Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/13366
DOI URL: http://dx.doi.org/10.21474/IJAR01/13366

RESEARCH ARTICLE

SMS CLASSIFICATION: CONJOINT ANALYSIS OF MULTINOMIAL NAIVE BAYES APPLICATION

Vedad Burgic and Dr. Dino Keco
Department of Information Technologies, Faculty of Engineeringamd Natural Sciences, International Burch University.

Manuscript Info

Manuscript History
Received: 30 June 2021
Final Accepted: 31 July 2021
Published: August 2021

Key words:-
SMS Trustworthiness Prediction, Ham Messages, Spam Messages, Machine Learning, Text Processing

Abstract

Nowadays there are ham and spam messages that are sent to the users via SMS. The aim of this article is to show how machine learning and text processing technologies can be used in order to predict the trustworthiness of SMS messages. The data we are going to use is collected from Kaggle. This study is very important because it helps us to understand how machine learning and text processing can be used in order to predict message trustworthiness. At the time of writing this article, there was not an article explaining how this can be done using the Multinomial Naive Bayes algorithm. The methodology we used in this article consists of dataset collection, data cleaning, data analysis, text preparation, and training model. This will be seen in the methodology section in great detail. At the end of this article, we will show to u the accuracy that we have got when implementing a Multinomial Naive Bayes algorithm for the classification of SMS messages. This study was quite beneficial because anyone can see how Multinomial Naive Bayes algorithm usage can be beneficial in order to predict the trustworthiness of SMS messages.

Introduction:-

SMS spam is any unwanted or not-required text message indiscriminately sent to your mobile phone, often for commercial purposes. You can get the form of a simple message, a link to a number to log in or text, a link to a site for more information, or a link to a site to download an application.(SMS Spam, n.d.) The goal of this article is to show how the Naive Bayes multinomial algorithm can be used to predict the reliability of SMS messages. This article is quite advantageous for science because it faces a very important problem that has not yet been studied. It is related to the application of the Naive Bayes multinomial algorithm in the messages of the SMS data set to predict ham or spam messages.

According to A. Harisinghaney, A. Dixit, S. Gupta, and A. Arora, electronic emails, text messages, and courier chats have become part of and plot of our lives. The main objective is to detect the text, as well as spam e-mails based on images. To reach the goal, three algorithms are applied (KNN algorithm, Naive Bayes algorithm, and DBScan reverse algorithm). They got good precision from all three algorithms. The results showed a comparison of the three algorithms applied in the same data set.(Text and Image Based Spam Email Classification Using KNN, Naïve Bayes and Reverse DBSCAN Algorithm, n.d.) The article was beneficial for science but it showed how multiple algorithms can be used to predict the classification of email but not of SMS.

Corresponding Author:- Vedad Burgic
Address:- Department of Information Technologies, Faculty of Engineeringamd Natural Sciences, International Burch University.

According to Ankit Kumar Jain, Diksha Goel, Sanjli Agarwal, Yukta Singh, and Gaurav Bajaj, content filtering is a popular approach to spam detection. They evaluated the use of social networking analysis measures to improve the performance of a content filtering model. Comparative analysis of different algorithms in which features are implemented is also included in the document. Furthermore, it has the contribution of different characteristics in spam detection. After the implementation and according to the selected feature set the algorithm of the neuronal artificial network using the technique of job propagation backward more efficiently than others. (Jain et al., 2019)

The literature that has been mentioned above is concerned with the classification of e-mail and not on the SMS classification, but we know that e-mail messages and SMS are two different worlds and that their content is not the same. Furthermore, they are not using the Multinomial Naive Bayes algorithm that we will use in this study. The central search problem This study address is the application of machine learning to predict the validity of SMS messages. You will use a Multinomial Naive Bayes algorithm with different characters to see the best results with data that is collected from Kaggle. This study consists of two parts. One is data collection and another is the processing of data that will yield results. This study is composed of reviewing the literature in which we will talk about other research and we will talk about its limitations. Another part is the methodology in which we will talk about data collection and data processing. In the results section, there are results that we collect methods we have used to predict the reliability of SMS messages. In the end, in the discussion section, there will be words about how this study was beneficial for science and how it was different from the other studies carried out by the topic to predict the reliability of messages.

Literature Review:-

Using Social Network Analysis for Spam Detection

According to DeBarr, Dave, and Harry Wechsler. "Using social network analysis for spam detection." International Conference on Social Computing, Behavioral Modeling, and Prediction. Springer, Berlin, Heidelberg, 2010., content filtering is a popular approach to spam detection. Focus on analyzing the content of the message to identify spam. Use of social network analysis measures to improve the performance of a content filter model was evaluated. They observed performance improvements for spam detection in repeated experiments by measuring the centrality of message transfer agents. For example, an increase of 70% in the proportion of spam was detected with a false positive rate of 0.1%. The messages that claim exceptionally long routes between the sender message transfer agent and the recipient's courier officer proved the spam. This article has quite good value but it does not implement the Multinomial Naive Bayes algorithm that we are going to implement here.

Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm

According to Harisinghaney, Anirudh, et al. "Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." 2014 International Conference on Reliability Optimization and Information Technology (ICROIT). IEEE, 2014., The Internet has changed the path of communication, which has become increasingly concentrated in emails. Emails, text messages, and online messaging conversations have become a part and part of our lives. Of all these communications, emails are more likely to exploit. Thus, several courier providers use algorithms to filter emails based on spam and ham. On paper, they intended to detect text emails and spam emails with images. To achieve the goal, three algorithms have been applied: KNN algorithm, Naïfe Bayes algorithm, and inverse DBScan algorithm. Pre-processing the e-mail text before performing algorithms that it is used to make them better predict. Their article uses Spam and Ham's Enron Corpus dataset. In the article, the performance of the comparison is provided from three algorithms based on four measurement factors: accuracy, specificity, sensitivity, and precision. They were able to attain good accuracy by all three algorithms. This article implements three different algorithms which are good because they are able to see which algorithm can yield the best results but the shortcoming of this article is that it talks only about email messages, not on SMS messages and our article is here to fill in the blanks and it will still use Multinomial Naive Bayes algorithm.

A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages

According to Nagwani, Naresh Kumar. "A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages." International Arab Journal of Information Technology (IAJIT) 14.4 (2017)., spam messages increase in the same proportion because billions of SMS messages are sent every day. Recent advance numbers occur in the SMS spam detection field and filtering. The purpose of this work is twofold, is first to identify

and sort the anti-spam messages from the SMS messaging collection and the second is to identify the important SMS messages of non-spam's non-filtered messages. The work is to classify SMS messages for effective management and SMS messaging. Work is planned at two levels of binary classification in which the first level of SMS classification are classified in the two SPAM and NON-SPAM classes using popular binary classifiers, then on the second sorting level, SMS messages are classified in high-priority and normal priorities. Four algorithms were used: Support Vector Machine, Latent Dirichlet Allocation, Naïve Bayes, and Non-negative Matrix Factorization to categorize the text messages. Experiments proved that the SVM algorithm performs the best. This article is similar to this article but our article is implementing a Multinomial Naive Bayes algorithm which is not implemented in this article. Aside from that, we are using a different dataset than the dataset in this article.

Predicting Spam Messages Using Back Propagation Neural Network

According to Jain, Ankit Kumar, et al. "Predicting Spam Messages Using Back Propagation Neural Network." Wireless Personal Communications 110.1 (2020): 403-422., the availability of low-cost messaging services has resulted in an increase in spam messages. Many mobile applications are developed to detect anti-spam messages on mobile phones, but there is still a lack of a complete solution. Their article presents an approach to detect spam. They have identified an effective set of features for text messages that sort messages in spam or ham with high precision. The resource selection procedure is implemented in text messages to get a resource vector for each message and the resulting resource vector is tested on a set of machine learning algorithms. The article also presents a comparative analysis of different algorithms in which resources are implemented. After implementation and depending on all selected resources, the artificial neural network algorithm using the propagation technique operates in the most efficient way. This article also talks about the SMS messages which is the theme we are working on here but they are not using any algorithm that we are using and aside from that they are not testing a dataset with different features in order to see which one will yield the best results.

Methodology:-

A. ***Dataset collection***

This article is using the dataset collected from Kaggle which has 5,574 messages which is of decent size. It consists of 4825 ham messages and 747 spam messages. Datasets were collected from the sources below:

- A collection of 425 SMS spam messages has been manually extracted from the Grumbletext Web site.
- A subset of 3375 SMS is taken from NUS SMS Corpus (NSC). The messages come largely from singapures and mainly from students who attend college. These messages were collected by the volunteers who knew the fact that their contributions would be made publicly available.
- A list of 450 SMS ham messages from Caroline Tag's Ph.D. thesis.
- SMS Spam Corpus v.0.1 Big has 1,002 SMS ham messages and 322 spam messages.

B. ***Dataset information***

This dataset has 4 columns and we will describe them below:

- v1: describes is message ham or spam.
- v2: the content of the message.
- Unnamed 1: unknown value
- Unnamed 2: unknown value

C. ***Data cleaning***

We will need to clean this dataset because it has vague and useless information that we do not need inside our training model. These values are “Unnamed 1” and “Unnamed 2”. Also we will rename the column values from “v1” to “label” and from “v2” to “sms”. This dataset has no empty values for label and SMS which means that we will not need to filter them out from our dataset. Command that we have used in order to drop unnecessary columns and rename vague columns is below:

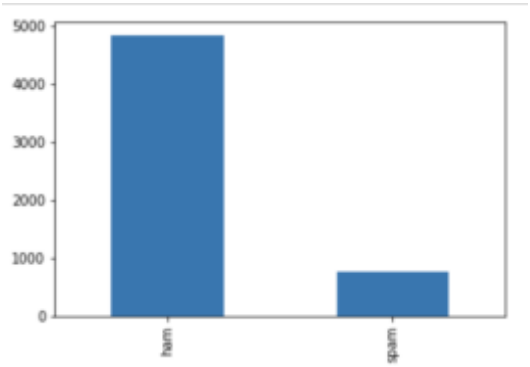
```
messages = messages.rename(columns={'v1': 'label', 'v2': 'sms'}).drop(columns=['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'])
```

D.

E.

F. ***Data analysis***

We will analyze and export important features from our dataset here. It will be of great benefit because it helps us to understand this dataset better. On the graph below the number of ham and spam messages in our dataset can be seen:



The number of ham messages is 4825 and the number of spam messages is 747.

1)

2) ***Most popular messages***

In the table below we can see the most popular ham messages and a number of their occurrences:

SMS	Count
Sorry, I'll call later	30
I cant pick the phone right now. Pls send a message	12
Ok...	10
Okie	4
Ok.	4

In the table below we can see the most popular spam messages and a number of their occurrence:

SMS	Count
Please call our customer service representative on FREEPHONE 0808 145 4742 between 9am-11pm as you have WON a guaranteed å£1000 cash or å£5000 prize!	4
I don't know u and u don't know me. Send CHAT to 86688 now and let's find each other! Only 150p/Msg rcvd. HG/Suite342/2Lands/Row/W1J6HL LDN. 18 years or over.	3
Camera - You are awarded a SiPix Digital Camera! call 09061221066 fromm landline. Delivery within 28 days.	3
Loan for any purpose å£500 - å£75,000. Homeowners + Tenants welcome. Have you been previously refused? We can still help. Call Free 0800 1956669 or text back 'help'	3

HMV BONUS SPECIAL 500 pounds of genuine HMV vouchers to be won. Just answer 4 easy questions. Play Now! Send HMV to 86688 More info:www.100percent-real.com	3

From the information above we can see that spam messages are lengthier. That may be the sign that we should include length features in our training model.

*3) **Most popular words***

In the table below we can see the most popular spam words and their occurrences:

Word	Count
call	346
free	219
txt	156
ur	144
i	144
mobile	123
text	121
stop	114
claim	113

In the table below we can see the most popular ham words and their occurrences:

Word	Count
u	989
gt	318
lt	316
get	301
ok	256
go	248
got	242
ur	237
know	236
Like	231

G. ***Text preparation***

We will need to remove punctuation and stopwords in order to get accurate predictions of our model. We will use the function below in order to achieve that:

```
def format_sms(sms):
    # Remove punctuation
    sms_formatted = "".join([character for character in sms if character not in string.punctuation]).split()
    # Remove stopwords
    sms_formatted = [word for word in sms.split() if word.lower() not in stopwords.words('english')]
    return " ".join(sms_formatted)
```

Later we are applying that function on our dataset:

```
messages['sms'].apply(format_sms).head()
```

H.

I. ***Train model***

1) ***Vectorization***

Vectorization is a technique with which you can run your code quickly. It is a very interesting and important way to optimize algorithms when you are implemented from scratch.(Mansoori, 2020) We will use the code below for vectorization:

```
vectorizer = TfidfVectorizer(input="english")
features = vectorizer.fit_transform(messages['sms'])
```

TFIDFVectorizer is used to convert a collection of raw documents into a TFIDF characteristics matrix. Equivalent to the protester followed by TfidFtransformer. Fit_Transform is used to learn vocabulary and IDF, return the document matrix. This is equivalent to the adjustment followed by the transformation but is implemented more efficiently. (Sklearn.Feature_extraction.Text.TfidfVectorizer — Scikit-Learn 0.24.2 Documentation, n.d.)

2)

3) ***Split data to test***

We are splitting vectorized features into 4 variables with the train_test_split method:

- features_train: independent train variables.
- features_test: independent test variables.
- labels_train: dependent train variables.
- labels_test: dependent test variables.

train_test_split is used to divide matrices into random train and test subsets. Rapid utility that surrounds the validation of the input and the next(shufflesplit().split(x)) and the application to enter data in a single call to divide data in a one-liner. (Sklearn.Model_selection.Train_test_split — Scikit-Learn 0.24.2 Documentation, n.d.)

We are using 70% of the data for training the model and 30% for testing data. We are passing features and labels so the method is able to split data properly:

```
features_train, features_test, labels_train, labels_test = train_test_split(features, messages['label'], test_size=0.3)
```

4) ***Training with Multinomial Naive Bayes***

Domingos and Pazzani (1996) discuss their intake of property independence and explains why the naive bays behave well for classification even with brute simplification. McCallum and Nigam (1998) have a multinomial model Naive Bayes for the classification of the text and shows better performance than the Bernoulli multivariate model due to the incorporation of frequency information. (Rennie et al., n.d.)

In the line below we are initializing the Multinomial Naive Bayes object and we are passing alpha 0.2 to it. Alpha is an additive smoothing parameter.

```
mnb = MultinomialNB(alpha=0.2)
```

In the code below we are training the model with the fit method and we are passing features_train and labels_train. The fit method fits Naive Bayes classifier according to X, y. (Sklearn.Naive_bayes.MultinomialNB — Scikit-Learn 0.24.2 Documentation, n.d.)
Later we are predicting the outcome with a test dataset using the predict method. Predict method performs classification on an array of test vectors X. Later we are using the accuracy_score method in order to see the score of our algorithm.

```
mnb.fit(features_train, labels_train)
pred = (mnb.predict(features_test))
score_mnb = accuracy_score(labels_test, pred)
print(score_mnb)
```

5)

6) ***Training model with length feature***

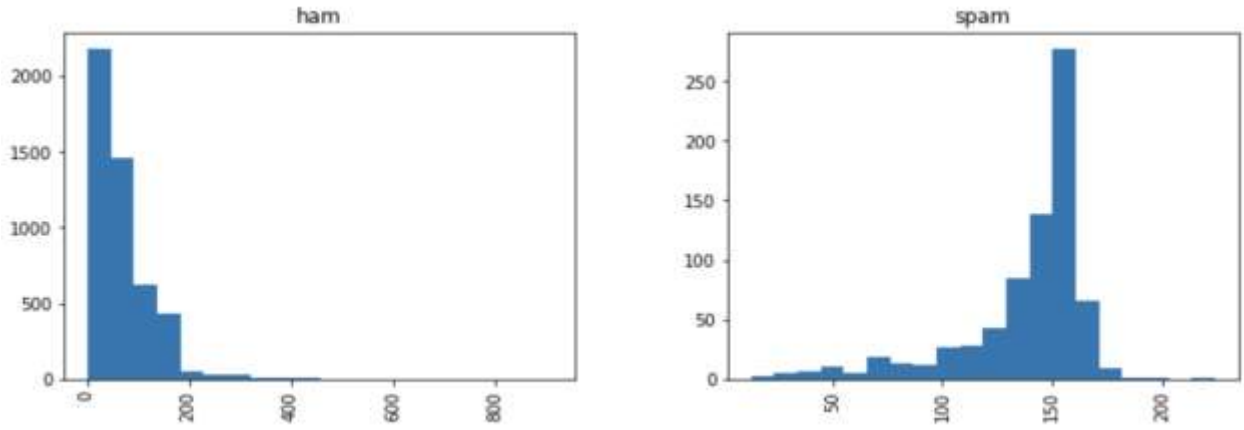
From the text above in the section most popular messages, we saw that spam messages are lengthier and we will add length features to our dataset in order to see if the accuracy of our model increases. We will copy the old dataset to the new dataset. SMS length feature is not natively added to our dataset but it is quite straightforward to add it as a column in our dataset with the simple function below:

```
messages_with_length = messages.copy()
messages_with_length['length'] = messages['sms'].apply(len)
```

It would be nice to see the length of both ham and spam messages on the graph. The length of the SMS messages will be visualized and we will have a better insight into the dataset. With the commands below we will be able to plot the graph:

```
messages_with_length.hist(column='length', by='label', figsize=(12,4), bins=20)
```

A histogram is a representation of data distribution. Function hist calls matplotlib.pyplot.hist(), in each series in the context of the data, resulting in column histogram. (Pandas.DataFrame.Hist — Pandas 1.3.3 Documentation, n.d.)



On the graph above we confirmed that spam messages are longer than ham messages. It was a good assumption stated above that spam messages are more lengthy than ham messages.

In the code below we are splitting new dataset into train and test data which is exactly same step that we did for the previous dataset:

```
features_train_with_length, features_test_with_length, labels_train_with_length, labels_test_with_length = train_test_split(features, messages_with_length['label'], test_size=0.3)
```

Below we are training models with a new dataset and we are predicting the outcome of the new dataset with test data. Later we are going to see what is the accuracy of the model in the results section.

```
mnb.fit(features_train_with_length, labels_train_with_length)
pred_with_length = (mnb.predict(features_test_with_length))
score_mnb_with_length = accuracy_score(labels_test_with_length, pred_with_length)
print(score_mnb_with_length)
```

Results:-

The aim of this article was to show how the Multinomial Naive Bayes algorithm can be used in order to predict SMS message ham or spam and we will show in this section that it can be of great benefit. At the beginning of the process, we gathered data, cleaned it, preprocessed it, and trained the model. Now we will show what are the results that we received from all the processes. The first model got an accuracy of 0.9826555023923444 which is very good accuracy. In the article, we showed in the histogram that spam messages tend to be longer than ham messages and we test the model with the length feature as well. When we trained the model with the length feature we got an accuracy of 0.9844497607655502 which is even better than one without length so our doubt was justified.

Discussion:-

The main purpose of this study was to show how the Multinomial Naive Bayes algorithm can be used in order to predict the trustworthiness of SMS messages. It was using different feature sets in order to see which one will provide the best results. This study was of great importance because there was no study done on how the Multinomial Naive Bayes algorithm can be used in order to predict the trustworthiness of SMS messages and also the concentration of most of the articles was on email messages not on SMS messages. This is something that we have already discussed in the Literature review. Below we will mention stated research without using new materials:

According to DeBarr, Dave, and Harry Wechsler. "Using social network analysis for spam detection." International Conference on Social Computing, Behavioral Modeling, and Prediction. Springer, Berlin, Heidelberg, 2010., evaluated the use of social networking analysis measures to improve the performance of a content filter model. This article has quite good value but it does not implement the Multinomial Naive Bayes algorithm that we are going to implement here.

According to Harisinghaney, Anirudh, et al. "Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." 2014 International Conference on Reliability Optimization and Information Technology (ICROIT). IEEE, 2014., Internet has changed the form of communication, which has been increasingly focused on e-mails. In the document, aim to detect spam e-mails based on the text and image. They used three algorithms: KNN algorithm, Naive Bayes algorithm, and DBscan reverse algorithm. In the document, a comparison performance of the three algorithms is provided on the basis of four measurement factors, ie, precision, sensitivity, specificity and precision. They could reach good precision from all three algorithms. (Text and Image Based Spam Email Classification Using KNN, Naïve Bayes and Reverse DBSCAN Algorithm, n.d.-a) This article implements three different algorithms which is good because they are able to see which algorithm can yield the best results but the shortcoming of this article is that it talks only about email messages, not on SMS messages and our article is here to fill in the blanks and it will still use Multinomial Naive Bayes algorithm.

According to Nagwani, Naresh Kumar. "A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages." International Arab Journal of Information Technology (IAJIT) 14.4 (2017)., SMS traffic is always day in day and billions of SMS are sent and received from billions of users every day. The goal of this work is double, first identifing and classifying spam messages from the collection of SMS messages and the second is to identify important SMS priority or messages from filtered nonSPAM messages. Four algorithms were

used: Naïve Bayes, Support Vector Machine, Latent Dirichlet Allocation and Non-negative Matrix Factorization. This article is similar to this article but our article is implementing a Multinomial Naive Bayes algorithm. Aside from that, we are using a different dataset than the dataset in this article.

According to Jain, Ankit Kumar, et al. "Predicting Spam Messages Using Back Propagation Neural Network." Wireless Personal Communications 110.1 (2020): 403-422., the increase in popularity of smartphones, text-based communication has also gained popularity. The availability of low-cost messaging services has led to an increase in spam messages. Their document has an approach to the detection of spam messages. The artificial neural network algorithm uses the technique of back propagation and works in the most efficient way. This article also talks about the SMS messages which is the theme we are working on here but they are not using any algorithm that we are using and aside from that they are not testing a dataset with different features in order to see which one will yield the best results.

The main limitation of this study is that it does not compare different algorithms but it can be done later by us or by someone else but it is very important that we have proven how the Multinomial Naive Bayes algorithm can be used in order to predict the trustworthiness of SMS messages. We also made a framework so everyone can check this article in order to implement their own algorithm for prediction.

References:-

1. Jain, Goel, Agarwal, Singh, & Bajaj. (2019). Predicting spam messages using back propagation neural network. Wireless Personal Communications, 110(1), 403–422. https://doi.org/10.1007/s11277-019-06734-y.
2. Mansoori, J. (2020, June 12). What is Vectorization in Machine Learning? Towards Data Science. https://towardsdatascience.com/what-is-vectorization-in-machine-learning-6c7be3e4440a.
3. Pandas.DataFrame.hist — pandas 1.3.3 documentation. (n.d.). Retrieved September 14, 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.hist.html.
4. Rennie, J., Shih, L., Teevan, J., & Karger, D. (n.d.). Tackling the Poor Assumptions of Naive Bayes Text Classifiers.
5. sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 0.24.2 documentation. (n.d.). Learn 0.24.2 Documentation. Retrieved September 14, 2021, from http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.
6. Sklearn.model_selection.train_test_split — scikit-learn 0.24.2 documentation. (n.d.). Learn 0.24.2 Documentation. Retrieved September 14, 2021, from http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.
7. Sklearn.naive_bayes.MultinomialNB — scikit-learn 0.24.2 documentation. (n.d.). Learn 0.24.2 Documentation. Retrieved September 14, 2021, from http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html.
8. SMS spam. (n.d.). Security Against SMS Spam. Retrieved September 14, 2021, from https://www.t-mobile.com/privacy-center/education-and-resources/sms-spam
9. Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm. (n.d.-a). IEEE Xplore. Retrieved September 14, 2021, from https://ieeexplore.ieee.org/document/6798302?reload=true&tp=&arnumber=6798302.
10. Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm. (n.d.-b). IEEE Xplore. Retrieved September 14, 2021, from https://ieeexplore.ieee.org/document/6798302.