

# *A Machine Learning based Spam Detection Mechanism*

Nikhil Govil<sup>1</sup>

Assistant Professor, Department of CEA  
IET, GLA University  
Mathura, India  
nikhil.govil@gla.ac.in

Kunal Agarwal<sup>2</sup>, Ashi Bansal<sup>3</sup>, Astha Varshney<sup>4</sup>

BTech (CSE) IV<sup>th</sup> year, Department of CEA  
IET, GLA University  
Mathura, India  
<sup>2</sup>kunal.agarwal\_cs16@gla.ac.in

**Abstract**—In today's internet-oriented data; receiving spam email messages are quite obvious. Most of the time such emails are commercial. But many times, such emails may contain some phishing links that have malware. This arises the need for proposing prudent mechanism to detect or identify such spam emails so that time and memory space of the system can be saved up to a great extent. In this paper, we presented the same mechanism which can filter spam and non-spam emails. Our proposed algorithm generates dictionary and features and trains them through machine learning for effective results.

**Keywords**—spam; machine learning; naive-bayes; data set

## I. INTRODUCTION

Spam is an internet terminology that refers to either unsolicited commercial email (UCE) or unsolicited bulk email (UBE). Internet surfers usually call such kind of emails as junk email which are forwarded for any commercial or business promotional purposes. Here unsolicited simply means that the recipient of such junk emails has not granted permission for the messages to the sender [1].

In practice, receiving spam email decrease the efficiency of the user and it is quite annoying to the user [5]. Usually, such emails are forwarded for profit-making purposes or even through fraud agencies that can snatch money through phishing. Spam detection identification has the major objective to inform users about fake-emails and/or relevant e-mails.

## II. NEED OF SPAM DETECTION

Consider a case in which someone over the internet is sending bulk emails regarding the promotion of their products for purchasing it, or someone sending a link to click or activate to win some lucrative prizes; such emails are generally considered as negative marketing strategies or fraud activities. As a receiver, you are helpless in this scenario [6]. These unwanted emails may consume a lot of memory of your system & also waste your precise time. It is also observed that one can be distributed by receiving such bogus emails again and again.

So there is a need [4] for some mechanism that can reduce or even provide some sort of panacea to from these spam emails. Keep this situation in mind, in this paper; we are presenting a machine learning-based spam detection mechanism that uses a dataset of approximately 6000 valid and invalid collection of emails. Our proposed model will first make a dictionary that remove helping verbs form the contents of the email. Now our proposed algorithm will run to check whether the entered email address is spam or not. By applying this mechanism, a user can work efficiently as comparatively fewer spam emails will be received. This mechanism also saves the time and memory of the system.

## III. PROPOSED MODEL

In this model, we are firstly creating a dictionary [11] [12] that includes a library named "stopwords" [7] [8] to remove all possible helping verbs from the content mentioned in the email. In the next phase, features are generating to train the dataset. After that our algorithm will be executed to check the possibility of an email to be spam or not. Finally, the machine learning model will be tested on a real-world emailing environment.

### A. Generate Dictionary

Firstly we have to prepare the data for generating a dictionary for our algorithm. This dictionary can be further utilized to extract the desired features which determine our algorithm. These are parameters through which a user can segregate spam emails and general or simply non-spam emails.

To segregate junk or non-junk emails, first of all, we have collected some specific words and insert them into the dictionary which will be utilized in the proposed model. In the literature, numerous word extracting methods exists. So there is a need to select the most suitable model precisely.

### B. Generating features

After generating a dictionary, now it is time to generate features. For this, we will utilize the data set which has been trained by applying machine learning processes. The

specific features can further be extracted. These extracted features must be thoroughly tested. When all the test cases are successful then this data can be passed to the Naive Bayes algorithm in the form of inputs.

Once the features are generated, it is now time to work on the prediction model for the algorithm. Through this prediction model, different words will be inserted as inputs. The model will calculate and produce the result as total occurrences of each and every word throughout the document inserted.

### C. Generating Machine Learning model

A data structure has been prepared that will run through the Naive Bayes algorithm. As an outcome, we can achieve a prediction model. In literature, the Naive Bayes algorithm is also termed as a "probabilistic classifier". The reason behind it is that it is based a great extent on performing the calculation the probability of an email which may be further classified into numerous domains.

### D. Testing of Machine Learning Model

This is a very crucial step in our model. In this phase, we test our proposed model as providing data set in the form of input.

## IV. WORKING OF MODEL

### A. Sending of data

For every client, there exists an option of composing an email through which a text email can be composed. This data can be further exchanged between different clients whenever required. This application provides one of the main features of the system, which is content spam filter. The feature of attaching the file is disabled in the application as the file may contain images. This is feature is intentionally disabled because it is quite possible that spam got failed to detect hidden objects of the image(s).

### B. Receiving the data

When a textual email is sent from one client then it reaches the inbox of other clients. It is quite obvious that any email service system user has the basic features of the email provided by the application being used. These features may include inbox, writing, sending, replying, forwarding, filtering and even deleting the emails.

The observation that should be made here is that if the content in the email represents spam, subsequently the message is blocked by the tool before it is delivered to the target client machine. This way only the non-spam emails are exchanged between clients.

### C. Spam report module

Working with our model is crucially dependent on the spam report module. In this section, the server system is activated and the spam report module is deployed in the active supervision of the server. As an output, the server system provides messages which can be categorized into spam and non-spam categories. This data can be further analyzed for the determination of compromised and non-compromised machines based on the well-defined degree of fault-tolerance.

Discrete results of spam and non-spam machines are also recorded. It also keeps track of the client's system details as the client's name, timestamping of emails, spam details, etc for analyzation purposes. These spams are detected when the filtering algorithm is applied to the system. To maintain the privacy issues of various clients; it is also mandatory to encrypt the messages before entering it to the network.

### D. Spam filtering algorithm

In today's digital era; the main challenge is to filter and detect the spam emails which are being forwarded uninterruptedly over the internet. The Bayesian spam filter can be applied to control such junk messages to a great extent. Most of the spammers are adapting to new technologies and are becoming more effective. This spam filtering approach identifies and controls spam emails. With the help of this algorithm, we can effectively differentiate between legitimate and illegitimate spam emails. It has been also noticed that the chances of getting spam messages are higher if the client receives such messages in the past as well. It means spam messages are growing successively by the time. The content of the messages must be verified by any source of the algorithm. So that internet surfer can work over the internet without any disturbance.

There can be some words which are spam for one organization but not in other organization. For these types of words, the algorithm verifies the frequency of the particular word the number of times it has occurred with respect to some organization and recognizes the spam based on this probability ratio. The most important part of this algorithm is, it detects the spam based on the patterns. It identifies the section of words against its spam city rather than individual words for more effective detection of spam content in an e-mail.

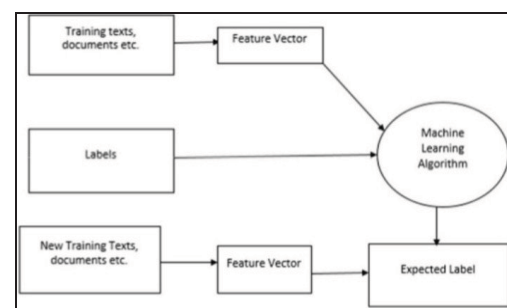


Fig. 1 Proposed Process architecture

Working of our proposed process model includes following basic steps:

1) *Training texts and document*: First we will obtain the data and then we will train the dataset according to the algorithm.

2) *Feature Vector*: Feature vector  $x$  composed of  $n$  words coming from spam emails. This vector comprises of  $n$  words coming from spam emails. The Naïve Bayes classifier makes the probability that the word is not dependent on each other. The main assumption of the classifier is each word is independent of each other.

3) *Machine Learning algorithm*: The Naive Bayes algorithm relies on Bayes Rule. This algorithm will look of the object will be  $n$  to each object by identifying all features separately. The algorithm shows us how we can calculate the posterior probability for one feature. The posterior probability will be calculated for a feature and these will be multiplied to get the final probability. This will be calculated on other classes.

4) *Expected Label*: This will give the outcome of the algorithm. We will let  $u$  know that the mail that is detected is spam or not.

Pseudocode for proposed algorithm

Begin:

```
create dictionary
seprate upper and lower case letters
import stopwords library
remove helping verbs through stopwords
function call naivebayes()
identifying spam and non-spam emails
display results
```

End;

#### E. Naive Bayes's classifier

Naive Bayes algorithm is a capability of refining filtering that uses features to recognize spam mails. It is an analytical technique used to classify text. The algorithm analyzes data and then apply naive Bayes to check whether an email is a spam or not. [2] [3].

It is the best approach to identify emails whether they are spam or not. This algorithm gives very low false positive spam detection rates and can change according to the need of the user. It is an intelligent approach that considers the complete message as the examining criterion rather than the single word thus saving a lot of time. The technique is highly scalable as it can be easily updated on the arrival of new data and adapts well to the future spam techniques. It is a less expensive approach as maximum-

likelihood training is done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation.

According to computer science, Naive Bayes models are also known as simple Bayes and independence Bayes models. The algorithm uses the references of Naïve Bayes; decision rule, but we know that naive Bayes is not a prominent method. It is a prominent method of technique for developing classifiers; models that assign class labels to problem instances represented as vectors of feature values, where the class labels are drawn from some finite set).

#### V. ADVANTAGES OF PROSOED ALGORITHM

- The algorithm has the feature of implementing according to the requirements of the user. Thus can be implemented in any organization.
- It is an intelligent approach as it considers the complete message as the examining criterion rather than a single word.
- It is highly scalable as it provides scalability of the algorithm that the user can utilize for better efficiency.
- The algorithm is easy to update on the arrival of new data. It is best suited for text classification.

#### VI. SPAM IDENTIFICATION METHODS

Numerous methods are presented in the literature to determine incoming emails that might be spam. The most prominent methods are Mail Header Analysis, Keyword Checking, URL Checking, etc.

##### A. Mail Header Checking

Mail header checking consists of well-defined protocols that, if a mail header matches, triggers the mail server to revert back the messages that have a blank "From" field. It also prepares a list that has numerous email addresses in the "To" from the same source, that has too many digits in email addresses. It also enables to return messages by matching the language code declared in the head [4].

##### B. Keyword Checking

This is also a widely used method for spam detection. Unlike mail header checking algorithm it scans both the body and the subject part of the email. Using rules such as combinations of keywords is a good solution to enhance filtering efficiency combinations of words can be specified and the list that must appear in the spam email can be updated further the messages with these words will be blocked [4].

### C. URL Checking

The URL checking method is also well-known spam detection method. In URL checking emails are classified on the basis of their URL. A list of addresses is recorded from which we never want to receive emails, any email coming from these addresses will be blocked [4].

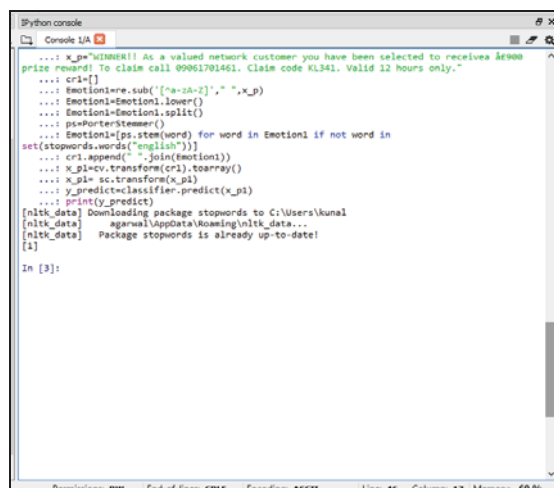
## VII. RESULTS

In our proposed process model, data is being trained so that Machine learning can work properly. Machine learning [9] [10] is an Artificial Intelligence approach that allows systems to learn automatically and improve from past experience. After applying the Naïve Bayes algorithm, emails are taken as inputs to the proposed model. Results are classified into 0 as non-spam and 1 as spam emails.



```
Python console
C:\Users\kunal agarwal\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15:
DeprecationWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in
0.23. Please import this functionality directly from joblib, which can be installed with: pip
install joblib. If this warning is raised when loading pickled models, you may need to re-
serialize these models with sklearn-learn 0.21+.
warnings.warn(msg, category=DeprecationWarning)
[0]
In [2]:
```

Fig. 2 Result as Non -Spam Mail showing “0”.



```
Python console
C:\Users\kunal agarwal\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15:
DeprecationWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in
0.23. Please import this functionality directly from joblib, which can be installed with: pip
install joblib. If this warning is raised when loading pickled models, you may need to re-
serialize these models with sklearn-learn 0.21+.
warnings.warn(msg, category=DeprecationWarning)
[0]
In [2]:
```

Fig. 3 Result as Spam Mail showing “1”.

## VIII. CONCLUSION

As of now recent days spam emails are increasing day by day and it is creating problem to the user so by spam detector, we will identify which mail is spam or not, by this efficiency of users will be increased. We are using the Naïve Bayes classifier that will give u the probabilistic index of that and will identify whether the mail is spam or not as per the shown results.

## REFERENCES

- [1] Sara Radicati, "Email Statistics Report, 2014-2018", The Radicati Group, Inc., Palo Alto, CA, USA, 2014.
- [2] Ramachandran, D. Dagon, and N. Feamster, "Can DNS-based blacklists keep up with bots?," in CEAS2006. The Second Conference on Email and Anti-Spam, 2006
- [3] G. Cormack, "Email spam filtering: A systematic review," Foundations and Trends in Information Retrieval, vol. 1, no. 4, pp. 335–455, 2008M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- [4] <http://www.securelist.com/en/threats/spam?chapter=97> last accessed on 23 February 2020.
- [5] A. Dalmia, M. Gupta, et al. Towards interpretation of node embeddings. In Companion of the The Web Conference 2018 on The Web Conference 2018, pages 945–952. International World Wide Web Conferences Steering Committee, 2018.
- [6] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada. Survey of review spam detection using machine learning techniques. J. Big Data, 2(1):1, 2015.
- [7] <https://pypi.org/project/stop-words/> last accessed on 03 March 2020.
- [8] <https://www.geeksforgeeks.org/removing-stop-words-nltk-python/> last accessed on 03 March 2020.
- [9] Manaranjan Pradhan and U Dinesh Kumar, "Machine Learning using Python", First Ed., Wiley, IIM Bangalore, (2019).
- [10] Tom M. Mitchell, "Machine Learning", McGraw Hill Education, (2017).
- [11] Duraipandian, M. "Performance Evaluation of Routing Algorithm for MANET based on the Machine Learning Techniques." *Journal of trends in Computer Science and Smart technology (TCSST)* 1, no. 01 (2019): 25-38.
- [12] Yashavant Kanetkar, "Let Us Python", First Ed., BPB Publication, (2019).