SPAM SMS CLASSIFIER USING MACHINE LEARNING ALGORITHMS

Manish Singh
Department of Computer
Science and Engineering
Chandigarh University
Gharuan, Mohali,India
manishsingh07092003@gmail.com

Natasha Sharma
Department of Computer
Science and Engineering
Chandigarh University
Gharuan, Mohali,India
natasha.sharma1003@gmail.com

Kumar Prasanjeet
Department of Computer
Science and Engineering
Chandigarh University
Gharuan, Mohali,India
prasanjeet1720@gmail.com

***Abstract*— Mobile SMS communication is insecure as a result of a significant problem with spam detection. A technique or model with high accuracy and precision is required to address this spam SMS issue. The amount of spam emails has dramatically increased over the last few years. SMS spam has major negative impacts since it harms both consumers and service providers, eroding their mutual trust to a great extent. Different types of classifier algorithm have been implemented like Naïve bayes, Random Forest, KNN and Support vector classifier on a raw dataset collected from UCI repository in this research. Metrices like Accuracy, Precision and Recall are takes as performance metrics for calculating the efficiency of the algorithm. After experimenting, the result of these algorithms and compared them with another models. We showed the comparison using Visualization Techniques.**

Keywords--- SMS, Machine learning, KNN, SVM, Naïve Bayes, Spam detection, Random Forest, Messages.

I. INTRODUCTION

SMS Spamming is very frustrating for users because it can cause numerous important and valuable messages to be lost. Phishing spam messages pose a real risk to users' security because they try to trick them into giving up personal information like passwords and record numbers by using parody messages that appear to come from reliable online organizations, like financial institutions. There are many reasons why the amount of spam messages is rising. First of all, a large portion of the global population uses mobile devices, making a large portion of that population susceptible to spam communications [1, 2, 6]. Second, the spammer may benefit from the low cost of sending spam messages [2, 4]. Machine learning has been one of the most discussed topics in recent years, and there are many classification applications based on machine learning that are used in a wide range of academic disciplines. In particular, spam detection is a very established field of study with a number of tried-and-true methods. The dataset is a large text file with the label and text message string of each message at the beginning of each line. After the data has been pre-processed and features have been retrieved, machine learning algorithms like SVM, Decision Tree, Naive Bayes and others are applied to the samples, and their results are compared. Specificity, accuracy, and sensitivity were taken into consideration when analyzing the proposed study's performance indicators [9]. Machine learning is the idea of learning how to use the data at hand to make decisions, predictions, and clusters. Additionally, it will develop itself to produce superior outcomes in a number of areas. Developing a classification algorithm that filters SMS spam would provide a useful tool for mobile phone manufacturers.

II. BACKGROUND AND RELATED WORK

Various types of models are mentioned below based on the detection of the type of dataset and which technique they are using to do that. A clear and basic overview of the models is given which are compared based on their accuracy, precision and recall score in detecting the output [12].

- KNN (K Nearest Neighbour) - K-Nearest Neighbour (K-NN) [5] stands as one of the most straightforward machine learning algorithms rooted in supervised learning. Its core premise lies in the assumption that new data points can be compared to existing ones. K-NN then categorizes the new data point into the category that best matches the existing categories [3]. This categorization hinges on assessing the similarity between the new data point and the stored dataset. In essence, the K-NN method enables the swift and precise classification of new data based on its likeness to previously gathered data.

- Naive Bayes is a supervised machine learning method primarily used for classification tasks, relying on Bayes' theorem [4]. This algorithm is widely used in text categorization, particularly with substantial training datasets. The Naive Bayes Classifier stands out as one of the simplest and one of the most effective classification algorithms currently. It plays a pivotal role in the development of fast machine learning models capable of making highly accurate predictions. Functioning as a probabilistic classifier, it makes predictions by assessing the likelihood of an event or object's occurrence.

- Logistic regression- The logistic function is used in this machine-learning approach to measure the connection between the categorical dependent variable and the independent variable [8]. It is a classification model which classifies the given input into their specific classes. In our model it will classify that whether the data is Spam or Ham.

III. Data Cleaning and Pre-processing

For creating an algorithm, the first step is to find the dataset which fits the requirement and then clean the raw data. Then the pre-processing of data is done so that it will ready to be used as parameter for training the model. Later, train the model and calculate the outputs. The first step to create the model is to get a dataset with all the required attributes for prediction. The raw dataset is collected from UCI repository [11, 14]. In the dataset there are 5572 rows and 5 columns, where 'spam' indicates that the message/SMS is spam or fraud and the 'ham' indicates that the message/SMS is genuine. The proportion of these spam and ham is in the ratio of 85:15 as shows in the Fig.1. So, the next step is data cleaning where the data quality is improved by removing all the null values and duplicate values from it. Only required columns are selected for usage and the rest are dropped using in-build drop function of pandas [11].

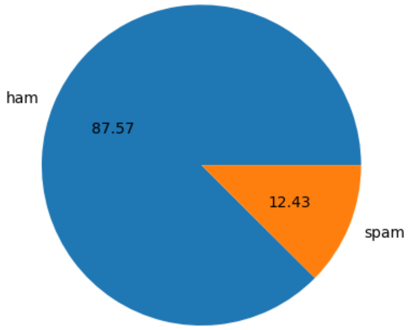


Fig-1: Pie chart of spam and ham data

Now that the data is ready for pre-processing, proceed further by encoding the data by 'spam'=1 and 'ham'=0 and replace them by spam and ham in the target column. As the data is imbalanced, it is unknown about which feature should be selected so copy the text attribute and split them into three different attributes which contains 'no. of sentences', 'no. of words' and 'no. of characters' [7]. The co-relation of the attributes will play a very important role for better performance of the algorithm. Hence, create a co-relation matrix of all these attributes as shown in Fig-2. By visualising the matrix, it can be seen that the co-relation between num_characters with num_sentences is 0.64, the co-relation between num_words and num_characters is 0.97. This shows that there is a heavy co-relation between these attributes.

Hence, all three attributes cannot be used together hence pick one attribute for further experiment. So, as the value of variation from the target is highest for num_characters, choose character attribute to build the model.

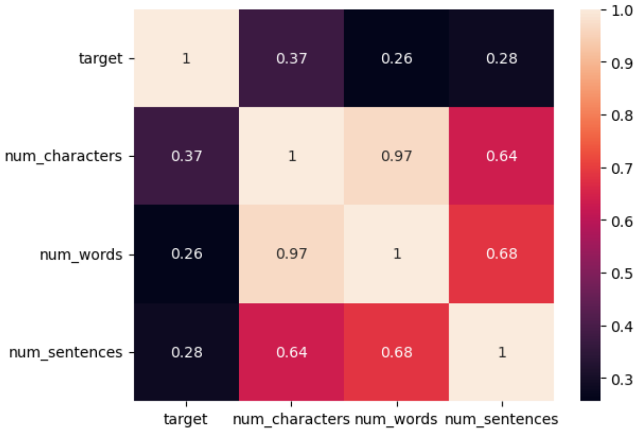


Fig-2 Co-relation matrix between spam and ham attributes

Now that selection of all the required attributes for the algorithm is done, begin the data pre-processing [14].
The work needs to be done in pre-processing the data are:

1. Lower case: converting all the alphabets into lowercase to improve the accuracy of algorithm.
2. Tokenization: Convert the text data into list of strings by using nltk.word_tokenize(text) which is an in-built function of nltk library.
3. Removing Special Characters: Remove all the special characters present in the text. Some of the special characters are: %, $, #, @, &, etc.
4. Removal of Stopwords and punctuations: Message contains a lot of punctuations and Stopwords which do not provide any meaning to the sentence i.e., are, is, a, the, (), {}, etc. They just increase the text size without providing and significant meaning to the text. Hence for better usage of the text data, remove all these stopwords from the text itself. It is done that by importing stopwords from nltk.corpus library and string.punctuation from string library. If the text contains any of the punctuation or stopwords, it will be removed from the text.
5. Stemming: There are words like dancing and loving where the substring 'ing' doesn't provide any meaning to the word. Hence, remove them so that they won't create any miscalculation during the prediction of algorithm. In this algo all these stemming words are removed by using Portstemmer which is imported from a library which is nltk.stem.porter.

   Ex- Input: Dancing
   Output: Danc

Now as all the pre-processing is done which was needed for the algorithm, dataset have a new_text attribute which contain the text which do not have any punctuation, special characters, any upper case letter and any suffix of words. Let's have a look at the most common or repeated spam and ham words in the data by using the WordCloud which is imported by word cloud library in the below Fig-3, 4.

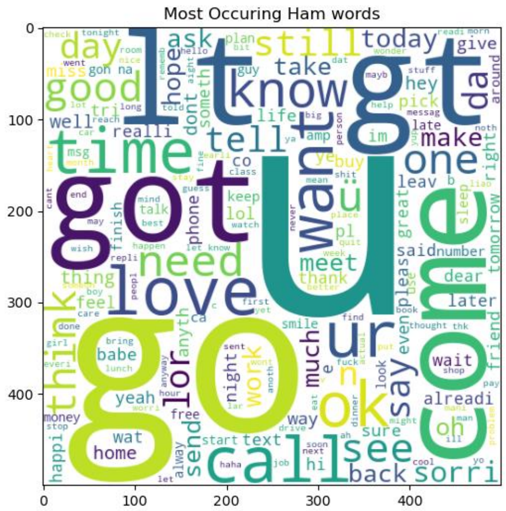


Fig-3: Most occurring Ham words

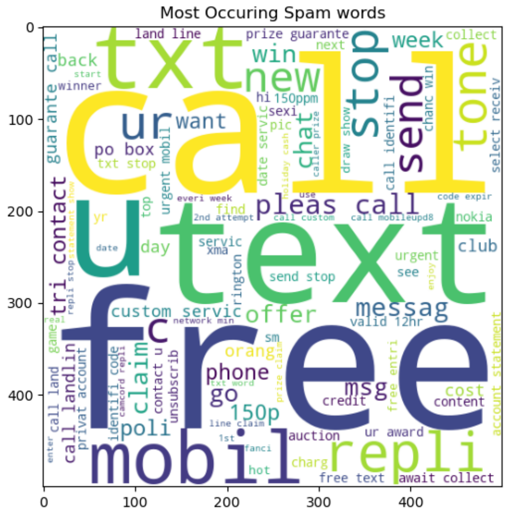


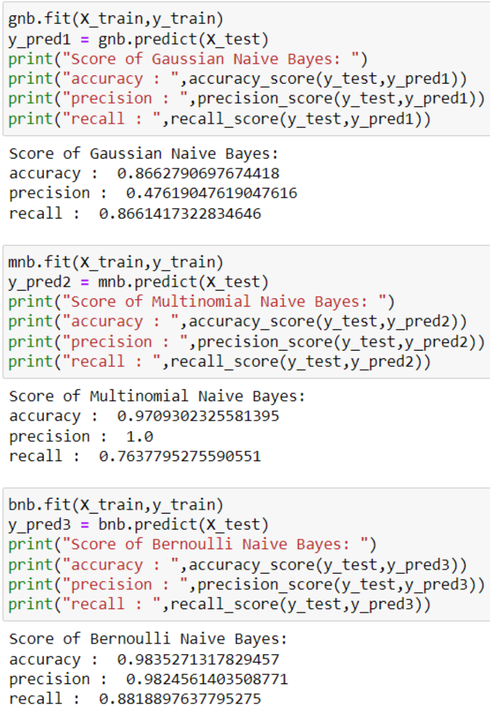
Fig-4: Most occurring Spam words

Now the most important part of the project which is the corpus from where the messages are going to be declared as ham and spam is created. Put all the spam words in a spam corpus and all the ham words in the ham corpus. The wordcloud which is shown in Fig-4,5 shows the words which occurs the most in their specified corpus. After creating both spam and ham corpus, we can now focus on building the prediction model.

IV. Model Training and Evaluation

The data is in string format but it is needed in numerical manner. Hence, first of all, import count vectorizer from sklearn library. The new_text is entered as a parameter in the countVectorizer() function and the new_text is converted into an array of numerical data. Because of this transformation now it has both X and Y parameter so imported train_test_split from sklearn and gave X and Y as a parameter in it [15].

There are a lot of models by which this algorithm can be implemented but the main problem is to select the best model which gives the best output. As given, Naïve Bayes is best classifier for text-based data so use that model for starting and later compare it with all other models by performing the same algorithm in them too [9,13]. The distribution of the data is not known hence try it on all the Naïve bayes model which are GuassianNB, MultinomialNB and BernouliNB as shown in Fig-6 [3, 8]. MultinomialNB gives Accuracy: 97.09%, Precision: 100% and Recall: 76.3%. BernouliNB gives Accuracy: 98.3%, Precision: 98.2% and Recall: 88.1%. GaussianNB gives Accuracy: 86.6, Precision:47.6, Recall: 86.6%. For prediction model, the most reliable factor for prediction is precision and it is clearly visible that Multinomial Naïve Bayes give the precision of 100% hence MNB is better as compared to others.



```
gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)
print("Score of Gaussian Naive Bayes: ")
print("accuracy : ",accuracy_score(y_test,y_pred1))
print("precision : ",precision_score(y_test,y_pred1))
print("recall : ",recall_score(y_test,y_pred1))
```

```
Score of Gaussian Naive Bayes:
accuracy :  0.8662790697674418
precision :  0.47619047619047616
recall :  0.8661417322834646
```

```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print("Score of Multinomial Naive Bayes: ")
print("accuracy : ",accuracy_score(y_test,y_pred2))
print("precision : ",precision_score(y_test,y_pred2))
print("recall : ",recall_score(y_test,y_pred2))
```

```
Score of Multinomial Naive Bayes:
accuracy :  0.9709302325581395
precision :  1.0
recall :  0.7637795275590551
```

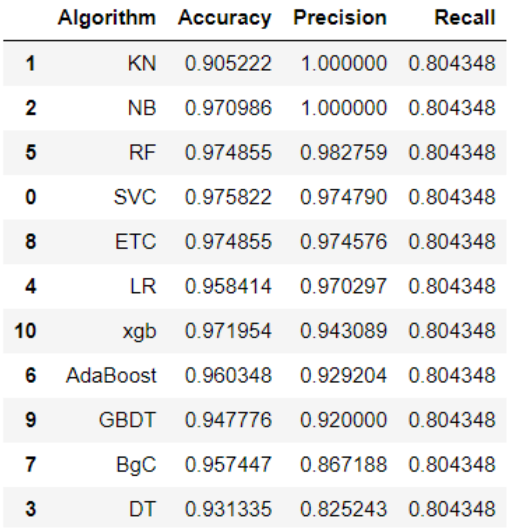
```
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print("Score of Bernoulli Naive Bayes: ")
print("accuracy : ",accuracy_score(y_test,y_pred3))
print("precision : ",precision_score(y_test,y_pred3))
print("recall : ",recall_score(y_test,y_pred3))
```

```
Score of Bernoulli Naive Bayes:
accuracy :  0.9835271317829457
precision :  0.9824561403508771
recall :  0.8818897637795275
```

Fig-6: Score of all Naïve Bayes models

V. Comparison with Existing Solutions

The identification of SMS spam is a relatively new topic of study, following the detection of spam in text messages, emails with social media attachments, tweets, and websites. Several studies on spam detection include [1, 2] and others. These studies are typically carried out after in the last few years. The usage of local and shortcut terminology, the limited message size, and the lack of complete slogan information are some of the challenges that recognized SMS spam detection techniques face. These problems must be resolved. There is currently a research gap in this area, and some studies have already been done. We mostly used Google Academic to look for relevant studies. We have collected a number of papers from it that have been published in additional conferences and journals, including IEEE explore, IJCSI ITJ ACM, and others. Google's educational tool There are numerous references in the collection of journals and conference papers that we have picked. We also looked for the cited publications, and we used a few of them as the basis for our own work [5, 7]. Our Assessment was carried out with the intention of reviewing all the methods and procedures applied in SMS spam identification. Numerous datasets were evaluated using various models to test the spam of SMS messages in various research publications, and it was determined which model provided the highest level of accuracy [10]. SVM, naive Bayes, decision trees, and k nearest neighbours are the models that have been employed most frequently in studies. We will be comparing these models considering various types of datasets. It has been observed that the data set used for the training and testing of the model in the majority of experiments contains a combination of ham and spam messages where more than 70% is ham messages and around 20-30% is spam messages. Because the output of the data is categorical either ham or spam so classification models are mostly considered for this research. In [2] the models taken for consideration are SVM (support vector machine), KNN (K nearest neighbour), NN (neural networks) the data set texts are been converted into numeric form to save time, after testing on the basis of evaluation the NN model shows the best accuracy of 95% but on the basis of considering all the other components which are precision, recall and f1 score its shown that the Naïve Bayes model is the best among the three. In [6], three models are taken for study which are LR (logistic regression), KNN and DT (decision tree), the dataset is split into 2 portions training data and testing data in the ratio 70:30. The accuracy of DT is observed to be the high enough that is 98% but it takes a lot of time, where else the highest accuracy is of LR that is 99% and it showed good performance in all overall conditions. There are 5 various models taken into research in [4] that are LR, KNN, NB (naïve bayes), SVM and DT. The best accuracy that has been observed is of SVM which is 98%, at second position we have NB with accuracy of 93% and taken the most less time than the other 4, so been regarded as the best one among all other. According to [3] SVM is the overall best model with respect to other models that are NB, KNN, Random Forest. Most of the researches and studies have used the above-mentioned models but, in some cases, it has been seen that some different models are considered for detection of SMS spam messages, like in [10] BiLSTM was also used with various other models that are NB, DT, bayes net. The BiLSTM was observed to have more accuracy than the other models that is of 94%. Another new model studied in [11] was maximum entropy classifier with 2 other models NB and SVM, but MEC shown the least accuracy here also SVM had the highest accuracy of 97%. According our study it has been remarked that in most of the researches the models which are most commonly used is SVM, KNN and NB among which SVM has been considered as the best according to the accuracy and other evaluations like time taken, recall, precision, f1score [5, 9]. The average accuracy of various models is given in Table-1.



	Algorithm	Accuracy	Precision	Recall
1	KN	0.905222	1.000000	0.804348
2	NB	0.970986	1.000000	0.804348
5	RF	0.974855	0.982759	0.804348
0	SVC	0.975822	0.974790	0.804348
8	ETC	0.974855	0.974576	0.804348
4	LR	0.958414	0.970297	0.804348
10	xgb	0.971954	0.943089	0.804348
6	AdaBoost	0.960348	0.929204	0.804348
9	GBDT	0.947776	0.920000	0.804348
7	BgC	0.957447	0.867188	0.804348
3	DT	0.931335	0.825243	0.804348

Table 1. Avg scores of models

In the above table we can see that by finding the average precision and other factors of different models KN, NB, RF are most accurate for detecting SMS spam messages. The graph in Fig-6 shows the comparison of all the model.

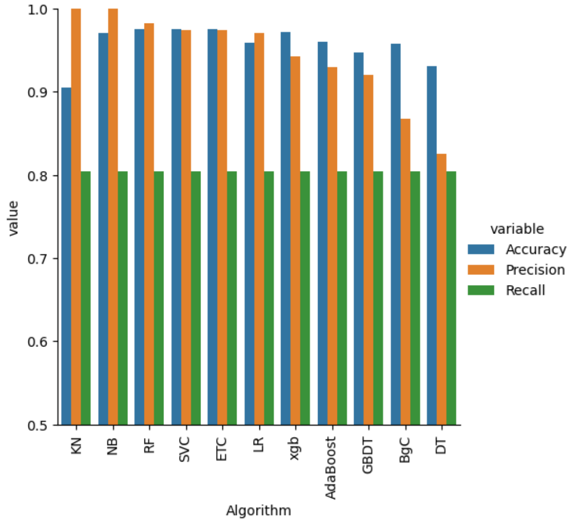


Fig-6: Comparison of all the model's scores

VI. CONCLUSION

This research paper proposes various methods to filter SMS spam messages by various machine learning algorithms. Several papers were studied and it's been observed that, Naïve Bayes, Random Forest and K-Nearest Neighbour has been taken into consideration for the testing the most of times. Among different models Multinomial Naïve bayes was seen the best algorithm to detect ham and spam messages with the best precision, accuracy and recall score.

VII. References

[1] T. J. Rani, "SMS Spam Detection Framework," *International Journal of Computer Science and Mobile Computing,* 2021.

[2] T. Krishna, "SMS Spam Detection," *Mathematical Statistician and Engineering Applications,* 2022.

[3] L. GuangJun, "Spam Detection Approach for Secure Mobile Message," *Security and Communication Networks,* 2020.

[4] M. Julis, "Spam Detection In Sms Using Machine Learning," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9,* 2020.

[5] A. Patel, "SMS Spam Detection using Machine Learning," *IJCRT,* 2021.

[6] O. Abayomi-Alli, "A deep learning method for automatic SMS spam classification:," *wiley,* 2022.

[7] S. Nagre, "Mobile SMS Spam Detection using," *2018 JETIR December,* 2018.

[8] A. Ora, "Spam Detection in Short Message Service," National College of Ireland, 2020.

[9] S. D. Gupta, "SMS Spam Detection Using Machine Learning," *Journal of Physics: Conference Series,* 2021.

[10] B. M. M. Hossain, "A Systematic Literature Review on SMS Spam Detection Techniques," *International Journal of Information Technology and Computer Sc,* 2017.

[11] M. Gupta, "A Comparative Study of Spam SMS Detection using," *eventh International Conference on Contemporary Computing (IC3),* 2018.

[12] S. Nyamathulla, "SMS Spam Detection with Deep Learning Model," *Department of Information Technology, Vignan’s Foundation for Science Technology,* 2022.

[13] Harsh, "SMS Spam Classifier Using Machine Learning," *International Journal of Research Publication and Reviews,* 2023.

[14] N. K. Nagwani, "A Bi-Level Text Classification Approach for SMS," *The International Arab Journal of Information Technology,,* 2017.

[15] S. Sumahasan, "Content-based SMS Spam Messages classification," *International Journal of Engineering Research in Computer Science and Engineering,* 2021.