Supervised Learning Approach for Spam Classification Analysis using Data Mining Tools

[1]R.Deepa Lakshmi
Mphil Scholar,
PSGR Krishnammal College for women, Coimbatore, India

[2]N.Radha
Sr.Lecturer,
G R Govindarajulu School of Applied Computer Technology,
PSGR Krishnammal College for Women,Coimbatore India

Abstract - **E-mail is one of the most popular and frequently used ways of communication due to its worldwide accessibility, relatively fast message transfer, and low sending cost. The flaws in the e-mail protocols and the increasing amount of electronic business and financial transactions directly contribute to the increase in e-mail-based threats. Email spam is one of the major problems of the today's Internet, bringing financial damage to companies and annoying individual users. Among the approaches developed to stop spam, filtering is the one of the most important technique. Many researches in spam filtering have been centered on the more sophisticated classifier-related issues. In recent days, Machine learning for spam classification is an important research issue. This paper explores and identifies the use of different learning algorithms for classifying spam messages from e-mail. A comparative analysis among the algorithms has also been presented.**

Keywords – **RapidMiner, Weka, Machine Learning techniques, J48, Spam Classification,.**

I. INTRODUCTION

This decade has widely been deemed as the internet era. The use of internet has been extensively increasing over the past decade and it continues to be on the ascent. Hence it is apt to say that the Internet is gradually becoming an integral part of everyday life. Internet usage is expected to continue growing and e-mail has become a powerful tool intended for idea and information exchange. Negligible time delay during transmission, security of the data being transferred, low costs are few of the multifarious advantages that e-mail enjoys over other physical methods. However there are few issues that spoil the efficient usage of emails. Spam email is one among them[6].

According to the data estimated by Ferris Research, spam accounts for 15% to 20% of email at U.S.-based corporate organizations [7]. In general, the sender of a spam message pursues one of the following tasks: to advertise some goods, services, or ideas, to cheat users out of their private information, to deliver malicious software, or to cause a temporary crash of a mail server. From the point of view of content spam is subdivided not just into various topics but also into several genres, which result from simulating different kinds of legitimate mail, such as memos, letters, and order confirmations.

II. LITERATURE SURVEY

Spam mail, also called unsolicited bulk e-mail or junk mail that is sent to a group of recipients who have not requested it. The task of spam filtering is to rule out unsolicited e-mails automatically from a user's mail stream. These unsolicited mails have already caused many problems such as filling mailboxes, engulfing important personal mail, wasting network bandwidth, consuming usesrs time and energy to sort through it, not to mention all the other problems associated with spam (crashed mail-servers, pornography adverts sent to children, and so on)[11]. According to a series of surveys conducted by CAUBE.AU 1, the number of total spams received by 41 email addresses has increased by a factor of six in two years (from 1753 spams in 2000 to 10,847 spams in 2001)[14]. Therefore it is challenging to develop spam filters that can effectively eliminate the increasing volumes of unwanted mails automatically before they enter a user's mailbox.

D. Puniskis [12] in his research applied the neural network approach to the classification of spam. His method employs attributes composed of descriptive characteristics of the evasive patterns that spammers employ rather than using the context or frequency of keywords in the message. The data used is corpus of 2788 legitimate and 1812 spam emails received during a period of several months. The result shows that ANN is good and ANN is not suitable for using alone as a spam filtering tool.

In [13] email data was classified using four different classifiers (Neural Network, SVM classifier, Naïve Bayesian Classifier, and J48 classifier). The experiment was performed based on different data size and different feature size. The final classification result should be '1' if it is finally spam, otherwise, it should be '0'. This paper shows that simple J48 classifier which make a binary tree, could be efficient for the dataset which could be classified as binary tree.

III. DATASET DESCRIPTION

The dataset that has been used for this work was acquired over a two months from various e-mail_ids. Around 20 attributes of the spam emails were identified and used in the dataset. From address, to address, type of spam received, organization from which the spam was received were few of the attributes used.

A. In-transit E-mail Packet Pre-classification

E-mail packets can be distinguished from other types of data packets based on the protocols. Widely used e-mail protocols, such as SMTP, Post Office Protocol (POP), and Internet Mail Access Protocol (IMAP); use the TCP as the transport protocol. TCP packets are distinguished from other types of transport protocol packets from the Protocol field. The e-mail protocols, SMTP for instance, normally uses port 25 for Destination port number in the TCP header.

Fig. 1 shows an e-mail packet pre-classification which is distributed over the network. In fact, e-mail packet pre-classification on a node before an MTA is sufficient for fast e-mail class estimation (at the receiving MTA) since e-mail packets have been pre-classified before reception by receiving MTAs[8].

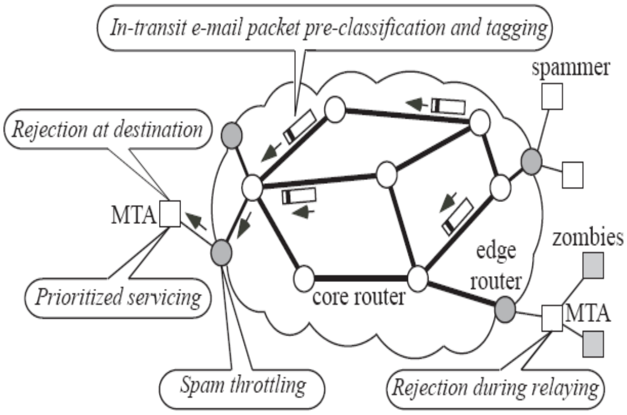

Fig. 1 Proactive spam control over the Internet

When e-mail packet pre-classification is performed at intermediate nodes, the spam pre-classification score can be passed to the receiving MTA. Appending additional bytes to packet header or payload (or simply packet marking) presents some challenges such as the possibility of packet fragmentation and rejection of packets with option field. The e-mail packets are likely to traverse several pre-classification nodes unless pre-classification is done at the ingress point of a corporate network (or an MTA).

B. E-mail Content Classification

Supervised-learning content classification techniques learn the distinctions among different e-mail classes. Once trained with examples to form a generative model, a supervised-learning e-mail content classification can recognize the exact or similar patterns observed during learning.

The accuracy of content classification using supervised-learning techniques depends on the quality, quantity, and timeliness of learning examples. The use of different learning datasets, each with different algorithms, makes difficult for different classifiers. A classifier that works well on certain learning data sets may not perform well on different data sets. There are several data sets, which can be used for evaluating spam content classifiers, such as the Spam Assassin and Ling Spam data sets.

E-mail content classification techniques, which originated from text classification techniques, dissect e-mails to estimate their classes [3][5]. E-mail header and body may contain several informative features, which distinguish non-spam from spam e-mails. The features can be extracted from the content of an e-mail, which could be characters, fixed-length strings, words, or phrases[9].

IV. METHODOLOGY

For analyzing real time dataset and to predict the performance, the supervised learning algorithms were adopted here.

Different algorithms use different biases for generalizing different representations of the knowledge. Therefore, they tend to error on different parts of the instance space. The combined use of different algorithms could lead to the correction of the individual uncorrelated errors. There are two main paradigms for handling an ensemble of different classification algorithms: Classifier Selection and Classifier Fusion. The first one selects a single algorithm for classifying new instances, while the latter fuses the decisions of all algorithms. This section presents the most important methods from both categories.

A. Classifier Selection

It is a very simple method, which produces Selection or Select Best. This method evaluates each of the classification algorithms on the training set and selects the best one for application on the test set. Although this method is simple, it has been found to be highly effective and comparable to other more complex state-of-the-art methods. Another line of research proposes the selection of a learning algorithm based on its performance on similar learning domains. Several approaches have been proposed for the characterization of learning domain, including general, statistical and information theoretic measures and model-based data Characterizations.

The selection of algorithms is based on their local performance, but not around the test dataset itself, and also comprising the predictions of the classification models on the test instance. Training data are produced by recording the predictions of each algorithm, using the full training data both for training and for testing.

B. Classifier Fusion

The classifier fusion approach is capable of taking several specialized classifiers as input and learning from training data how well they perform and how their outputs should be combined. In addition to data fusion, relies quite heavily on machine learning. This method assumes that the classifiers in the pool are trying to solve the same classification problem. As a result, only adequate fusing classifiers that can attempt to detect the entire set.

C. Classification Algorithms

The main motivation for different classification algorithms is accuracy improvement. Each method has its own variety of algorithms. Various algorithms of these methods were used to predict the accuracy of the dataset. Different classification methods used are MLP, k-NN and SVM. Each model is associated with a coefficient, usually proportional to its classification accuracy.

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model”. The Naive-Bayes inducer computes conditional probabilities of the classes given the instance and picks the class with the highest posterior. Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting.

The basic concept of it is to find whether an e-mail is spam or not by looking at which words are found in the message and which words are absent from it. In the literature, the NB classifier for spam is defined as follows

$$C_{NB} = \arg\max_{c_i \in T} P(c_i) \prod_k P(w_k \mid c_i)$$

where *T* is the set of target classes (spam or non-spam), and *P*(*wk* | *ci*) is the probability that word *wk* occurs in the e-mail, given the e-mail belongs to class *ci*. The likelihood term is estimated as

$$P(w_k \mid c_i) = \frac{n_k}{N}$$

where *nk* is the number of times word *wk* occurs in emails with class *ci*, and *N* is the number of words in emails with class *ci*.

Linear discriminant analysis (LDA) and the related Fisher's linear discriminant are methods used in statistics and machine learning to find a linear combination of features which characterize or separate two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification.

A multilayer perceptron is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. To ensure the classification of a new occurrence, training sample is adjusted for each iteration.

For instance let *x* be a vector that the perceptron fails to classify, and wi, bi the vector of weight and bias which corresponds to the ith iteration. We have sign (wix+bn) ≠ c where c is the sign corresponding to the real class of the message that has the characteristic vector x. The new vectors wi+1 and bi+1 are calculated as follows:

$$wi+1 = wi + cx \text{ and } bi+1 = bi + c$$

The training continues until the perceptron manages to classify correctly all the messages of the training sample.

J48 builds decision trees from a set of training data using the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. J48 examines the normalized information gain that results from choosing an attribute for splitting the data. The splitting procedure stops if all instances in a subset belong to the same class. Then a leaf node is created in the decision tree telling to choose that class. In this case J48 creates a decision node higher up in the tree using the expected value of the class. Further it provides an option for pruning trees after creation.

V. RESULT EVALUATION

The data set was separated into two parts, one part is used as training data set to produce the prediction model, and the other part is used as test data set to test the accuracy of our model. The Training data set contains feature values as well as classification of each record. Testing is done by 10-fold cross validation method.

A. Measuring the Performance

The meaning of a good classifier can vary depending on the domain in which it is used. For example, in spam classification it is very important not to classify legitimate messages as spam as it can lead to. e.g. economic or emotional suffering for the user.

B. Precision and Recall

A well employed metric for performance measurement in information retrieval is precision and recall. These measures have been diligently used in the context of spam classification.

Recall is the proportion of relevant items that are retrieved, which in this case is the proportion of spam messages that are actually recognized.

In the spam classification context, precision is the proportion of the spam messages classified as spam over the total number of messages classified as spam. Thus if only spam messages are classified as spam then the precision is 1. As soon as a good legitimate message is classified as spam, the precision will drop below 1.

Formally:

Let ngg be the number of good messages classified as good (also known as false negatives).

Let ngs be the number of good messages classified as spam (also known as false positives).

Let nss be the number of spam messages classified as spam (also known as true positives).

Let nsg be the number of spam messages classified as good (also known as true negatives).

The precision (p) and recall (r) are defined as

$$p = \frac{n_{ss}}{n_{ss} + n_{gs}} = \frac{1}{1 + \frac{n_{gs}}{n_{ss}}}$$

$$r = \frac{n_{ss}}{n_{ss} + n_{sg}} = \frac{1}{1 + \frac{n_{sg}}{n_{ss}}}$$

The precision calculates the occurrence of false positives which are good messages classified as spam. When this happens p drops below 1. Such misclassification could be a disaster for the user whereas the only impact of a low recall rate is to receive spam messages in the inbox. Hence it is more important for the precision to be at a high level than the recall rate.

A problem when evaluating classifiers is to find a good balance between the precision and recall rates. Therefore it is necessary to use a strategy to obtain a combined score. One way to achieve this is to use weighted accuracy.

C. Weighted Accuracy

To reflect the difference in misclassifying a good message and a spam message a cost sensitive evaluation is used to measure the performance of the classifier.

The weighted accuracy of a classifier is defined as

$$W_{acc} = \frac{\lambda \cdot n_{gg} + n_{ss}}{\lambda \cdot n_g + n_s}$$

where g n is the total number of messages and s n is the total number of spam messages. 1 is the weight of each good message. Each misclassification of a good message counts as l misclassifications of spam.

D. Cross Validation

There are several means of estimating how well the classifier works after training. The easiest and most straightforward means is by splitting the dataset into two parts and using one part for training and the other for testing. This is called the holdout method. The disadvantage is that the evaluation depends heavily on which samples end up in which set. Another method that reduces the variance of the holdout method is k -fold cross-validation.

In k-fold cross-validation, M is split into k mutually exclusive parts, M1, M2...Mk. The inducer is trained on Mi \ M and tested against Mi. This is repeated k times with different i such that Îi {1, 2... k}. Finally the performance is estimated as the mean of the total number of tests. For a k-folded test the precision *p* and the recall *r* are defined as

$$p = \frac{1}{n} \sum_{i=1}^{k} p_i$$

$$r = \frac{1}{n} \sum_{i=1}^{k} r_i$$

where *pi* and *ri* are the precision and recall for each of the k tests. This Research has shown that k = 10 are a satisfactory total, therefore 10-fold cross validation was used throughout the experiments in this thesis.

The predictive performance of the classifiers for weka, Rapid Miner is shown below [1][2][10].

Table 1 depicts the results obtained for the dataset using WEKA software. Three classifier algorithms viz. J48, MLP, Simple logistic were employed and the above tabulated results have been obtained. The J48 took less time to build the model and also had pretty good prediction accuracy associated with it. The number of correctly and incorrectly classified instances associated with each of the classifiers could also be seen from the table.

TABLE I
WEKA: EVALUATION CRITERIA

Evaluation Criteria	Classifiers		
	J48	MLP	Simple Logistic
Time taken to build the Model	0.05	44.13	4.38
Correctly Classified Instances	187	184	184
Incorrectly Classified Instances	13	16	16
Prediction Accuracy	93 %	92 %	92 %

TABLE: II
RAPID MINER: EVALUATION CRITERIA

Evaluation Criteria	Classifiers		
	NB	MLP	LDA
Time taken to build the Model	8	8	1.23
Error Rate	0.936	0.651	0.762
Prediction Accuracy	90%	93%	92%

Finally, in case of the Rapid Miner, the NB, MLP and LDA classifier algorithms were used for classifying the dataset. The results can be characterized to having pretty low time taken to build the model and good prediction accuracy. However the error rate was observed to be on the slightly higher end. This is clearly seen from Table 2.

Thus various criteria have been used for evaluation of the classifiers. Having evaluated the classifiers for a trained and established dataset, efforts were assiduously made to examine their performance for a test dataset. The results and predictive performance of the classifiers are shown in the table. The same evaluation criteria viz. time taken to build the model, number of correctly classified instances, number of incorrectly classified instances and prediction accuracy were used during analysis. However there were no major changes in the order of precedence among the algorithms.

From Table 1, 2 it is seen that three algorithms are compared in each tool. It is important to note that the time taken for total number of instances have been varied and increased to a higher amount. Usually it is very tough to predict large dataset due to randomness in data. Hence testing for larger datasets would give us the flexibility to analyze each algorithm's real effectiveness in prediction.

VI. RESULTS AND DISCUSSION

To get a insightful view of the matters at hand, the final and the most important evaluation criteria was established namely the predictive accuracy. The predictive accuracy was calculated using the formula shown below.

$$\text{Prediction accuracy} = \frac{\text{Number of Correctly Classified Instances}}{\text{Total Number of Instances}}$$

$$\text{Total Number of Instances} = \text{Correctly Classified Instances} + \text{Incorrectly Classified Instances}$$

The predictive accuracy is a parameter that delineates how accurate an algorithm predicts the required data.

The performance of the datasets were evaluated which was based on the three criteria namely, the prediction accuracy, learning time and error rate. The results of the experiments are given below:

Result obtained using Weka

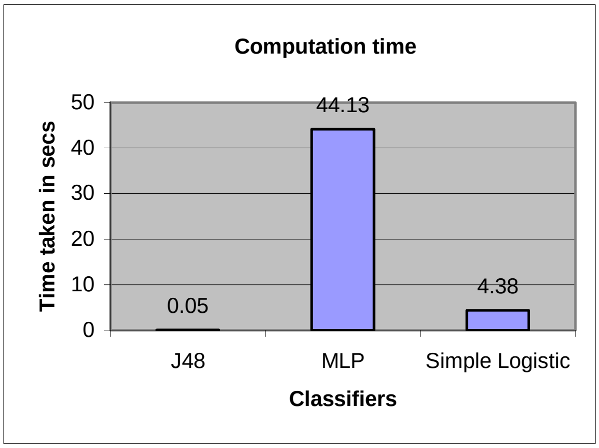

Fig. 2 Time taken to build the model

The time taken to build the model gives an idea on how fast the classifier works on he given dataset. In the above figure, the time taken to build model is plotted in the shape of a bar graph and compared for various algorithms. For this criterion j48 took the least time and hence it is the useful in time critical applications where the time required to build the model plays a significant role in its efficiency.

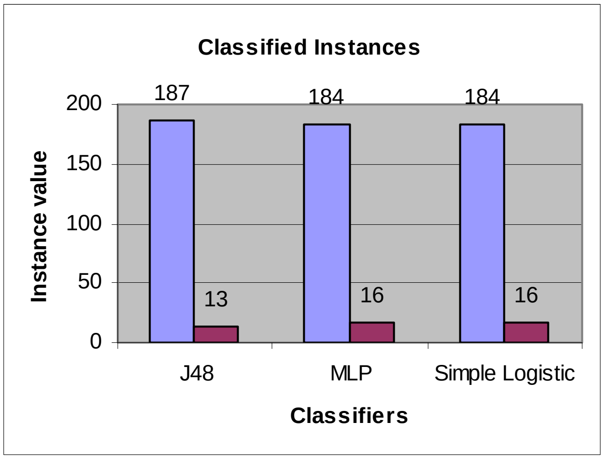

Fig. 3 Classified Instances

In figure 3, The parameters 'The Number of Correctly classified Instances' and 'The Number of incorrectly classified instances' have been compared for various classifier algorithms in form of a bar diagram. The blue bar gives the correctly classified instances and the red bar stands for the incorrectly classified instances. Since the same dataset has been used for comparing the performance of all three algorithms, the total number of instances in each of the three cases should be logically the same in number. This is verified from the diagram. We find that in each of the three cases, the number of instances in the blue bar and the red bar sums up to an equal number. Hence this facilitates a direct comparison among the classifier algorithms. As expected, using the J48 algorithm resulted in a high number of correctly classified instances and a correspondingly lower number of incorrectly classified instances. The performance of the other two classifiers was similar. However, it is worth reminding that the MLP took a larger time to build compared to Simplistic logic, though both displayed similar performance characteristics.

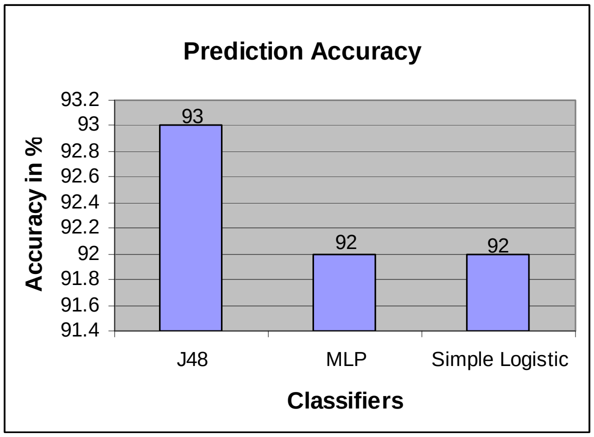

Fig. 4 Prediction Accuracy

The most critical parameter among the evaluation criteria is the prediction accuracy. The Prediction accuracy shows how accurate an algorithm can predict the required data. The predictive accuracy for various algorithms is shown in the above graph. We know that the algorithm that has higher number of correctly classified instances should have the highest accuracy. Hence, as expected, J48 that had the highest number of correctly classified instances has the highest predictive accuracy too. From the above graph this can be vividly observed. Hence the result obtained in the figure # is corroborative to what has been shown in this figure.

Result obtained using Rapid Miner

Coming to rapid miner software, the evaluation parameters can be analyzed as follows

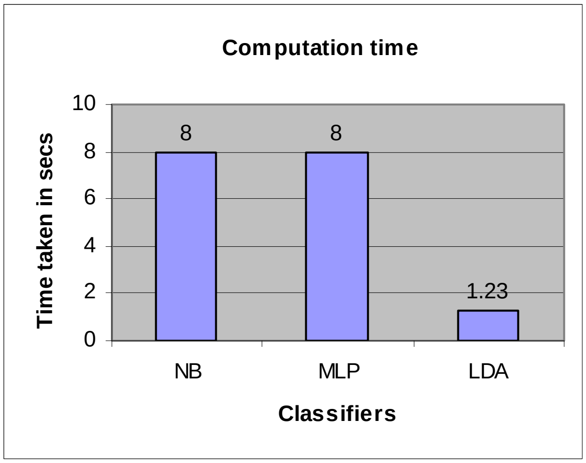

Fig. 5 Computation Time

The above diagram lists the computation time taken by each of the algorithm to build the model of the dataset. The LDA algorithm consumes very less time among its peers. The NB an MLP classifiers are not very time efficient when run using the Rapid miner software.

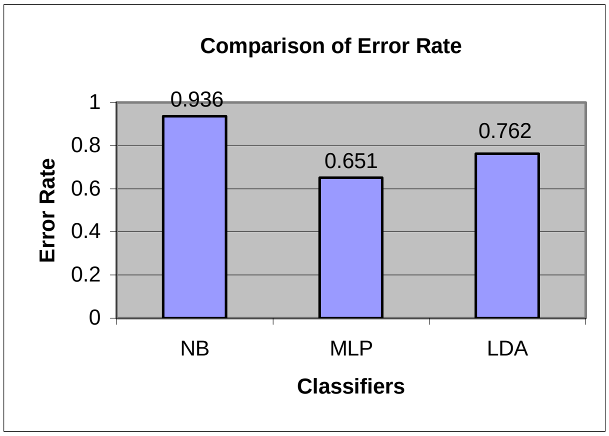

Fig. 6 Comparison of Error Rate with classifiers

Although the MLP takes some time to build the model, when it comes to comparing the error rates, it has the least value. LDA, that took the minimum time to build the model was comparatively more error prone. The NB algorithm took more time to build and also was prone to errors and this was evident from its high error rates. The figure above depicts the same.

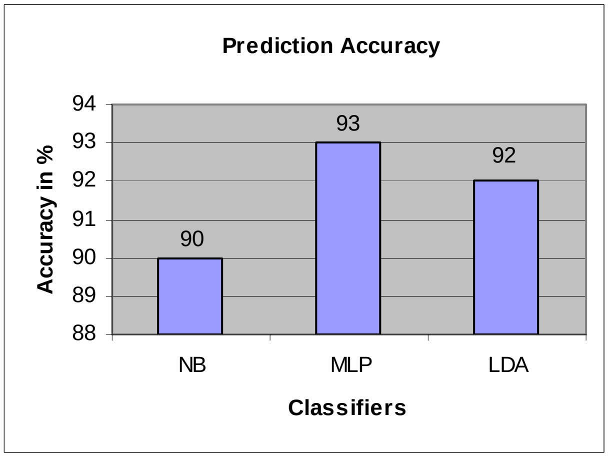

Fig. 7 Prediction Accuracy

The prediction accuracy of the algorithms are thus easily estimated and compared. The MLP algorithm, with the low error rate, thus has the highest prediction accuracy. The LDA comes as the close second followed by the NB classifier.

VII. CONCLUSION

Thus through this paper a comprehensive analysis of various classifiers using different software tools viz. WEKA, RapidMiner was implemented on a common dataset. The results were compared based on a fore mentioned evaluation criteria. The study revealed that the same classifier performed dissimilarly when run on the same dataset but using different software tools. Some of those classifiers to different software tools for one would expect the classifiers to be consistent as the test was done on the same dataset. Classifier like LDA is a good example. However some classifiers like NB and Simple Logistic performs well. But when it is compared with MLP it seems not to be better. Thus from all perspectives MLP were top performers in all cases and thus can be deemed consistent. Further it is observed that for this dataset the error rate irrespective of the classifier for MLP yielded excellent error rates compared to other algorithms.

REFERENCES

[1] Witten I. & Frank E., 2000 *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Mateo.

[2] M. Embrechts, B. Szymanski, K. Sternickel, T. Naenna, and R. Bragaspathi, 2003. Use of Machine Learning for Classification of Magnetocardiograms, *Proc. IEEE Conference on System, Man and Cybernetic*s, Washington DC.

[3] Cohen, W. 1996. Learning rules that classify e-mail*. In Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*. Palo Alto, California.

[4] Sebastiani, F. 2002 *Machine learning in automated text categorization. ACM Computing Surveys* 34, 1, 1–47.

[5] Patrick Pantel and Dekang Lin. Spamcop: 1998. A spam classification and organization program. *In Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[6] C. Pu and S. Webb. 2006. Observed trends in spam construction techniques: A case study of spam evolution. *In Proc. of the 3rd Conf. on E-Mail and Anti-Spam.*

[7] Ferris 2003. *Research. Spam Control: Problems & Opportunities*.

[8] Perkins, A. The classification of search engine spam. http://www. ebrand management.Com/ white paper rs/spam classification/.

[9] Kiritchenko, S., Matwin, S., and Abu-Hakima, S. 2004. Email Classification with Temporal Features. *Proceedings of the International Intelligent Information Systems* (IIS04), Zakopane, Poland, 523-533.

[10] James Carpinter, Ray Hunt, Tightening the net: A review of current and next generation spam filtering tools, Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand computers & security 25(2006) 566 – 578, journal homepage: www.elsevier.com/locate/cose, ELSEVIER.

[11] Duncan Cook, Catching Spam before it arrives: Domain Specific Dynamic Blacklists, Australian Computer Society, 2006, ACM.

[12] D. Puniškis, R. Laurutis, R. Dirmeikis, An Artificial Neural Nets for Spam e-mail Recognition, electronics *and electrical engineering ISSN 1392 – 1215 2006. Nr. 5(69)*

[13] Youn and Dennis McLeod, " A Comparative Study for Email Classification, Seongwook Los Angeles" , CA 90089, USA, 2006.

[14] Bekker, S 2003, Spam to Cost U.S. Companies $10 Billion in 2003, ENT News, viewed May 11 2005, ttp://www.entmag.com/news/article.asp?EditorialsID =5651>.

[15] Levent Ozgur, Tunga Gungor, Fikert Gurgen, Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish, Elsevier, 2004.

[16] Julia Itskevitch. Master Thesis, A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Computing Science, Simon Fraser University, "Automatic Hierarchical E-Mail Classification Using Association Rules", July 2001.

[17] Gauthronet, S & Drouard, E 2001, Unsolicited Commercial Communications and Data Protection, Commission of the European Communities.

[18] James Clark, Irena Koprinska, Josiah Poon, A Neural Network Based Approach to Automated E-mail Classification.

[19] Ian Stuart, Sung-Hyuk Cha, and Charles Tappert, A Neural Network Classifier for Junk Email. Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 7th, 2004

[20] D. Puniškis, R. Laurutis, R. Dirmeikis, An Artificial Neural Nets for Spam e-mail Recognition, electronics *and electrical engineering ISSN 1392 – 1215 2006. Nr. 5(69)*

[21] Duhong Chen, Tongjie Chen, and Hua Ming, Spam Email Filter Using Naïve Bayesian, Decision Tree, Neural Network, and AdaBoost

[22] Christine E. Drakeand Jonathan J. Oliver, Eugene J. Koontz. Anatomy of a Phishing Email. Proceedings of the First Conference on Email and Anti-spam (CEAS), 2004.