Journal of Network and Computer Applications 32 (2009) 357–366

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

An innovative analyser for multi-classifier e-mail classification based on grey list analysis

Md Rafiqul Islam [a,*], Wanlei Zhou [a], Minyi Guo [b], Yang Xiang [c]

[a] School of Engineering and Information Technology, Deakin University, Burwood, VIC 3125, Australia
[b] Department of Computer Software, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan
[c] School of Management and Information Systems, Central Queensland University, QLD 4702, Australia

ARTICLE INFO

Article history:
Accepted 20 February 2008

Keywords:
E-mail
TP
TN
Spam
FP
GL

ABSTRACT

In this paper, we propose a new technique of e-mail classification based on the analysis of grey list (GL) from the output of an integrated model, which uses multi-classifier classification ensembles of statistical learning algorithms. The GL is the output of a list of classifiers which are not categorized as true positive (TP) nor true negative (TN) but in an unclear status. Many works have been done to filter spam from legitimate e-mails using classification algorithms and substantial performance has been achieved with some amount of false-positive (FP) tradeoffs. However, in spam filtering applications the FP problem is unacceptable in many situations, therefore it is critical to properly classify e-mails in the GL. Our proposed technique uses an innovative analyser for making decisions about the status of these e-mails. It has been shown that the performance of our proposed technique for e-mail classification is much better than the existing systems, in terms of reducing FP problems and improving accuracy.

1. Introduction

Spam has become one of the biggest world wide problems facing the Internet today. The Internet is becoming an integral part of our everyday life and the e-mail has been a powerful tool for idea and information exchange, as well as for users' commercial and social lives. Due to the increasing volume of spam, the users as well as internet service providers (ISPs) are facing a lot of problems. The cost to corporations in bandwidth, delayed e-mail, and employee productivity has become a tremendous problem for anyone who provides e-mail services. However, it is amazing that despite the increasing development of anti-spam services and technologies, the number of spam messages continues to increase rapidly.

E-mail classification techniques are able to control the problem in a variety of ways. Detection and protection of spam e-mails from the e-mail delivery system allows end-users to regain a useful means of communication. Many researches on content-based e-mail classification have been centred on the more sophisticated classifier-related issues. Currently, applying machine learning techniques for spam classification is an important research issue. The success of machine learning techniques in text categorization has led researchers to explore advanced learning algorithms in spam filtering (Zhang et al., 2003; Drucker et al., 2002; Islam et al., 2005a, b; Ali and Xiang, 2007; Cheng and Li, 2006; Zi-Qiang Wang et al., 2006; Koprinska et al., 2007).

In this paper, we proposed an effective and efficient e-mail classification technique by adopting a grey list (GL) analyser through an integrated classification system. The main focus of our paper is to generate a list of misclassified e-mails called GL e-mails by classification ensembles technique and analyse them by an analyser. The GL is the list of e-mails which are not considered as true positive (TP) nor true negative (TN). The term GL is related to black-list (BL) and white-list (WL) and considered as the middle of them, i.e. not sure which one to belong. In our proposed system, the GL is considered as a list of e-mails where no uniform decision comes from all the classifiers. The analysis of GL e-mails are based on two premises: (i) user feedback technique, i.e. the user will give feedback to the analyser about the status of these e-mails and (ii) sender verification technique, i.e. the system will send the e-mail back to the sender and wait for a certain timeframe. If response comes within the predefined timeframe then it will be treated as a TP otherwise it will be treated a TN. This technique is so-called rule-based challenge/response (C/R) technique.

The organization of the rest of the paper is as follows: Section 2 outlines the related work on e-mail classification techniques and Section 3 describes the GL generating process. Section 4 presents details of analysing the GL and Section 5 presents the experimental results. Finally, the paper ends with conclusion in Section 6.

* Corresponding author.
E-mail addresses: rmd@deakin.edu.au, mdrafiqul@gmail.com (M.R. Islam), wanlei@deakin.edu.au (W. Zhou), minyi@u-aizu.ac.jp (M. Guo), y.xiang@cqu.edu.au (Y. Xiang).

1084-8045/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.jnca.2008.02.023

2. Related work

This section briefly overview different e-mail classification techniques, including rule-based techniques and content-based techniques.

2.1. Rule-based techniques

Rule-based filtering techniques use a set of rules to classify e-mail as spam e-mail or legitimate e-mail and they can be applied at either the mail user agent (MUA) level or the mail transfer agent (MTA) level. E-mail clients contain an element at the MUA level for categorizing e-mail based on a set of rules determined by the user. These rules can be constructed to examine an e-mail message's header and body, for keywords or phrases given by the end-user. A common use of such rules is to categorize newly arrived e-mail into a specific folder. For example, the user could create a folder called spam and define a number rules that would transfer a newly arrived e-mail to the spam folder if it were triggered. Such rules could look for specific words in the content of the e-mail, look for punctuation being used in the subject of the e-mail, or note the content type of the e-mail. While this technique does work well, it also has a serious problem. The rule set needs constant updating and refinement because most spammers use obfuscation techniques. For example, one of the common obfuscation used is misspelling words.

Filtering at the MTA level can achieve some economies of scale but it also triggers some problems. Since by nature, spam is sent in bulk, blocking the sender can dramatically reduce the number of spam needed to be stored and delivered. Some of the techniques described for MUA rule-based filtering can be applied at the MTA level (Islam et al., 2005b; Islam and Chowdhury, 2005; Islam and Zhou, 2007; Liu and Yu, 2005; Ali and Xiang, 2007; Zi-Qiang Wang et al., 2006; Kong et al., 2006).

2.1.1. White-list

WL is an MUA level rule-based filtering technique, where a WL is a register containing a collection of contacts from which e-mail messages can be accepted. If an e-mail arrives but does not come from one of the contacts in the WL, then it is treated as spam and placed in the spam folder. While this technique is effective for some users, it has also drawbacks. Any e-mail sent by a stranger will simply be incorrectly classified as FP. However, there is a scheme that incorporates a C/R mechanism to allow new senders to be added to a user's WL.

2.1.2. Black-list

BL contains lists of known spammers. Essentially when a user gets spam, the user adds the sender of the spam to the BL. The entire domain of the sender of the spam can be added to the BL. Newly arrived e-mails are checked, and if the sender is on the BL, the e-mail is automatically classified as spam. The major problem stems from the fact that spammers tend to forge header information in their spam. The sender information is generally forged, meaning that perhaps innocent people are added to a BL but more importantly the effect which the BL will have is diminished dramatically.

A distributed BL is a network tool for anti-spam engines. Distributed BLs maintain a collection of common spam messages on a central server. The filter is shared amongst the subscribers, so if one person identifies a message as spam then all others benefit. When a message arrives, it is compared to the digest of known spam and deleted if a match is found. This method is low in FP, but false-negatives (FNs) tend to be high so often another filtering technique is required to work in conjunction. The central repository must be maintained by an unbiased organization (Islam et al., 2005a, b; Islam and Chowdhury, 2005; Islam and Zhou, 2007).

2.2. Content-based techniques

Spam typically has a distinctive content, which should be able to distinguish from legitimate e-mail. Categorizing e-mail based on its content seems like a logical progression from simplistic rule-based approaches. This would help reduce error rates as legitimate e-mail would not be blocked even if the ISP from which it originated, is on a real-time block list. In addition, the presence of a single token should not cause the e-mail to be classified as spam.

2.2.1. Using classification algorithms

Classification algorithms such as support vector machine (SVM), Naïve Bayes (NB) and Boosting, etc., are used for content-based spam filtering. Each algorithm can be viewed as searching for the most appropriate classifier in a search space that contains all the classifiers it can learn. Classification algorithms require the instance representation to classify the contents of the e-mails. The instances are messages and each message is transformed into a vector $(x_1,\ldots, x_m)$, where $x_1,\ldots, x_m$ are the values of the attributes $X_1,\ldots, X_m$, much as in the vector space model in information retrieval (Zhang et al., 2003; Drucker et al., 2002). In the simplest case, each attribute represents a single token (e.g., "money"), of Boolean variables (1-contains token/0-otherwise). Instead of Boolean attributes, another two attribute vector representations such as frequency attributes and N-gram attributes are also considered (Islam and Chowdhury, 2005; Liu and Yu, 2005). The key concepts of e-mail classification using machine learning algorithms can be categorized into two classes, $y_i \in \{-1,1\}$, and there are N labelled training examples: $(x_1, y_1),\ldots,(x_n, y_n)$, $x \in R^d$ where d is the dimensionality of the vector (Drucker et al., 2002; Islam et al., 2005a, b; Islam and Zhou, 2007; Cristianini and Shawe-Taylor, 2000; Hunt and Carpinter, 2006; Zi-Qiang Wang et al., 2006).

SVM is a new learning algorithm which has some attractive features, such as eliminating the need for feature selections, which makes for easier e-mail classification. SVMs are a range of classification and regression algorithms that have been based on the Structural Risk Minimization (SRM) principle from statistical learning theory formulated by Vapnik (Drucker et al., 2002; Cristianini and Shawe-Taylor, 2000). The SVM aims to select the hyperplane that separates the training instances (messages) of the two categories with maximum distance. This target hyperplane is found by selecting two parallel hyperplanes that are each tangential to a different category—that is, they include at least one training instance of a different category, whilst providing perfect separation between all the training instances of the two categories. The training instances that lie on, and thus define the two tangential hyperplanes are the support vectors. The distance between the two tangential hyperplanes is the margin. Once the margin has been maximized, the target hyperplane is in the middle of the margin (Drucker et al., 2002; Islam and Zhou, 2007; Cristianini and Shawe-Taylor, 2000; Hunt and Carpinter, 2006). The SRM is to find an optimal hyperplane that can guarantee the lowest true error.

SVM is based on the idea that every solvable classification problem can be transformed into a linearly separable one by mapping the original vector space into a new one, using non-linear mapping functions. More formally, SVMs learn generalized linear discriminant functions such as: $f(\vec{x}) = \sum_{i=1}^{m'} w_i h_i(\vec{x}) + w_0$, where m' is the dimensionality of the new vector space, and $h_i(\vec{x})$

are the non-linear functions that map the original attributes to the new ones. The higher the order of the $h_i(\vec{x})$ functions, the less linear the resulting discriminant. The type of $h_i(\vec{x})$functions that can be used is limited indirectly by the algorithm's search method, but the exact choice is made by the person who configures the learner for a particular application. The function $f(\vec{x})$ is not linear in the original vector space, but it is linear in the transformed one (Drucker et al., 2002; Islam et al., 2005b; Cristianini and Shawe-Taylor, 2000; Koprinska et al., 2007). The NB learner is the simplest and most widely used algorithm that derives from Bayesian decision theory (Islam and Chowdhury, 2005; Islam and Zhou, 2007; Sahami et al., 1998). A Bayesian classifier is simply a Bayesian network applied to a classification task. It contains a node C representing the class variable and a node X_i for each of the features. From Bayes' theorem and the theorem of total probability $P(C=c_k|X=x)$ for each possible class c_k, the probability that a message with vector $\bar{x}=(x_1,\ldots,x_m)$ belongs in category c is

$$P(C=c|\vec{X}=\vec{x})=\frac{P(C=c)\cdot P(\vec{X}=\vec{x}|C=c)}{\sum_{c'\in\{c_L,c_S\}}P(C=c')\cdot P(\vec{X}=\vec{x}|C=c')}.$$

The boosting algorithms, like SVMs, learn generalized linear discriminates of the form of equation $f(\vec{x})=\sum_{i=1}^{m'}w_i\cdot h_i(\vec{x})+w_0$. In boosting algorithms, however, the mapping functions $h_i(\vec{x})$ are themselves learnt from data by another learning algorithm, known as weak learner. A common weak learner is decision stump induction (Islam et al., 2005b; Ali and Xiang, 2007; Koprinska et al., 2007), which constructs a one-level decision tree that uses a single attribute from the original attribute set to classify the instance $\vec{x}$ to one of the two categories. In the case of continuous attributes, the decision tree is a threshold function on one of the original attributes.

Furthermore, the mapping functions $h_i(\vec{x})$ are learnt by applying iteratively (for $i=1,\ldots,m'$) the weak learner, in our case regression stump induction, to an enhanced form of the training set, where each training instance $\vec{x}_j$ carries a weight $v_i(\vec{x}_j)$. At each iteration, the weights of the training instances are updated, and, hence, applying the weak learner leads to a different mapping function $h_i(\vec{x})$. This iterative process is common to all boosting methods, where each $h_i(\vec{x})$ can be thought of as a weak classifier that specializes in classifying correctly training instances that the combination of the previous weak classifiers $h_i(\vec{x}_k)(k=1,\ldots,i-1)$ either misclassifies or places close to the classification boundary. This is similar to the behaviour of SVMs, which focus on instances that are misclassified or support the tangential hyper planes (Islam and Zhou, 2007; Ali and Xiang, 2007; Hunt and Carpinter, 2006; Koprinska et al., 2007).

3. Proposed technique for generating GL

This section presents an innovative analyser, called GL analyser, for analysing GL e-mails produced by our multi-classifier ensembles technique. The main aim of our proposed e-mail classification technique is to reduce the false-positive (FP) problems through the use of different aspects of anti-spam filtering techniques, especially the learning-based anti-spam filter. Much work has been done using machine learning techniques for spam filtering and achieved high accuracy but the FP problems still remain, and the consequences are generally expensive in real world. One misclassified legitimate e-mail could sometimes cause great trouble to a user. We have studied extensively through different classification algorithms and found that sometimes classification algorithms vary for producing the coherent result with the same e-mail corpora. Keeping this in mind, we have proposed an innovative analyser, which will collect a classifier's output and analyse it, especially the GL output of classifiers. The foremost focus of GL analyser is to look into the ambivalent e-mails and provide a final pronouncement regarding the classification of the e-mails. This technique reduces the FP problems considerably and enhances the overall performance of a spam filtering system.

Fig. 1 shows the overview of the GL e-mail generating process for n $(n=3)$ classifiers. Every classifier generates two sets of output data, as a binary classification, one is true legitimate set (L_t) and another is true spam set (S_t) as shown in Fig. 1(a)–(c), respectively. Fig. 1(d) shows the collective multi-classifier classification approach with n $(n=3)$ classifiers. The intersection regions of three output sets of S_T and L_T represent the TN and TP, respectively, because all the three classifiers give the same result. The reaming regions of the output sets, as shown in Fig. 1(d) represent the GL e-mails, because of the diverse predictions, not exclusive pronouncements come from the n classifiers.

The total number of output e-mails and output data sets E_{out} of individual classifier from Fig. 1(a)–(c) can be represents mathematically as follows:

$$\begin{aligned}&\text{Classifier } C_1: E_{out}\Rightarrow n(Lc_1\cup Sc_1)\\&\text{Classifier } C_2: E_{out}\Rightarrow n(Lc_2\cup Sc_2)\\&\vdots\\&\text{Classifier } C_n: E_{out}\Rightarrow n(Lc_n\cup Sc_n)\end{aligned}\quad(1)$$

where $n(Lc_1\cup Sc_1)$ represents the total number of output e-mails.

In the case of multiple classifier selection, as shown in Fig. 1(d), the output of the classifiers can be categorized in the following three different sets.

True legitimate outputs L_T: This is the common legitimate output from all n classifiers and this type of output is considered as TP. Mathematically the number of outputs can be represented as follows:

$$\begin{aligned}&C_1\cup C_2: L_T\Rightarrow n(Lc_1\cap Lc_2)\\&C_1\cup C_3: L_T\Rightarrow n(Lc_1\cap Lc_3)\\&C_2\cup C_3: L_T\Rightarrow n(Lc_2\cap Lc_3)\\&C_1\cup C_2\cup C_3: L_T\Rightarrow n(Lc_1\cap Lc_2\cap Lc_3)\end{aligned}\quad(2)$$

True spam outputs S_T: This is the common spam output from all n classifiers, i.e. the intersection region of n sets, as shown in Fig. 1(d). This sort of output is considered as TN. The number of S_T and their combinations are as follows:

$$\begin{aligned}&C_1\cup C_2: S_T\Rightarrow n(Sc_1\cap Sc_2)\\&C_1\cup C_3: S_T\Rightarrow n(Sc_1\cap Sc_3)\\&C_3\cup C_2: S_T\Rightarrow n(Sc_3\cap Sc_2)\\&C_1\cup C_2\cup C_3: S_T\Rightarrow n(Sc_1\cap Sc_2\cap Sc_3)\end{aligned}\quad(3)$$

Grey list output: These are the mixed outputs from different classifiers, that is, the classifiers arrived different conclusions for the same e-mail. These sorts of output are considered neither TP nor TN but in the middle of them, so belong to the GL. The total number and their combinations of output category are as follows:

$$\begin{aligned}&C_1\cup C_2: n(Sc_1\cup Lc_2+)n(Sc_2\cup Lc_1)\\&C_1\cup C_3: n(Sc_1\cup Lc_3)+n(Sc_3\cup Lc_1)\\&C_3\cup C_1: n(Sc_3\cup Lc_2)+n(Sc_2\cup Lc_3)\\&C_1\cup C_2\cup C_3: n(Lc_3\cup(Sc_1\cap Sc_2)+Lc_1\cup(Sc_3\cap Sc_2)\\&\qquad+Lc_2\cup(Sc_3\cap Sc_1))+n(Sc_3\cup(Lc_1\cap Lc_2)\\&\qquad+Sc_1\cup(Lc_3\cap Lc_2)+Sc_2\cup(Lc_3\cap Lc_1))\end{aligned}\quad(4)$$

Since every classifier has two sets of outputs C_il and C_is. So the total number of output sets are 2^n, where n is the number of classifiers. For n classifiers, $n\geqslant 2$, the final output terms can be

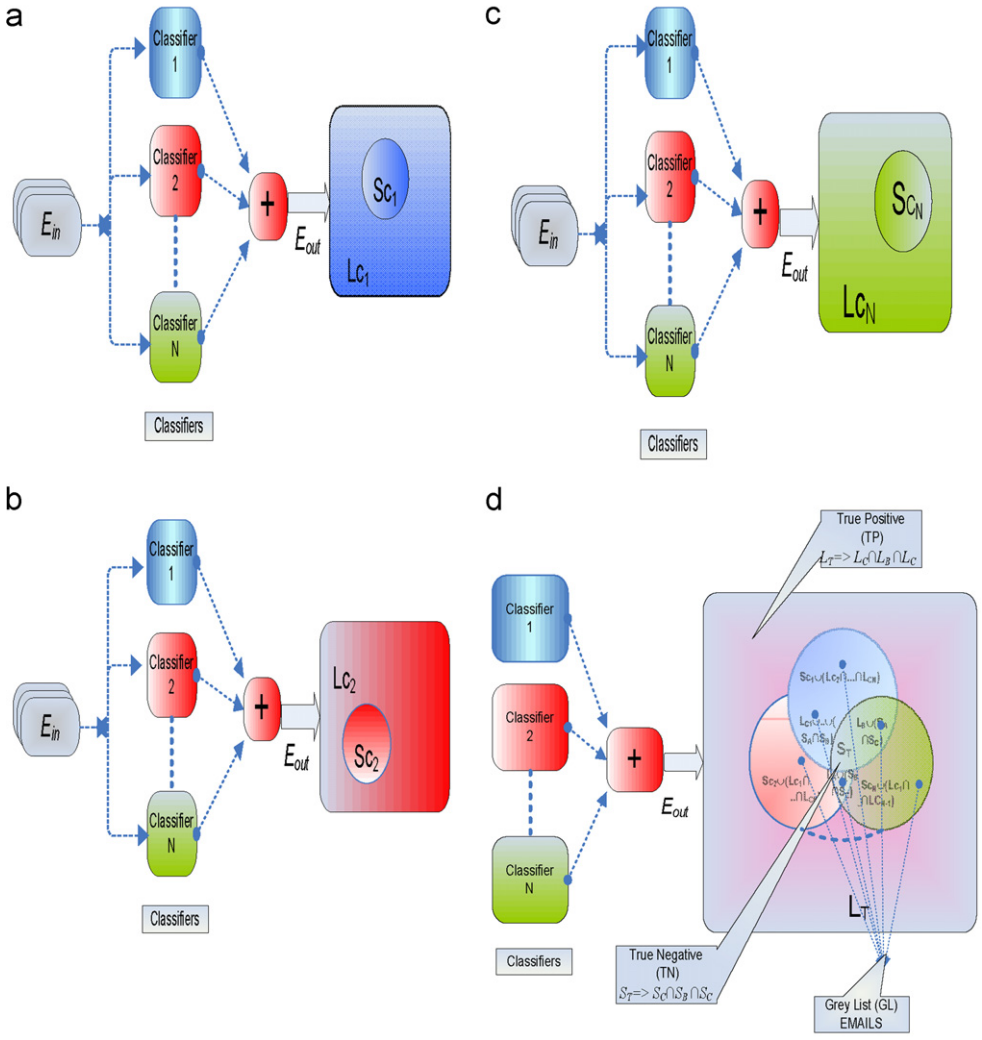

Fig. 1. Output sets of single and multiple classifiers: (a) classifier-1, produces two sets of outputs, $E_{out} = Lc_1 \cup Sc_1$; (b) classifier-2, produces two sets of outputs $E_{out} = Lc_2 \cup Sc_2$; (c) classifier-*N*, produces two sets of outputs $E_{out} = Lc_n \cup Sc_n$; and (d) multiple classifiers, where $n = 3$, produces three different sets of outputs $E_{out} = L_T \cup S_T \cup GL$.

represented using the following equations:

$$\int (C_1, C_2, \ldots, C_n) \Rightarrow \prod_{i=1}^{n} C_i(l) + \prod_{i=1}^{n} C_i(s) + \sum_{x=1}^{m} \prod_{j=1}^{p} C_j(s) \prod_{k=1}^{q} C_k(l) \tag{5}$$

where $C_1, C_2, \ldots, C_n$ are the classifiers, $j \neq k$ and $p+q = m$. From Eq. (5) we can derive the TP, TN and GL as follows:

$$\begin{aligned} \text{TP} &\Rightarrow \int (C_1, C_2, \ldots, C_n) \\ &\Rightarrow \prod_{i=1}^{n} C_i(l) \\ &\Rightarrow C_1(l) C_2(l) \ldots\ldots C_n(l) \end{aligned} \tag{6}$$

It can be represented using set theory as

$$\forall_{(l)}, y_i(l); \text{ and } y_i(l) = \{l : C_1(l) \wedge C_2(l) \wedge \cdots \wedge C_N(l)\} \tag{7}$$

$$\begin{aligned} \text{TN} &\Rightarrow \int (C_1, C_2, \ldots, C_n) \\ &\Rightarrow \prod_{i=1}^{n} C_i(s) \\ &\Rightarrow C_1(s) C_2(s) \ldots C_n(s) \end{aligned} \tag{8}$$

It can be represented using set theory as

$$\forall_{(s)}, y_i(s), \text{ and } y_i(s) = \{s : C_1(s) \wedge C_2(s) \wedge \cdots \wedge C_N(s)\} \tag{9}$$

$$\begin{aligned} \text{GL} \Rightarrow \int (C_1, C_2, \ldots, C_n) &\Rightarrow \sum_{x=1}^{m} \prod_{j=1}^{p} C_j(s) \prod_{k=1}^{q} C_k(l) \\ &\Rightarrow C_1(l) C_2(l) \cdots C_{m-1}(l) C_m(s) + C_1(l) C_2(l) \ldots \\ &C_{m-1}(s) C_m(l) + \cdots + C_1(s) C_2(s) \ldots \\ &C_{m-2}(s) C_{m-1}(l) C_m(s) + C_1(s) C_2(s) \ldots \\ &C_{m-2}(s) C_{m-1}(s) C_m(l) \end{aligned} \tag{10}$$

It can be represented using set theory as

$$\exists_{(s)}, y_i(gl) \text{ or } \exists_{(l)}, y_i(gl) \tag{11}$$

So, it is clear from the above equation that the number of GL terms increases exponentially (2^n) as the number of classifiers increases. In Eqs. (5) and (10) the term p represents the upper bound of GL terms which is $2^n - 2$. However, our experiment shows that the number of GL e-mails, in multi-classifier ensembles technique, actually depend on the selecting the classifiers and its corresponding parameters.

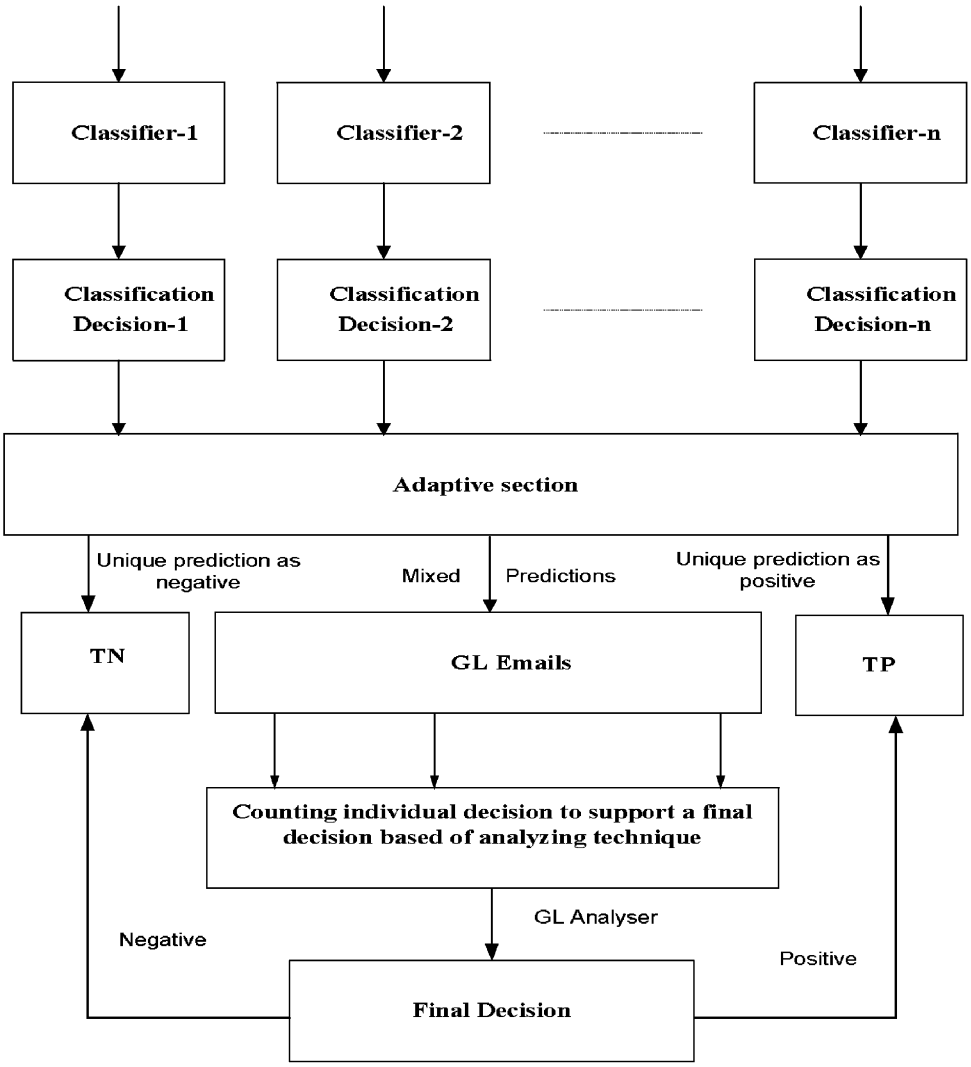

Fig. 2. Flow diagram of our proposed multi-classifier classification system.

4. Grey list analysis

This section describes the analysis of the GL e-mails generated by the multiple classifiers classification ensembles technique. The main objective of the analysis of GL e-mail is to reduce the FP problems and to achieve better accuracy. Fig. 2 shows the block diagram of the analyser. We call the component of the multiple classifiers as the adaptive section as a single e-mail can go through multiple classifiers and the categorization is depended on the output labelling, given by each classifier. The prime function of this adaptive section is to sort out the e-mail and send it to the subsequent three mailboxes, TP, TN and GL.

The analyser will collect the output of the GL e-mails from individual classifier's prediction. This process will also depend on the selection of the classifier by the user using a power user interface (PUI). In our system we use a PUI to give users the flexibility to select individual or combined classifier/(s). In the case of single classifier classification, the analysis is very simple and it is the same as existing systems (Islam et al., 2005b; Androutsopoulos et al., 2004).

But for the multi-classifier classification approach, the system will differentiate the classifiers' output into three different categories as discussed before. Actually, the outputs of the proposed systems are considered in the following three different types TP, TN and GL as shown in Fig. 2. The TP and TN are the e-mails where all classifiers come to the same conclusion but in terms of GL e-mails no uniform decision has been made so the system needs to analyse them further to make the decision. So, in our system an analyser will be used to make a final conclusion whether an e-mail in GL is spam or legitimate. Two techniques are used to analyse GL e-mails. One is the user selection technique and another is the sender verification technique. The flow diagram of the analyser is shown in Fig. 3.

As shown in Fig. 3, initially the filtering system has the preference to set the analysing procedure, whether it will be scrutinized by using user's feedback technique or sender verification technique. The first selection is the concept of personalized spam detection and the other is based on the concept of so-called C/R technique. After receiving the response from any of the techniques the system will send the e-mails to the subsequent mailboxes based on the credentials of the system. The following section describes the analysing techniques.

4.1. User selection process

It is a straightforward approach to analyse the e-mail. As considered, the user is the final source and more authentic way to

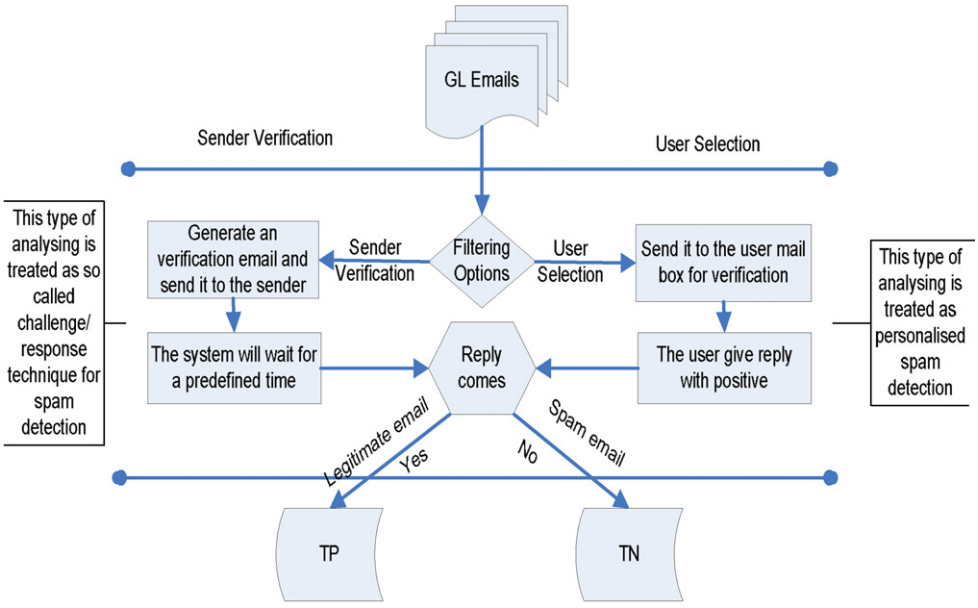

Fig. 3. GL e-mail analysing technique.

detect the e-mail whether it is spam or legitimate. It is based on the concept of personalized spam filtering technique. One of the issue here is that, this is not a collaborative approach or globalization approach to detect spam because it violets the general definition of spam (unsolicited e-mails), while adopting a principle of "somebody's spam could be the other user ham".

In this process, when the GL e-mail comes to the analyser then first the analyser will check it out to the TP database whether the header/domain exists or not, as shown in Fig. 4. If exists then the e-mail will be treated as positive and will not to be sent to the user further. Alternatively the GL e-mail will be dispatched to the user for user response in relation to the final status of this e-mail. The user will identify the e-mail and make decision whether it is spam or legitimate. After getting appropriate response from the user it will be sent to the spam or legitimate database, respectively. This user selection process is quite simple but more effective in terms of accuracy.

The underlying premise behind this technique is that users have their personal, often conflicting, opinions as to what constitutes spam, because not everyone has the same opinion of whether a given topic or e-mail is of interest to them. The key advantage of this system is its resilience and adaptability. Spam has been shown to exhibit concept drift, which is the change in the characteristic content of spam over time (due to new products like Viagra, or in response to changes in spam filters to work around them). Because there is no static knowledge base, personalized filters can respond to these changes as they occur.

4.2. Sender verification

The second option is a bit intricate compared to the first one. This process is based on what we call a C/R technique. The basic scenario is that when a stranger sends a message, the MTA/MUA will automatically respond with a challenge and until such time when the stranger responds with the correct answer to the challenge, the e-mail is not delivered. The analogy is like that: *X* sends an e-mail to *Y*, who uses a C/R system. Since *X*'s address is not yet on *Y*'s TP database so the system will automatically send a message to *X* for verification. Once the C/R system does not receive any valid response from *X*′ side within a certain timeframe, the e-mail will be classified as spam.

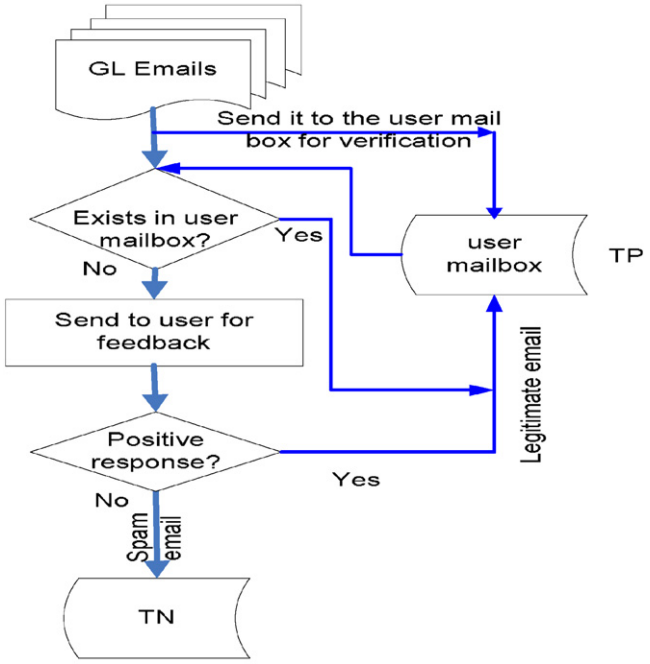

Fig. 4. Block diagram of user feedback process.

Fig. 5 illustrated the flow diagram of sender verification technique. In this technique, when the GL e-mail comes to the analyser, the analyser will check it out to the TP database whether it exists or not. If it exists in the user mailbox then it will not be sent to the sender for feedback. In that case the e-mail will be detected as legitimate and will send to the legitimate database. Alternatively the GL e-mail will be send to the sender for verification. On the other hand, If not exist in the TP database, the analyser will automatically send a message to the sender for authentication; until the sender responds with the correct answer within a certain timeframe, the e-mail will remain as GL. If the sender responds with correct answer then the e-mail is considers as TP and the user mailbox will be updated accordingly. Otherwise the message will be treated as TN and will be sent to the spam

mailbox. On the other hand, if the time expires then it is also considered as TN.

This technique has some limitations, especially when spammers use falsified e-mails addresses or have hijacked legitimate e-mails addresses. In addition to that, it requires additional software and extra time overhead for generating a verification e-mail to the sender and sender may simply not respond to the verification due to lack of motivation, which may result in FP situation.

After getting the proper labelling of GL e-mail by the analyser, the system will consider the feature of these e-mails and the classifier will be trained periodically from the newly generated features in a dynamic fashion. This is called DFS technique presented in Islam et al. (2007).

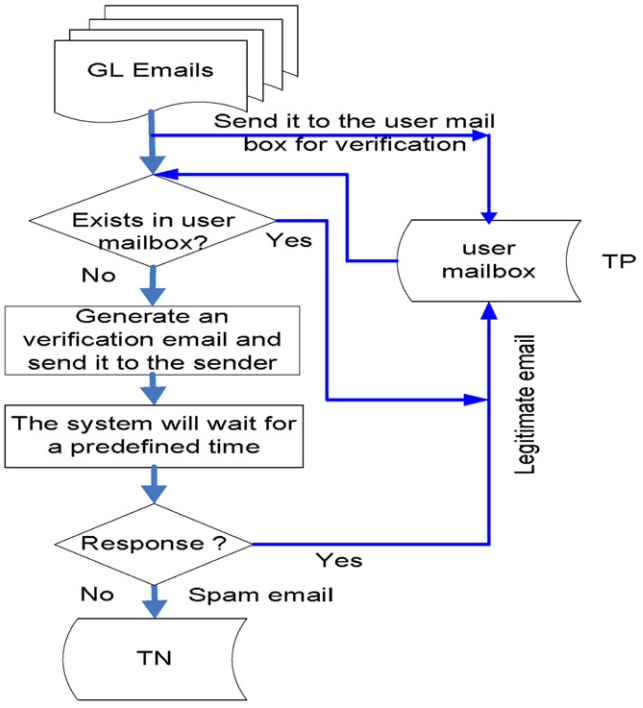

Fig. 5. Block diagram of sender verification process.

5. Experimental results

This section presents the experimental results of our proposed system. In our experiment, we have used three different classification algorithms: NB, SVM and AdaBoost. Firstly user e-mails (both spam and legitimate) are initially transformed and indexed, which is considered as an initial transformation. After initial transformation the e-mail corpus will be classified by the classifier/(s). A user interface is used in our system to give options to the user for selecting individual/combined classifiers. After e-mails are classified through classifiers they are send to corresponding mailboxes of TP, TN and GL, respectively, and e-mails in GL will be analysed further.

We used the public data sets PUA-1–2–3 (Androutsopoulos et al., 2004) in our experiments and converted the data sets based on our experimental design and environment. Firstly we have encoded the whole data sets, both train and test sets, then indexed every e-mail for test data sets and finally recorded the output according to the index value.

Table 1 shows the receiver operating characteristic (ROC) report of our experiment. In this table we only used three important measurement values in our ROC report, AUC (area under an ROC curve) estimation values a, 95% of confidence limit (CI) both lower and upper limit, and 1-sided probability values.

The AUC is a popular measure of the accuracy of experiment. Other things being equal, the larger the AUC, the better the experiment is as predicted the existence of the classification. The possible values of AUC range from 0.5 (no diagnostic ability) to 1.0 (perfect diagnostic ability). The confidence intervals (CI) option specifies the value of alpha to be used in all CI. The quantity (1-Alpha) is the confidence coefficient (or confidence level) of all CI. The *p*-value represents the hypotheses tests for each of the criterion variables.

A ROC curve shows the characteristics of our experiment by graphing the FP rate (1-specificity) on the horizontal axis and the

Table 1
The ROC report of three classifiers (SVM, AdaBoost and NB) with our proposed technique

Data sets	ROC estimation	SVM	AdaBoost	NB	Proposed
PUD1	AUC	0.88889	0.94444	0.88889	0.94444
	95% of CI	0.62324–0.97062	0.64675–0.99242	0.62324–0.97062	0.64675–0.99242
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)
PUD2	AUC	0.80159	0.92857	0.84524	1.00000
	95% of CI	0.50496–0.92888	0.55918–0.99034	0.57778–0.94877	–
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)
PUD3	AUC	0.82792	0.90909	0.74675	0.90909
	95% of CI	0.59766–0.93201	0.68239–0.97626	0.51229–0.87762	0.68239–0.97626
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)
PUD4	AUC	0.88889	0.85764	0.88889	1.00000
	95% of CI	0.62324–0.97062	0.59957–0.95415	0.62324–0.97062	–
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)
PUD5	AUC	0.84295	0.80769	0.80449	0.92308
	95% of CI	0.62434–0.93911	0.61966–0.90804	0.58805–0.91341	0.72564–0.98007
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)
PUD6	AUC	0.86667	0.80000	0.78333	0.95000
	95% of CI	0.62464–0.95679	0.57441–0.91265	0.53931–0.90599	0.67868–0.99315
	p-Value	(<0.001)	(<0.001)	(<0.001)	(<0.001)

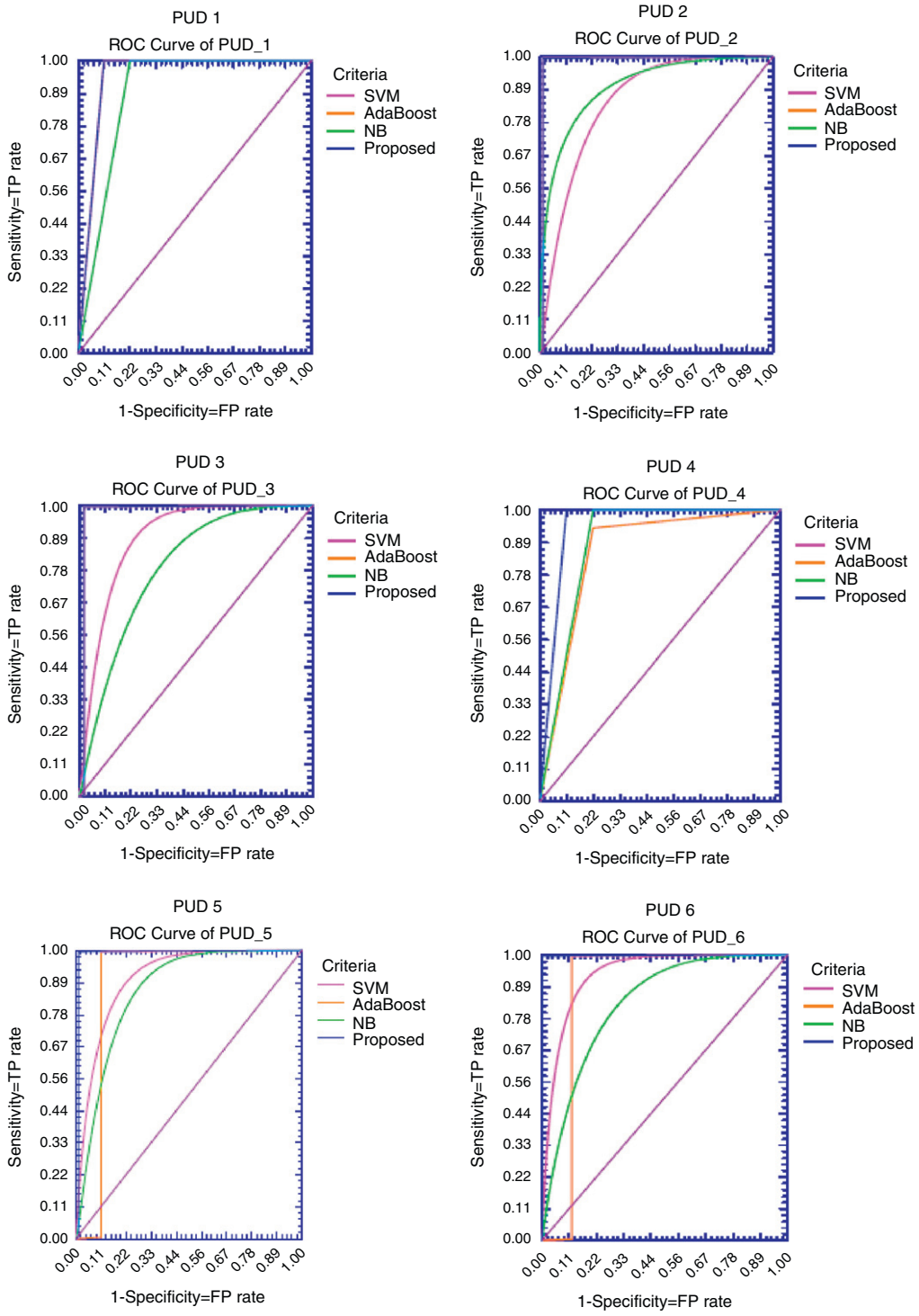

Fig. 6. ROC curve for comparative analysis of single classifier and multi-classifier classification ensembles.

true-positive rate (sensitivity) on the vertical axis for various cutoff values. Fig. 6 shows the ROC curves of our experimental data PUA1-6. Each point on the ROC curve represents a different cutoff value. Cutoff values that result in low FP rates tend to result low true-positive rates as well. As the true-positive rate increases, so does the FP rate.

Fig. 6 shows the comparisons of three algorithms, SVM, NB and AdaBoost with our proposed classifier ensembles technique with data sets PUA1-6. It is clear from the figure that the proposed technique gives better performance for every data sets compared to any individual algorithms, particularly in reducing FP rate. The AUC of the ROC curve is also better in proposed technique, which proofs the classification accuracy of our multi-classifier ensembles technique.

Obviously, a useful experiment should have a cutoff value at which the true-positive rate is high and the FP rate is low. In fact, a near-perfect classification would have an ROC curve that is almost vertical from (0,0) to (0,1) and then horizontal to (1,1). The diagonal line serves as a reference line since it is the ROC curve of experiment that is useless in determining the classification.

Table 2
The comparison of FP with individual and combined classification approach

Data sets	Comparison of FP			
	SVM	AdaBoost	NB	Proposed
	FP	FP	FP	FP
PUD1	0.0	0.0	0.181	0.0
PUD2	0.091	0.091	0.18	0.0
PUD3	0.181	0.121	0.0	0.0
PUD4	0.181	0.0	0.12	0.0
PUD5	0.181	0.091	0.09	0.0
PUD6	0.181	0.191	0.12	0.0
AVG	0.136	0.082	0.115	0.0

Table 3
The comparison of FN with individual and combined classification approach

Data sets	Comparison of FN			
	SVM	AdaBoost	NB	Proposed
	FN	FN	FN	FN
PUD1	0.091	0.09	0.27	0.09
PUD2	0.09	0.091	0.270	0.0
PUD3	0.181	0.036	0.350	0.036
PUD4	0.181	0.0	0.181	0.0
PUD5	0.355	0.09	0.36	0.090
PUD6	0.360	0.09	0.36	0.09
AVG	0.21	0.066	0.299	0.051

Table 4
The comparison of misclassification (MC) cost and the percentage of GL outputs

Data sets	MC cost				GL
	SVM	AdaBoost	NB	Proposed	
PUD1	0.091	0.09	0.272	0.091	0.181
PUD2	0.09	0.091	0.27	0	0.27
PUD3	0.181	0.363	0.36	0.18	0.18
PUD4	0.272	0	0.18	0	0.273
PUD5	0.363	0.091	0.364	0.09	0.273
PUD6	0.364	0.09	0.36	0.091	0.273
AVG	0.227	0.121	0.301	0.075	0.242

Table 2 shows the comparative result of FP for three classifiers SVM, AdaBoost, NB and our proposed technique. It has been shown that the output of FP is zero for all data sets in our proposed technique. But there is still some FP for other classifiers. FP is considered one of the important tradeoffs of spam filtering. In our experiment it shows zero, which is more convincing and proves the success of our design.

Similarly Table 3 shows the comparative result of false negative for three classifiers SVM, AdaBoost, NB and our proposed technique. It has been shown that the output of FN is much lower (~5.1%) compared to any of the individual algorithms. It is much higher in NB and SVM but lower in AdaBoost.

Table 4 shows the percentage of misclassification (MC) cost and the GL e-mails. The MC cost is the ratio of the misclassified e-mails, both spam and legitimate, from the classifier. It has been shown that the average MC cost in our proposed technique is much lower (~0.075) compared to any individual algorithms. The comparison of average MC cost is also graphically illustrated in Fig. 7, which indicates that our proposed MC cost is always lower compared to others.

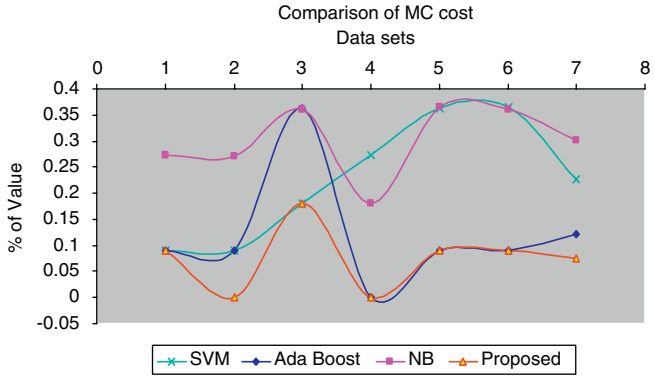

Fig. 7. Comparison of MC cost.

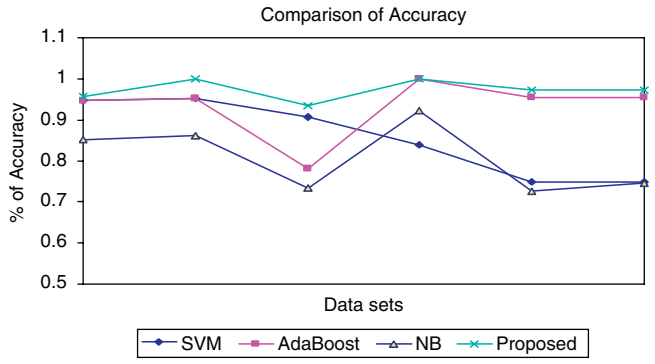

Fig. 8. Comparison of accuracy.

Table 4 also shows the list of GL e-mails which are somehow misclassified by any of the algorithm/(s). In the existing techniques, these sorts of e-mails are considered either TP or TN, based on the decision made by algorithms. We have investigated it and found that an individual e-mail is misclassified by one algorithm but not for another algorithm. So, there is a chance to reduce the rate FP or FN. In our experiment, we considered this sort of e-mails as GL and analysed it using the technique discussed above which will increase the performance of our whole e-mail classification system.

Fig. 8 shows the comparison of accuracy of our experiment. It is clear that the accuracy of our proposed system (~97%) is much better than the other classifier algorithms, which proves the success of our spam filtering technique.

6. Conclusion and future work

In this paper, an innovative spam filtering technique has been proposed based on the multiple classification approach. Emphasis has been given in this paper by using different aspects of learning-based anti-spam filtering algorithms for reducing FP problems and getting better performance compared to existing techniques. In our technique, the multiple classifiers will produce a list of e-mails; those are misclassified by any of the classifier, known as GL e-mail. An analyser for analysing the produced GL e-mails has also been proposed in our paper. Our experimental result proves the success and effectiveness of our proposed technique. However, the GL analyser has added complexity and cost in terms of software and time overhead, as we have discussed which will reduce the filtering speed. We are working on it and also working to find out the lower bound of GL terms and minimizing the GL e-mails. We will explore it in our future work.

Acknowledgements

The author would like to thank the anonymous reviewers for their valuable comments and constructive suggestions that helped to improve the quality of this paper.

References

Ali S., Xiang Y. Spam classification using adaptive boosting algorithm. In: Sixth IEEE/ACIS international conference on computer and information science, 2007. p. 972–6.

Androutsopoulos I, et al. Learning to filter unsolicited commercial email. NCRS, T. Report 2004.

Cheng V, Li CH. Personalized spam filtering with semi-supervised classifier ensemble. In: IEEE/WIC/ACM International Conference on Web Intelligence, 2006. p. 195–201.

Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge: Cambridge University Press; 2000.

Drucker H, Shahrary B, Gibbon DC. Support vector machines: relevance feedback and information retrieval. Inform Process Manage 2002;38(3):305–23.

Hunt R, Carpinter J. Current and new developments in spam filtering. In: 14th IEEE international conference on networks, ICON'06, vol. 2, 2006. p. 1–6.

Koprinska I, Poon J, Clark J, Chan J. Learning to classify e-mail. J Inform Sci 2007;77(10):2167–87.

Islam M, Chowdhury M. Spam filtering using Ml algorithms. In: Proceedings of the IADIS international conference on WWW/Internet. USA: International Association for Development of the Information Society Press; 2005. p. 419–26.

Islam M, Zhou W. Architecture of adaptive spam filtering based on machine learning algorithms. In: ICA3PP 2007, Lecture Notes in Computer Science, vol. 4494. Berlin: Springer; 2007. p. 458–69.

Islam R, Chowdhury M, Zhou W. An analysis of spam and its classification techniques based on statistical learning algorithms. Technical report TRC 05/06, Deakin University, Australia; 2005a.

Islam R, Chowdhury M, Zhou W. An Innovative spam filtering model based on support vector machine. Proceedings of the IEEE international conference on intelligent agents, web technologies and internet commerce, vol. 2, 28–30, Austria; 2005b. p. 348–53.

Islam R, Zhou W, Chowdhury M. Dynamic feature selection for spam filtering using SVM. In: ACIS ICIS'07. IEEE; 2007. p. 757–62.

Kong JS, Rezaei BA, Sarshar N, Roychowdhury VP, Boykin PO. Collaborative spam filtering using E-mail networks. J Comput 2006;39(8):67–73.

Liu Huan, Yu Lei. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowledge Data Eng 2005;17(4).

Sahami M, Dumais S, Heckerman D, Horvitz E. A Bayesian approach to filtering junk e-mail. In: Learning for text categorization: papers from the workshop, Wisconsin, Madison. AAAI technical report WS-98-05, 1998.

Zhang J, et al. A modified logistic regression: an approximation to SVM and its applications in large-scale text categorization. In: Proceedings of the 20th international conference on machine learning. AAAI Press; 2003. p. 888–95.

Zi-Qiang Wang, Xia Sun, Xin Li, De-Xian Zhang. An efficient SVM-based spam filtering algorithm. In: International conference on machine learning and cybernetics, 2006. p. 3682–6.