INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 3)
Available online at: www.ijariit.com

Survey of machine learning methods for spam e-mail classification

Sanjana Reddy
sanjanasr.cs18@rvce.edu.in
R. V. College of Engineering, Bengaluru, Karnataka

Navya Priya N
navyapriyan.cs18@rvce.edu.in
R. V. College of Engineering, Bengaluru, Karnataka

Varsha R Jenni
varsharjenni.cs18@rvce.edu.in
R. V. College of Engineering, Bengaluru, Karnatak

ABSTRACT

The humongous volume of unsolicited bulk e-mail (spam) which is further increasing, is the major cause for developing anti-spam protection filters. Machine learning provides a very optimized approach to automatically filter spams at a very successful rate. Here, in this paper we survey some of the most popular machine learning algorithms (Naïve Bayes, k-NN, SVMs and ANN) and their applicability to the problem of spam e-mail classification. Descriptions of the algorithms are presented, and the comparison of their performance on the UCI spam-base dataset is presented.

Keywords— *Spam, E-mail classification, Machine learning algorithms, k-NN, SVM, Naïve Bayes, ANN.*

1. INTRODUCTION

The spam e-mails that we get these days are causing a serious issue and they can't be neglected. They constitute upto 75-80% of the total number of e-mail messages. More precisely, spams can cause direct financial losses, misuse of traffic, storage space and computational power. Spam makes users look through and sort out additional e-mail, not only wasting their time and causing loss of work productivity and violating their privacy rights. In 2005, according to Ferris Research Analyzer Information Service, the worldwide financial loss was close to $50 billion caused primarily by spams.

The two mostly used approaches for e-mail spam filtering are knowledge engineering and machine learning. In knowledge engineering, a set of rules has to be defined according to which e-mails are labelled as spam or ham. These rules have to be defined by the user who uses the filter or some other authority (example: a company which provides rule-based spam filtering tool). The results are not very promising, as the rules should be constantly updated and maintained, which causes inconvenience to the users.[1] Machine learning is more efficient when compared to knowledge engineering. The results are not based on any set of rules. Hence it is more optimised unlike knowledge engineering. Here, a set of pre-classified and pre-processed e-mail messages (training samples) and a specific algorithm is used to learn the classification rules from these training samples. A lot of machine learning algorithms provide an efficient way to filter out spams, to name a few- Naïve Bayes, Support Vector Machine, k-Nearest Neighbours, Artificial Neural Networks etc. [1]

2. LITERATURE SURVEY

There is a rapid increase in the interests toward e-mail spam filtering. Many researchers and data scientists have presented their methods for e-mail spam classification. Leug [2] presented a detailed survey to find whether information filtering and retrieval could help in improving the efficiency of the model in detecting spam. However, the paper did not present the details of various machine learning algorithms that can be used. Wang [3] surveyed the various methods that could be adopted to identify unsorted spams. The paper also presented ways to categorize spam e-mails into different hierarchical folders and also regulate tasks to reply to the appropriate e-mails.

Sanz, Hidalgo, and Pérez [4] detailed the research issues associated with e-mail spams, in what way it affects users, and by what means users and providers can reduce its effects. The paper also enumerates the legal, economic, and technical measures to mediate the e-mail spams. They acknowledged that supported technical measures, content analysis filters are extensively used and proved to possess a reasonable percentage of accuracy and precision as a result, the review focused more on them, detailing how they work. The research work explained the organization and therefore the procedure of the many machine learning approaches utilized for the aim of filtering e-mail spams. However, the review didn't cover recent research articles during this area because it was published in 2008 and comparative analysis of the various content filters was also missing. A quick study on e-mail image spam filtering methods was presented by [5]. The study targeting e-mail anti spam filtering approaches to transfer from text-based techniques to image-based methods. Spam and therefore the spam filters are premeditated to reducing it have spawned an upsurge in creativeness and inventions. However, the study didn't cover machine learning techniques, simulation tools, dataset corpus, and therefore the architecture of e-mail spam filtering techniques.

3. MACHINE LEARNING IN E-MAIL CLASSIFICATION

E-mail spam filtering is a type of unsupervised learning, where features like a bunch of words or subject line analysis can help us label the e-mails. The input to e-mail classification task can be considered as a matrix(2-D) whose axes are features and messages(mails). The main classification task can be divided into subtasks. Data collection and representation which are problem specific come under the first subtask and e-mail feature selection and an attempt to reduce features in order to reduce the dimensionality (i.e. the number of features) for the steps remaining, comes under the second subtask. Finally, in the last phase(classification) of the process there is an actual mapping done between training set and testing set. In the following section we will review some of the most popular machine learning methods.[1]

3.1 Dataset

We use the spam-base UCI dataset to evaluate the algorithms described here. It contains a set of spam and non-spam emails. This dataset consists of 4,601 instances each with 57 attributes as shown in Table 1. The last column of this database denotes whether the e-mail is spam or not. Most of the attributes indicate whether a particular word or character is frequently occurring in the e-mail or not. The run length attributes measure the length of sequences of consecutive capital letters. Pre-processing is one of the major tasks for data. In this dataset, data normalization is done before performing any further processing. Data cleaning, integration, transformation and reduction are those major tasks in pre-processing. [6]

Table 1: Dataset description

Attributes	Data type	Interval	Description
48	Continuous real	[0,100]	Percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail.
6	Continuous real	[0,100]	Percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurrences) / total characters in e-mail
1	Continuous real	-	Average length of uninterrupted sequences of capital letters
1	Continuous integer	-	Length of longest uninterrupted sequence of capital letters
1	Continuous integer	-	Sum of length of uninterrupted sequences of capital letters: total number of capital letters in the e-mail
1	Nominal	{0,1}	Denotes whether the e-mail was considered spam (1) or not (0)

3.2 SVM

A support vector machine is most widely used for binary classification of data. SVM classifies the data by forming the most accurate hyper plane between the two classes. The hyperplane should be at the maximum distance from both classes which allows us to classify the new data with confidence. Hyperplanes are decision boundaries that help in classifying the data. The dimension of the decision boundary depends on the number of features of the dataset. The points closer to the hyperplane affects the position of the dataset to a greater extent. With the help of these support vectors, we build our large margin hyperplane.[7]

3.2.1 Implementation: There are two types of SVM Model

a. Linear: Linear SVM is used when the two classes in the dataset are linearly separable.
b. Non-Linear: Non-Linear SVM is used when the two classes in the dataset are not linearly separable. The different types of non-linear kernels are:
   - Polykernel
   - Sigmoid
   - Gaussian

Algorithm of SVM is as follows:
Step 1: Convert the data sample into a vector
Step 2: Divide the dataset for training and testing
Step 3: Train the data by building the large margin hyperplane (decision boundary)
Step 4: Test the model built using the testing det of the data

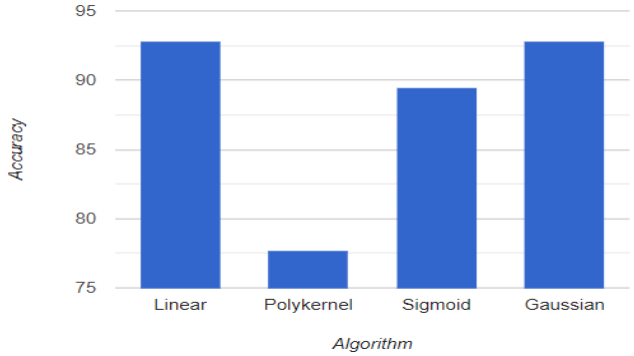

Fig 1: SVM Algorithms and their accuracy

Table 2: SVM Algorithm and their accuracy

Algorithm	Accuracy (%)
Linear	92.8
Polykernel	77.7
Sigmoid	89.5
Gaussian	92.8

From the above Figure 1 and Table 2, it is clear that Linear and Gaussian give the highest accuracy of 92.8%. The type of kernel to be used depends on the nature of the dataset and for spam dataset these two kernels work the best.[8]

3.2 Naïve Bayes

The Naïve Bayes algorithm uses the Bayes Rule, as shown in Equation 1, for classification. This algorithm classifies each object by individually looking at each of its features. Here the object is an e-mail and the features are the unique words in the e-mail. The posterior probability is calculated for each unique word in the e-mail. These probabilities are multiplied to get a final probability for both the classes. The class that has the highest probability will determine which class the object belongs to.[9]

$$P(A|B)=(P(B|A)*P(A))/P(B) \quad (1)$$

A and B are events
P(B) is not equal to zero.
P(A|B) is a conditional probability: the likelihood of an event occurring given that is true.
P(B|A) is a conditional probability: the likelihood of an event occurring given that is true.
P(A) and **P(B)** are the marginal probability.[10]

Applying Bayes rule in spam filtering, as shown in Equation 2 and 3:

$$P(SPAM|WORD) = (P(WORD|SPAM) * P(SPAM))/P(WORD) \quad (2)$$

$$P(WORD) = (P(WORD|SPAM) * P(SPAM) + (P(WORD|NON\text{-}SPAM) * P(NON\text{-}SPAM) \quad (3)$$

3.2.1 Implementation: There are three types of Naïve Bayes Model:
a. Multinomial Naïve Bayes is applied when the features have discrete frequency counts.
b. Bernoulli Naïve Bayes is used when the dataset has binary features and predictions are done with the binary features.
c. Gaussian Naïve Bayes is applied if the dataset features are normally distributed. [10]

Algorithm of Naïve Bayes is as follows:
Step 1: Data set is converted into a frequency table. Here we use the UCI spam-base dataset which has already been pre-processed. Data has undergone data normalization, data cleaning, integration, transformation and reduction.
Step 2: Prepare a Likelihood table after finding the corresponding probabilities.
Step 3: Use Naïve Bayesian equations to calculate the posterior probability for each class. The class that has the highest posterior probability is the predicted class.

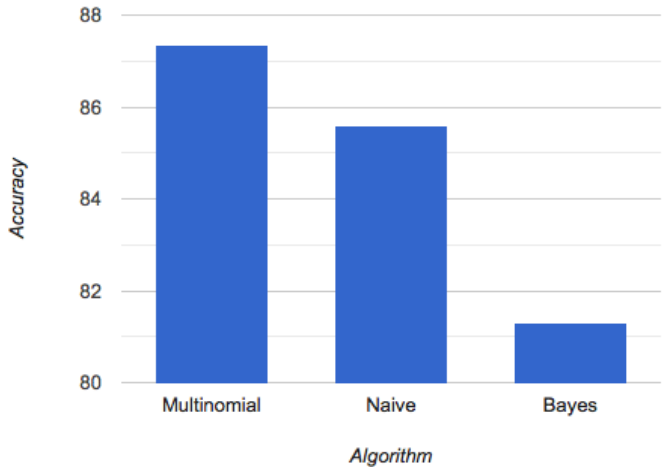

Fig. 2: Naïve Bayes Algorithm and their accuracy

Table 3: Naïve Bayes Algorithm and their accuracy

Algorithm	Accuracy (%)
Multinomial Naïve Bayes	87.36%
Bernoulli Naïve Bayes	85.58%
Gaussian Naïve Bayes	81.30%

It can be seen from Figure 2 and Table 3 that Multinomial Naïve Bayes achieves the highest accuracy. Although Multinomial Naïve Bayes provides greater accuracy but the difference in accuracies is not very significant as Bernoulli Naïve Bayes and Gaussian

Naïve Bayes. All of these algorithms are used for document classification but considerably differ in their approaches to classify the text. [11]

3.3 K-Nearest Neighbor(k-NN)

k-Nearest Neighbor algorithm is a clustering technique, where groups of patterns are classified into related classes called clusters [7]. This algorithm is an example based classifier where the training samples are used for mere comparison and not to form explicit categories. Therefore, the classification model is not built from data, rather classification is carried out by matching the test instance with K training examples and decision is made as to which group it belongs to depending on the resemblance to k closest neighbors. The justification for using nearest neighbours is best exemplified by the following: "If it walks like a duck, quacks like a duck, and looks like a duck, then it is most likely a duck".

The Euclidean distance between the feature vectors of the messages can be used to figure out how 'close' the messages are to each other. Based on how 'close' they are, we try to classify a message according to the classes of its nearest neighbours in the training set.[12]

3.3.1 Implementation:

Training: Store the training messages. Rename the columns and change the original data frame only if the particular columns are true. Pre-process the data into usable data-frames.
Classification: The Euclidean distance between each test row and train row is obtained, after which k neighbors are returned. Now, if at least t messages in k neighbors of the message m are unsolicited, then m is an unsolicited e-mail, otherwise, it is legitimate.[13]

The algorithm of k-NN is as follows:

Step 1: Import the dataset and adjust the columns as necessary
Step 2: Preprocess the data into usable data frames and split the dataset into testing(D) and training dataset(M)(here it is split into half).
Step 3: The number of nearest neighbours(k) is taken, here it is 10 and all other testing(x_test and y_test) and training variables(x_train and y_train) are initialised.
Step 4: The data is classified using distance calculations and by employing a majority vote amongst k-nearest-neighbors.
Step 5: The euclidean distance function computes the euclidean distance , these distances between each test row again every train row helps to compute k nearest neighbours.
Step 6: for each d in D and each m in M do
Step 7: Neighbors(d) = {}
Step 8: if |Neighbors (d) | < k then
Step 9: Neighbors(d) = Closest (d, m)∪ Neighbors(d)
Step 10: end if
Step 11: if |Neighbors(d)| ≥ k then
Step 12: restrain(M, xj, yj)
Step 13: end if
Step 14: end for
Step 15: Count of the occurrences of the labels(t) is recorded, the maximum count is retrieved.
Step 16: The key associated with the maximum value is the predicted classification(ham or spam).

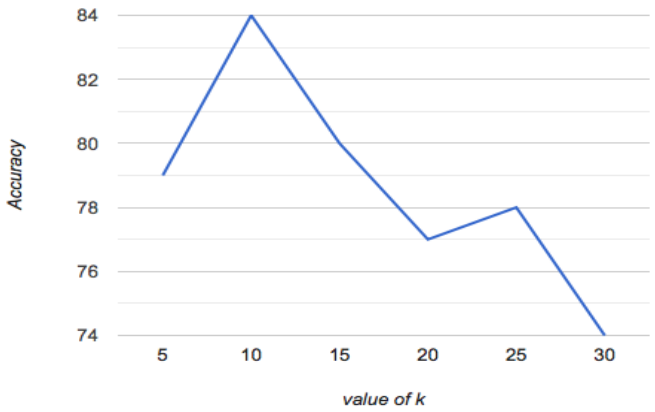

Fig. 3: value of k and the corresponding accuracy

Table 4: Value of k and the corresponding accuracy

Value of k	Accuracy
5	79%
10	84%
15	80%
20	77%
25	78%
30	74%

On implementing the model, the accuracy varied on the value of k(number of nearest neighbours) as shown in Fig 3 and Table 4. When k was 10, an accuracy of about 84% was achieved and an error rate of 16%, which is quite a promising result. When k is taken to be 5, accuracy of about 79% was achieved and when k is 20 , accuracy of 77% was achieved. On an average accuracy of 79% was achieved. The above results show that an optimum value of k gives a good accuracy. Hence, choosing the value of k also, is quite a task.[13]

3.4 Artificial Neural Network

Artificial neural networks (ANN) are non-linear statistical data modeling tools that try to simulate the functions of biological neural networks. It consists of interconnected collections of artificial neurons or perceptrons that process information.[14]

The idea of the artificial neuron is to find a linear function of the feature vector as shown in Eqn 4 such that it is greater than zero for vectors of one class, and less than zero for vectors of other class. If we denote the classes by numbers +1 and -1, we can state that we search for a decision function given by Eqn 5. The perceptron learning is done with an iterative algorithm. It begins with randomly chosen parameters (w0,b0) and updates them in every iteration. On one of the iterations, say the n-th iteration, of the algorithm a training sample (x,c) is chosen such that it's decision does not classify it correctly. The parameters (wn,bn) are then updated using Eqn 6 and 7.[15]

$$f(x) = wT\ x + b. \quad (4)$$

Where w = (w1 w2,...wm) , b=bias

$$d(x) = \text{sign}\ (wT\ x + b) \quad (5)$$

$$w(n+1) = w\ (n + cx) \quad (6)$$

$$b(n+1) = b(n + c) \quad (7)$$

3.4.1 Implementation: The algorithm of Artificial Neural Networks is as follows:

Training:

Step 1: Initialize w and b (to random values or to 0).
Step 2: Find a training example (x,c) for which sign (wT x + b).
Step 3: If there is no such example, then training is completed, store the final w and stop.
Step 4: Else go to the next step.
Step 5: Update (w,b): w := w + cx, b := b + c. Go to the previous step.

Filtering:

Step 6: Given a message x, determine its class as sign (wT x + b)

The network created is a 3-layer neural network for spam detection:

1. The first layer is the input layer which has 57 nodes, 1 node for each feature of the e-mail
2. The second layer is the middle layer which has 4 node.
3. The final layer is the output layer which has just 1 node.

The input layer takes in the 57 features of the e-mail as a vector and passes it to the middle layer. Finally, the output layer outputs a real number in the interval (0, 1) which determines whether the mail is spam or not.[16]

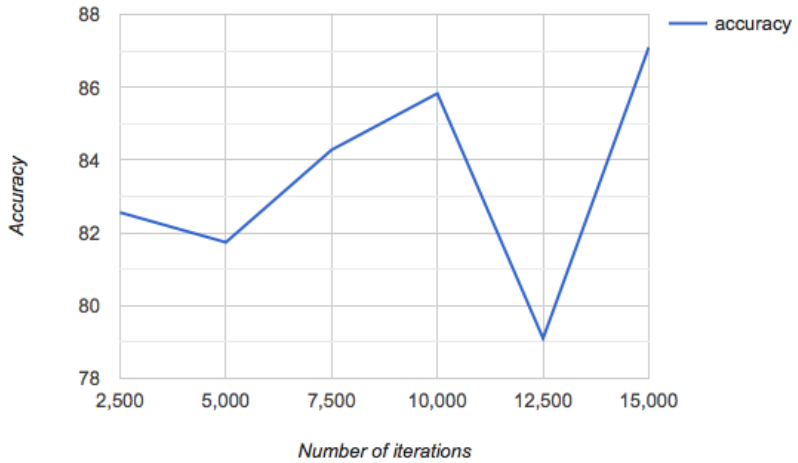

Fig. 4: Number of iterations and corresponding accuracy

Table 5: Number of iterations and corresponding accuracy

Number of Iterations	Accuracy
2500	82.56%
5000	81.74%
7500	84.28%
10000	85.83%
12500	79.10%
15000	87.10%

It can be seen that an accuracy of 87% can be achieved with 15000 iterations. Artificial Neural Networks give good results with lesser number of iterations as well as shown in Figure 4 and Table 5. Accuracy can be increased but it comes at a cost of increasing the number of iterations which is time consuming.[16]

4. PERFORMANCE COMPARISON

We summarize the performance result of four algorithms in terms of accuracy. Table 6 and Figure 5 summarize the results of the four classifiers on the UCI spambase dataset. In terms of accuracy we can find that the SVM method is the most accurate while the Naïve Bayes and Artificial Neural Network give us approximately the same lower percentage. k-NN method gives us the lowest accuracy.[1]

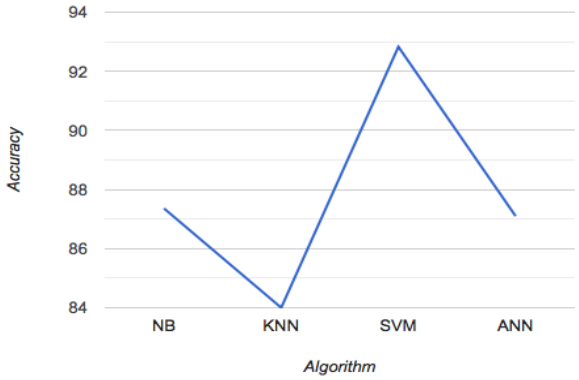

Fig. 5: Algorithm versus accuracy

Table 6: Algorithm and their corresponding accuracy

Algorithm	Accuracy
Naïve Bayes	87.36%
KNN	84%
SVM	92.83%
ANN	87.10%

Table 7: Classification principle, advantage and disadvantage of the algorithms

S.no	Algorithm	Classification principle	Advantages	Disadvantages
1.	Naive Bayes	Works on Bayes theorem	It has high accuracy and speed when used for large datasets.	Assumption is made that events occurring are mutually exclusive.
2.	Support Vector Machine	Non-linear mapping	Highly efficient and accurate classifier, less prone to overfitting.	Complex algorithm, difficult to understand. Training time is more.
3.	k-Nearest Neighbors	Learning by analogy and distance-based comparison	Less work on training the data sets but more work on classification.	Computationally expensive. Requires efficient storage techniques.
4.	Artificial Neural Networks	Works by finding a linear function of the feature vector.	Requires fewer input features to achieve the same results produced by other classifiers, medium for fine-tuning weights of the connections available.	Not commonly used because research needs to be done for the network configuration, momentum, and learning rate. Supervision needed to enhance precision performance.

In Table 7, classification principles of each classification technique are highlighted with their advantages and disadvantages.

5. CONCLUSION

In this paper we review some of the most popular machine learning algorithms and their application in filtering spam in e-mails. Here, the algorithms are described and comparison of their performance on the UCI spambase dataset is presented. The table gives the gist of all the algorithms described. It can be seen that the SVM algorithm has a very satisfying performance with minimal cons whereas k-NN, though universally consistent doesn't give much accuracy and is time consuming. Hence, on further research, there can be escalation with respect to the performances of all the algorithms. [1]

6. REFERENCES

[1] Machine Learning Methods For Spam E-Mail Classification W.A. Awad and S.M. ELseuofi, International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011

[2] C.P. Lueg, From spam filtering to information retrieval and back: seeking conceptual foundations for spam filtering, Proc. Assoc. Inf. Sci. Technol., 42 (1) (2005)

[3] X.L. Wang, Learning to classify e-mail: a survey, 2005 International Conference on Machine Learning and Cybernetics (Vol. 9, pp. 5716-5719), IEEE (Aug 2005)
[4] E.P. Sanz, J.M.G. Hidalgo, J.C.C. Péreze-mail spam filtering Adv. Comput., 74 (2008), pp. 45-114
[5] S. Dhanaraj, V. Karthikeyani, A study on e-mail image spam filtering techniques, Paper presented at the International Conference on Pattern Recognition, Informatics and Mobile, Engineering (PRIME) (2013)
[6] Ham and Spam E-Mails Classification Using Machine Learning Techniques by Mahmoud Bassiouni, Mayar Aly Shafaey ,El-Sayed A. El-Dahshan,Journal of Applied Security Research
[7] Machine learning for email spam filtering: review, approaches and open research problems, by Emmanuel Gbenga Dadaa, Joseph Stephen Bassia, Haruna Chiromab, Shafi'i Muhammad Abdulhamidc ,Adebayo Olusola Adetunmbi ,Opeyemi Emmanuel Ajibuwae.
[8] https://github.com/subhinA/classify-emails-as-spam-or-ham
[9] Spam Mail Detection Using Artificial Neural Network and Bayesian Filter by Levent Özgür, Tunga Güngör, and Fikret Gürgen
[10] Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification,by Gurinder Singh,Bhawna Kumar,Loveleen Gaur,Akriti Tyagi,2019 International Conference on Automation, Computational and Technology Management (ICACTM),Amity University
[11] https://github.com/78526Nasir/Spam-Detection-Using-Different-Naive-Bayes-Classifier
[12] Romero, C., Garcia Valdez, M., & Alanis, A. (2010). A comparative study of machine learning techniques in blog comments spam filtering. The 2010 International Joint Conference on Neural Networks (IJCNN).
[13] https://github.com/stanley-c-yu/k-nearest-neighbors
[14] E-mail Spam Classification With Artificial Neural Network and Negative Selection Algorithm by Ismaila Idris.
[15] Spam Detection using a Neural Network Classifier by David Ndumiyana, Munyaradzi Magomelo,Lucy Charity Sakala.
[16] Spam detection using neural networks in Python by Aman Goel, Medium article.

