Supervised Machine Learning techniques for Spam Email Detection

Dr. Swapna Borde[1], Utkarsh M. Agrawal[2], Viraj S. Bilay[3], Nilesh M. Dogra[4]
[1, 2, 3, 4] Department Of Computer Engineering,
[1, 2, 3, 4] Vidyavardhini's College of Engineering and Technology, Vasai, Palghar, India.

Abstract- *Unsolicited e-mail (Spam) has become a major issue nowadays for each e-mail user. Through email, companies and individuals send advertisements for various products, undesirable harmful news, and contents, and fake proposals etc. The spam emails result in unnecessary consumption of network bandwidth resulting blocking email servers. Traditional systems make it very difficult to detect spam as these emails are written or generated in a very special way so that anti-spam filters cannot detect such emails. This paper proposes a system that uses supervised machine learning algorithms to detect and filter spam develop a classification model that is trained using Enron dataset and will able to predict whether an e-mail is spam or not.*

Keywords- Classification, Machine Learning Algorithms, Spam-Email, Filtering, Supervised Learning.

I. INTRODUCTION

In this digital age, is the time of computers, one of the well-organized and easier modes of communication is the email. Reading an email is becoming a regular habit of many people. This is an efficient, fast and cheaper means of communication. Email formulates it desired both in professional and personal associations [1]. One of the major issues for any category of users of email and Internet is receiving spam email.

Through email, companies and individuals send advertisements for various products, undesirable harmful news, and contents, and fake proposals etc. These spam emails irritate email users and waste their precision time. For non-serious and nontechnology savvy users, these emails create big problems as the users get misguided by these emails. The spam emails result in unnecessary consumption of network bandwidth resulting blocking email servers. In order to address this growing problem, each organization must analyze the tools available to determine how best to counter spam in its environment. Tools, such as the corporate e-mail system, e-mail filtering gateways, blacklist filter, greylist, contracted zanti-spam services, and end-user training and many other techniques provide an important arsenal for any organization. However, users cannot avoid the very serious problem of attempting to deal with large amounts of spam (bulk email) on a regular basis. If this problem is not tackled and there are no anti-spam activities, spam will inundate network systems, hinder employee productivity, steal bandwidth, and still be there tomorrow. The difficulty of undesired electronic messages is nowadays a serious issue, as spam makes up 75-80% of total amount of emails. The spam causes several problems may result in direct financial losses and also causes misuse of traffic.

In order to address this issue, a significant research on anti-spam techniques has been taken place and various kinds of anti-spam software have been developed and used by email users. Spam filter techniques include both manual and automatic methods. In manual methods, negative lists of spammers, list of authentic senders, and selected list of words in email content or subject are considered for developing anti-spam filter. In recent years, machine learning technique, a better technique compare to manual methods, is used to detect and classify spam emails automatically [2].

In supervised or inductive machine learning, the algorithms learn from the training dataset that contains both inputs and outputs (results) and a model is created. The model is then tested for new samples for classification. In case of binary classification, the output belongs to two classes. In recent days e-mail spam filtering is one of the important research field.

II. RELATED WORK

In machine learning, a set of rules is created according to which messages are then categorized as spam or legitimate mail (ham). A set of rules should be created either by the user of the filter, or by some other authority (e.g. A software company that provides a particular rule-based spam-filtering tool). The major drawback of this method is that the set of rules that are defined must be constantly updated, and maintaining it is not convenient for most users. The rules could, of course, be updated either in a centralized manner by the maintainer of the spam filtering tool, or there is even a peer-2-peer knowledgebase solution, but when the rules are publicly available, the spammer can adjust the text of his

message so that it would pass the filter. Therefore it is better when spam filtering is customized on a per-user basis. The machine learning approach does not require specifying any rules explicitly. Instead, a set of pre-classified samples is needed. Then any appropriate algorithm is used to "learn" the classification rules from this data.The subject of machine learning has been widely studied and there are lots of algorithms suitable for this task [3].

Over the period of time several machine learning techniques such as neural network, Bayes algorithm, SVM, lazy algorithms, decision trees and artificial-immune systems etc. have been used in classifying spam email datasets. All these techniques use different approaches to solve the problem such as Neural Net, where it tries to model the data similar to human brain processing information. The model is built and applied with minimum statistical or mathematical knowledge. The model implicitly learns the linear or non-linear mapping from the given input to the object values using backpropagation algorithm. It provides a guaranteed local minima and has excellent representation power of various functions [2].

In neural networks, neural net is applied on dataset using algorithms such as Perceptron or Back-propagation algorithm. In Perceptron algorithm 'learning' process is performed by using binary classifiers (functions that can decide whether an input in the form vector, belongs to some specific class or not). It is a type of linear classifier, i.e. a classification algorithm that makes its predictions that are based on a linear predictor function and combines a set of weights with the feature vector. The algorithm processes elements in the training set one at a time. In context of neural networks, a perceptron is an artificial neuron that uses the unit step function as its activation function. As a linear classifier, the simplest feedforward neural network is the single-layer perceptron. Whereas in Back Propagation the simplest feedforward is an expression for the partial derivative of the cost function with respect to any weight (or bias) in the network. The expression tells us how quickly the cost changes when we change the weights and biases. And while the expression is somewhat complex, with each element having a natural, intuitive interpretation. And so backpropagation isn't just as fast algorithm for learning. It actually gives us detailed insights into how changing the weights and biases changes the overall behavior of the network.

In Naïve Bayes, the main aim is to obtain a simple probabilistic classifier by calculating a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses the Bayes theorem and assumes all attributes to be independent given the value of the class variable. This conditional independence assumption rarely holds true in real world applications, hence the characterization as Naive yet the algorithm tends to perform well and learn rapidly in various supervised classification problems [4].

Recently machine learning techniques with feature selection methods have been studied. Abductive network ensemblers (committees) based networks, a set of inductive machine learning techniques are applied to classify UCI public domain spam email dataset using feature reduction (82.5% reduction of original data) technique [5] and found 91.7% classification accuracy with false positives 4.3%. The performance of these GMDH based algorithms is found to be better than other techniques such as MLP based neural net and Naïve Bayes algorithm. A revised back-propagation algorithm along with thesaurus of keywords and related keywords is used on public domain spam email dataset Ling-Spam corpus and found that the performance in terms of spam accuracy is better than that of simple back propagation neural net algorithm [6].

A decision tree approach can also be applied for classification of spam emails. A decision tree includes a rule set by which objective functions can be predicted. These algorithms mine the dataset for information and calculate gain and entropy values which are then used form multiple rules. These multiple rules are then combined to form a decision tree on which the data is tested against to classify the spam emails. Various algorithms that are based on decision tree approach such as ID3 which calculates gain as deciding factor or C4.5 (Extension to ID3) which uses gain ratio as its deciding factor to define the rules and form the decision tree.

A. Artificial Neural Networks - Perceptron Algorithm

Perceptron networks come under single-layer feed-forward networks and are also called simple perceptrons [7]. The perceptron is an algorithm for learning a binary classifier i.e. a function that maps its input x (a real-valued vector) to an output value f(x) (a single binary value):

$$f(x) = \begin{cases} 1 & \text{if } w.x + b > \theta \\ 0 & \text{otherwise} \end{cases}$$

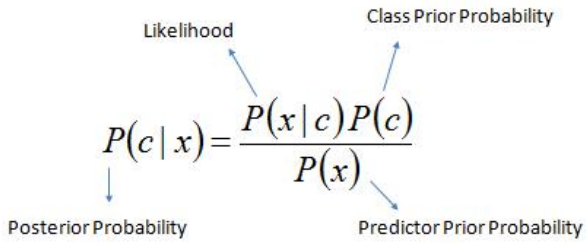
where w is a vector of real-valued weights, w.x is the dot product $\sum_{i=0}^{m} [\![w_i x_i]\!]$, where m is the number of inputs to the perceptron, b is the bias and θ is the fixed threshold. The perceptron learning rule is used in weight updation. For each training input, the output value f(x) is calculated which determine whether or not an error has

occurred. The error calculation is based on the comparison of the values of targets with those of the calculated outputs. The weights will be adjusted on the basis of the learning rule if an error has occurred for a particular training example. Although the perceptron rule finds a successful weight vector when the training examples are linearly separable, it can fail to converge if the examples are not linearly separable i.e. it will never reach a point where all vectors are classified properly. The most famous example of the perceptron's inability to solve problems with linearly non-separable vectors is the Boolean exclusive-or problem. The solution spaces of decision boundaries for all binary functions and learning behaviors are studied in the reference [8].

B. Naïve Bayes Algorithm

The Naive Bayesian classifier is based on Bayes' theorem with independent assumptions between predictors. A Naive Bayesian model is easy to build and doesn't include any complicated iterative parameter estimation which makes it particularly useful for very large datasets. Even though, the Naive Bayesian classifier is one of the simplest, it often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. To classify the e-mail filtering, according to the Naive Bayes algorithm on the message set and training of continuous learning, statistical model, model storage class a priori probabilities and lexical features of the a posteriori probability, when new mail arrives, according to the stored in the model of the probability of new emails calculated the probability value, and then decide which belong to the category.

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 1.

- P(c|x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

C. C4.5 Algorithm

The C4.5 algorithm builds decision trees from a set of training data in the same way as ID3 [12], using the concept of information entropy. The training data is a set S= s1, s2…… of already classified samples. Each sample si consists of a p-dimensional vector {x1,i, x2,i, …… xp,i} where the xj represent attribute values or features of the sample, as well as the class in which si falls. At each node of the tree, C4.5 algorithm chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sub lists. One drawback or limitation of ID3 is that it is overly sensitive to features with large numbers of values. Since for spam email classification there would be large number of emails to be classified and hence large number of values ID3 proves to be inefficient. To overcome this problem, C4.5 uses "Information gain," This computation does not, in itself, produce anything new. However, it allows to measure a gain ratio [13].

Gain ratio, is defined as follows:

$$\text{GainRatio}(p, T) = \frac{Gain(p,T)}{SplitInfo(p,T)}$$

Where SplitInfo is:

$$\text{SplitInfo}(p, T) = -\sum_{j=1}^{n} p'\left(\frac{j}{p}\right) * \log(p'\left(\frac{j}{p}\right))$$

C4.5 addresses the following issues not dealt with by ID3:

- Avoiding overfitting the data
- Determining how deeply to grow a decision tree.
- Reduced error pruning.
- Rule post-pruning.

III. PERFORMANCE MEASURES FOR SUPERVISED MACHINE LEARNING METHODS

In order to measure the performance of supervised machine learning methods various performance measures are

used such as recall, precision, false positive rate, accuracy, specificity, and F-measure [5] [6]. These measures can easily be derived from the confusion matrix of the model. The overall performance of the model is analyzed considering its performance both on training and testing data. A model that is built on training data by learning each and every case or peculiarities precisely (fitting best way) may not perform well on test data. High performance only on training data is not a good indicator of overall performance of the model. Overfitting of model is one of the issues in model building exercise. A good model must be able to generalize well on test data where test data is completely different from training data. True positives (TP), true negatives (TN), false positive (FP) and false negatives (FN) are four components of confusion matrix. The calculation [5] of various parameters are given below:

a) Recall = TP/ (TP + FN). It explains how good a test is at detecting the positives. i.e. predicting positive observations as positive.. A high recall is desired for a good model. Recall is also known as sensitivity or TP Rate.

b) FP Rate = FP/ (FP+ TN). It explains how good a model is at detective the negatives. A model predicting as positive when actually it is negative, is not desirable. This measure is also evaluated as 1- Specificity where specificity (TN Rate)= TN/ (TN+FP). A high specificity (predicting all negatives correctly) is desirable.

c) Precision = TP/ (TP + FP). It determines how many of the positively classified are relevant. It is the percentage of positive predictions correct. A high precision is desirable.

d) Accuracy = TP + TN/ (TP+TN+FP+FN). It tells how well a binary classification test correctly i.e. what percentage of predictions that are correct. Accuracy alone is not a good indicator, as it does not tell how well the model is in detecting positives or negatives separately.

e) F-Measure = 2 * (Precision X Recall)/ (Precision + Recall). F-measure is a good indicator as it considers both precision and recall. A high F-measure is desirable.

f) Precision, recall, f-measure, and false positive rates are calculated for both the outcome classes (e.g. both yes and no) for and a weighted average is considered while experimenting in each technique.

g) In an experiment no single measure tells how good a model is. Different measures discussed above have been considered in our experiment for evaluating various machine learning models for classifying spam email corpus.

IV. EXPERIMENTATION AND COMPARISON OF PERFORMANCE

The machine learning classification experimentation on dataset consists of three steps: preparing the data, classification experiments using various machine learning classifiers and evaluating the performance of machine learning classifiers.

A. Dataset Preparation

The spam e-mail database considered for experimentation is collected from Enron Spam datasets [9] to develop a model and to determine whether a given email is spam or not using the model developed. The dataset was created in 2006 by V. Metsis, I. Androutsopoulos and G. Paliouras and presented at 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006. It consists of 5172 instances of legitimate and spam email messages with 29.0% being spam. The observations consist of 25030 input attributes i.e. words and performance measures as output. In our experimentation different indicators of spam (unsolicited commercial email) or non-spam have not been considered. The dataset consists of a total of 5172 emails with 25030 attributes, out of which 1500 (29.0%) instances are spam and 3672 (71.0%) are non-spam. The dataset consists of an outcome variable (spam or non-spam), frequency count of various words, and length of sequence of consecutive capital letters. The dataset is high-dimensional and complex in nature where spammers have used different strategies so that it would be difficult to identify a spam email. A good classifier builds model that is capable of generalizing and not overfitting to the training dataset. In this paper, holdout method is adopted where the entire dataset is divided to two mutually exclusive data sets: training and testing. The model is built on training data and then evaluated or tested on test data. It is found in literature that for a classifier, a 2/3 to 1/3 training-to-test set random split provides good result i.e. near optimal mean squared error of the prediction accuracy [10] [11]. For our experimentation, 3361 (65%) observations are considered as training and 1811 (35%) as test data set have been considered with a random selection procedure. The distribution of dataset is presented in Table-I. The same dataset without changing the instances is used for experimenting all the techniques to avoid any type of biasness.

Table 1. DISTRIBUTION OF DATASET

Dataset	Spam	Non-Spam	Total
Corpus Dataset	1500 (29.0%)	3672 (71.0%)	5172
Training Dataset	945	2416	3361 (65%)
Test Dataset	555	1256	1811 (35%)

B. Comparison of Performance

Supervised machine learning techniques is applied in the experiment to the Spam Email dataset. The performance of the techniques is discussed below.

Out of all machine-learning techniques considered in the experiment, Naïve Bayes was found to be the best in terms F-measure (87%) and FP rate (7%) followed by Perceptron with F-measure (84.5%) and FP- rate (6.8%).

Table 2. COMPARISON OF NAÏVE BAYES AND PERCEPTRON

	Naïve Bayes	Perceptron	C4.5
Recall	0.978	0.845	0.699
FP-Rate	0.070	0.068	0.123
Precision	0.795	0.845	0.726
Accuracy	0.916	0.905	0.820
F-measure	0.877	0.845	0.712

V. CONCLUSION

In this paper, we compare classification algorithms on Enron spam e-mail dataset through experimentation and it is found that neural network provides the best result among all the classifiers (96% recall, 96% precision) keeping FP to a minimum (0.07%). Neural network model is robust compare to all the category of algorithms considered because of its ability to predict best in spite of the fact that the dataset is noisy and sparse Neural network model has also the best scalability features compare to other algorithms considered as it could able to construct the model efficiently when the dataset considered for experimentation is very large. On the other hand the interpretability of neural network model is very low compare to decision tree, decision rules and Naïve Bayes algorithms. Neural network model worked as the dataset is complex with high dimensionality, and missing data. One of the limitations of this study is that the machine learning algorithms are tested on one dataset and need to on several public domain spam email datasets.

REFERENCES

[1] R.Malarvizhi and K.Saraswathi, "Content-Based Spam Filtering and Detection," International Journal of Engineering Trends and Technology (IJETT), vol. 4, no. 9, pp. 4237-4242, 2013.

[2] P. K. Panigrahi, "A Comparative Study of Supervised Machine Learning Techniques for Spam E-Mail Filtering," in Fourth International Conference on Computational Intelligence and Communication Networks, 2012.

[3] K. Tretyakov, "Machine Learning Techniques in Spam Filtering," in Data Mining Problem-oriented Seminar, 2004.

[4] J. A. A. a. C. M. J. George Dimitoglou, "Comparison of the C4.5 and a Naïve Bayes Classifier for the Prediction of Lung Cancer Survivability".

[5] E. S. M. &. A.-A. R. E. El-Alfy, "Using GMDH-based networks for improved spam detection and email feature analysis. Applied Soft Computing," pp. 477-488, 2011.

[6] H. &. Y. B. Xu, "Automatic thesaurus construction for spam filtering using revised back propagation neural network," Expert Systems with Applications, pp. 18-23, 2010.

[7] D. Sivanandanam, Principles of Soft Computing.

[8] J.-W. L. a. C.-Y. L. Daw-Ran Liou, "Learning Behaviors of Perceptron," ISBN 978-1-477554-73-9.

[9] I. A. a. G. P. V. Metsis, "The Enron-Spam datasets," [Online]. Available: http://www.aueb.gr/users/ion/data/enron-spam/.

[10] D. &. R. S. Kevin, "Optimally splitting cases for training and testing high dimensional classifiers," BMC Medical Genomics, 2011.

[11] K. T. G. &. V. I. Sechidis, "On the stratification of multi-label data. Machine Learning and Knowledge Discovery in Databases," pp. 145-158.

[12] "ID3 algorithm," [Online]. Available: https://en.wikipedia.org/wiki/ID3_algorithm.

[13] A. M. H. E. a. M. E. B. HSSINA, "A comparative study of decision tree ID3 and C4.5," ID3 and C4.5, pp. 13-19.