SPAM CLASSIFICATION BASED ON SUPERVISED LEARNING USING MACHINE LEARNING TECHNIQUES

T. Hamsapriya[1], D. Karthika Renuka[2] and M. Raja Chakkaravarthi[3]
Department of Information Technology, PSG College of Technology, Tamil Nadu, India
E-mail: [1]hamsapriya.t@gmail.com, [2]karthirenu@yahoo.com and [3]chakkaravarthiraja@ymail.com

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. Spam emails are invading users without their consent and filling their mail boxes. They consume more network capacity as well as time in checking and deleting spam mails. The vast majority of Internet users are outspoken in their disdain for spam, although enough of them respond to commercial offers that spam remains a viable source of income to spammers. While most of the users want to do right think to avoid and get rid of spam, they need clear and simple guidelines on how to behave. In spite of all the measures taken to eliminate spam, they are not yet eradicated. Also when the counter measures are over sensitive, even legitimate emails will be eliminated. 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. The effectiveness of the proposed work is 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:

Machine Learning, J48, MLP, Naive Bayesian, Spam Classification, FBL, Feature Subset Selection

1. INTRODUCTION

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 [1]. In recent years, spam email or more properly, Unsolicited Bulk Email (UBE) is a widespread problem on the Internet. Spam email is so cheap to send, that unsolicited messages are sent to a large number of users indiscriminately. When a large number of spam messages are received, it is necessary to take a long time to identify spam or non-spam email and their email messages may cause the mail server to crush.

To solve the spam problem, there have been several attempts to detect and filter the spam email on the client-side. In previous research, many Machine Learning (ML) approaches are applied to the problem, including Bayesian classifiers as Naive Bayes, C4.5 and Support Vector Machine (SVM) etc [2]. In these approaches, Bayesian classifiers obtained good results by many researchers so that it widely applied to several filtering software's. However, almost approaches learn and find the distribution of the feature set in only the spam and the non-spam messages. Today, there are many type of spam email, for example, advertisements for the purpose of making money or selling something, urban legends for the purpose of spreading hoaxes or rumors etc. Moreover, there are HTML mails contains web bug which is a graphic in an email message designed to monitor who is reading the message. Therefore, some of spam mails are judged to be non-spam email even if we use the existing filtering techniques. 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.

2. 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 users 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)[3]. According to a series of surveys conducted by CAUBE.AU 1, the number of total spasm 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)[4]. 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 [5] 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 [6] 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.

3. DATASET DESCRIPTION

The dataset that has been used for this work was acquired over a two months from various e-mail_ids. Around 57 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.

Datasets for machine learning techniques can be gathered from UCI Machine Learning Repository. Spam dataset collected from UCI consists of data extracted from 4601 email messages. Each instance in Spam dataset consists of 58 attributes. Most of the attributes represent the frequency of a given word or character in the email that corresponds to the instance.

- Word freq *w:* 48 attributes describing the frequency of word w, the percentage of words in the email.
- Char freq *c:* 6 attributes describing the frequency of a character c, defined in the same way as word frequency.
- Char freq cap: 3 attributes describing the longest length, total numbers of capital letters and average length.
- Spam class: the target attribute denoting whether the email was considered spam or no spam.

4. METHODOLOGY

For analyzing real time dataset and to predict the performance, the supervised learning algorithms were adopted here [7]. 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. Classifier Selection 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. 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.

4.1 CLASSIFICATION ALGORITHMS

The text classification techniques have been used to filter spam emails. It includes keyword-based, phrase-based, and character-based studies. Machine learning for spam classification has been proposed for filtering spam emails. WEKA is a collection of machine learning algorithms implemented in Java. A comparative analysis among different learning algorithms for classifying spam messages from e-mail are done through WEKA tool.

The dataset gathered from UCI repository has 2788 legitimate and 1813 spam emails received during a period of several months. Using this dataset as training dataset, models are build for classification algorithms.

- MLP classifier
- J48 classifier
- Naïve Bayesian Classifier

4.1.1 Multilayer Perceptron (MLP)-classifier:

A multilayer perceptron is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. The multilayer perceptron consists of three or more layers an input and an output layer with one or more hidden layers. Learning through back propagation occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result.

Neural networks have been attracting more and more researches since the past decades. In recent years there has been a shift towards the use of artificial neural networks for image classification because machine learning has the ability to learn complex data structures and approximate any continuous mapping. They have the advantage of working fast even with large amount of data. The BPNN has generalized capability in solving different problems. Back propagation is a structure of small processing units called neurons connected in a systematic manner. The back propagation neural networks, also known as multi layer perceptron. The neurons are arranged in layers typically there is one input layer, one or more hidden layers and one layer for output neurons which is interconnected to the following layer. Each neuron has its associated weight. By adjusting the weights during the training, the actual result is compared with target value to perform the classification.

4.1.2 J48-classifier:

J48 builds decision trees from a set of training data using the concept of information entropy. J48 examines the normalized information gain that results from choosing an attribute for splitting the data. 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 classifier recursively classifies until each leaf is pure, meaning that the data has been categorized as close to perfectly as possible. J48 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1,s_2,...$ of already classified samples. Each sample $s_i = x_1,x_2,...$ is a vector where $x_1,x_2,...$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1,c_2,...$ where $c_1,c_2,...$ represent the class to which each sample belongs. At each node of the tree, J48 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest

normalized information gain is chosen to make the decision. The J48 algorithm then recurs on the smaller sublists.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, J48 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, J48 creates a decision node higher up the tree using the expected value.

4.1.3 Naive Bayes-classifier:

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. Naïve Bayes classifies can handle an arbitrary number of independent variables whether continuous or catrgorial. Given a set of variables, X={X1, X2…..Xd}, we can construct the posterior probability for the event Cj among a set of possible outcomes C = {c1, c2……cd}

$$p(X \mid C_j) \alpha \prod_{k=1}^{d} p(xk \mid C_j) \quad (1)$$

Now rewrite the posterior as,

$$p(C_j \mid X) \alpha \; p(C_j) \prod_{k=1}^{d} p(xk \mid C_j) \quad (2)$$

Using Bayes rule, we can label the new case with a class C_j that achieves the highest posterior probability.

5. FBL ALGORITHM

Filtered Bayesian Learning (FBL) Algorithm is used to increase the performance of Naive Bayes-classifier. The additional flow required by FBL to classify instances is represented in Fig.1. It filters out the dependent attributes of a given dataset, as a result, the set of attributes used to represent the data is modified. Then it transforms the original data set so it complies with the new representation. The Naive Bayes Classifier works under the assumption of independent attributes, and that is why we perform a first stage where we detect all the dependencies between attributes for a later processing, trying to achieve a representation free of dependent attributes. This is performed at the first stage called "Dependency Analysis" [8]. The complete dependency search and clean algorithm can be decomposed into four main steps

- Definitions and initialization
- Dependencies analysis
- Dependency based filtering
- IG based filtering

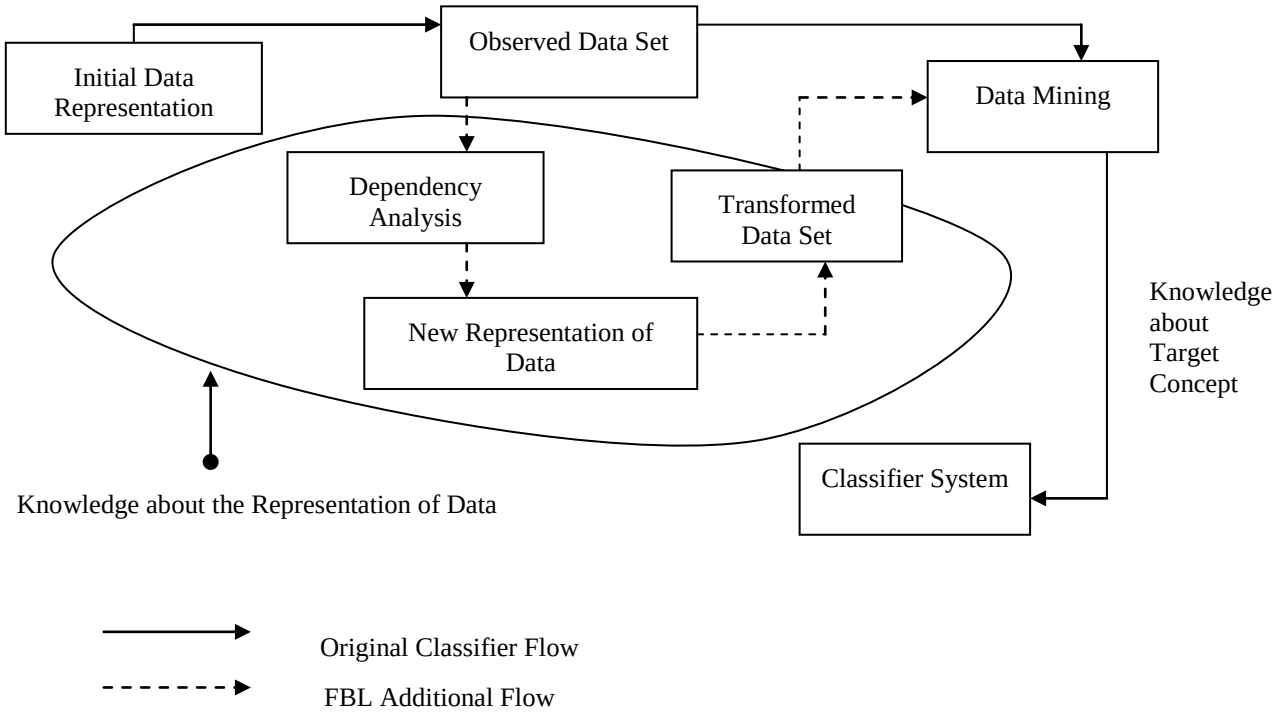

Fig.1. Naive Bayes original data flow and FBL extended flow for performing a complete classification

6. 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.

6.1 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.

6.2 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,

$$p = n_{ss} / \left(n_{ss} + n_{gs}\right) = 1 / \left(1 + \left(n_{gs} / n_{ss}\right)\right) \quad (3)$$

$$r = n_{ss} / \left(n_{ss} + n_{sg}\right) = 1 / \left(1 + \left(n_{sg} / n_{ss}\right)\right) \quad (4)$$

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 [9]. Therefore it is necessary to use a strategy to obtain a combined score. One way to achieve this is to use weighted accuracy.

6.3 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 kfolded test the precision *p* and the recall *r* are defined as,

$$p = \frac{1}{n}\sum_{i=1}^{k} P_i \quad (5)$$

$$r = \frac{1}{n}\sum_{i=1}^{k} R_i \quad (6)$$

where, p_i and *r*i 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.

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 Naive Bayes took less time to build the model and J48 has pretty good prediction accuracy. The number of correctly and incorrectly classified instances associated with each of the classifiers could also be seen from the table.

Table.1. Weka: Evaluation Criteria

Evaluation Criteria	**Classifiers**		
	J48	**Naïve Bayes**	**MLP**
Time taken to build the Model	0.06	0.02	9.48
Correctly Classified Instances	4233	4095	4279
Incorrectly Classified Instances	368	506	322
Prediction Accuracy	92%	89%	93%

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.1. 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 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.

7. 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.

$$P.A = \frac{\text{Number of Correctly Classified Instances}}{\text{Total Number of Instances}} \quad (7)$$

Total Number of Instances = Correctly Classified Instances+ Incorrectly Classified Instances

P.A = Prediction Accuracy

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 result of the experiments in WEKA Tool is shown in Fig.2 & Fig.3.

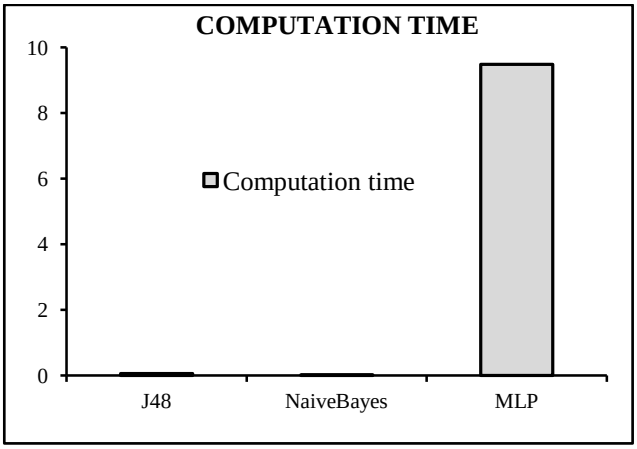

Fig.2. Time taken to build the model

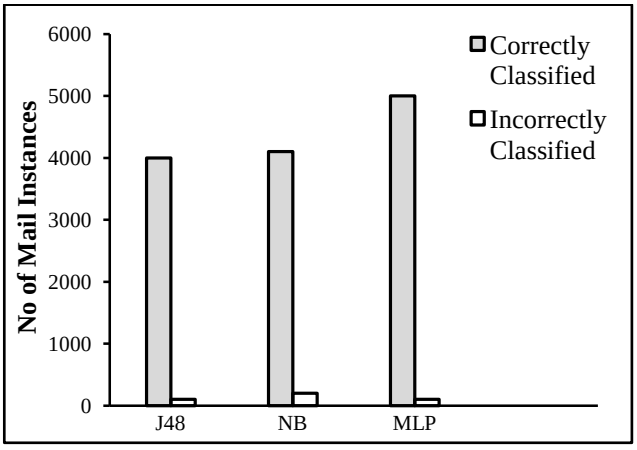

Fig.3. Classified Instances

8. RESULT AND EVALUATION OF FBL ALGORITHM

The effectiveness of FBL can be evaluated by comparing the FBL attributes selection to the best possible subset selection. Each instance in Spam dataset consists of 58 attributes, to improve the performance of naive bayes FBL is used which removes 12 attributes which are dependent to one another .FBL is able to find a subset of attributes that allows the Naive Bayes Classifier to perform better than using the original ones. Due to the usage of FBL algorithm accuracy is improved and number of attributes is reduced in Naive bayes[8]. The results and predictive performance of the classifiers Naive Bayes after apply FBL algorithm is shown in the Table.2.

Table.2. FBL: Evaluation Criteria

Evaluation Criteria	Classifiers Naive Bayes	
	Before	After
Attributes	58	45
Correctly Classified Instances	4095	4187
Incorrectly Classified Instances	506	414
Prediction Accuracy	89%	91%

9. CONCLUSION AND FUTURE WORK

Thus through this paper a comprehensive analysis of various classifiers using WEKA, 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 Naive Bayes is a good example. However some classifiers like J48 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 .In our work in order to increase the performance of the Naive Bayes FBL algorithm is used to produce better result.

ACKNOWLEDGMENT

The authors gratefully acknowledge our Institution and our Department for their valuable support.

REFERENCES

[1] C. Pu and S. Webb, "Observed trends in spam construction techniques: A case study of spam evolution", *Proceeding of 3rd Conference on E-Mail and Anti-Spam*, 2006.

[2] M. Embrechts, B. Szymanski, K. Sternickel, T. Naenna, and R. Bragaspathi,"Use of Machine Learning for Classification of Magnetocardiograms", *Proceedings of IEEE Conference on System, Man and Cybernetics, Washington DC*, pp. 1400-05, 2003.

[3] Duncan Cook, Jacky Hartnett, Kevin Manderson and Joel Scanlan, "Catching Spam before it arrives: Domain Specific Dynamic Blacklists", *in ACSW Frontiers, Australian Computer Society*, Vol. 54, pp. 193 – 202, 2006.

[4] Bekker S, "Spam to Cost U.S. Companies $10 Billion in 2003",ENTNews, http://www.entmag.com/news/article.asp?EditorialsID=5651>.

[5] D. Puniškis, R. Laurutis and R. Dirmeikis, "An Artificial Neural Nets for Spam e-mail Recognition", *Electronics and electrical engineering*, Vol. 69, No. 5, pp. 73 – 76, 2006.

[6] Youn and Dennis McLeod, "A Comparative Study for Email Classification", *Proceedings of International Joint Conferences on Computer, Information, System Sciences and Engineering*, 2006.

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

[8] Jose C. Cortizo and Ignacio Giraldez, "Discovering Data Dependencies in Web Content Mining", *Proceedings of IADIS International Conference* WWW/Internet *iteration*, 2004.

[9] Upasana Pandey and S. Chakraverty "A Review of Text Classification Approaches for E-mail Management", in *IACSIT International Journal of Engineering and Technology*, Vol. 3, No. 2, 2011.