-1261.

[11] Pritesh A. Patil, Prayag P. Bhosale, Literature Survey on Spam Email Detection, International Journal of Research Publication and Reviews, 2022, 3(11):2688-2694, 2022.

[12] Asif Karim, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab, A Comprehensive Survey for Intelligent Spam Email Detection, IEEE 2019, 7:168261-168295.

[13] Murali B, Dr. C. Nagaraju, Tomato plant leaf disease classification by using morphological operations and machine learning techniques, Indian journal of natural sciences, 2023, 13(76):51994 – 52002.

[14] Murali B, Dr. C. Nagaraju, Image processing and machine learning techniques for detection of mosaic disease in banana, Indian Journal of Natural Sciences, 2022, 12(70):38654 – 38661, .