# Detecting Spam Emails/Sms Using Naive Bayes And Support Vector Machine

**Prachi Gupta, Ratnesh Kumar Dubey, Dr. Sadhna Mishra**

**Abstract:** SMS or Email spams are dramatically increasing year by year because of the expansion of movable users round the world. Recent reports have clearly indicated an equivalent. Email or SMS spam may be a physical and thriving drawback because of the actual fact that bulk pre-pay SMS packages are handily obtainable recently and SMS is taken into account as a trusty and private service. SMS spam filtering may be a relatively recent trip to deal such a haul. The amount of information traffic moving over the network is increasing exponentially and therefore the devices that are connected thereto are considerably vulnerable. Thus there's a bigger have to be compelled to secure our system from this kind of vulnerability, here network security play a really vital role during this context. In this paper, a SMS spams dataset is taken from UCI Machine Learning repository, and after perform pre-processing and different machine learning techniques such as Naive Bayes (NB) and Support Vector Machine (SVM) are applied to the dataset are applied and compute the performance of these algorithms.

**Index Terms** : data mining, review spam, classification, comparative analysis, spam detection, sentiment analysis, naive bayes, SVM.

## 1 INTRODUCTION

One of the foremost outstanding ways of communication between an enormous numbers of people is SMS, wherever transmission of messages should happen as per the communication commonplace protocols. Consequently, there's a demand for text categorization algorithms which will be used as a region of classifying the messageseither to ham or spam messages. whereas spam messages aren't tempting, ham messages are the one that's created by real users. thus spam messages should be known and detached once they reach the mobile station, example of spam messages are those created by promotional corporations. the extra consequences of the SMS spam are frustrating, they're conjointly intense longer, resources, cash and network information measure, in any case the accessibility of spam filtering computer code for characteristic SMS spam are restricted. Moreover, there's a further misclassification drawback that will be arising once ham messages are eliminated and blocked as spam [2]. Users get irritated of Email and SMS spam and ends up in performance humiliation of the service [3], SMS spam sometimes influence a cluster of individuals and disseminated through mobile networks, conversely World Wide net transmits email spam. Still, numerous solutions for SMS spam detection were nonheritable from email spam filtering and classification methodologies [4]. Also, there are completely different problems that are sweet-faced by the research worker analysts of SMS spam discovery like the restriction of brazenly accessible dataset.

## 2 LITERATURE REVIEW

According to [1], presents detection of Spam and ham messages using various supervised machine learning algorithms like naïve Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm and compares their performance in filtering the Ham and Spam messages. As people indulge more in Web-based activities, The fact that an email box is flooded with unsought emails and with rising sharing of private – data by companies, SMS spam is very common. SMS spam filter inherits much functionality from E-mail Spam Filtering. In the developing period of the Internet, individuals are involving increasingly in free online services. Individuals tend to share their data on different sites, though that data is imparted to different organizations that spam individuals to offer their services.According to [2], In todays world where large part of communication takes place within the kind of SMS or emails. However, because of advertising agencies and social networking websites most of the emails circulated contain unwanted info that isn't relevant to the user. Spam SMS or emails square measure a kind of email correspondence wherever the user receives unsought messages via email. Spam emails cause inconvenience and loss to the recipients therefore there's a necessity to filter them and separate them from the legitimate emails. several algorithms and filters are developed to observe the spam emails however spammers unendingly evolve and sophisticate their spamming techniques because of that the prevailing filters have become less effective. the tactic projected during this paper involves making a spam filter victimization binary and continuous likelihood distributions. According to [3], the short electronic messaging service (SMS) spam is known the unsought or unsought messages received on mobile phones. These SMS spams represent a veritable nuisance to the mobile subscribers. This promoting observe additionally worries service suppliers in sight of the actual fact that it upsets their purchasers or maybe causes them lose subscribers. By manner of mitigating this observe, researchers have projected many solutions for the detection and filtering of SMS spams. In this paper, they tend to gift a review of the presently obtainable strategies, challenges and future analysis directions on spam detection techniques, filtering and mitigation of mobile SMS spams.

## 3 PROBLEM DEFINITION

makes it potential for the account holder to miss a crucial message; thereby defeating the aim of getting AN email address for effective communication. These junk emails from on-line promoting campaigns, on-line fraudsters among others is one in every of the explanations for this paper. We try to obtain the feature sets that can best represent and distinguish the spams from ham(non-spam). We then follow both supervised and unsupervised methodology to obtain spams from the dataset. We also include sentiment analysis


- *Prachi, Gupta, Mtech(Research Scholar), CSE LNCT Bhopal*
- *Prof. Ratnesh Kumar Dubey, Assistant Professor, CSE LNCT Bhopal*
- *Dr. Sadhna Mishra, Professor & Head, CSE LNCT Bhopal*

methodology into our spam detection. Lastly, we compare our analysis obtained from taking various types of feature sets based on text, sentiment scores, reviewer features, as well as the combined method.

## 4 PROPOSED WORK

A machine learning techniques have been proposed for detecting and classify the reviews through various processing steps which is shown in figure 1.

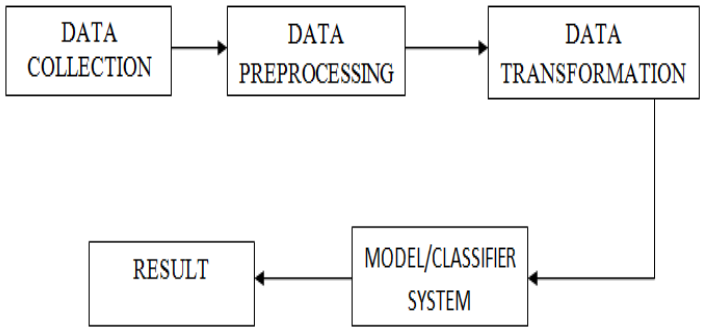


***Fig. 1**. Proposed Flow Diagram*

Our Steps or Algorithm Steps will follow:

1. Data Collection:- The SMS collection dataset we have taken from UCI machine learning repository.
2. Data Preprocessing: Data preprocessing is the most important phase in detection models as the data consists of ambiguities, errors, redundancy which needs to be cleaned beforehand.
3. Data Transformation: Data is transformed into lowercase and change the data types according to algorithm needs.
4. Classification System: The attributes are identified for classifying process and system perform feature extraction and then these classification system classify the content into spam or ham.

## 5 EXPERIMENTAL ANALYSIS

The experimental and result analysis is done by using intel i5-2410M CPU with 2.30 GHz processor along with 4 GB of RAM and the windows operating system is running. For result analysis we use R and R studio for processing the data and then we load the sms dataset which consist a 5574 observation with no missing are present in the dataset. Figure 2 shows the dataset has been loaded.

```
> data_text <- read.delim("SMSSpamCollection", sep="\t", header=F, colClasses="chara
cter", quote="")
>
> str(data_text)
'data.frame':   5574 obs. of  2 variables:
 $ V1: chr  "ham" "ham" "spam" "ham" ...
 $ V2: chr  "Go until jurong point, crazy.. Available only in bugis n great world la
 e buffet... Cine there got amore wat..." "Ok lar... Joking wif u oni..." "Free entr
y in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to recei
ve entry question("| __truncated__ "U dun say so early hor... U c already then sa
y..." ...
>
> head(data_text)
   V1
1 ham
2 ham
3 spam
4 ham
5 ham
6 spam

                                                      V2
1                                 Go until jurong point, crazy.. Availab
le only in bugis n great world la e buffet... Cine there got amore wat...
2                                 Ok lar... Joking wif u oni...
3 Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 871
21 to receive entry question(std txt rate)T&C's apply 08452810075over18's
4                       U dun say so early hor... U c already then say...
5                 Nah I don't think he goes to usf, he lives around here though
6       FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like
 some fun you up for it still? Tb ok! XxX std chgs to send, Â£1.50 to rcv
```

***Fig. 2**. Loading a dataset*

Once the dataset has been loaded it need to be transform by providing proper attributes names and also convert the data types of the attributes. These data we have taken is collected from various sources so its need to be pre-processed before classify the data. tm packages is used to pre-processed the data which contains various inbuilt functions through which we can easily pre-processed the data by converting into lower case, removing unwanted urls, numbers , white spaces, etc. Various data cleaning steps are shown in figure 3.

```
      ham      spam
0.8659849 0.1340151
> library(tm)
Loading required package: NLP
>
> library(SnowballC)
> corpus = VCorpus(VectorSource(data_text$Text))
> as.character(corpus[[1]])
[1] "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine th
ere got amore wat..."
>
> corpus = tm_map(corpus, content_transformer(tolower))
> corpus = tm_map(corpus, removeNumbers)
> corpus = tm_map(corpus, removePunctuation)
> corpus = tm_map(corpus, removeWords, stopwords("english"))
> corpus = tm_map(corpus, stemDocument)
> corpus = tm_map(corpus, stripWhitespace)
> as.character(corpus[[1]])
[1] "go jurong point crazi avail bugi n great world la e buffet cine got amor wat"
>
>
> #Creating the Bag of Words for the model
>
> dtm = DocumentTermMatrix(corpus)
> dtm
<<DocumentTermMatrix (documents: 5574, terms: 6981)>>
Non-/sparse entries: 43801/38868293
Sparsity           : 100%
Maximal term length: 40
Weighting          : term frequency (tf)
>
> dtm = removeSparseTerms(dtm, 0.999)
>
```

***Figure 3**. Data pre-processing*

After pre-processing the data we create a document term matrix through which we can calculate the term frequency means how many times a term is coming in spam or ham messages. So we keep a record for those words which is coming more than 60 times. Figure 4 shows the wordcloud of the dataset.

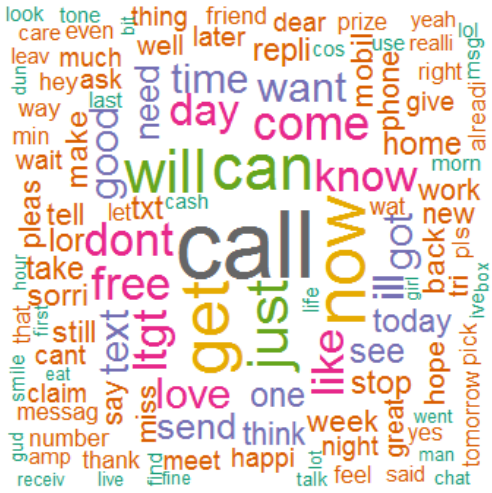


***Figure 4**. Presenting the word frequency as a word cloud*

Now we can split the dataset into two parts one from training data and second for testing data. Training data is for to train the machine learning model and once the model is trained we can test the model performance on testing dataset. In these we can build the two machine learning model which are SVM and Naive Bayes.

**Naive Bayes**

These classifier is based on the assumptions of conditional probability which is based on bayes theorem, These algorithm is very fast in training and the performance outcomes of the model on testing dataset are shown in figure 5.

```
> nb_pred = predict(classifier_nb, type = 'class', newdata = test_set)
>
> confusionMatrix(nb_pred,test_set$Class)
Confusion Matrix and Statistics

          Reference
Prediction  ham spam
      ham  1195    7
      spam    0  183

               Accuracy : 0.9949
                 95% CI : (0.9896, 0.998)
    No Information Rate : 0.8628
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9783

 Mcnemar's Test P-Value : 0.02334

            Sensitivity : 1.0000
            Specificity : 0.9632
         Pos Pred Value : 0.9942
         Neg Pred Value : 1.0000
             Prevalence : 0.8628
         Detection Rate : 0.8628
   Detection Prevalence : 0.8679
      Balanced Accuracy : 0.9816

       'Positive' Class : ham
```

***Figure 5.*** *Performance Measure of Naive bayes*

**Support Vector Machine**

These classifier is used to classify the data into classes based on the hyperplane which is nothing but a line which differentiate the data into two classes . The figure shows the performance outcomes of the model on the testing dataset.

```
> svm_pred = predict(svm_classifier,test_set)
>
> confusionMatrix(svm_pred,test_set$Class)
Confusion Matrix and Statistics

          Reference
Prediction  ham spam
      ham  1195  189
      spam    0    1

               Accuracy : 0.8635
                 95% CI : (0.8443, 0.8812)
    No Information Rate : 0.8628
    P-Value [Acc > NIR] : 0.4882

                  Kappa : 0.009

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.000000
            Specificity : 0.005263
         Pos Pred Value : 0.863439
         Neg Pred Value : 1.000000
             Prevalence : 0.862816
         Detection Rate : 0.862816
   Detection Prevalence : 0.999278
      Balanced Accuracy : 0.502632

       'Positive' Class : ham
```

***Figure 6****. Performance measure of SVM*

**Performance Measure**

We used accuracy, which are derived using confusion matrix.

**Table-1** Confusion Matrix

| | Classified as Normal | Classified as Attack |
|---|---|---|
| Normal | TP | FP |
| Attack | FN | TN |

Where

TN -Instances correctly predicted as ham.

FN - Instances wrongly predicted as ham.

FP -Instances wrongly predicted as spam.

TP -Instances correctly predicted as spam.

$$\text{Accuracy} = \frac{\text{Number of samples correctly classified in test data}}{\text{Total number of samples in test data}}$$

**Table 2**. Accuracy of the models

| Model | Accuracy |
|---|---|
| Naive Bayes | 99.49% |
| Support Vector Machine | 86.35% |

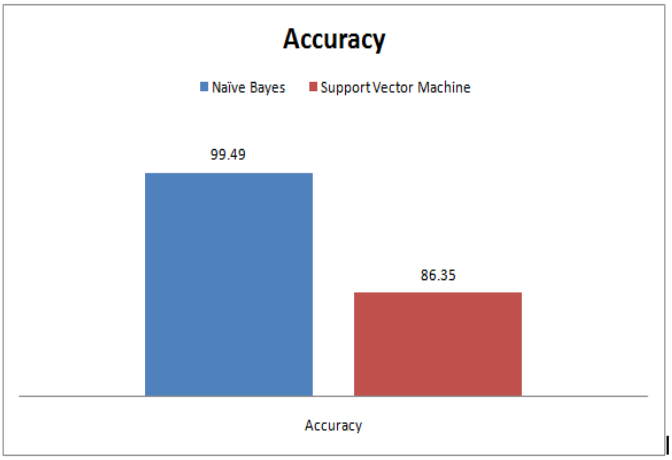


***Figure 7.*** *Accuracy Comparison*

## 6 CONCLUSION

In this paper, we propose a machine learning technique for SMS Spam filtering based on algorithms namely Naïve Bayes, and Support vector machine. The dataset that we have used in our work consists of 5574 observations of 2 variables. The first variable is the content of the emails and the second variable the target variable, which is the class to be predicted. In this paper, The Naive Bayes performed exceptionally well as compared to SVM.

## REFERENCES

[1] Pavas Navaney, Gaurav Dubey, Ajay Rana, "SMS Spam Filtering using Supervised Machine Learning Algorithms" in IEEE 2018.

[2] Shubhi Shrivastava, Anju R, "Spam Mail Detection through Data Mining Techniques" in IEEE 2017.

[3] Shafi'i Muhammad Abdulhamid, Muhammad Shafie, "A Review on Mobile SMS Spam Filtering Techniques " in IEEE 2017.

[4] Jindal, Nitin, Bing Liu. "Mining comparative sentences and relations." in AAAI, 2006.

[5] Jindal Nitin, LIU Bing. "Review spam detection" in ACM Press ,2007.

[6] Jindal Nitin, Liu Bing. "Opinion spam and analysis" in ACM Press , 2008.

[7] [Xie Sihong, WANG Guan "Review spam detection via temporal pattern discovery" in ACM Press , 2012.

[8] Lim Ee-Peng, Nguyen Viet-An "Detecting product review spammers using rating behaviors" in ACM Press , 2010.
[9] Jindal Nitin, Liu Bing "Finding unusual review patterns using unexpected rules" in ACM Press , 2010.
[10] Wang Guan, Xie Sihong "Identify online store review spammers via social review graph" in 2011.
[11] Nitin Indurkhya, Fred J Damerau. "Handbook of natural language processing" in Chapman and Hall 2010.
[12] Ohana B, Tierney B. "Sentiment classification of reviews using SentiWordNet" in 2009.
[13] Hu X, Tang J, Gao H "Unsupervised sentiment analysis with emotional signals" in 2013.
[14] Pak A, Paroubek P. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining" in 2010.