SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:4013–4026
Published online 25 August 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1584

RESEARCH ARTICLE

An intelligent three-phase spam filtering method based on decision tree data mining

Jyh-Jian Sheu[1], Yin-Kai Chen[2], Ko-Tsung Chu[3]*, Jih-Hsin Tang[4] and Wei-Pang Yang[2]

[1] College of Communication, National Chengchi University, Taipei, Taiwan
[2] Department of Information Management, National Dong Hwa University, Hualien, Taiwan
[3] Department of Finance, Minghsin University of Science and Technology, Hsinchu, Taiwan
[4] Department of Information Management, National Taipei University of Business, Taipei, Taiwan

ABSTRACT

In this paper, we proposed an efficient spam filtering method based on decision tree data mining technique, analyzed the association rules about spams, and applied these rules to develop a systematized spam filtering method. Our method possessed the following three major superiorities: (i) checking only an e-mail's header section to avoid the low-operating efficiency in scanning an e-mail's content. Moreover, the accuracy of filtering was enhanced simultaneously. (ii) In order that the probable misjudgment in identifying an unknown e-mail could be "reversed", we had constructed a reversing mechanism to help the classification of unknown e-mails. Thus, the overall accuracy of our filtering method will be increased. (iii) Our method was equipped with a re-learning mechanism, which utilized the supervised machine learning method to collect and analyze each misjudged e-mail. Therefore, the revision information learned from the analysis of misjudged e-mails incrementally gave feedback to our method, and its ability of identifying spams would be improved. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
spam; data mining; decision tree

***Correspondence**
Ko-Tsung Chu, Department of Finance, Minghsin University of Science and Technology, Hsinchu, Taiwan.
E-mail: ktc1009@must.edu.tw

1. INTRODUCTION

With the advance of Internet technologies, e-mail has become one of the major communication channels in modern society. Due to its low cost and convenience, e-mail has become an important media for spreading advertisements, viruses, and detrimental information. The unsolicited e-mails or called spams have occupied network bandwidth, decreased people's work efficiency, and even leaked personal information. According to Hong Kong Anti-SPAM Coalition report dated in 2004 [1], it was estimated that $9 billion was needed to deal with the impact brought about by spam on an annual basis. Furthermore, Symantec spam report dated in May 2014 stated that the global spam rate is 60.6% for the month of May [2]. In other words, almost 61 in 100 e-mails are spams, which is a serious problem.

Various mechanisms have been proposed to filter out the spammy e-mails, including white/blacklisting, greylisting [3], rule learning [4,5], and the methods based on text classification, such as naïve Bayes [6–9], support vector machine [10], and boosting trees [11,12], multi-agent [13,14], and genetic algorithm [15]. Other approaches combine two mechanisms or users' experiences to increase the filtering accuracy such as collaborative filtering techniques [16].

Among these mechanisms, content-based filtering methods, which scan an e-mail's content, are widely used and characterized by their effectiveness [6]. However, the process of scanning content will increase complexity and reduce operation efficiency. Recently, some efficient header-based methods of analyzing only an e-mail's header section have been proposed [4,5,17,18]. However, the header-based methods perform better in efficiency but probably poor to maintain their accuracy because an e-mail's header has less information than an e-mail's content.

The machine learning methods are to collect existing data (denoted as "training data") and choose useful attributes of the data to generate meaningful rules or models, which can be applied to predict the newly arrived data [19–21]. Machine learning techniques can be divided into two categories: unsupervised and supervised. In

unsupervised learning, no labels are used on the training data to be classified. On the other hand, the supervised learning methods learn the classification by using a set of man-made examples [22]. Supervised learning algorithms are now applied frequently [23], such as support vector machines [10], random forests [24], and decision trees [4,5,17,25,26].

Machine learning methods have two major phases: (i) the training phase and (ii) the classification phase [19–22]. In the training phase, the prior estimates are captured by building a model from the training data. The model built using training data is then applied by a classifier to classify the unknown data in the second phase (i.e., the classification phase). However, at the end of training phase, the model or rules are learned from previous data, whose knowledge may be outdated. If the spammers design newly spamming techniques, the classifier could not detect and filter these novel spams.

This study aims to propose an efficient spam filtering mechanism based on machine learning technique. We will apply the uncomplicated decision tree data mining algorithm to find association rules about spams from the training e-mails. Based on these association rules, we propose a systematized three-phase spam filtering method with the following major superiorities:

(1) Checking only an e-mail's header section in order to avoid the low-operating efficiency in scanning an e-mail's content. On the other hand, the accuracy of filtering will be enhanced simultaneously.
(2) A reversing mechanism is constructed to avoid misjudgment in identifying unknown e-mails. In order that the probable misjudgment can be "reversed", we establish a reversing mechanism, which will calculate a supplementary score to help the classification of each unknown e-mail. Thus, the overall accuracy of our filtering method will be increased.
(3) A re-learning mechanism is designed to incrementally improve our method. We utilize the supervised machine learning to collect and analyze each misjudged e-mail that resulted from our method. Therefore, the revision information learned from the analysis of misjudged e-mails can incrementally give feedback to our method and improve its ability of identifying spams.

The remainder of this paper is organized as follows: Section 2 discusses the decision tree data mining algorithm. Section 3 presents the descriptions of our proposed mechanism. The experimental results of our method are shown in Section 4. Section 5 concludes this paper.

2. DECISION TREE DATA MINING ALGORITHM

Decision tree is one of the data mining methods upon the tree data structure. The general statistical methods usually can only analyze the distribution of the surface of data, whereas decision tree algorithms can find the potential association rules between the important attributes from the existing data. Moreover, the prediction of classification of the unknown data can be further acquired by comparing their related attributes' values to these association rules.

An example of a tree is illustrated in Figure 1. There is a start node called the "*root node*" in each tree. If there is any node under, the bottom nodes will be the "*children nodes*" of the above one, and the above one will be the "*parent node*" of the bottom ones. For example, node *A* is the root node of this tree in Figure 1, *B* and *C* are children nodes of *A*, and *A* is the parent node of *B*. Moreover, each node without children is called a "*leaf node*", such as *C*, *D*, and *E* in Figure 1.

The Iterative Dichotomiser 3 (called ID3 for short) is one of the most well-known and effective decision tree algorithms, [25,26]. In 1999, Stark and Pfeiffer [27] studied the behavior of ID3 and pointed out that ID3 was better than other decision tree methods, such as C4.5, CHAID, and CART. As compared with the improved methods of ID3 (for example, C4.5), Ohmann *et al*. demonstrated that the quantity of association rules computed by ID3 was not as numerous as that of C4.5 [28]. In other words, considering the simplicity of rule quantity, the ID3 algorithm possessed the superior feature. Hence, we choose ID3 as the data mining technique in this research.

Let "*target attribute*" be the attribute which is concerned objective of our research. For example, we suppose that the attribute "e-mail type" ("S" means it is spammy; "L" means it is legitimate) is the target attribute in this study. And let "*critical attributes*" be the other important attributes which interest us in this research. The construction process of decision tree will start from root node. Note that all of the data instances are initially contained in the root node. The ID3 algorithm will select an unselected critical attribute with the maximum "*information gain*" (the detailed process will be described later). Then, the ID3 algorithm will divide all data instances into children nodes according to their values of the selected critical attribute. Subsequently, each children node respectively repeated the same process for its own data instances.

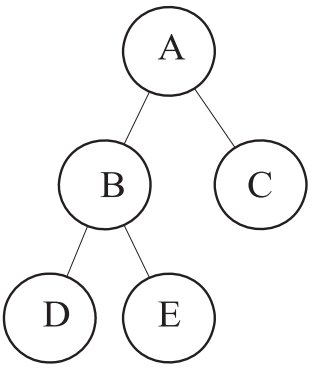

Figure 1. An example of tree.

In the ID3 algorithm, there are two conditions to end the construction process of a decision tree: (i) All of the critical attributes are selected; and (ii) target attribute's values of all data instances in this children node are exactly the same. If any of the two conditions is satisfied, this children node will be signified as a leaf node. Given a leaf node C, it will be labeled by the value of target attribute possessed by the majority of data instances in C, which is denoted as $Label(C)$. And let $|Label(C)|$ be the number of data instances whose target attribute's value is equal to $Label(C)$ in C. Then, we calculate C's degree of purity (denoted as $Purity(C)$) and degree of support (denoted as $Support(C)$), and end this node's execution of the ID3 algorithm. The formulas of $Purity(C)$ and $Support(C)$ are defined as follows:

$$Purity(C) = (|Label(C)|/|C|)*100\%$$
$$Support(C) = (|C|/N)*100\%$$

where $|C|$ is the number of data instances contained in node C and N is the number of total data instances.

The detailed process of the ID3 algorithm is summarized as follows. Note that we have modified step 4 of the ID3 algorithm by adding a stop condition in order to avoid the inordinate branching. And the variables P_{lower}, P_{upper}, and S_{lower} indicate the threshold values of stopping computation.

Step 1. If the target attribute's values of all data instances in node C are exactly the same, then set C to be a leaf node, compute $Purity(C)$ and $Support(C)$, and stop.

Step 2. If all critical attributes are "selected", then set C to be a leaf node, let $Label(C)$ be the value of target attribute possessed by the majority of data instances in C, compute $Purity(C)$ and $Support(C)$, and stop.

Step 3. Compute the information gain $G(A)$ for each unselected critical attribute A and select the one with maximum information gain. Divide all data instances contained in node C into disjoint children nodes according their values of the select attribute A;

Step 4. Treat each children node branched in step 3 as node C. If $P_{\text{lower}} > Purity(C)$ or $Purity(C) > P_{\text{upper}}$ or $Support(C) < S_{\text{lower}}$, then stop; else, continue the algorithm recursively from step 1.

Considering a certain critical attribute A on node C, its information gain $G(A)$ concerns the "entropy" of node C, which is denoted as $E(C)$ and calculated by the following formula:

$$E(C) = -\sum_{i=1}^{t} \frac{p_i}{n} \times \log_2 \frac{p_i}{n}$$

where t is the number of target attribute's values, p_i is the total number of data instances corresponding to the ith value of the target attribute in C, and n is the number of data instances in C. Then, the information gain $G(A)$ of critical attribute A is calculated by using the following formulas:

$$G(A) = E(C) - E^{+}(A)$$
$$E^{+}(A) = \sum_{j=1}^{k} (n_j/n) \times E(C_j)$$

where k is the number of values of critical attribute A, C_j with $1 \le j \le k$ is a subset of C including the data instances corresponding to the jth value of critical attribute A, and n_j is the total number of data instances contained in C_j.

Finally, each leaf node in the resulted decision tree will be labeled as a value of the target attribute. And each path constructed from root node to leaf node will form an association rule. In other words, all of the internal nodes on the path constructed a row of "if" judgment of several critical attributes. With the "then" result presented by the labeled value (i.e., $Label(C)$) of the leaf node, there is the association rule of "if–then" pattern constructed.

3. SYSTEM ARCHITECTURE

In this paper, we propose a systematized three-phase spam filtering method based on decision tree data mining technique. In the method, we construct a reversing mechanism to avoid the misjudgment of unknown e-mails and design a re-learning mechanism to incrementally improve the filter's ability to identify spams accurately. As shown in Figure 2, our method can be divided into the following three phases:

(1) Training phase: The purpose of this phase is to find association rules about spams by analyzing only the header sections of training e-mails. And the association rules will be applied to classify unknown e-mails in the second phase. There exist two major modules in the training phase: *rule constructing module* and *reversing mechanism's setup module*. The rule constructing module will check the critical attributes of e-mails and apply the decision tree algorithm ID3 to compute the potential association rules of the "if–then" pattern, which will be stored into the *rule database*. And the reversing mechanism's setup module is designed to initialize the parameters of our reversing mechanism.

(2) Classification phase: This phase is to classify the unknown e-mails. Each unknown e-mail will be scored by applying the rule database and the reversing mechanism together. According to the computed score, each unknown e-mail can be classified as either a legitimate e-mail or a spam.

(3) Re-learning phase: This phase will incrementally learn the revision information by analyzing misjudged e-mails resulted from the classification phase to improve our filtering method. Thus, those misjudgments will give feedback to our filtering method and strengthen its ability of classifying unknown e-mails accurately.

Occasionally, the header-based method may unavoidably suffer from deficiency of information in identifying

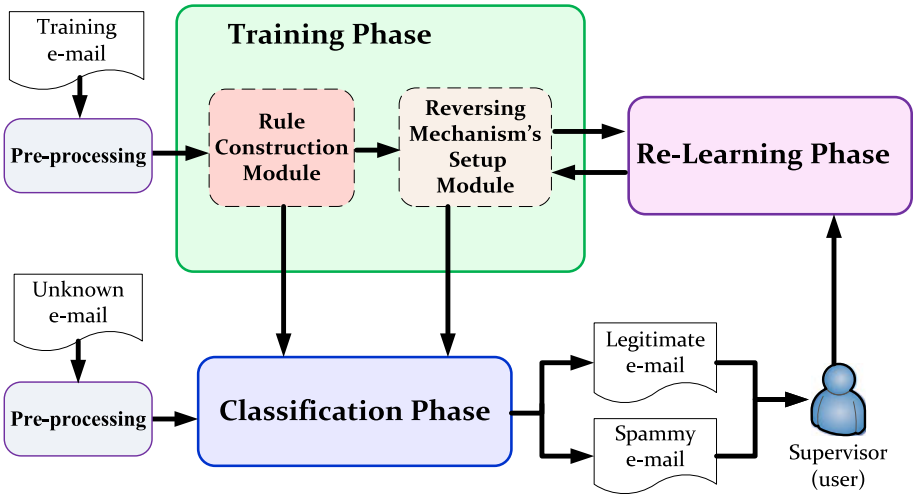

Figure 2. Architecture of the three-phase spam filtering method.

unknown e-mails. In order that the misjudgment can be "reversed", we establish a reversing mechanism with the *reversing database* in our filtering system. Our reversing mechanism contains the following three major tasks:

(1) In the training phase, the reversing mechanism's setup module will initialize the reversing database, which will be applied to compute an auxiliary score for each unknown e-mail in the classification phase.
(2) In the classification phase, each unknown e-mail is first examined and scored by applying the rule database to compute its *original score*. Note that the original score of an unknown e-mail implies its tendency to be identified as a spam. Obviously, if the computed original score of a legitimate e-mail is high, then it should be decreased. On the other hand, if the computed original score of a spammy e-mail is low, then it should be increased. Therefore, we arrange that each unknown e-mail should be examined once again to compute its *additional score* by checking the corresponding parameters of the reversing database. The deciding classification of this unknown e-mail will be judged according to the sum of the original score and the additional score. Thus, the latent misjudgment of this unknown e-mail is likely to be "reversed" by the effect of the additional score.
(3) The re-learning phase utilizes the supervised machine learning method to analyze each misjudged e-mail resulted from the classification phase. According to the revision information learned from analysis of those misjudged e-mails, our re-learning mechanism will improve the reversing database. Hence, the misjudged e-mails can incrementally give feedback to strengthen our reversing mechanism.

Then, we will introduce the detailed procedures of the training phase, classification phase, and re-learning phase. Note that each training e-mail or unknown e-mail is handled first by the pre-processing procedure. In the pre-processing procedure, the header section of each e-mail will be examined. First, the meaningless stop words will be removed from the header section. Then, apply the Porter stemming algorithm [29] to strip suffixes from English words. Thus, the noise in fields of an e-mail's header section will be reduced.

3.1. Training phase

In the training phase, numerous e-mails collected in advance are taken as the training data for our spam filtering method. The main purpose of this phase is to seek for association rules between the critical attributes and the target attribute of training e-mails. Then, these rules will be applied to classify unknown e-mails in the classification phase. As shown in Figure 3, the training phase contains two major modules: rule constructing module and reversing mechanism's setup module, which will be introduced as follows.

3.1.1. Rule construction module

We set the attribute "e-mail type" to be the target attribute in this study. If the training e-mail is spammy, then its e-mail type will be denoted as "S." On the other hand, if the e-mail is legitimate, then its e-mail type will be denoted as "L." Moreover, as shown in Table I, nine critical attributes of binary values are defined by surveying the important fields of an e-mail's header section and referring to the related researches [4,5,18]. These nine critical attributes are divided into three categories of "sender", "title", and "time and size." We will apply the ID3 decision tree algorithm to analyze the associative rules between the nine critical attributes and the target attribute of the training e-mails.

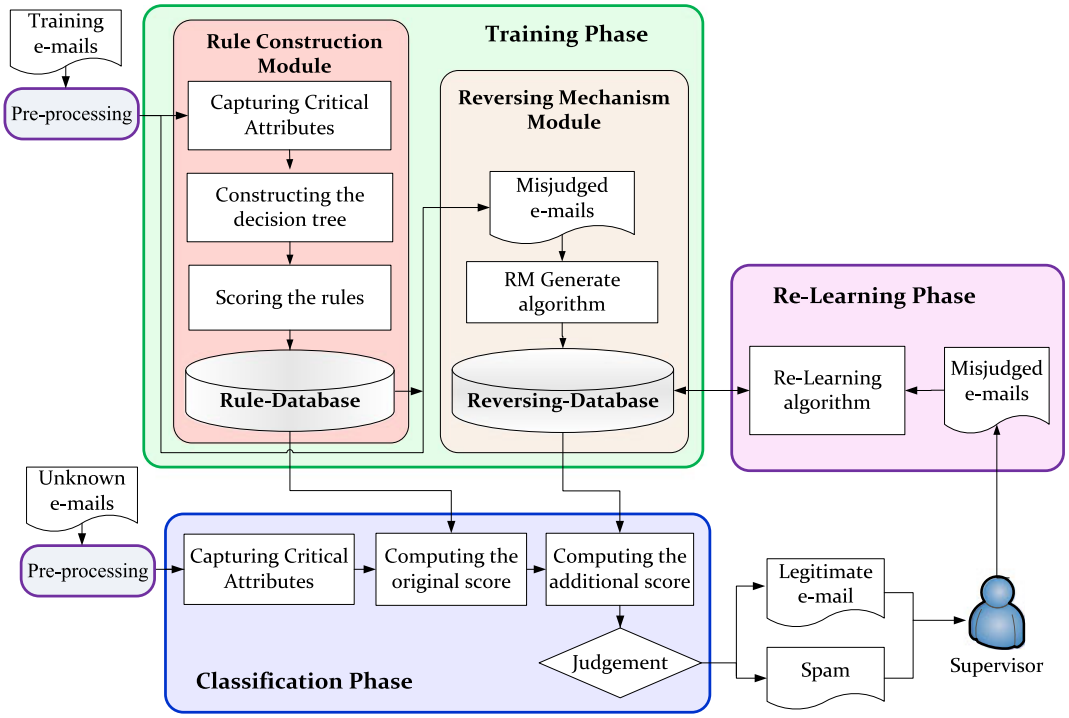

Figure 3. Detailed process of the three-phase spam filtering method.

Table I. The nine critical attributes of e-mail.

Attribute categories	Critical attribute	Value
Sender	Length of sender's name is abnormal	If the length of sender's name is more than nine characters, then it is set at 1 (True), otherwise 0 (False).
	Either sender's name or address is abnormal	If any of sender's name and address is blank or contains abnormal symbol, then it is set at 1 (True), otherwise 0 (False).
	Spam keyword is found in sender's name or address	If any spam keyword is found, then it is set at 1 (True), otherwise 0 (False).
Title	E-mail's title is abnormal	If the title is blank or contains more than three wrong (or unknown) words, then it is set at 1 (True), otherwise 0 (False).
	E-mail's title includes spam keyword (type I)	If an e-mail's title has a spam keyword, then it is set at 1 (True), otherwise 0 (False).
	E-mail's title includes spam keyword (type II)	If an e-mail's title has at least three spam keywords, then it is set at 1 (True), otherwise 0 (False).
Other	Sending date and receiving date are abnormal	If the date of sending distinctly differs from the date of receiving, then it is set at 1 (True), otherwise 0 (False).
	E-mail's size is abnormal	If an e-mail's size is equal to or larger than 8000, then it is set at 1 (True), otherwise 0 (False).
	E-mail's format	If this e-mail's format is HTML or contains attachments, then it is set at 1, otherwise 0.

The detailed process of the rule construction module is described by the following stages.

Stage 1. Capturing critical attributes

In this stage, each training e-mail will be checked to capture the values of all necessary critical attributes. For each training e-mail, the target attribute's value depends on its type ("S" means it is spammy; "L" means it is legitimate). And the nine critical attributes are defined as shown in Table I, whose values are decided by checking the corresponding fields of header section and looking up the spam keywords table (if necessary). The table of spam keywords contains suspicious keywords found frequently in spams. In this study, we will take the spam keyword table proposed by Sanpakdee *et al.* [30].

Stage 2. Constructing the decision tree

This stage employs the decision tree data mining algorithm ID3 to look for the association rules between the target attribute and critical attributes. The captured attributes of the training e-mails mentioned in the preceding texts will be input into the algorithm ID3 to build a decision tree, which will bring out the potential association rules of the "if–then" pattern between the nine critical attributes and the target attribute.

Stage 3. Scoring the rules

Then, we will score each rule by using the formulas based on the values of its degree of support and degree of purity. Given an association rule R, we assume that C is its leaf node, n is the number of e-mails whose the target attribute's value is "S" in node C, and $Support(R)$ records the degree of support of this rule. We compute the values of degree of support for all rules and denote the maximum one as $Support_{MAX}$ and the minimum one as $Support_{MIN}$. Let $Support(C)$, $Purity(C)$, and $Label(C)$ be defined as mentioned earlier. Before describing the scoring formula of rules, we have to introduce the following three important functions: $Spam.Tendency(R)$, $W(Rule.Support(R))$, and $S(Rule.Support(R))$.

The function $Spam.Tendency(R)$ implies the rule's "intensity" to classify e-mails as spams, which is defined as follows:

$$Spam.Tendency(R) = Purity(C) \text{ if } Label(C) = "spam";$$
$$\text{and } Spam.Tendency(R) = \left(\frac{n}{|C|}\right)*100\% \text{ otherwise},$$

where $|C|$ is the number of e-mails contained in leaf node C.

The function $W(R)$ records the weighted value of rule R, which is computed as follows:

$$W(R) = \frac{Support(C)}{Support_{MAX} + Support_{MIN}} \times 100\%.$$

Assume that W_{MAX} is the maximum one and W_{MIN} is the minimum one of weighted values of all the rules computed by the formula in the preceding texts. Then, the function $S(Rule.Support(R))$ will record the score of $Rule.Support(R)$, which is relative to the ranking of weighted value of rule R.

$$S(Rule.Support(R)) = \frac{W(R) - W_{MIN}}{W_{MAX} - W_{MIN}} \times 100\%.$$

Now, we can compute the score of rule R, which is recorded by the function $Rule.Score(R)$. It is composed of $Spam.Tendency(R)$ and $Rule.Support(R)$ in a ratio of 7:3, which is defined as follows:

$$Rule.Score(R) = (0.7 \times Spam.Tendency(R) + 0.3 \times S(Rule.Support(R))) \times 100.$$

After computing the scores, all of the rules are stored into the rule database, which keeps the extracted association rules and will be accessed by the classification phase to classify unknown e-mails. Moreover, we choose out the minimum rule's score from the rules with $Spam.Tendency(R)$ more than 80% and set it as the *threshold* λ for judging whether the unknown e-mail is spam.

3.1.2. Reversing mechanism's setup module

As mentioned earlier, this module is designed to initialize the parameters of the reversing database, which will be applied by the classification phase to calculate an additional score for each unknown e-mail. Thus, the latent misjudgment of this unknown e-mail is likely to be "reversed" by the effect of additional score.

In the reversing database, we construct a *reversing table* for each rule of the rule database. Each reversing table records the nine items of critical attributes as mentioned in Table I. Moreover, each item in this table has two parameters: *plus value* and *minus value*, which record the adjustment values that will be increased additionally to unknown e-mails. The plus value records a positive integer value, which implies a supplement to the score of an unknown e-mail. Moreover, the minus value records a negative integer value, which implies a subtraction from the score of an unknown e-mail. An example of reversing table is shown in Table II. Given a rule R_i, we denote its corresponding reversing table as $RT(R_i)$. Moreover, we denote the nine items of reversing table $RT(R_i)$ as $RT(R_i)[j]$ for $1 \le j \le 9$, whose the two parameters plus value and minus value are named as $RT(R_i)[j].plus$ and $RT(R_i)[j].minus$, respectively.

By using the scored rules of rule database to examine the attributes of training e-mails, we can classify them and collect the misjudged ones. The algorithm *RT_Initial* will initialize the reversing tables by applying these misjudged training e-mails. The process of algorithm *RT_Initial* is illustrated in Figure 4. Note that each initial value of $RT(R_i)[j].plus$ and $RT(R_i)[j].minus$ for $1 \le j \le 9$ is set as zero before performing *RT_Initial*. Moreover, the parameters I^+ and I^- are two positive integers, which are basic units to adjust the values of $RT(R_i)[j].plus$ and $RT(R_i)[j].minus$, respectively.

The detailed process of *RT_Initial* is summarized as follows.

Step 1. Check the critical attributes of this misjudged training e-mail.

Step 2. According to the values of critical attributes, this training e-mail will dovetail with some rule, say R_i, in the rule database. Then, the corresponding reversing table $RT(R_i)$ associated with this dovetailed rule R_i will be chosen from the reversing database.

Step 3. If this misjudged training e-mail is "false positive" (a legitimate e-mail to be judged as a spam), do the following operations:

Table II. An example of reversing table $RT(R)$.

No.	Item (True/False)	Revised values: Plus value	Revised values: Minus value
1	Length of sender's name is abnormal	+4	−4
2	Either sender's name or address is abnormal	+4	−2
3	Spam keyword is found in sender's name or address	+4	−4
4	E-mail's title is abnormal	+4	−2
5	E-mail's title includes spam keyword (type I)	+4	−2
6	E-mail's title includes spam keyword (type II)	+8	−4
7	Sending date and receiving date are abnormal	+10	0
8	E-mail's size is abnormal	+4	−3
9	E-mail's format	+4	−2

For $1 \le j \le 9$, check whether this misjudged training e-mail satisfies the statement of $RT(R_i)[j]$ ("True" or "False"):

If True then do

If $RT(R_i)[j].plus \ge I^-$, then $RT(R_i)[j].plus = RT(R_i)[j].plus - I^-$;

Else (False) then do

$RT(R_i)[j].minus = RT(R_i)[j].minus - I^-$.

Step 4. If this misjudged training e-mail is "false negative" (a spam to be judged as a legitimate e-mail), do the following operations:

For $1 \le j \le 9$, check whether this misjudged training e-mail satisfies the statement of $RT(R_i)[j]$ ("True" or "False"):

If True, then do

$RT(R_i)[j].plus = RT(R_i)[j].plus + I^+$;

Else (False), then do

If $RT(R_i)[j].minus + I^+ \ge 0$, then $RT(R_i)[j].minus = RT(R_i)[j].minus + I^+$.

Step 5. Store $RT(R_i)$ back to the reversing database.

3.2. Classification phase

The task of this phase is to classify each unknown e-mail to be either a legitimate e-mail or a spam according to the association rules learned in the training phase. The first step is to extract the critical attributes of each unknown e-mail and find the dovetailed association rule in the rule database to compute the original score for this e-mail. Then, this unknown e-mail's attributes will be examined once again to compute its additional score by checking the corresponding reversing table in the reversing database. Finally, this unknown e-mail can be classified according to the total score composed of original and additional scores. We describe the process of this phase in the following stages:

Stage 1. Capturing critical attributes

In this stage, each unknown e-mail will be examined to capture the values of nine critical attributes as shown in Table I. The values of these nine critical attributes are decided by checking the corresponding fields of an unknown e-mail's header section and looking up the spam keywords table (if necessary).

Stage 2. Computing the original score

According to the values of critical attributes, this unknown e-mail will dovetail with some association rule, say, R_i in the rule database built in the training module. And we will set the original score of the unknown e-mail to be $Rule.Score(R_i)$, which is as mentioned previously in the training phase. Assume that the original score of this unknown e-mail is named as $score_A$

Stage 3. Computing the additional score

Assume that the additional score of this unknown e-mail is named as $score_B$. In this stage, we will access the corresponding reversing table of dovetailed rule R_i. First, the reversing table $RT(R_i)$ is picked form the reversing database. Then, this unknown e-mail's attributes will be

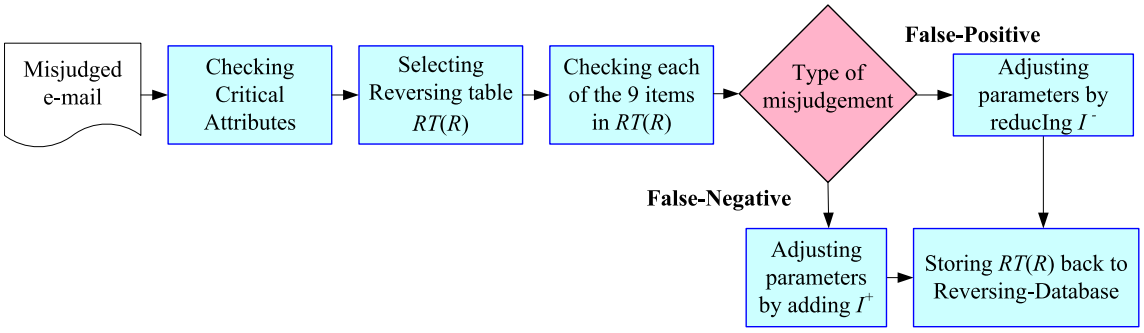

Figure 4. The flow of *RT_Initial*.

examined again to compute its additional score (i.e., $score_B$) by executing the following steps:

Step 1. Check the critical attributes of this unknown e-mail.
Step 2. For $1 \le j \le 9$, check whether this unknown e-mail satisfies the statement of $RT(R_i)[j]$ ("True" or "False"):

If True, then $score_B = score_B + RT(R_i)[j].plus$;
Else (False), then $score_B = score_B - RT(R_i)[j].minus$.

Stage 4. Judging classification of this unknown e-mail

Now, the total score of this unknown e-mail can be obtained by adding up $score_A$ and $score_B$. Then, this unknown e-mail will be classified as a spam if $score_A + score_B \ge \lambda$, and a legitimate e-mail otherwise. In this research, we apply the supervised machine learning method to collect the misjudged e-mails for further analysis in the re-learning phase. Therefore, in this stage, the supervisor will monitor the classification results of all unknown e-mails and collect the misjudged ones.

3.3. Re-learning phase

During the re-learning phase, the misjudged e-mails collected in the phase mentioned in the preceding texts (the classification phase) will be used by algorithm *RT_Modify* to modify the reversing tables in the reversing database. The process of *RT_Modify* is similar to *RT_Initial* but not exactly the same. In *RT_Modify*, the parameters M^+ and M^- are basic units to adjust the values of $RT(R_i)[j].plus$ and $RT(R_i)[j].minus$, respectively. The detailed process of *RT_Modify* is summarized as follows.

Step 1. Check the critical attributes of each misjudged e-mail.
Step 2. According to the values of critical attributes, this misjudged e-mail will dovetail with some rule, say R_i, in the rule database. Then, choose the corresponding reversing table $RT(R_i)$ associated with this dovetailed rule R_i from the reversing database.
Step 3. If this misjudged e-mail is "false positive" (a legitimate e-mail to be judged as a spam), do the following operations:

For $1 \le j \le 9$, check whether this misjudged e-mail satisfies the statement of $RT(R_i)[j]$ ("True" or "False"):

If True, then do
If $RT(R_i)[j].plus \ge M^-$, then $RT(R_i)[j].plus = RT(R_i)[j].plus - M^-$;
Else (False), then do
$RT(R_i)[j].minus = RT(R_i)[j].minus - M^-$.

Step 4. If this misjudged e-mail is "false negative" (a spam to be judged as a legitimate e-mail), do the following operations:

For $1 \le j \le 9$, check whether this misjudged e-mail satisfies the statement of $RT(R_i)[j]$ ("True" or "False"):

If True, then do
$RT(R_i)[j].plus = RT(R_i)[j].plus + M^+$;
Else (False), then do
If $RT(R_i)[j].minus + M^+ \ge 0$, then $RT(R_i)[j].minus = RT(R_i)[j].minus + M^+$.

Step 5. Store $RT(R_i)$ back to the reversing database.

4. EXPERIMENTAL RESULTS

In this section, we perform experiments to confirm the accuracy and efficiency of our spam filtering method. We employ two spam datasets as experimental data: SpamAssassin [31] and Enron-Spam [32], which are commonly applied in research papers about spams. The dataset of SpamAssassin consists of 6827 e-mails (4894 legitimate e-mails and 1933 spams), and the dataset of Enron-Spam consists of 52 076 e-mails (19 088 legitimate e-mails and 32 988 spams). Moreover, we have collected 10 502 e-mails (4401 legitimate e-mails and 6101 spams) in the recent period as supplements to the two datasets mentioned in the preceding texts. Therefore, the total number of experimental e-mails is 69 405 (28 383 legitimate e-mails and 41 022 spams). We denote these e-mails as the third dataset: *Mixed-Set*, which will be adopted in this section.

The experiments will be proceeded through the following steps: First, we will introduce the efficacy assessment indexes used in this paper. And we optimize the parameters (I^+, I^-, M^+, M^-) used in the algorithms *RT_Initial* and *RT_Modify*. Then, we conduct a series of experiments to measure the filtering performance of the proposed method.

4.1. Assessment indexes used in this study

To evaluate the performance for our spam filtering method proposed in this paper, we employ the following efficacy assessment indexes: "precision", "recall", and "F-measure", which are commonly used for document classification. The decision confusion matrix, as shown in Table III, is used to explain the calculation equations listed as follows [33–35]. Note that all the four cases A, B, C, and D in Table III are recorded by the quantity of e-mails.

(1) Accuracy: the percentage of total e-mails that are correctly recognized. It is defined by the following formula:

Table III. Four cases of judgment.

	E-mail's categorization in reality	
	Spammy	Legitimate
To be judged as spam	*A*	*B*
To be judged as legitimate e-mail	*C*	*D*

$$Accuracy = \frac{A + D}{A + B + C + D}.$$

(2) Precision: It calculates the ratio of the e-mails classified correctly in the e-mails judged as the certain category, representing the filter's capabilities of classifying correctly such category of e-mails. In this study, we calculate the "*spam precision*" from the perspective of identifying spams and the "*legitimate precision*" from the perspective of identifying legitimate e-mails. And the value of "*precision*" is set as the mean of spam precision and legitimate precision. The formulas are listed as follows:

$$Spam\ Precision = \frac{A}{A + B};$$
$$Legitimate\ Precision = \frac{D}{C + D};$$
$$Precision = \frac{Spam\ Precision + Legitimate\ Precision}{2}.$$

(3) Recall: It refers the ratio of the e-mails classified correctly. The "*spam recall*" is defined as the probability of classifying correctly spammy e-mails as spams, and the "*legitimate recall*" is defined as the probability classifying correctly legitimate e-mails. Then, we set the "*recall*" value as the mean of the spam recall and legitimate recall. The formulas are listed as follows:

$$Spam\ Recall = \frac{A}{A + C};$$
$$Legitimate\ Recall = \frac{D}{B + D};$$
$$Recall = \frac{Spam\ Recall + Legitimate\ Recall}{2}.$$

(4) F-measure: the harmonic mean of the precision and recall with equation listed as follows:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

(5) FP-rate and FN-rate: FP-rate defines the ratio of misjudging legitimate e-mails as spammy, and FN-rate defines the ratio of misjudging spams as legitimate. The formulas are listed as follows:

$$FP\text{-}rate = \frac{B}{B + D}$$
$$FN\text{-}rate = \frac{C}{A + C}$$

4.2. Optimization of parameters

Before performing the experiments, we must optimize the important parameters used in this research. First off, we should optimize the parameters $(P_{\text{lower}}, P_{\text{upper}}, S_{\text{lower}})$, which indicate the threshold values used in step 4 of the ID3 algorithm. By performing many experiments, we found that the size of the decision tree built by the ID3 algorithm can be pruned acceptably if we set $(P_{\text{lower}}, P_{\text{upper}}, S_{\text{lower}})$ as (20, 90, and 2.5%). Therefore, we use these threshold values as stop conditions in step 4 of the ID3 algorithm. That is, step 4 can be interpreted as follows: "If $20\,\% > Purity(C)$ or $Purity(C) > 90\,\%$ or $Support(C) < 2.5\,\%$, then stop."

Then, we must optimize four important parameters (I^+, I^-, M^+, M^-). The parameters I^+ and I^- are used by algorithm *RT_Initial* in the training phase, and M^+ and M^- are used by *RT_Modify* in the re-learning phase. We set the initial values of (I^+, I^-, M^+, M^-) as (1, 1, 1, 1). Then, we adopted all the 6827 e-mails as testing data from the SpamAssassin dataset to observe the variation of the accuracy of our spam filtering system (using both reversing mechanism and re-learning mechanism). We fixed the three parameters (I^+, I^-, M^-) at (1, 1, 1) and gradually changed M^+ to investigate the variation of the accuracy of our method. The results were shown in Table IV. From the experimental results, we observed that the higher value of M^+, the higher value of accuracy. While the M^+ reached 10, the accuracy could not be improved further. Thus, we chose 10 as the optimal value of M^+.

Similarly, we manipulated M^- and fixed the other three parameters (I^+, I^-, M^+) at (1, 1, 10) to observe the variation of accuracy, which was shown in Table V. Obviously, the accuracy could not be improved further, while the M^- reached 7. Therefore, we chose 7 as the optimal value of M^-.

By applying the similar way, we chose 12 as the optimal value of I^-. Then, we tried to optimize the value of I^+. However, the accuracy would decrease while we began to adjust I^+. Hence, we kept I^+ as 1 and finished the optimization work. Finally, we set the optimal values of

Table IV. Optimization of parameter M^+ with $(I^+, I^-, M^-) = (1, 1, 1)$.

M^+	Accuracy of method III
4	0.7486
8	0.7438
9	0.7515
10	0.9054
11	0.9054
–	–

Table V. Optimization of parameter M^- with $(I^+, I^-, M^+) = (1, 1, 10)$.

M^-	Accuracy of method III
4	0.9250
6	0.9442
7	0.9518
8	0.9518
9	0.9518
–	–

parameters as (1, 7, 10, 12). Therefore, we would adopt the optimized parameters in the experiments of the next subsection.

4.3. Experimental analysis

In this subsection, we perform the following three experiments to confirm the efficiency of our spam filtering method proposed in this paper: (A) using SpamAssassin as experimental dataset; (B) using Enron-Spam as experimental dataset; and (C) using Mixed-Set as experimental dataset.

Note that the ratio of legitimate e-mails to spams used in the training phase was different from that in the classification phase. In the training phase, we took randomly 1000 training e-mails in the ratio of 1:1 (500 legitimate e-mails and 500 spams) from the experimental dataset adopted in each experiment. Moreover, in the classification phase, we would continuously increase the amount of testing data (unknown e-mails) to observe the performances of our filtering method. Note that those unknown e-mails were taken randomly from the adopted experimental dataset without any predefined proportion of legitimate e-mails to spams.

In each experiment, the performance of our spam filtering system will be verified by adopting different methods which are combinations of the mechanisms proposed in our filtering system: (I) using neither reversing mechanism nor re-learning mechanism; (II) using only reversing mechanism; and (III) using both reversing mechanism and re-learning mechanism. Note that method III is actually equivalent to the whole filtering system proposed in this paper. The experimental results are discussed as follows.

4.3.1. The result of experiment A

In this experiment, we chose SpamAssassin as the experimental dataset to observe the performance of our filtering system. The result of this experiment was shown in Figure 5. Obviously, the curve of the accuracy of method I, which used neither reversing mechanism nor re-learning mechanism, was lower than those of methods II and III. With the assistance of reversing mechanism in classifying unknown e-mails, the curve of the accuracy of method II was better than that of method I. Moreover, with both of reversing mechanism and re-learning mechanism, method III obtained the most outstanding accuracy. After applying the re-learning mechanism increasingly, the accuracy of method III reached 0.9675, which implied an excellent result.

The performances of method III evaluated in various efficacy assessment indexes were shown in Figure 6. Each curve of the four indexes had outstanding exhibition. Among these four curves, the curve of recall obtained the highest values, which implied that our filtering system would classify the certain category (spammy or legitimate) of e-mails correctly.

4.3.2. The result of experiment B

In this experiment, we chose Enron-Spam as the experimental dataset. The Enron-Spam dataset collected a larger number of e-mails as compared with the SpamAssassin. The experimental results were shown in Figure 7. Obviously, the considerable quantities of testing data had made a great impact on the behaviors of methods I, II, and III. With the support of re-learning mechanism, method III incrementally learned knowledge from numerous unknown e-mails and obtained the best accuracy. As compared with experiment A, the curve of accuracy of method III in this experiment was lifted up and reached 0.9765.

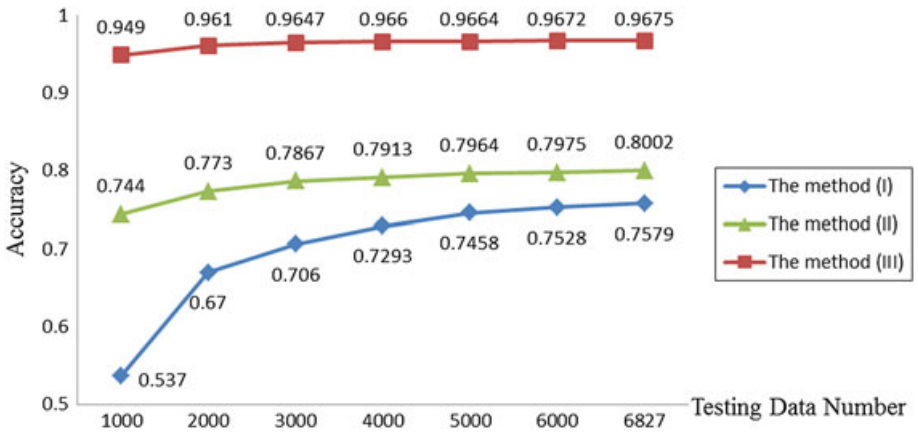

Figure 5. The result of experiment A.

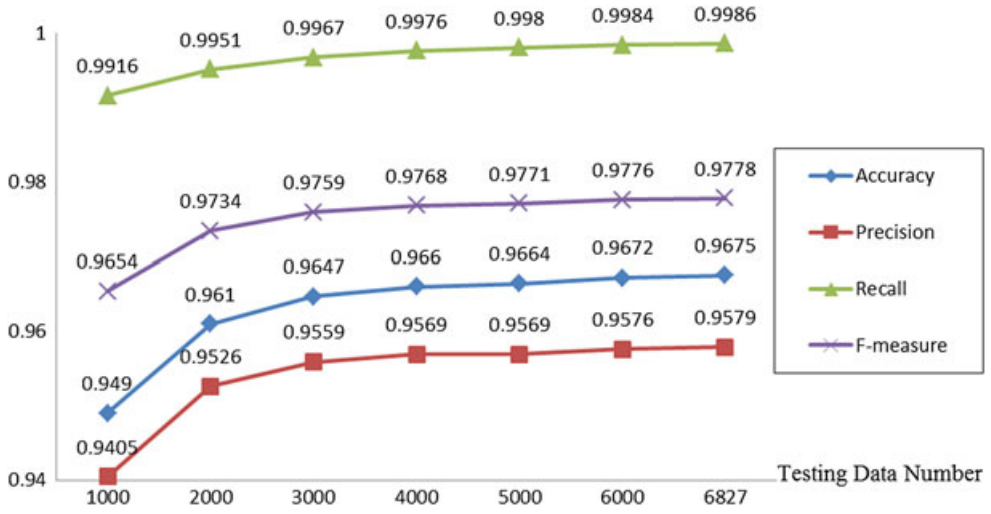

Figure 6. The performance of method III in experiment A.

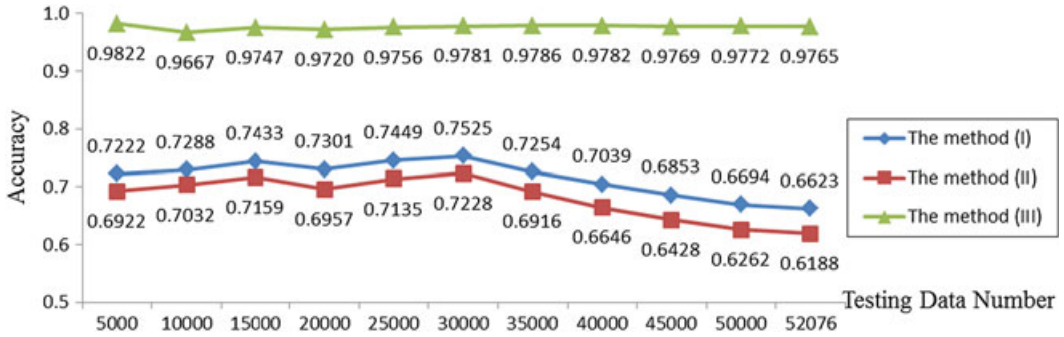

Figure 7. The result of experiment B.

The performances of method III recorded in various efficacy assessment indexes were shown in Figure 8. Compared with Figure 6, the curves of the four indexes in this experiment had better exhibitions. After re-learning from a plenty of unknown e-mails, all curves became stable and reached the desirable values. Moreover, the ratio of

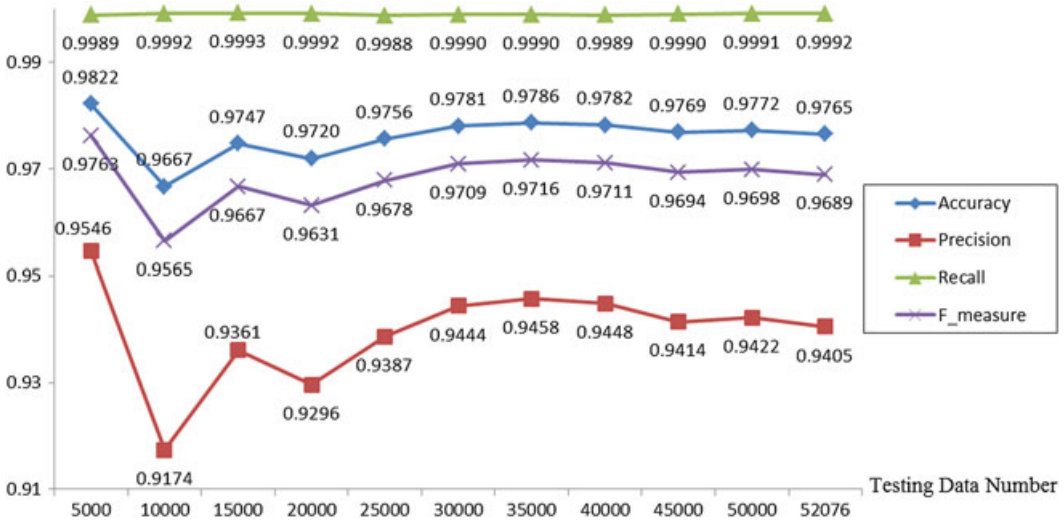

Figure 8. The performance of method III in experiment B.

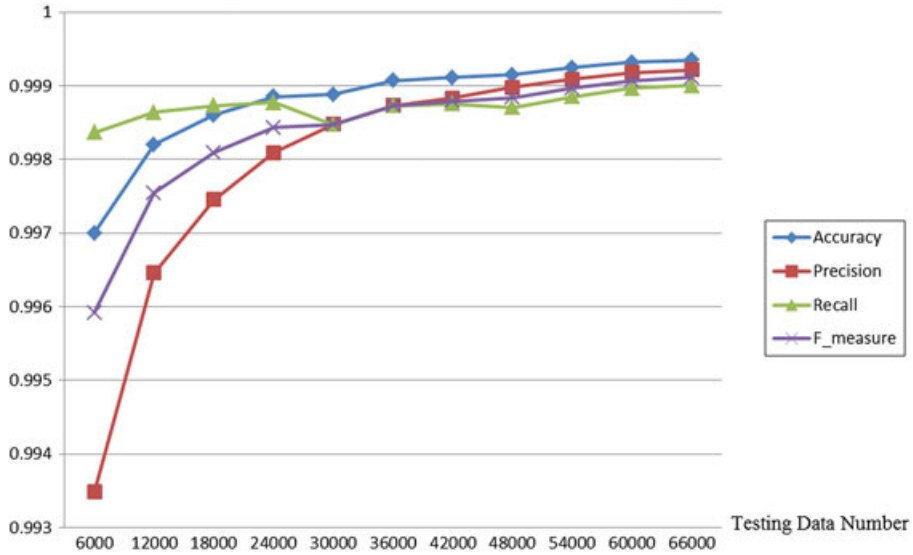

Figure 9. The result of experiment C.

recall eventually reached the highest value of 0.9992, which indicated that our system classified the unknown e-mails perfectly.

4.3.3. The result of experiment C

The previous two experiments had confirmed that the performance of method III, the filtering system proposed in this paper, was excellent. In this experiment, we applied only method III to observe exhibitions of various efficacy assessment indexes. To emphasize the influence of a great quantity of testing data, we adopted the Mixed-Set as experimental dataset and chose a double amount of training e-mails in this experiment. Precisely, in the training phase, we took randomly 2000 training e-mail from the Mixed-Set without any predefined proportion of legitimate e-mails to spams. Then, in the classification phase, we continuously increased the amount of unknown e-mails taken randomly from the Mixed-Set to observe the performances of our spam filtering method.

The results of this experiment were shown in Figure 9. By taking more training data from a plentiful dataset, the ratio of each assessment index acquired a greatly high value, more than 0.99, in the beginning of this experiment. After applying a large number of testing data, all curves of the four assessment indexes approached more closely and reached the extremely desirable numerical values. Moreover, the results of the FP-rate (the ratio of misjudging legitimate e-mails as spammy) and FN-rate (the ratio of misjudging spams as legitimate) in method III of the three experiments were shown in Table VI. Obviously, both the FP-rate and FN-rate were ideal even though the proposed method applied a small experimental dataset. For example, the FP-rate in experiment A was only 0.0014, which showed that our filtering method infrequently misjudged

Table VI. The experimental results of FP-rate and FN-rate.

Experiment	FP-rate	FN-rate
A	0.0014	0.1112
B	0.000786	0.0367
C	0.00099539	0.000454711

Table VII. Comparison of filtering methods.

Methods	Recall (%)	Precision (%)	Accuracy (%)	Checking the whole content of e-mail
The method of Bayes classifier proposed by Tretyakov [21]	87.44	100	94.49	Yes
Incremental clustering-based classification (ICBC) proposed by Hsiao and Chang [36]	96.1	95.76	94.73	Yes
Genetic algorithm proposed by Sanpakdee *et al.* [30]	75.71	89.83	85.53	Yes
The decision tree method proposed by Sheu [4]	96.35	96.67	96.5	No
The decision tree method proposed by Sheu *et al.* [5]	94.00	97.96	96.17	No
The proposed method in this paper: experimental results of method III in experiment B	99.92	94.05	97.65	No
The proposed method in this paper: experimental results of experiment C	99.90	99.92	99.93	No

the legitimate e-mails (that are of vital importance) as spammy ones. Moreover, the considerable quantities of data instances had made a great and incredible improvement on the misjudgment rate, which implied that the filtering method proposed in this paper possessed outstanding performances.

Table VII was the comparison between some spam filtering methods proposed in literatures. Note that some of these methods had to check the whole content of e-mail, whereas our method would check the e-mail's header only. Obviously, our method revealed better accuracy and recall. We could observe that our precision rate of method III in experiment B was inferior to those of the other methods. However, our precision rate of experiment C reached an almost perfect numerical value, which implied that our method could incrementally learn classification knowledge under a great quantity of unknown e-mails to improve itself.

5. CONCLUSIONS

In this research, we proposed an efficient spam filtering method based on the decision tree data mining technique, analyzed the association rules among about spams, and applied these rules to develop a systematized spam filtering method. Different from content checking, we classified e-mails simply by analyzing their basic header data only. Our method possessed the following three major superiorities: (i) checking only the e-mail's header section to avoid the low-operating efficiency in scanning the e-mail's content. Moreover, the accuracy of filtering was enhanced simultaneously. (ii) In order that the probable misjudgment in identifying an unknown e-mail could be "reversed", we had constructed a reversing mechanism to help the classification of unknown e-mails. Thus, the overall accuracy of our filtering method will be increased. (ii) Our method was equipped with a re-learning mechanism, which utilized the supervised machine learning method to collect and analyze each misjudged e-mail. Therefore, the revision information learned from the analysis of misjudged e-mails incrementally gave feedback to our method, and its ability of identifying spams would be improved.

The results of the experiments with a large number of testing data showed that the ratios of assessment indexes —accuracy, recall, precision, F-measure, FP-rate, and FN-rate—approached more closely and reached the extremely desirable numerical values, which implied that the filtering method proposed in this paper possessed outstanding performances. Note that one of advantages of our method was to reduce the calculation cost. Therefore, the method proposed in this paper can classify unknown e-mails precisely and not consume too many system resources, which will be extremely useful in resolving the requirement of judging a large number of unknown e-mails nowadays.

ACKNOWLEDGEMENTS

This work is partially supported by the Ministry of Science and Technology, Taiwan, R.O.C. under grant no. MOST 103-2410-H-004-112.

REFERENCES

1. Hong Kong Anti-SPAM Coalition (HKASC). Legislation: one of the key pillars in the fight against spam. White Paper, 2004.
2. Symantec intelligence report: May 2014. Symantec, 2014.
3. Cook D, Hartnett J, Manderson K, Scanlan J. Catching spam before it arrives: domain specific dynamic blacklists. *In Proceedings of the 2006 Australasian Workshops on Grid Computing and E-Research* 2006; **54**: 193–203.
4. Sheu JJ. An efficient two-phase spam filtering method based on e-mails categorization. *International Journal of Network Security* 2009; **8**(3): 334–343.
5. Sheu JJ, Chu KT. An efficient spam filtering method by analyzing e-mail's header session only. *International Journal of Innovative Computing, Information and Control* 2009; **5**(11): 3717–3731.
6. Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD. An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *In Proceedings of the 23rd Annual International ACM SIGR Conference on Research and Development in Information Retrieval*, 2000; 160–167.
7. Guo Y, Zhou L, He K, Gu Y, Sun Y. Bayesian spam filtering mechanism based on decision tree of attribute set dependence in the mapreduce framework. *Open Cybernetics & Systemics Journal* 2014; **8**: 435–441.
8. Han J, Kamber M. *Data Mining Concepts and Techniques*. Morgan Kaufman: USA, 2001; 284–287.
9. Zhou B, Yao Y, Luo J. Cost-sensitive three-way email spam filtering. *Journal of Intelligent Information Systems* 2014; **42**(1): 19–45.
10. Drucker H, Wu D, Vapnik V. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks* 1999; **10**(5): 1048–1054.
11. Carreras X, Marquez L. Boosting trees for anti-spam email filtering. *4th International Conference on Recent Advances in Natural Language Processing* (*RANLP*), Bulgaria, Sep. 5–7, 2001; 58–64.
12. DeBarr D, Wechsler H. Spam detection using random boost. *Pattern Recognition Letters* 2012; **33**(10): 1237–1244.
13. Islam MR, Zhou W, Guo M, Xiang Y. An innovative analyser for multi-classifier e-mail classification based

on grey list analysis. *Journal of Network and Computer Applications* 2009; **32**: 357–366.
14. Shih DH, Chiang HS, Lin B. Collaborative spam filtering with heterogeneous agents. *Expert Systems with Applications* 2008; **35**(4): 1555–1566.
15. Shrivastava JN, Bindu MH. E-mail spam filtering using adaptive genetic algorithm. *International Journal of Intelligent Systems and Applications (IJISA) 2014* 2014; **6**(2): 54–60.
16. Golbeck J, Hendler J. Reputation network analysis for email filtering. *In Proceedings of the First Conference on Email and Anti-Spam* (*CEAS*), 2004.
17. Liu YN, Han Y, Zhu XD, He F, Wei LY. An expanded feature extraction of e-mail header for spam recognition. *Advanced Materials Research* 2013; **846**: 1672–1675.
18. Wang CC, Chen SY. Using header session messages to anti-spamming. *Computers & Security* 2007; **26**: 381–390.
19. Lai CC. An empirical study of three machine learning methods for spam filtering. *Knowledge-Based Systems* 2007; **20**(3): 249–254.
20. Sebastiani F. Machine learning in automated text categorization. *ACM Computing Surveys* 2002; **34**(1): 1–47.
21. Tretyakov K. Machine learning techniques in spam filtering. Technical report, Institute of Computer Science, University of Tartu, 2004.
22. Jayaraj A, Venkatesh T, Murthy CSR. Loss classification in optical burst switching networks using machine learning techniques: improving the performance of tcp. *IEEE Journal on Selected Areas in Communications* 2008; **26**(6): 45–54.
23. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer: New York, 2001.
24. Breiman L. Random forests. *Machine Learning* 2001; **45**: 5–32.
25. Quinlan JR. Induction of decision trees. *Machine Learning* 1986; **1**(1): 81–106.
26. Quinlan JR. *C4.5: Programs for Machine Learning*. Morgan Kaufmann: San Mateo, 1993.
27. Stark KD, Pfeiffer DU. The application of non-parametric techniques to solve classification problems in complex data sets in veterinary epidemiology — an example. *Intelligent Data Analysis* 1999; **3**(1): 23–35.
28. Ohmann C, Moustakis V, Yang Q, Lang K. Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. *Artificial Intelligence in Medicine* 1996; **8**(1): 23–36.
29. Porter MF. An algorithm for suffix stripping. *Program* 1980; **14**: 130–137.
30. Sanpakdee U, Walairacht A, Walairacht S. Adaptive spam mail filtering using genetic algorithm. *The 8th International Conference Advanced Communication Technology*. Phoenix Park, Korea, Feb. 20–22, 2006; 441–445.
31. Schwartz A. *SpamAssassin*. O'Reilly, 2004.
32. Enron-SPAM datasets: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html
33. Delany SJ, Cunningham P, Tsymbal A, Coyle L. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems* 2005; **18**: 187–195.
34. Fdez-Riverola F, Iglesias EL, Díaz F, Me'ndez JR, Corchado JM. Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems with Applications* 2007; **33**(1): 36–48.
35. Fdez-Riverola F, Iglesias EL, Díaz F, Me'ndez JR, Corchado JM. Spamhunting: an instance-based reasoning system for spam labelling and filtering. *Decision Support Systems* 2007; **43**(3): 722–736.
36. Hsiao WF, Chang TM. An incremental cluster-based approach to spam filtering. *Expert Systems with Applications* 2008; **34**:1599–1608.