Comparative Study of Classification Algorithms for Spam Email Detection

Aakanksha Sharaff, Naresh Kumar Nagwani and Abhishek Dhadse

Abstract Spam in emails has become a major issue. Spam messages consume space, network bandwidth and are of no use to the receiver. It is very difficult to filter spam as spammers try to tackle the processes carried out by the filtering mechanism. Various classification algorithms are used to classify a mail as spam or non-spam (ham). The present paper compares and discusses the effectiveness of four machine learning classification algorithms, belonging to different categories (Probabilistic, Decision Tree, Vector Machines and Lazy Algorithms) on the basis of various performance measures, using WEKA, a data mining tool to analyze different algorithms. Enron dataset is taken in a processed form from Athens University of Economics and Business and it is found that J48 and BayesNet algorithms perform better than SVM.

Keywords Email classification · Performance measures · Spam · Email spam filtering

1 Introduction

With the advent of email, communication has become a lot easier than before. Emails have become an important part of communication in any organization among the world. Originally, email was designed to create, send and receive messages from one another. However the increase in the popularity of emails has also resulted in spam e-mails over the course of time. Spam mails are the messages which are sent in bulk, i.e. sent to many users at the same time for commercial purposes. The sender of spam mails has no relation with the receivers. The spammers obtain a list of mails from various sources such as address books, so as to send mails. On clicking of spam mail, the spammers get benefit. These mails can also be infected, so upon clicking, these can be sent from your address to other

A. Sharaff (✉) · N.K. Nagwani · A. Dhadse
Department of CSE, National Institute of Technology, Raipur 492010, India

N.R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, DOI 10.1007/978-81-322-2553-9_23

contacts in your list, or simply introduce viruses in your system as well. These can also contain phishing links [6] aimed to retrieve your passwords. Overall, this reduces a firm's productivity regarding communication, and is very risky regarding security [14]. Hence study of email and spam detection has become an important area in the field of computer science. Each message cannot be detected as spam or not by human alone because that would be very time consuming task. Hence classification algorithms are used for the purpose.

Many classification algorithms can be used for the task of text classification, such as probabilistic algorithms, decision tree algorithms, lazy algorithms etc. [8, 10]. Despite of so many algorithms, the task of complete accuracy is not yet fulfilled. So these algorithms need to be compared for performance with each other. In this paper, we have used four classification algorithms, namely BayesNet, SVM, J48 (C4.5) and Lazy IBK for comparison. Spam dataset is a processed dataset taken from the library of Natural Language Processing Group, Department of Informatics—Athens University of Economics and Business. These mails are analyzed using WEKA, a data mining tool from University of Waikato, New Zealand. In this work, the same dataset is used for training and testing in the form of folds. The contents of a mail are used to train the four above mentioned algorithms, which learn from the dataset [4, 5]. The results obtained are used to decide the performance of algorithms.

2 Related Work

Over the time, due to the risk factor involved in the reception of spam mails, many works have been carried out to counter these mails. Many spam classification systems are designed for this purpose. The goal of such systems is to check whether a mail is spam or not, and if it is spam, move it to another destination or folder. Many research works are carried out to search for the best classification algorithm. But it mainly depends on data. The four algorithms we have used are of four different categories, i.e. Probabilistic, Decision Tree, Lazy algorithms and Support Vector Machines. Many studies have been done to compare the performances of these algorithms, the classification algorithms are used to test whether a mail is spam or not and based on the results, the algorithms are ranked. The various performance measures included accuracy, TP rate, precision, recall and F-measure in in paper [1, 11].

Spam Classifiers are built and tested on publically available datasets for evaluation. For example, J48 is used for medical diagnosis, which decides whether a person has certain disease or not on the basis of his symptoms [2]. A study of Naive Bayes is done in [3] which consider six types of Naive Bayes on six datasets, derived from the original Enron corpus [12], and are made publically available, which contain proper ham to spam ratio and are more realistic than previous comparable benchmarks. In this paper, a comprehensive review of recent machine learning approaches to Spam filters was presented in which a quantitative analysis of the use of feature selection algorithms and datasets was conducted [16].

3 Dataset

Enron dataset [12] is taken in a processed form from Athens University of Economics and Business. The dataset used in our study contains ham and spam messages of particular users of Enron. The ratio between ham and spam is maintained. These messages contain body and different headers such as subject, sender etc. Out of which we have considered subject and body for our study.

4 Classification Algorithms

As mentioned before, we have used four algorithms, each belonging to different category.

1. **J48** is the open source java implementation of C4.5 algorithm which is an improvement over ID3 algorithm, and it works on data by splitting into different parts. Each node of the tree splits its instances into one part or other. It is based on an impurity measure, called entropy. The difference in entropy, called information gain is calculated and the node providing maximum information gain is used to split the data. It handles both continuous and discrete values. It handles missing values as well.
2. **SVM**, i.e. Support Vector Machines are the supervised machine learning models which work by classification and regression analysis. SVM considers data as points in space mapped in a way such that the difference between the closest data points is maximum [7]. New examples are then put into the graph depending on which side of the margin they fall on. A good separation has more distance between the closest data points, since larger margin indicates the lower generalization error.
3. **BayesNet**, which stands for Bayesian Network is a probabilistic classifier model which works on the data by creating directed acyclic graphs using their conditional dependencies. The conditional probabilities are associated with the words in the email. These are then used to determine a mail is spam or not. Because a Bayes net only relates nodes that are probabilistically related by some sort of causal dependency, it can save enormous amount of time.
4. **LazyIBK** belongs to the lazy class of algorithms [9]. It is also called the K Nearest Neighbour algorithm, which works by classifying an instance depending on the majority in the nearest neighbours [15]. It works directly on test-data. If the majority of nearest mails are spam, the new email is likely to be classified as spam. LazyIBK is the Weka [13] implementation of KNN algorithm. It may return more than k neighbours if there are ties in the distance. Lazy algorithms are suitable when instances are not available beforehand, but occur online one by one.

5 Performance Measures

The classification results into a confusion matrix, which consists of four parts, True Positive (TP), True Negative, (TN), False Positive (FP) and False Negative (FN) [11]. These values can be used to determine performance measures like Accuracy, FP rate, Precision, Recall and F-Measure. The following performance measures are used for comparison of algorithms-

1. Accuracy:
   Accuracy = (TP + TN)/(TP + TN + FP + FN)
   It tells how much classification is done correct. TP and TN together is the correct number of classifications done by the classifier. It does not consider positives and negatives separately and hence other measures are also required for the analysis other than accuracy.
2. FP Rate:
   FP Rate = FP/(FP + TN)
   It tells how the model has performed in detecting the negatives. A low FP rate is desirable as a model classifying positives as negatives is not desired.
3. Precision:
   Precision = TP/(TP + FP)
   It indicates how many instances, which are classified as positive, are actually relevant. A high precision is desirable because high relevancy in detecting positives is desired, i.e. less FP is desired.
4. Recall:
   Recall = TP/(TP + FN)
   It is also called as TP rate, and is an indication of how good a system can detect positives.
5. F-Measure:
   F-Measure = 2*(Precision*Recall)/(Precision + Recall)
   Since a high Precision and Recall is desired, hence high F-Measure is also desired.

6 Methodology

The methodology of our experimentation consists of four main parts, preparation of dataset, pre-processing, application of algorithms, and performance evaluation based on the above performance measures.

1. **Preparation of Dataset** The dataset needs to be in the proper format for applying machine learning algorithms. The dataset is converted into .arff file using weka library, with 4307 mails, out of which 635 are spam. The first attribute contains the filename, which is ignored later, second attribute consists

of the subject and body of the email, and the last attribute of the dataset is the nominal attribute, which consists of the value whether a mail is spam or not.

2. **Pre-Processing** The dataset is loaded into the WEKA tool and the first attribute is removed, as it does not contribute to spam detection. The second attribute, which is a string of words, has to be converted into a vector of words, which is done by string to word vector. Then the words are separate entities. These words are then passed through stopword removal, which consists of WEKA's list of stopwords, and then Snowball stemming algorithm is applied to the data.
3. **Application of Algorithms** The algorithms discussed above are applied with the use of filtered classifier. The split between training and test data set is done using 10-Fold Cross validation.
4. **Performance Evaluation** The results obtained are then compared on the basis of the performance measures discussed above. The results obtained are shown below.

A. **Accuracy**

In terms of accuracy, BayesNet and J48 performed better than SVM and Lazy IBK. This indicates a deviation in the existing trend that SVM performs better than most of the algorithms. Here, J48 has the highest accuracy, followed by BayesNet and lazyIBK while SVM has the lowest accuracy. Figure 1 shows the accuracy chart over different classifiers.

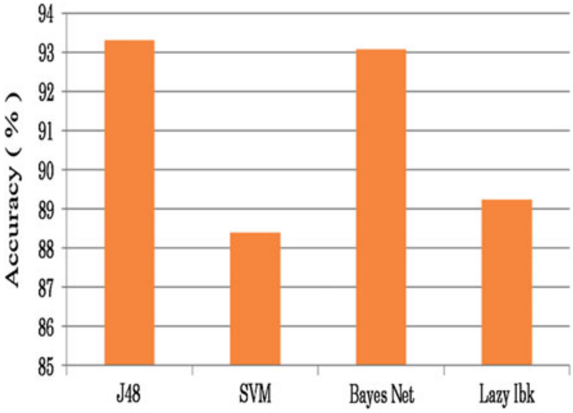
B. **Precision and Recall**

The same type of result is obtained when precision and recall are considered shown in Fig. 2. BayesNet performs the best, followed by J48 and Lazy IBK while SVM has the lowest precision and recall values. This is also against the existing norms. Usually SVM performs the best amongst all the above mentioned algorithms.

C. **TP Rate and FP Rate:**

In terms of TP Rate, J48 has the highest value, followed by BayesNet and LazyIBK and again SVM has the lowest value of all as shown in Fig. 3. It is

Fig. 1 Accuracy



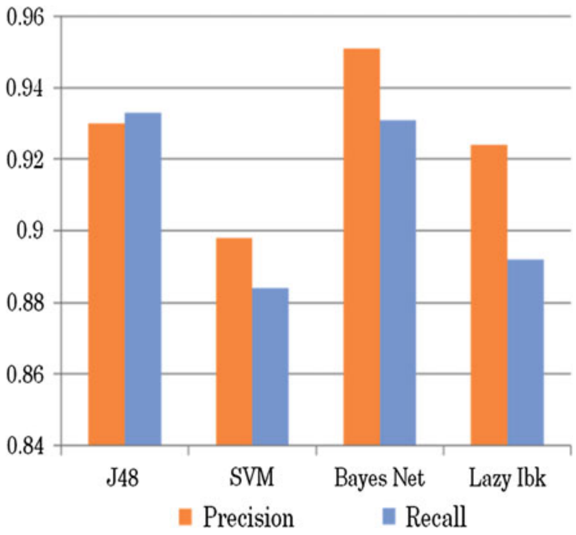


Fig. 2 Precision and recall

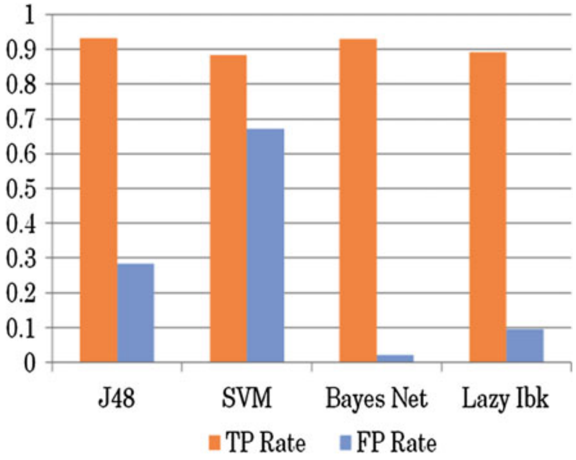


Fig. 3 TP rate and FP rate

equivalent to Recall. While in terms of FP Rate, Bayesnet again performs the best, followed by LazyIBK and J48, while SVM is found to have exceptionally high FP Rate.

D. **F-Measure**

F-Measure is dependent on Precision and Recall, hence the result is almost the same as above obtained results, i.e. BayesNet performs best followed by J48 and Lazy IBK and after that SVM. Figure 4 shows the F-measure performance of different classifiers

E. **Overall Results**

Table 1 describes the overall performance of different classifiers (J48, SVM, BayesNet, LazyIBK)

Fig. 4 F-measure

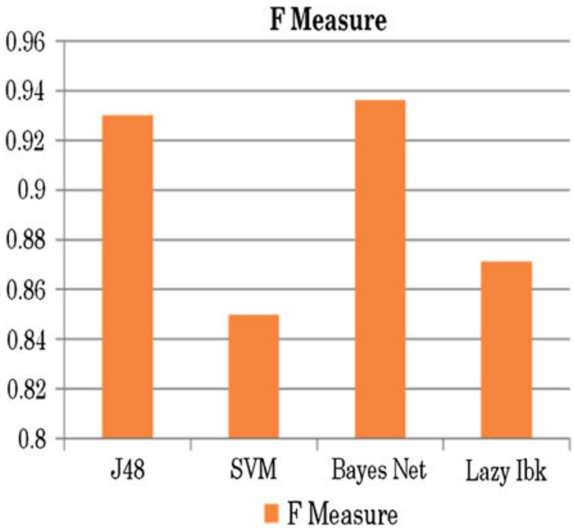


Table 1 Describes the overall performance of different classifiers (J48, SVM, BayesNet, LazyIBK)

Classification algorithms	Accuracy (in %)	Precision	Recall	FP Rate	F-measure
J48	93.3132	0.93	0.933	0.284	0.93
SVM	88.391	0.898	0.884	0.671	0.85
BayesNet	93.081	0.951	0.931	0.022	0.936
LazyIBK	89.236	0.924	0.892	0.095	0.901

7 Conclusion

In this paper, four classification algorithms, belonging to four different categories are compared against various performance measures. The results obtained indicate a deviation from the pre-established norms, i.e. SVM is one of the best classification algorithms, but here the results indicate otherwise. BayesNet and J48 perform the better than SVM. This deviation indicates that the performance of algorithm depends on data more than the algorithms. LazyIBK since belongs to Lazy algorithms, works at prediction time, i.e. on test data, hence it cannot be used as a basis to judge the performance of SVM. But J48 and BayesNet are having better results indicate that the algorithm performance depends on data. In future, we would like to optimize the classification time by the use of sub classifiers and then evaluate the performance of various classification algorithms. This can be done by using incremental sub-classifiers. Each sub-classifier represents a small part of a typical classifier and hence takes less time. This will improve the classification results which will be compared again with the existing results.

Acknowledgments Aakanksha Sharaff, Naresh Kumar Nagwani would like to thanks National Institute of Technology, Raipur, India for providing their support, imparting their knowledge, and making all the required facilities available for completing this research.

References

1. Panigrahi, P.K.: A comparative study of supervised machine learning techniques for spam email filtering. In: 2012 Fourth International Conference on Computational Intelligence and Communication Networks, 2012
2. Alvestad, S.: Early warnings of critical diagnoses, 2009
3. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam filtering with Naive Bayes—which Naive Bayes?. In: Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006
4. Ackoff, R.L.: From data to wisdom. J. Appl. Syst. Anal. **16**(1), 3–9 (1989)
5. Sharma, N.: The origin of the data information knowledge wisdom hierarchy. Data Inf. Knowl. Wisdom hierarchy (2008)
6. Chandrasekaran, M., Narayanan, K., Upadhyaya, S.: Phishing email detection based on structural properties. In: Proceedings of 9th Annual NYS Cyber Security Conference, June 2006
7. Ozarkar, P., Patwardhan, M.: Efficient spam classification by appropriate feature selection. Global J. Comput. Sci. Technol. Softw. Data Eng. **13**(5) (2013)
8. Lim, T.S., Loh, W.Y., Shih, Y.S.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Mach. Learn. **40**(3), 203–228 (2000)
9. Hsiao, W.F., Chang, T.M.: An incremental cluster-based approach to spam filtering. Expert Syst. Appl. **34**(3), 1599–1608 (2008)
10. Awad, W.A., Elseuofi, S.M.: Machine learning methods for E-mail classification. Int. J. Comput. Appl. **16**(1), 39–45 (2011). doi:10.5120/1974-2646
11. El-Alfy, E.S.M., Abdel-Aal, R.E.: Using GMDH-based networks for improved spam detection and email feature analysis. Appl. Soft Comput. **11**(1), 477–488 (2011)
12. Ion Androutsopoulos, Enron Dataset http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html
13. Weka- Data Mining Tool tutorials, documentation http://www.cs.waikato.ac.nz/ml/weka/documentation.html
14. Ahmed, Kh: An overview of content-based spam filtering techniques. Informatica **31**(3), 269–277 (2007)
15. Geetha Ramani, R., Sivagami, G.: Parkinson disease classification using data mining algorithms. Int. J. Comput. **32**(9) (2011)
16. Guzella, T.S., Caminhas, W.M.: A review of machine learning approaches to Spam filtering. Expert Syst. Appl. **36**, 10206–10222 (2009)