A novel approach for Spam Email Filtering Using Machine Learning

Subhalaxmi Sahoo[1], Sudan Jha[2], Deepak Prashar[3*]

[1]Research Scholar, Electrical Engineering, subhalaxmisahoo166@gmail.com
[2]School of Computer Science & Engineering, Lovely Professional University, jhasudan@hotmail.com
[3*]School of Computer Science & Engineering, Lovely Professional University, deepak.prashar@lpu.co.in

Abstract: Spam emails also known as unsolicited emails (maybe commercial or maybe not) i.e. those mails which are sent without our request or concern. Email spam is the practice of sending unwanted emails, mostly contains commercial messages to randomly generated persons. In the internet email spam is widespread because of such low cost of sending emails other than any other means of communication. It is important to filter spam emails because most of the malicious activities performed in the internet done through email spamming. Though there are many spam filters are available we still get huge amount of spam emails. This is not because the filters are not accurate & effective; the reason is the generation of quick and effective counters of the algorithm used in the filters. In our project we used mainly three supervised learning algorithms namely Linear SVC, Multinomial NB, and k-NN to implement the filter. We used these algorithms to train the system about spam email by using the feature called word count vector which is generated by processing a dataset filled with existing emails containing both spam and legitimate emails. The full process of the project and the result of the execution by implementing the three models/algorithms are discussed.

Keywords: Word Count Vector, Linear SVC, Multinomial NB, KNN.

1. Introduction

Electronic spam is the most troublesome Internet phenomenon challenging large global companies, including AOL, Google, Yahoo and Microsoft. Spam causes various problems that may cause economic losses. Spam causes traffic problems and bottlenecks that limit memory space, computing power and speed. Spam causes users to spend time removing it. For analyzing it we have used machine learning algorithms for detecting spam emails efficiently. We tend to determine whether these messages are either mails or spams or a legitimate mail by examining the structure of e-mails using Liner SVC, Multinomial NB and KNN algorithm.

The Internet is considered a very powerful tool. Email is an efficient way to exchange information. Considering the growth of the Internet and wide use of email, the rate of increase of spam is of great concern. Spam may originate from anywhere in the World Wide Web. Despite tools to prevent spam, it has been increasing daily. One way to assess the current situation is that organizations examine available means that can be used to even count the amount of spam. These means include corporate email systems, gateways, spam filtering and end user training. Internet users cannot disregard this important problem of the modern Internet world. Lack of mechanized systems to prevent spam will result in a spam-saturated World Wide Web, destruction of Internet products and severe loss of bandwidth.

DOI: 10.5281/zenodo.3715417

1.1 E-mail (electronic - mail)

Electronic mail (or e-mail or email) is an Internet Service that allows people who have an e-mail address (accounts) to send and receive electronic letters. Those are much like postal letters, except that they are delivered much faster than snail mail when sending over long distances, and are usually free. Like with regular mail, users may get a lot of unwanted mail. With e-mail, this is called spam. Some programs used for sending and receiving mail can detect spam and filter it out nearly completely.

Email operates across computer networks, which today is primarily the Internet. Some early email systems required the author and the recipient to both be online at the same time, in common with instant messaging. Today's email systems are based on a store-and-forward model.

1.2 Spam Email

Email spam, also known as junk email, is a type of electronic spam where unsolicited messages are sent by email. Many email spam messages are commercial in nature but may also contain disguised links that appear to be for familiar websites but in fact lead to phishing web sites or sites that are hosting malware. Spam email may also include malware as scripts or other executable file attachments (trojans). Spam is named after spam luncheon meat by way of a Monty Python sketch in which Spam in the sketch is ubiquitous, unavoidable and repetitive.

Email spam has steadily grown since the early 1990s. Botnets, networks of virus -infected computers, are used to send about 80% of spam. Since the expense of the spam is borne mostly by the recipient, it is effectively postage due advertising. This makes it an excellent example of a negative externally.

1.3 Ham Email

Ham email is a term sometimes used as opposed to spam messages. Ham is then all "good" legitimate email messages, that is to say, all messages solicited by the recipient through an opt-in process. In simple words – "E-mail that is generally desired and isn't considered spam".

1.4 Classification

In general, in classification we have a set of predefined classes and want to know which class a new object belongs to. It is a supervised learning process. It uses predictive methods.

1.4.1 Types of Classifiers:

1. Logistic Regression
2. Naïve Bayes
3. Stochastic Gradient Descent
4. K-Nearest Neighbours
5. Decision Tree
6. Random Forest &
7. Support Vector Machine

We have used Naïve Bayes, K-Nearest Neighbours & SVM for this project as mentioned in the paper.

1.4.2 <u>Naïve Bayes Classifier:</u>

This method uses Bayes Theorem to determine how often **A** happens *given that **B** happens*, written **P(A|B),** when we know how often **B** happens *given that **A** happens*, written **P(B|A)**, and how likely **A** and **B** are on their own.

Bayes Theorem –

$$p\,(B \mid A) = \{p\,(A \mid B) * p\,(B)\} / p\,(A)$$

Where-

DOI: 10.5281/zenodo.3715417

- P(A|B) is "Probability of A given B", the probability of A given that B happens
- P(A) is Probability of A
- P(B|A) is "Probability of B given A", the probability of B given that A happens
- P(B) is Probability of B
  - So according to the Project context we can say- If **S** is the event of a given e-mail being spam and **W** is a word in the e-mail, we will classify it as spam with probability:

p (S|W) = {p(W|S) P(S)} / {p(W|S) ·p(S)+p(W|$\bar{S}$)·p ($\bar{S}$)}

Where:

- p(S) is the anterior probability, which is set to the expected ratio of spam.
- p(W|S) and p(W|$\bar{S}$) are easily calculated by simply counting the occurrence of each word in spam and non-spam e-mails in the training data.
- p(S|W) is called the posterior probability, which is calculated using the anterior probability of being spam and the probability of the given word occurring in spam and non-spam e-mails. So the classifier is trained using some data to determine these word probabilities, which can also be adjusted when a user indicates a new e-mail to be spam or vice-versa.

1.4.3 K-Nearest Neighbors:

Neighbors based classification is one type of lazy learning, as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point. The *advantage* is that this algorithm is simple to implement, robust to noisy training data and effective if training data is large. On the other hand, the lacuna is there is a need to determine the value of K and the computation cost is high as it needs to computer the distance of each instance to all the training samples.

Process: KNN algorithm determines which class a new data point belongs to if there are given set of classes. It consists of well-defined steps –

*1. Choose the number **K** of neighbors. Most common default value for **K** is **5**.*

2. Take the K nearest neighbors of the new data point according to Euclidean Distance (can use any distance calculation mechanism).

3. Among the K neighbors, count the number of data points in each category/class.

4. Assign the new data point to the category where the no. of counted neighbors is most.

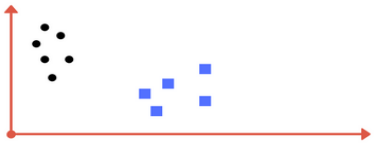
1.4.3 SVM (Support Vector Machine):

A Support Vector Machine (**SVM**) is a discriminative classifier formally defined by a separating hyper plane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyper plane which categorizes new examples. The advantage of SVM is generalized as the effective algorithm in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient. On other hand, this algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

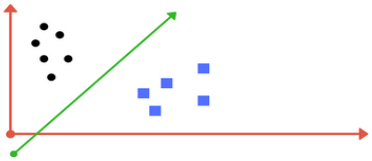
Method: SVM does the job of separating two classes, shown below –

Let us have two classes in a one dimensional space like the following picture –

DOI: 10.5281/zenodo.3715417



What SVM does is that it finds out a line/ hyper-plane (in multidimensional space) between two classes -



To find this hyper - plane there are some tuning parameters to be concerned about-

Kernel - The learning of the hyper plane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role-

For **linear kernel,** the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

f(x) = B (0) + sum (ai * (x, xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

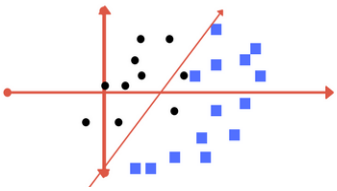
The **polynomial kernel** can be written as –

$K(x, xi) = 1 + sum(x * xi)\ ^d$

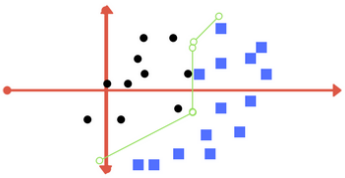
& **exponential** as

$K(x, xi) = exp(-gamma * sum((x - xi^2))$

Regularization- The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.



Pic: A

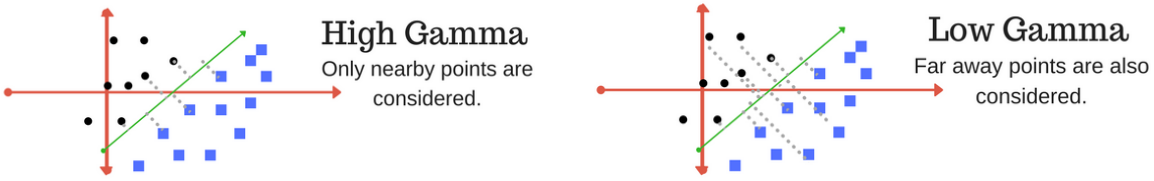


Pic: B

In the above picture it is shown that- A: low regularization value, B: high regularization value

DOI: 10.5281/zenodo.3715417

Gamma- The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible line are considered in calculation.



Margin- A *margin is a separation of line to the closest class points.* This is the very important characteristic of SVM classifier. SVM core always tries to achieve a good margin.

1.5 Restrictions that has to be considered while filtering

False-Positive – This is the case where good mails are mistakenly considered as spam by the filter.

False-Negative - This is the case where spam mails are considered to be good by the filter.

So, the restriction is that the designed filter should never give any False-Positive results but it is possible to give False-Negative results.

1.6 Confusion Matrix

In the field of machine learning and specifically the problem of statistical classification, a **confusion matrix**, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes.

We have used predictive methods in this paper as the algorithms that are used, all based on classification.

1.7 Text Mining

Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.

Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and interpretation of the

DOI: 10.5281/zenodo.3715417

output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness.

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics.

1.8 Packages Required

1.8.1 ANACONDA (Python Distribution):

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment.
Package versions are managed by the package management system - conda, which makes it quite simple to install, run, and update complex data science and machine learning software libraries like Scikit-learn, TensorFlow, and SciPy.

Features: The major ones are:

High-performance Distribution - Easy to Install 1,000+ data science packages.
Package Management - Many ways to manage packages, dependencies and environments with conda.
Portal to Data Science - Various methods to uncover insights in the data and create interactive visualizations.

1.8.2 Scikit-Learn:

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Features:

1. Simple and efficient tool for data mining and data analysis.
2. Accessible to everybody, and reusable in various contexts.
3. Built on NumPy, SciPy, and matplotlib
4. Open Source, commercially useable – BSD license.

2. RELATED WORK

2.1 The Framework

The essential issue lies behind data mining where the objective is to sensibly manage the trades as awesome 'ol shaped or enchanting. For get-together issues varying execution measures are portrayed a goliath bit of which are associated with right number of cases requested successfully. An all the all the all the also fitting measure is required in setting of the trademark structure of charge card trades. Right when a card is reiterated or stolen or lost and gotten by fraudsters it is every now and again used until its open inspiration driving limitation is depleted. In like path, instead of the measure of sensibly requested trades, an answer which moves the total open inspiration driving spread on cards subject to mutilation is more unmistakable.

Since the weakening seeing sales issue has all things considered been delineated as a get-together issue, paying little personality to some quantifiable structures accumulated data mining numbers have been

DOI: 10.5281/zenodo.3715417

proposed to direct it. Among these, decision trees and fake neural frameworks are the clearest ones. The examination of Bolton and Hand [1] gives a better than normal once-over of making on pound zone issues.

2.2 The Gained Estimation

Gained estimations in this paper go for displaying at change approaches as time advances. Since their first introduction by Holland [2], they have been viably associated with various issue locale from cosmology to sports, from move to programming building, et cetera. They have in like way been used as a touch of data burrowing unavoidably for variable decision and are for the most part joined with other data mining estimations. In this study, we endeavor to deal with our gathering issue by using only a trademark number system.

2.3 The Request Approach

Request is used for picking the best individuals, that is, for picking those chromosomes with higher achievement regards. The decision operation takes the present masses and passes on a 'mating pool' which contains the thorough social gathering which will go over. There are a couple decision structures, as uneven demand, discretionary decision, roulette wheel decision, address decision. In this work the running with decision bits are used.

2.3.1. Discuss Decision

Wrangle about decision has been used as a touch of this as it picks regard individuals from fluctuating get-togethers. It picks individuals from the present people continually at sporadic, shapes a requirement and the best individual of a party wins the resistance and is put into the mating pool for recombination. This framework is repeated the measure of times key to finish the pined for size of midway masses. The resistance review controls the decision quality. The more essential the control gage, the more grounded is the confirmation approach.

2.3.2. Elitist Affirmation

Keeping an eye on an irrefutable objective to guarantee that the best individuals of the methodology are passed to further periods, and should not be lost in discretionary affirmation, this decision administrator is used. So, we used a couple best chromosomes from each time, in setting of the higher accomplishment regard and are surrendered to the end coming time of people.

2.3.3. Duplication

To make a minute time span people of courses of action from those picked through trademark experts: mix (additionally called recombination), and what's more change. For each new response for be made, a couple "parent" charts is decided for rising from the pool picked early. By passing on a "tyke" methodology using the above structures for cross breed and change, another system is made which regularly shares limitless attributes of its "family". Inexperienced gatekeepers are decided for each new tyke, and the method continues until another mass of structures of fitting size is passed on. Notwithstanding the way that change systems that rely on upon the usage of two gatekeepers are more "science animated", some examination proposes more than two "watchmen" are perfect to be used to duplicate a respectable quality chromosome.

These structures finally result in the bleeding edge people of chromosomes that are not the same as the secured period. All around the general achievement will have related by this framework for the masses, since in a general sense the best life shapes from the first are decided for raising, nearby genuinely level of less fit rationalities, for reasons starting at now said above. Disregarding the way that Cream and Change are known as the fundamental common specialists, it is possible to use moving heads, for instance, regrouping, colonization-end, or change in characteristic numbers.

2.3.4. End

This generational system is stressed until an end condition has been come to. Standard peak conditions are: (1) An answer is found that satisfies scarcest criteria; (2) Settled number of periods completed; (3) Scattered

DOI: 10.5281/zenodo.3715417

spending technique (estimation time/money) accomplished; (4) The most raised overseeing structure succeeding is coming to or has satisfied a level with the veritable focus on that uncommon emphases at no time later on invigorate happens obviously; (5) Manual outline; last however not the base (6) Blends of the above.

3. PREPARING THE TEXT DATA

We divided the downloaded Euron-spam corpus (Data Set) containing 33716 emails in 6 directories. Each of 6 directories contains 'Ham' & 'Spam' folders. Total number of non-spam emails and spam emails are 16545 and 17171 respectively. This corpus is divided into training set and test set in 60:40 split respectively. In any text-mining problem, text cleaning is the first step where we remove those words from the document, which may not contribute to the information we want to extract. Emails may contain a lot of undesirable characters like punctuation marks, stop words, digits, etc. which may not be helpful in detecting the spam email. The emails in Euron-spam corpus have been processed in the following ways:

a) Removal of stop words – Stop words like "and", "the", "of", etc. are very common in all English sentences and are not very meaningful in deciding spam or legitimate status, so these words have been removed from the emails.

b) Lemmatization – It is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. For example, "include", "includes," and "included" would all be represented as "include". The context of the sentence is also preserved in lemmatization as opposed to stemming (another buzz word in text mining which does not consider meaning of the sentence).

```
[('order', 1414), ('address', 1293), ('report', 1216), ('mail', 1127), ('send', 1079), ('language', 1072), ('email', 1051),
('program', 1001), ('our', 987), ('list', 935), ('one', 917), ('name', 878), ('receive', 826), ('money', 788), ('free', 762)
```

4. PREPARING CREATING WORD DICTIONARY

A sample email in the data-set looks like this:

```
Subject: posting

hi, ' m work phonetics project modern irish ' m hard source. anyone recommend book article english? ',
specifically, interest palatal (slender) consonant, work helpful too. thank! laurel Sutton (sutton @ garnet. berkeley.
edu
```

The first line of the mail is subject and the 3rd line contains the body of the email. We only performed text analytics on the content to detect the spam mails. As a first step, we created a dictionary of words and their frequency. For this task, training set of 20229 mails is utilized. This python function creates the dictionary –

Once the dictionary is created we added just a few lines of code written below to the above function to remove. We have also removed absurd single characters in the dictionary which are irrelevant here. We found some absurd word counts to be high. Our dictionary has some of the entries given in next page as most frequent words. Here we have chosen 3000 most frequently used words in the dictionary.

5. FEATURE EXTRACTION PROCESS

DOI: 10.5281/zenodo.3715417

We have extracted **word count vector** (our feature here) of 3000 dimensions for each email of training set. Each **word count vector** contains the frequency of 3000 words in the training file. The below python code generates a feature vector matrix whose rows denote 20229 files of training set and columns denote 3000 words of dictionary. The value at index 'ij' is the number of occurrences of j^{th} word of dictionary in i^{th} file.

```
def extract_features(mail_dir):

files = [os.path.join(mail_dir,fi) for fi in os.listdir(mail_dir)]

   features_matrix = np.zeros((len(files),3000))

   docID = 0;

   for fil in files:

     with open(fil) as fi:

       for i,line in enumerate(fi):

         if i == 2:

           words = line.split()

           for word in words:

             wordID = 0

                  for i,d in enumerate(dictionary):

               if d[0] == word:

                 wordID = i

                 features_matrix[docID,wordID] = words.count(word)

       docID = docID + 1

   return features_matrix
```

6. FEATURE TRAINING THE CLASSIFIERS

Here, we have used **scikit-learn ML library** for training classifiers. We have trained three models here namely Linear SVC, Naive Bayes and K-Neighbors Classifier. The decision function of SVM model that predicts the class of the test data is based on support vectors and makes use of a kernel trick.

DOI: 10.5281/zenodo.3715417

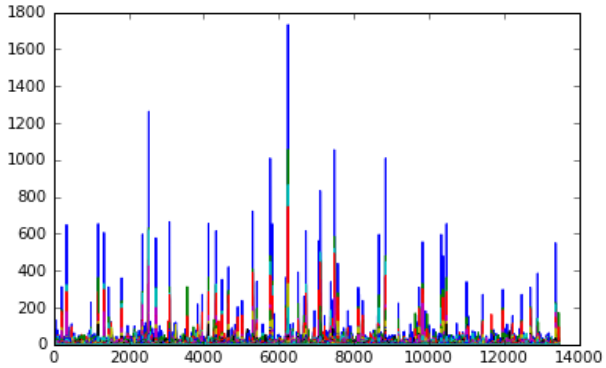
Once the classifiers are trained, we checked the performance of the models on test-set. We have extracted word count vector for each mail in test-set and predicted its class (ham or spam) with the trained NB classifier, SVM model and KNN. Below is the full code for spam filtering project. We have included the three functions that have been defined before.

```
import os
import numpy as np
from collections import Counter
from sklearn.naive_bayes import MultinomialNB, GaussianNB, BernoulliNB
from sklearn.svm import SVC, NuSVC, LinearSVC


from sklearn.metrics import confusion_matrix

train_dir = 'train-mails'

dictionary = make_Dictionary(train_dir)

train_labels = np.zeros(702)

train_labels[351:701] = 1

train_matrix = extract_features(train_dir)

model1 = MultinomialNB()
model2 = LinearSVC()

model1.fit(train_matrix,train_labels)
model2.fit(train_matrix,train_labels)

test_dir = 'test-mails'
test_matrix = extract_features(test_dir)
test_labels = np.zeros(260)
test_labels[130:260] = 1

result1 = model1.predict(test_matrix)
result2 = model2.predict(test_matrix)
print confusion_matrix(test_labels,result1)
print confusion_matrix(test_labels,result2)
```

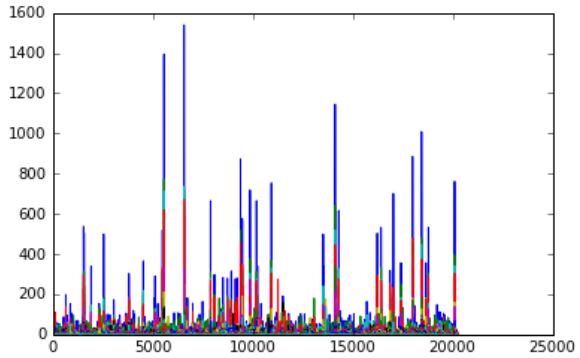
7. RESULTS

The results of running the same dataset in different implemented algorithm is shown below –

DOI: 10.5281/zenodo.3715417



Graph for Test Set - This is the Graph by plotting the test dataset we have taken.



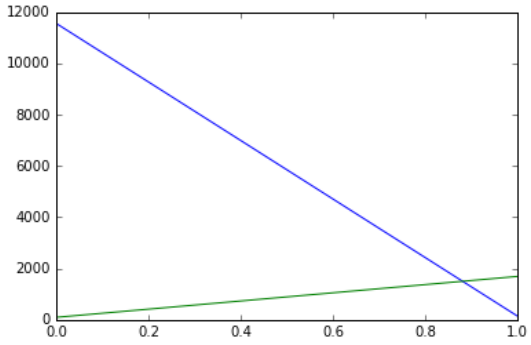
Graph for Train Set - This is the Graph by plotting the training dataset we have taken.

1. **Linear SVC –** The results for Linear SVC algorithm is shown below **–**

Confusion Matrix –

	Ham	**Spam**
Ham	**11553**	**95**
Spam	**150**	**1689**

Resulting Graph –



Accuracy Achieved - 98.18%

False Positive – 1.8%

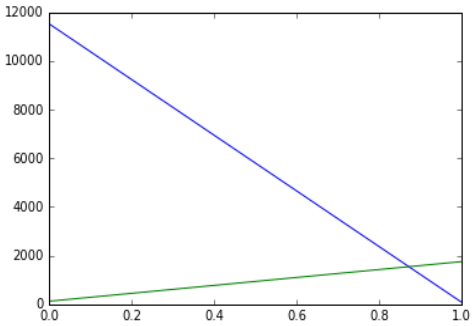
1. **Multinomial NB -**The results for Multinomial Naïve Bayes algorithm is shown below-

Confusion Matrix –

DOI: 10.5281/zenodo.3715417

	Ham	**Spam**
Ham	**11524**	**124**
Spam	**88**	**1751**

Resulting Graph –



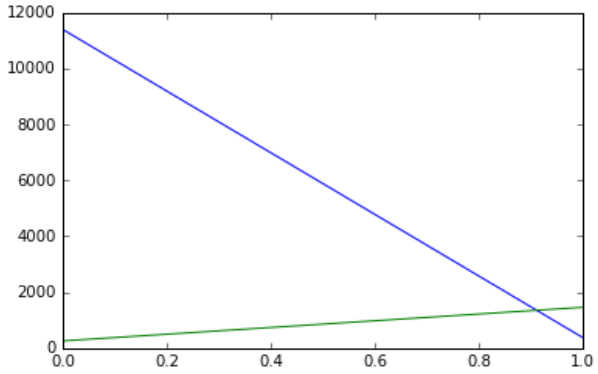
Accuracy Achieved – 98.42%
False Positive – 1.5%

2. **K-Nearest Neighbors -** The results for k-NN algorithm is shown below –

Confusion Matrix –

	Ham	**Spam**
Ham	**11391**	**257**
Spam	**381**	**1458**

Resulting Graph –



Accuracy Achieved – 95.26%
False Positive – 4.7%

8. SUMMARY AND CONCLUSION

There are many ways to filter Internet spam. Considering the daily growth of spam and spammers, it is essential to provide effective mechanisms and to develop efficient software packages to manage spam. Using valid emails and spam the present study extracted data from emails using machine learning algorithms to develop a new model. Measuring the rate of various classes of valid emails and running on test data, the Naive Bayes model demonstrated higher efficiency than other two classifier algorithms and with a low rate of false positive. The proposed algorithm can be modelled to be implemented on a Mail Server and Mail

DOI: 10.5281/zenodo.3715417

Client in order to eliminate problems, such as bandwidth reduction and very low efficiency, from which users usually suffer.

References

[1] Nadji, Y., Antonakakis, M., Perdisci, R., Dagon, D., & Lee, W. (2013, November). Beheading hydras: performing effective botnet takedowns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 121-132). ACM.

[2] Cho, C. Y., Caballero, J., Grier, C., Paxson, V., & Song, D. (2010). Insights from the Inside: A View of Botnet Management from Infiltration. *LEET*, *10*, 1-1.

[3] Dittrich, D. (2012, April). So You Want to Take Over a Botnet... In *LEET*.

[4] Goodman, N. (2017). A Survey of Advances in Botnet Technologies. *arXiv preprint arXiv:1702.01132*.

[5] Schiavoni, S., Maggi, F., Cavallaro, L., & Zanero, S. (2014, July). Phoenix: DGA-based botnet tracking and intelligence. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 192-211). Springer, Cham.

[6] Dagon, D. (2005, July). Botnet detection and response. In *OARC workshop* (Vol. 2005).

[7] Micro, T. (2006). Taxonomy of botnet threats. *Whitepaper, November*.

[8] Dagon, D., Gu, G., Lee, C. P., & Lee, W. (2007, December). A taxonomy of botnet structures. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual* (pp. 325-339). IEEE.

[9][Nazario, J. (2008). Bot and botnet taxonomy. *Computer Security Institute. Computer Security Institute Security Exchange*.

[10] Al-Jarrah, O. Y., Alhussein, O., Yoo, P. D., Muhaidat, S., Taha, K., & Kim, K. (2016). Data randomization and cluster-based partitioning for botnet intrusion detection. *IEEE transactions on cybernetics*, *46*(8), 1796-1806.

[11] Plohmann, D., Gerhards-Padilla, E., &Leder, F. (2011). Botnets: Detection, measurement, disinfection &defence. *European Network and Information Security Agency (ENISA)*, *1*(1), 1-153.

[12] Khattak, S., Ramay, N. R., Khan, K. R., Syed, A. A., &Khayam, S. A. (2014). A taxonomy of botnet behavior, detection, and defense. *IEEE communications surveys & tutorials*, *16*(2), 898-924.

[13] Anagnostopoulos, M., Kambourakis, G., &Gritzalis, S. (2016). New facets of mobile botnet: architecture and evaluation. *International Journal of Information Security*, *15*(5), 455-473.

[14] Kwon, J., Lee, J., Lee, H., &Perrig, A. (2016). PsyBoG: a scalable botnet detection method for large-scale DNS traffic. *Computer Networks*, *97*, 48-73.

[15] Maratea, A., Petrosino, A., & Manzo, M. (2014). Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, *257*, 331-341.

[16] Ng, A. (2011). Advice for applying machine learning.

[17] The CAIDA UCSD Dataset 2008-11-21, 2008. https://data.caida.org/datasets/security/telescope-3days-conficker/

[18] Singh, K., Guntuku, S. C., Thakur, A., &Hota, C. (2014). Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences*, *278*, 488-497.

[19] Abu Rajab, M., Zarfoss, J., Monrose, F., & Terzis, A. (2006, October). A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (pp. 41-52). ACM.

DOI: 10.5281/zenodo.3715417

[20] Feily, M., Shahrestani, A., &Ramadass, S. (2009, June). A survey of botnet and botnet detection. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on* (pp. 268-273). IEEE.

DOI: 10.5281/zenodo.3715417