Journal of Universal Computer Science, vol. 17, no. 6 (2011), 944-960
submitted: 15/5/10, accepted: 30/11/10, appeared: 28/3/11 © J.UCS

Cost-Sensitive Spam Detection Using Parameters Optimization and Feature Selection

Sang Min Lee
(Department of Computer Engineering, Korea Aerospace University, Seoul, Korea
minuri33@kau.ac.kr)

Dong Seong Kim
(Department of Electrical and Computer Engineering, Duke University, Durham, USA
dongseong.kim@duke.edu)

Jong Sou Park
(Department of Computer Engineering, Korea Aerospace University, Seoul, Korea
jspark@kau.ac.kr)

Abstract: E-mail spam is no more garbage but risk since it recently includes virus attachments and spyware agents which make the recipients' system ruined, therefore, there is an emerging need for spam detection. Many spam detection techniques based on machine learning techniques have been proposed. As the amount of spam has been increased tremendously using bulk mailing tools, spam detection techniques should counteract with it. To cope with this, parameters optimization and feature selection have been used to reduce processing overheads while guaranteeing high detection rates. However, previous approaches have not taken into account feature variable importance and optimal number of features. Moreover, to the best of our knowledge, there is no approach which uses both parameters optimization and feature selection together for spam detection. In this paper, we propose a spam detection model enabling both parameters optimization and optimal feature selection; we optimize two parameters of detection models using Random Forests (RF) so as to maximize the detection rates. We provide the variable importance of each feature so that it is easy to eliminate the irrelevant features. Furthermore, we decide an optimal number of selected features using two methods; (i) only one parameters optimization during overall feature selection and (ii) parameters optimization in every feature elimination phase. Finally, we evaluate our spam detection model with cost-sensitive measures to avoid misclassification of legitimate messages, since the cost of classifying a legitimate message as a spam far outweighs the cost of classifying a spam as a legitimate message. We perform experiments on Spambase dataset and show the feasibility of our approaches.

Keywords: Feature Selection, Intrusion Detection, Parameters Optimization, Random Forests, Spam Detection, Spambase
Categories: I.2.6, I.5.1, K.6.5, L.4.0

1 Introduction

An electronic mail (e-mail) is an efficient and increasingly popular communication method. Concern about the proliferation of unsolicited bulk e-mail, commonly referred to as "spam", has been steadily increasing [Cranor and LaMacchia 98]. When people receive in a small amount of spam, it rarely poses a significant problem.

However, as the quantities of spam have been increased tremendously because of bulk mailing tools (bulk-mailers), the recipients become increasingly annoyed and Internet Service Providers (ISPs) have been deluged with complaints and spam places a considerable burden on the system. Moreover, virus attachments, spyware agents and phishing have become the most serious security threats to individuals and businesses recently.

A variety of technical and regulatory countermeasures against spam have been proposed. First step to counteract to spam is to detect it. Spam detection models are mainly divided into two approaches: non-statistical and statistical approaches. The latter is generally more powerful than the former. Existing statistical detection models search for particular keyword patterns in e-mails. A number of spam detection models using machine learning techniques have been proposed. It is necessary to reduce the resources for processing to detect spam because it should keep up with the huge amounts of e-mails by bulk-mailers. To decrease the amount of consuming resources with guaranteeing high detection rates, parameters optimization of spam detection models using machine learning techniques (e.g., threshold function value, the number of hidden layers in artificial neural networks) and feature selection of audit data (which figures out what feature of audit data is more important and needs to be selected in detection of spam mail) can be used. Parameters optimization is used to find out optimal parameters of spam detection models. Previous approaches [Abu-Nimeh *et al*. 08; Zhao 04] considered the parameters optimization of spam detection models but they did not show the details of it. Feature selection is used to find out only important features or feature set out of all the features of audit data. The feature selection enables one to eliminate irrelevant features to avoid processing overheads. Previous feature selection approaches [Bursteinas and Long 00; Thota *et al*. 09; Zhao and Zhu 06; Zhu 08] were proposed but they did not provide how they performed the feature selection. [Liang *et al*. 08] performed the feature selection via feature ranking algorithm but detection rates were not improved. Especially, there is no approach that shows how the number of features is determined in their experiments. Furthermore, there is no work incorporating both parameters optimization and feature selection together.

In this paper, we propose a novel spam detection model which uses both parameters optimization and optimal feature selection in a unified manner. We adopt Random Forests (RF) which is a stage-of-the-art machine learning algorithm [Breiman 01]. For parameters optimization of spam detection, two parameters (*mtry* and *ntree*) of RF are optimized to maximize spam detection rates. For feature selection, we concurrently provide variable importance of individual feature to select the important features on a scale between 0 and 1. This variable importance represents how each feature is significant for spam detection so that our approach can select relevant features and remove irrelevant ones. We then figure out the optimal number of selected features using two methodologies: (i) only one parameters optimization during overall feature selection and (ii) parameters optimization in every feature elimination phase. According to these procedures, our approach can detect spam with low processing overheads while guaranteeing high detection rates.

We take into account cost-sensitive measures for our spam detection model, since the cost of misclassifying legitimate messages can be much higher than the cost of misclassifying spam messages.

A preliminary version of this paper appeared in [Lee *et al*. 10].

The rest of the paper is as organized as follows. In [Section 2], related work and brief description of RF are presented. Our proposed spam detection model is presented in [Section 3] and evaluation results and analysis are followed in [Section 4]. Finally, we conclude the paper in [Section 5].

2 Background

2.1 Related Work

Spam is generally defined as "unsolicited, usually commercial, e-mail sent to a large number of recipients". To cope with it, spam detection models which automatically classifies spam or non-spam have been researched under two categories; non-statistical and statistical approaches. The problem with non-statistical approaches is that there is no learning component to admit messages whose content 'look' legitimate. This may lead to undetected spam and a frustrating proliferation of automatic answer-seeking replies [Ravi Kiran and Atmosukarto 05].

Due to the above limitations, researches have used machine learning algorithms. One of the widely used methods, Bayesian classification, attempted to calculate the probability that a message is spam based upon previous feature frequencies in spam and non-spam [Androutsopoulos *et al*. 00a; Graham 03; Sahami *et al*. 98]. A notable example is the open source software SpamBayes. Support vector machines (SVM) was used in spam classification [Drucker *et al*. 99; Zhu 08]. Other popular learning algorithms applied to spam detection are Boosting [Carreras and Marquez 01] and Artificial Neural Networks (ANN).

However, those approaches still impose large overheads due to heavy computation and low detection rates because they did not use feature selection i.e., they used irrelevant features [Duda *et al*. 01]. Also, they did not optimize the parameters in machine learning algorithm. The objective of parameters optimization is to adjust the value of several parameters in machine learning algorithms and to figure out optimal values of them. For example, the weight values and a number of hidden layers on ANN, value of parameters of kernel function of SVM [Salem and Stolfo 10; Xie 07] and so on. [Abu-Nimeh *et al*. 08; Zhao 04] showed parameters optimization on their algorithms but they did not explain how they computed the optimal parameters values. In addition, those approaches did not apply feature selection to spam detection. The objective of feature selection is to figure out relevant features among whole features of audit data to decrease processing time and improve detection rates. All features are not essential to classify whether an e-mail is a legitimate or spam, because irrelevant features not only increase computational costs (such as, time and resources) but also decrease the classification rate. [Bursteinas and Long 00; Thota *et al*. 09; Zhao and Zhu 06; Zhu 08] performed feature selection but they did not mention how they decided the number of important features, and they did not provide variable importance of each feature as a numerical value. Although [Liang *et al*. 08] performed the feature selection using feature ranking algorithm but the detection rates are very low. Especially, there is no approach which used both parameters optimization and feature selection together.

In this paper, we present a new spam detection which incorporates both parameters optimization and feature selection using RF. We introduce RF in brief in next section.

2.2 Overview of Random Forests

Random Forests (RF) is a special kind of ensemble learning techniques and robust concerning the noise and the number of attributes [Breiman 01]. RF builds an ensemble of CART tree classifications using bagging mechanism [Duda *et al*. 01]. By using bagging, each node of trees only selects a small subset of features for the split, which enables the algorithm to create classifiers for high dimensional data very quickly. This counterintuitive strategy turns out to perform very well compared to the state-of-the-art methods in classification and regression. Also, RF runs efficiently on large data sets with many features [Zhang and Zulkernine 05] and its execution speed is fast [Yang *et al*. 08]. RF produces additional facilities, especially the variable importance by numerical values [Breiman 01]. Thus, the variable importance is able to make one easily figure out which features are important or not. Using this facility, we can figure out important features and eliminate the irrelevant features. In addition, since RF has only two parameters, it is relatively easy to regulate them; the number of variables in the random subset at each node (*mtry*) and the number of trees in the forest (*ntree*), and RF is usually not very sensitive to their values. However, it is important to optimize those two parameters to maximize the classification accuracy. In this paper, we use the variable importance for the optimal feature selection phase and optimize *mtry* and *ntree* in parameters optimization phase. The next Section presents our proposed spam detection model.

3 Proposed Spam Detection Model

3.1 Overall Flow of Proposed Spam Detection Model

An overall flow of our proposed approach is shown in [Fig. 1]. At first, experimental dataset [Spambase 99] is divided into training set and testing set. An initial spam detection model is built by performing only parameters optimization on training set. Then, detection rates are compared with a predefined threshold value of detection rates, *T*. If it satisfies a design requirement of spam detection model, it finishes the phase. Otherwise, it continues its phases to rebuild a spam detection model with optimal feature selection. We propose two approaches for optimal feature selection. First approach, (a) in [Fig. 1], is to use initial parameters values of spam detection model until rebuilding process is finished (i.e., parameters optimization is performed once, then the optimal parameters values are used during one overall phases). Second approach, (b) in [Fig. 1], is to perform parameters optimization whenever we eliminate an irrelevant feature using the feature selection results. These two approaches will be described in [Section 3.2.5] in detail. After a final spam detection model is constructed, it is evaluated once more by 5-fold cross validation for an unbiased detection evaluation. At last, the detection model is evaluated using testing set in terms of cost-sensitive measures to take into account the high cost of misclassifying legitimate mails.

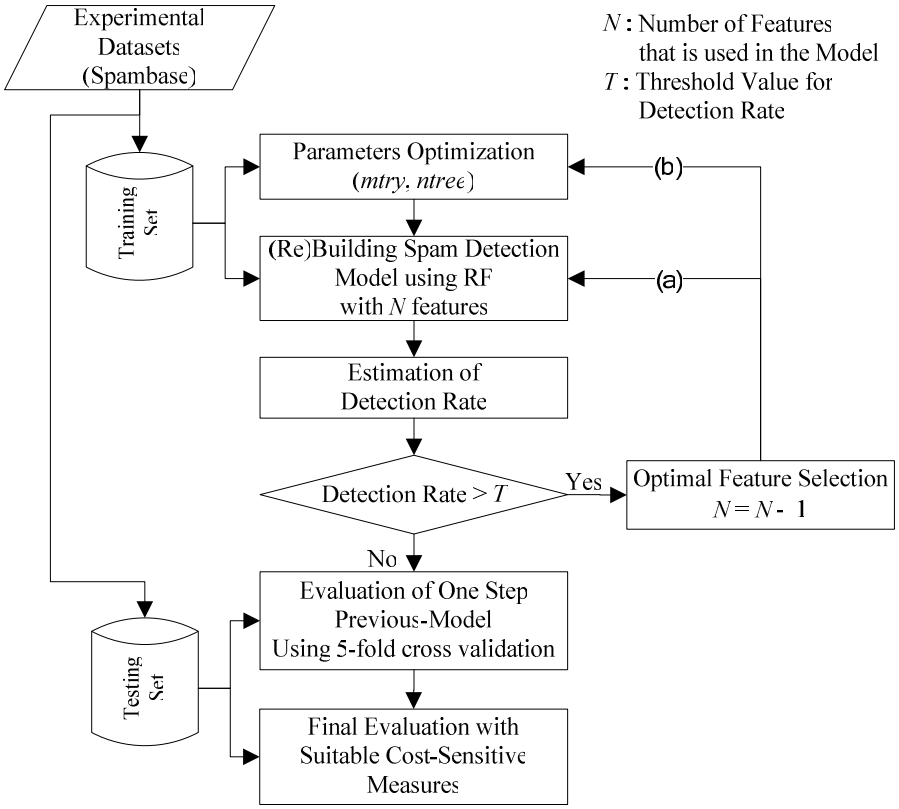

Figure 1: Overall flow of proposed approach

3.2 Detailed Description of Proposed Model

3.2.1 Parameters Optimization of Spam Detection model

It is important to guarantee high detection rates. We optimize two parameters of RF: *mtry* and *ntree*. One specific function is adopted to get the optimal value of *mtry*. This will be described in [Section 4] in detail. For the optimal value of *ntree*, we carry out experiments with large enough *ntree* value. Then, we choose the optimal *ntree* value when the detection rates are highest and the *ntree* value is stable and lowest simultaneously. We can figure out two optimal parameters so that it may reduce computational overheads and guarantee high detection rates.

3.2.2 Building Initial Spam Detection Model

In this phase, an initial spam detection model is built using RF with all *N* number of features of dataset. We use the optimal parameters for the detection model which is

constructed from the previous phase. Once again, an initial spam detection model does not use feature selection and it uses whole features variable of dataset.

3.2.3 Evaluation using Detection Rates

Through the previous phases, a confusion matrix (which shows true positive, true negative, false positive and false negative value of classification (detection) results of RF) is generated. Generally, the cost of losing a legitimate message (false positives) is much greater than that of allowing a spam message (false negatives). In this paper, however, we only estimate the detection rates (accuracy) of the spam detection model since we only focus on spam detection and not cost. The detection rates (accuracy, *Acc*) are defined as equation (1):

$$\frac{\text{Number of true positives + Number of true negatives}}{\text{Total number of all instances in a dataset}} \tag{1}$$

The error rate (*Err* = 1 - *Acc*) is defined as equation (2):

$$\begin{aligned}&\frac{\text{Number of false positives + Number of false negatives}}{\text{Total number of all instances in a dataset}}\\ &= 1 - \text{Equation (1)}\end{aligned} \tag{2}$$

Then, we compare the computed result with a predefined certain threshold value, *T*. Here, *T* value is determined by considering tradeoffs between detection rates and processing resources. In this paper, the number of features which is used for the final experiment is considered as processing overheads since the irrelevant features can cause processing resources. If the detection rates are greater than *T* value, it goes to optimal feature selection phase. It means that the spam detection model can be rebuilt with less number of features until detection rates are high enough to satisfy the design criteria. Otherwise, we finish the iteration and evaluate the previous-spam detection model of which detection rates are greater than *T* value because the model which has too low detection rates are useless even though it consumes less processing resources.

3.2.4 Feature Selection

RF is able to compute each feature importance as a numerical value. We rank the whole features in descending order with respect to their feature importance value, and eliminate an irrelevant feature which is the lowest ranked. In other words, we can select important features. This enables our approach to reduce computational overheads of dataset as well as to enhance the detection rates.

3.2.5 Rebuilding Spam Detection Model

We propose two approaches in rebuilding spam detection model to decide the optimal number of selected features.

- *(a) Only One Parameters Optimization during Overall Feature Selection*

First approach is to only perform parameters optimization of detection model once at an initialization. Then, the computed optimal parameters values are used until the rebuilding process is finished. This approach has an advantage of saving experiment time (processing resources) but the initial optimal parameters values may not be the optimal parameters values of the rebuilt models because the number of features of rebuilt models is changed in every feature elimination phase.

- *(b) Parameters Optimization in Every Feature Elimination Phase*

Second approach is to perform parameters optimization in every feature elimination phase. This approach may take longer time to finish its phases but it is able to design a more optimal spam detection model compared to the first approach.

3.2.6 Evaluation of Previous Model

To verify the effectiveness of our approach using parameters optimization and feature selection simultaneously, we perform 5-fold cross validation. The previous researches performed either parameters optimization or feature selection. However, our approach uses both of them. It is able to not only reduce computational overheads but also increase detection rates.

3.2.7 Final Evaluation with Suitable Cost-Sensitive Measures

Detection model is usually evaluated in terms of detection rates (accuracy) and false rate (false positives). We incorporate cost-sensitive evaluation measures that assign false positives a higher cost than false negatives because blocking legitimate messages (false positives) mistakenly is more severe than letting spam messages pass the spam detector (false negatives). This is based on the assumption that most users can tolerate a small percentage of mistakenly admitted spam messages, while they consider losing legitimate messages much more damaging. We use Weighted Accuracy (W_{Acc}) and Weighted Error Rate (W_{Err} = 1 – W_{Acc}) used in several approaches [Androutsopoulos *et al*. 00b; Carreras and Marquez 01; Sakkis *et al*. 03; Zhang *et al*. 04]. Let us denote S and L for spam and legitimate messages, respectively. Also, let us denote $n_{L\to L}$, $n_{S\to S}$ numbers of legitimate and spam messages correctly classified by the system, respectively. And let us denote $n_{L\to S}$ the number of legitimate messages misclassified as spam (false positives), and $n_{S\to L}$ is the number of spam messages wrongly treated as legitimate (false negatives). Then, W_{Acc} and W_{Err} (1 - W_{Acc}) is defined as:

$$W_{Acc} = \frac{\lambda \cdot n_{L\to L} + n_{S\to S}}{\lambda \cdot N_L + N_S}, \qquad W_{Err} = \frac{\lambda \cdot n_{L\to S} + n_{S\to L}}{\lambda \cdot N_L + N_S} \tag{3}$$

Where N_L is the total number of legitimate messages and N_S is the total number of spam messages. W_{Acc} treats each legitimate message as if it were λ messages: when false positive occurs, it is counted as λ errors; and when it is classified correctly, these counts as λ successes. The higher λ is, the more cost is penalized on false positives.

[Androutsopoulos *et al*. 00b; Zhang *et al*. 04] introduced three different values of λ: λ = 1, 9 and 999. When λ is set to 1, spam and legitimate mails are weighted equally; when λ is set to 9, a false positive is penalized nine times more than a false negative; for the setting of λ = 999, more penalties are put on false positives: misblocking legitimate messages is as bad as letting 999 spam messages pass the detector. This cost introduces a very high bias for classifying messages as legitimate, which may be reasonable when blocked messages are deleted automatically without further processing because most users would consider losing legitimate messages from their mailboxes unacceptable.

In practice, when λ is assigned a high value (such as λ = 999), W_{Acc} can be so high that it tends to be easily misinterpreted. To avoid this problem, it is better to compare the weighted accuracy and error rate to a simplistic baseline. As suggested in [Androutsopoulos *et al*. 00c; Zhang *et al*. 04], we use the case where no detector is present as baseline: legitimate messages are never blocked and spam messages can always pass the detector. Then the baseline versions of weighted accuracy and weighted error rate are:

$$W_{Acc}{}^{b} = \frac{\lambda \cdot N_L}{\lambda \cdot N_L + N_S}, \qquad W_{Err}{}^{b} = \frac{N_S}{\lambda \cdot N_L + N_S} \tag{4}$$

To allow easy comparison with the baseline, we use Total Cost Ratio (*TCR*) [Androutsopoulos *et al*. 00c; Zhang *et al*. 04] as a single measurement of the spam detecting effects:

$$TCR = \frac{W_{Err}{}^{b}}{W_{Err}} = \frac{N_S}{\lambda \cdot n_{L \to S} + n_{S \to L}} \tag{5}$$

Here higher *TCR* values indicate better detection performance. If *TCR* is less than 1.0, then the baseline (not using the detector) is better. If cost is proportional to wasted time, an intuitive meaning for *TCR* is the following: it measures how much time is wasted to delete manually all spam messages when no filter is used (N_S), compared to the time wasted to delete manually any spam messages that passed the detector ($n_{S \to L}$) plus the time needed to recover from mistakenly blocked legitimate messages ($\lambda \cdot n_{L \to S}$). An effective spam detector should be able to achieve a *TCR* value higher than 1.0 in order to be useful in real world applications.

In addition, it is vital to compare false positive rates of detection models since false positive rate is more expensive than false negative rate. The Receiver Operating Characteristic (ROC) curve is a graph to plot false positive rate vs. true positive rate, in which various threshold values are compared. In consequence, we compute the Area under the ROC curve (AUC) for our detection model.

4 Experiments and Analysis

In this Section, we show experimental results on Spambase dataset [Spambase 99]. The dataset is split into training set and testing set. Training set is used to construct a

spam detection model and conduct parameters optimization and feature selection. Testing set is used for a final cost-sensitive evaluation. We first present experiments on parameters optimization for RF and describe feature selection experiments to figure out important features and to eliminate irrelevant features. Then, we figure out an optimal number of selected features. Finally, we evaluate our approach with cost-sensitive measures to take into account the high cost of misclassifying legitimate mails and show the results. Before we present the experimental results, the next Section describes evaluation dataset and experimental environments.

4.1 Experimental Data and Environments

We used the Spambase dataset. The Spambase dataset is an e-mail message collection containing 4601 messages, being 1813 (39%) marked as spam messages, was created by Hopkins *et al*. [Spambase 99]. The legitimate messages were donated by Forman and collected from a single mailbox. The collection comes in pre-processed (not raw) form, and its instances have been represented as 58-dimensional vectors (a.k.a, 58 feature variables, in short, features). The first 48 features are words extracted from the original messages, without stop list nor stemming, and selected as the most unbalanced words for the spam class. The next 6 features are the percentage of occurrences of the special characters ";", "(", "[", "!", "$" and "#". The following 3 features represent different measures of occurrences of capital letters in the text of the messages. Finally, the last feature is the class label which indicates whether an instance is a spam or legitimate mail. Some researchers considered the Spambase dataset obsolete, as it does not represent the state of practical spam messages; however, others considered it is a good test bed (dataset) for evaluating learning techniques [Koprinska *et al*. 07]. Spambase dataset, which is divided into training set and testing set, was used for the experiments. Open source R-project (R version 2.9.1) [Xie 07; R Project 09] and WEKA 3.6.1 [WEKA 08] tools were used to perform experiments.

4.2 Experimental Results and Analysis

We build spam detection model using RF. There are two parameters to be regulated in RF: (i) the number of variables in the random subset at each node (*mtry*) and (ii) the number of trees in the forest (*ntree*). To get the best detection rates, it is essential to optimize two parameters. We can find out an optimal *mtry* value by using 'tuneRF()' function which is provided in randomForest package in R-project [Xie 07; R Project 09] and the computed *mtry* values are shown in [Tab. 1]. The optimal *mtry* value was proportional to the number of features. In case of *ntree*, there was no specific built-in function, so we regulated an optimal *ntree* value by carrying out experiments as varying *ntree* values ranging from 0 to 500 (Note that detection rates did not increase even we increase *ntree* value over 500). The optimal *ntree* value was evaluated with respect to detection rates. The experimental results for determination of the optimal *ntree* value are depicted in [Fig. 2]. It shows that detection rates of detection model (in terms of accuracy) tuned out the highest and stable when *ntree* = 140. As the result of experiments for the initial spam detection model, we set two optimized parameter values; *mtry* = 7, *ntree* = 140.

Number of Features	Optimal *mtry* value
57 ~ 49	7
48 ~ 34	6
33 ~ 25	5
24 ~ 16	4
15 ~ 9	3
8 ~ 4	2
3 ~ 2	1

Table 1: The optimal mtry value

Figure 2: ntree values VS. accuracy

Now, we perform feature selection. The first approach for feature selection is to perform only one-time parameters optimization during overall feature selection procedure. The second approach for feature selection is to compute optimal parameters values (e.g., *mtry and ntree* values) in every feature elimination phase, since we assume that optimal *mtry and ntree* values may vary with respect to the number of selected features. The experimental results of both approaches are shown in [Tab. 2]. It presents the optimal *mtry* and *ntree* values and the detection rates. The detection rates were computed using 5-fold cross validation by WEKA [WEKA 08]. The threshold value of detection rates, *T* are presented in [Section 3.2.3], was set to 95%. We compared our approaches with previous approaches [Abu-Nimeh *et al*. 08; Bursteinas and Long 00; Fontana 08; Liang *et al*. 08; Thota *et al*. 09; Xie 07; Zhao

and Zhu 06; Zhu 08] in terms of detection rates and processing overheads (the number of features in this paper). Only [Abu-nimeh *et al*. 08] had little higher detection rates than 95% and the rests of them had less than 95%. Also, most of them used whole features (processing overheads). For these two reasons, we set the threshold value to 95%. According to the experimental results, the optimal number of selected important features was 26, 19 for first and second approach, respectively. Second approach consumes less processing resources when it builds a detection model than first approach. Even though the second approach takes more time to rebuild the spam detection model, the optimal spam detection model can be used continuously once it is built. [Tab. 2] shows that the initial spam detection model that used all 57 features of the Spambase dataset [Spambase 99] and the rebuilt models used the reduced number of features by using two feature selection approaches.

Number of Used Features	**First Approach (*mtry* = 7, *ntree* = 140 are set using whole the feature)**	**Second Approach (*mtry* and *ntree* are regulated whenever feature selection is performed)**		
	Detection Rates	***mtry***	***ntree***	**Detection Rates**
57	95.4358	7	140	95.4358
56	95.3054	7	381	95.5227
55	95.4358	7	352	95.4358
…				
27	95.0446	5	342	95.2402
26	**95.1098**	5	246	95.1967
25	**94.8707**	5	168	95.1532
…				
20	94.7837	4	318	95.0446
19	94.6968	4	168	**95.0011**
18	94.6968	4	212	**94.6533**
17	94.7185	4	382	94.5447
16	94.3056	4	113	94.3491

Table 2: The experimental results of both approaches

Optimal feature selection of Spambase dataset was carried out employing the variable importance of features, supported by RF [Breiman 01]. As the results, feature importance of each individual feature was computed as a numerical value and we ranked features with respect to the average variable importance. We partially show the top 10 and bottom 10 features and their average variable importance in [Tab. 3]. The most important feature of the optimal spam detection model was 'char_freq_!' which represents "percentages of characters in the e-mail that match !" property and next feature was 'word_freq_remove' which represents "percentage of words in the e-mail that match remove" and so on. As the above, most of the attributes indicate

whether a particular word or character was frequently occurring in e-mails. In general, spam messages usually include a lot of special characters [Ravi Kiran and Atmosukarto 05]. Especially, spam messages include a lot of '!' (an exclamation mark) since it sometimes means emphasis and helps messages to be conspicuous. Moreover, the objective of the bulk of spam mails is related with sales, so phrases including financial words or characters, such as 'credit' and '$' are often enough. The most irrelevant feature of our proposed approach was 'word_freq_table' which represents "percentage of words in the e-mail that match table" and next feature was 'word_freq_parts' which represents "percentage of words in the e-mail that match parts" and so on.

Rank	**Features**	**Average Variable Importance**
1	char_freq_!	0.5021
2	word_freq_remove	0.4838
3	word_freq_credit	0.4740
4	char_freq_$	0.4739
5	word_freq_hp	0.4725
6	word_freq_edu	0.4687
7	capital_run_length_longest	0.4644
8	word_freq_free	0.4490
9	capital_run_length_total	0.4448
10	word_freq_george	0.4431
…		
48	word_freq_conference	0.2270
49	word_freq_cs	0.2106
50	word_freq_make	0.2044
51	word_freq_addresses	0.1994
52	word_freq_857	0.1755
53	word_freq_415	0.1527
54	word_freq_direct	0.1521
55	word_freq_3d	0.1246
56	word_freq_parts	0.1241
57	word_freq_table	0.0440

Table 3: Rank of average variable importance

In [Tab. 4], we present the comparisons between our approaches and previous approaches. [Abu-Nimeh *et al*. 08; Xie 07] performed the parameters optimization and [Bursteinas and Long 00; Liang *et al*. 08; Thota *et al*. 09; Zhao and Zhu 06; Zhu

08] performed the feature selection. However, there are no approaches that used both of them together. It is very unique that we perform both of parameters optimization and feature selection together. [Abu-Nimeh *et al*. 08] showed the highest detection rates (95.43%) but they used all 57 features using RF. It is the same result with our approach's when we also used all 57 features [see Tab. 2]. Also, in [Tab. 2], our second approach's detection rates using 56 features were higher (95.5227%). Although we did not show the whole results in [Tab. 2], there are more results which detection rates are higher than 95.43% and they used less features than all 57 features. If we set the threshold value as 95.5%, our results would be the best result. However, we set the threshold value as 95% with considering other previous approaches' results and tradeoffs between detection rates and processing resources. Even though the final experimental results of our approaches show a little degradation on detection rates, it is marginally small (because threshold value was set to 95%) and we consume less processing resources (because we use less features). In the rest of previous approaches, the detection rates of our approach are higher than them. Furthermore, we provide how the optimal number of selected features is determined by using variable importance and two threshold values. In summary, these results proved that our approaches outperform than the others.

Approaches	**Feature Selection**	**Parameters Optimization**	**Detection Rates**
[Abu-Nimeh *et al*. 08]	X	O	95.43
[Bursteinas and Long 00]	Multivariate Decision Trees	X	87.54
[Fontana 08]	X	X	92.97
[Liang *et al*. 08]	Distance Discriminant	X	92.00
[Thota *et al*. 09]	False Discovery Rate	X	93.00
[Xie 07]	X	O	94.00
[Zhao and Zhu 06]	Forward selection	X	94.90
[Zhu 08]	Rough set theory	X	94.60
Our First Approach	Variable importance	O	95.11
Our Second Approach	Variable importance	O	95.00

Table 4: Comparisons with the previous approaches

In addition, we evaluated our approaches with cost-sensitive measures to consider the high cost of misclassifying legitimate messages. [Abu-Nimeh *et al*. 08; Zhang *et al*. 08] evaluated their approaches on Spambase dataset [Spambase 99] with

employing suitable cost-sensitive measures. [Abu-Nimeh *et al.* 08] carried out adopting Random Forests (RF), Support Vector Machines (SVM), Neural Networks (NNet), Classification and Regression Trees (CART), Logistic Regression (LR) and Naive Bayes (NB). [Zhang *et al.* 08] carried out using SVM, Naive Bayes and cost-sensitive Multiobjective Genetic Programming (csMOGP). [Tab. 5] shows the results of them and ours. [Abu-Nimeh *et al.* 08] did not calculate *TCR* so we could only compare the W_{Acc}. Our approach is comparable to the RF result and superior to the others. In the case of [Zhang *et al.* 08], our approach also outperformed the results of [Zhang *et al.* 08] except the result of csMOGP (λ = 999) but the difference is subtle. The set of values for the *TCR* measure is shown in [Tab. 5]. Here, higher values imply better spam detection performance. Our approaches give the highest *TCR* scores for all combinations with exception of SVM (λ = 999).

	W_{Acc} (λ = 1)	*TCR*	W_{Acc} (λ = 9)	*TCR*	W_{Acc} (λ = 999)	*TCR*
RF [Abu-Nimeh *et al.* 08]	95.43	*N/A*	96.78	*N/A*	97.18	*N/A*
BART [Abu-Nimeh *et al.* 08]	93.50	*N/A*	96.27	*N/A*	96.84	*N/A*
SVM [Abu-Nimeh *et al.* 08]	93.70	*N/A*	95.22	*N/A*	95.54	*N/A*
NNet [Abu-Nimeh *et al.* 08]	94.72	*N/A*	95.36	*N/A*	95.54	*N/A*
CART [Abu-Nimeh *et al.* 08]	89.93	*N/A*	92.18	*N/A*	92.64	*N/A*
LR [Abu-Nimeh *et al.* 08]	92.70	*N/A*	94.72	*N/A*	95.14	*N/A*
NB [Abu-Nimeh *et al.* 08]	73.67	*N/A*	62.02	*N/A*	59.64	*N/A*
SVM [Zhang *et al.* 08]	89.1	2.37	94.8	2.93	93.3	4.31
Naive Bayes [Zhang *et al.* 08]	79.3	1.99	71.2	0.36	70.7	2.44
csMOGP [Zhang *et al.* 08]	78.8	1.54	95.8	0.00	99.9	1.66
Our First Approach	95.1098	8.058	96.6606	5.755	98.3752	2.018
Our Second Approach	95.0011	7.883	96.4933	5.682	98.1960	2.017

Table 5: Final cost-sensitive evaluation results and comparisons with the previous approaches

Finally, we show the Receiver Operating Characteristic (ROC) curves of our two approaches in [Fig. 3]. We computed the Area under the ROC curve (AUC) of them; First Approach = 0.985, Second Approach = 0.9853. In general, the AUC illustrates the efficiency of classifiers in regard to false positives. The calculated results show that they are close to the area under the ideal curve (AUC = 1). Therefore, the results

can prove that our approaches are reasonably competitive and practical for spam detection.

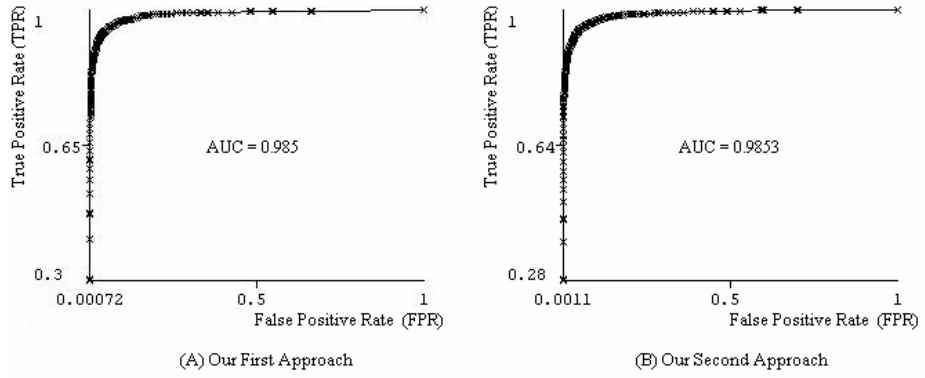

Figure 3: ROC curves of our two approaches

5 Conclusions

In this paper, we presented a new spam detection model using RF. To the best of our knowledge, it is the first time that we perform parameters optimization and optimal feature selection together in spam detection. We evaluated our proposed model using Spambase dataset and compared our approaches with previous approaches. The contributions of this paper are summarized as four-folds: It is capable of (i) optimizing the parameters of RF (ii) identifying important features as a numerical value (iii) determining the optimal number of selected features by using variable importance and two threshold methods (iv) detecting spam with low processing overheads and high detection rates through all of the above.

Acknowledgements

This work was supported by 2009 Korea Aerospace University Faculty Research Grant.

References

[Abu-Nimeh *et al*. 08] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: "Bayesian Additive Regression Trees-Based Spam Detection for Enhanced Email Privacy"; Proc. the 3rd Int. Conf. on Availability, Reliability and Security (ARES 2008), IEEE Computer Society, (2008), 1044–1051.

[Androutsopoulos *et al*. 00a] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., Spyropoulos, C. D.: "An Evaluation of Naïve Bayesian Anti-Spam Filtering"; Proc. the 11th European Conf. on Machine Learning, (2000), 9–17.

[Androutsopoulos *et al*. 00b] Androutsopoulos, I., Paliouras, G., Karkaletsis V., Sakkis, G., Spyropoulos, C. D., Stamatopoulos, P.: "Learning to Filter Spam E-mail: A Comparison of a

Naive Bayesian and a Memory-Based Approach"; Proc. the Workshop on Machine Learning and Textual Information Access, (2000), 1–13.

[Androutsopoulos *et al*. 00c] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Spyropoulos, C. D.: "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages"; Proc. the 23rd ACM Int. Conf. On Research and Development in Information Retrieval, (2000), 160–167.

[Breiman, 01] Breiman, L.: "Random Forests"; *Machine Learning*, 45, (2001), 5–32.

[Bursteinas and Long 00] Bursteinas, B., Long, J. A.: "Transforming Supervised Classifiers for Feature Extraction"; Proc. the 12th IEEE Int. Conf. on Tools with Artificial Intelligence, IEEE Computer Society, (2000), 274–280.

[Carreras and Marquez 01] Carreras, X., Marquez, L.: "Boosting Trees for Anti-Spam Email Filtering"; Proc. the 4th Int. Conf. on Recent Advances in Natural Language Processing, (2001).

[Drucker *et al*. 99] Drucker, H., Wu, D., Vapnik, V.: "Support Vector Machines for Spam Categorization"; *IEEE Transactions on Neural Networks*, 10, 5 (1999), 1048–1054.

[Duda *et al*. 01] Duda, R. O., Hart, P. E., Stork, D. G.: "*Pattern Classification*"; 2nd ed., John Wiley & Sons, Inc., (2001).

[Cranor and LaMacchia 98] Cranor, L. F., LaMacchia, B. A.: "SPAM!"; *Communications of the ACM*, 41, 8 (1998), 74–83.

[Fontana 08] Fontana, P.: "A Combination of Decision Trees and Instance-Based Learning"; Master's Scholarly Paper, Univ. of Maryland, (2008).

[Graham 03] Graham, P.: "Better Bayesian Filtering"; Proc. the 1st Annual Spam Conf., MIT Press, (2003).

[Koprinska *et al*. 07] Koprinska, I., Poon, J., Clark, J., Chan, J.: "Learning to Classify E-mail"; *Information Sciences Including Special Issue on Hybrid Intelligent Systems*, 177, 10 (2007), 2167–2187.

[Lee *et al*. 10] Lee, S., Kim, D., Kim, J., Park, J.: "Spam Detection Using Feature Selection and Parameters Optimization"; Proc. the 4th Int. Workshop on Intelligent, Mobile and Internet Services in Ubiquitous Computing, IEEE Computer Society, (2010), 883–888.

[Liang *et al*. 08] Liang, J., Yang, S., Winstanley, A.: "Invariant Optimal Feature Selection: A Distance Discriminant and Feature Ranking based Solution"; *Pattern Recognition*, 41, (2008), 1429–1439.

[Ravi Kiran and Atmosukarto 05] Ravi Kiran, S. S., Atmosukarto, I.: "Spam or Not Spam – That is the question"; Technical Report, University of Washington, (2005).

[Sahami *et al*. 98] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: "A Bayesian Approach to Filtering Junk E-Mail"; Proc. AAAI Workshop on Learning for Text Categorization, AAAI Techincal Report WS-98-05, (1998).

[Sakkis *et al*. 03] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., Stamatopoulos, P.: "A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists"; *Information Retrieval*, 6, (2003), 49–73.

[Salem and Stolfo 10] Salem, M. B., Stolfo, S. J.: "Detecting Masqueraders: A Comparison of One-Class Bag-of-Words User Behavior Modeling Techniques"; *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 1, 1 (2010), 3–13.

[Thota *et al*. 09] Thota, H., Miriyala, R. N., Akula, S. P., Rao, K. M., Vellanki, C. S., Rao, A. A., Gedela, S.: "Performance Comparative in Classification Algorithms Using Real Datasets"; *Journal of Computer Science and Systems Biology*, 2, 1 (2009), 97–100.

[Xie 07] Xie, Y.: "An Introduction to Support Vector Machine and Implementation in R"; (2007).

[Yang *et al*. 08] Yang, B.-S., Di, X., Han, T.: "Random Forests Classifier for Machine Fault Diagnosis"; *Journal of Mechanical Science and Technology*, 22, (2008), 1716–1725.

[Zhang and Zulkernine 05] Zhang, J., Zulkernine, M.: "Network Intrusion Detection using Random Forests"; Proc. the 3rd Annual Conf. on Privacy, Security and Trust, (2005).

[Zhang *et al*. 04] Zhang, L., Zhu, J., Yao, T.: "An Evaluation of Statistical Spam Filtering Techniques"; ACM Transactions on Asian Language Information Processing, 3, 4 (2004), 243–269.

[Zhang *et al*. 08] Zhang, Y., Li, H.Y., Niranjan, M., Rockett, P.: "Applying Cost-Sensitive Multiobjective Genetic Programming to Feature Extraction for Spam E-mail Filtering", Proc. EuroGP, *Lecture Notes in Computer Science*, 4971, (2008), 325–336.

[Zhao 04] Zhao, C.: "Towards better accuracy for Spam predictions"; Technical Report, University of Toronto, (2004).

[Zhao and Zhu 06] Zhao, W., Zhu, Y.: "Classifying Email Using Variable Precision Rough Set Approach"; Proc. the 1st Int. Conf. on Rough Sets and Knowledge Technology, *Lecture Notes in Artificial Intelligence*, 4062, (2006), 766–771.

[Zhu 08] Zhu, Z.: "An Email Classification Model Based on Rough Set and Support Vector Machine"; Proc. the 5th Int. Conf. on Fuzzy Systems and Knowledge Discovery, 5, (2008), 236–240.

[R Project 09] R Project for Statistical Computing (R version 2.9.1), http://www.r-project.org/, Jun 2009

[Spambase 99] Spambase Dataset, ftp://ftp.ics.uci.edu/pub/machine-learningdatabases/spambase/, Jul 1999.

[WEKA 08] WEKA Software Stable GUI version 3.6.1, http://www.cs.waikato.ac.nz/ml/weka, Dec 2008.