

# A Study of Spam Mail Detection System before Receiving the Message Body

Yukiko Sawaya<sup>1</sup>, Yutaka Miyake<sup>1</sup>

<sup>1</sup> KDDI R&D Laboratories, Inc., 2-1-15 Ohara, Fujimino-shi Saitama 356-8502, Japan  
{yu-sawaya, miyake}@kddilabs.jp

**Abstract.** Because the amount of spam mail has been increasing and it currently occupies 85% of all mail traffic, SMTP servers are frustrated with the task of filtering. To address this problem, it has been proposed that the SMTP server blocks spam mail before receiving the message body at the SMTP session by using host information or behaviors of spam sender clients. Although these proposals block spam mail at a high rate, they may also block legitimate mail, which causes a great deal of trouble to legitimate SMTP clients because they are forced to resend mail. We propose an efficient spam mail-blocking system based on information obtained before receiving the DATA command at the SMTP session. This method suppresses the false positive rate minimally and works as a lightweight primary filter. The preliminary result of our method was that it could detect 95% of spam mail.

**Keywords:** spam mail detection, SMTP protocol, binary decision tree

## 1 Introduction

The amount of spam mail has been increasing, and according to one report [1], spam mail currently occupies 85% of all mail traffic. Therefore, SMTP servers are becoming frustrated with their task of filtering. To address this problem, it is necessary to detect spam mail before receiving message bodies so that the resource of filtering spam mail need not be used. It has been proposed that SMTP servers detect mail from the SMTP clients that send spam mail before receiving the message body at the SMTP session. This method detects spam mail by the behaviors or host information of SMTP clients before receiving the DATA command that intends to send message bodies at the SMTP session. By using these methods, as spam mail can be detected and denied before receiving message bodies, SMTP servers are prevented from wasting resources for processing unnecessary spam mail. To estimate whether the SMTP clients send spam mail or not, the DNS black/blackhole list (DNSBL) is often used. SMTP servers block mail according to whether it is on the list of IP addresses of malicious hosts. Although this method is efficient for well-known spam sender clients, it is not efficient for bot-infected hosts. To make matters worse, as DNSBLs often register a wide range of IP addresses, legitimate IP addresses may be determined as malicious IP addresses in that case. Other methods such as greylisting [2] urge SMTP clients to resend mail and recognize SMTP clients that do not resend mail as spam senders. Although this method efficiently detects spam sender clients

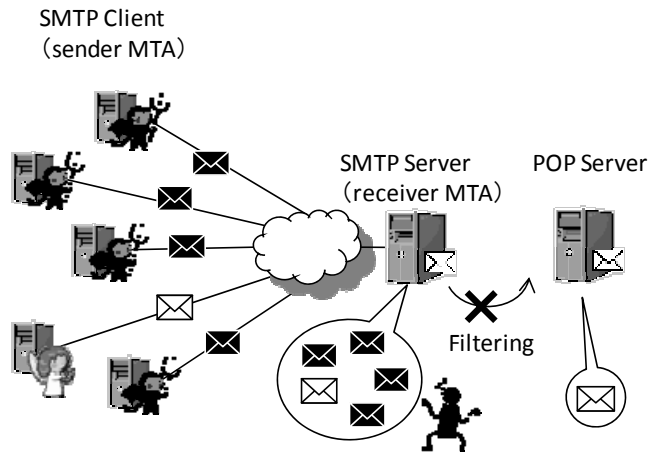
and prevents spam mail being received, many legitimate SMTP clients are forced to resend mail. To avoid this, SMTP servers usually create and manage whitelists of legitimate SMTP clients. This method can block spam mail at a high rate while it also blocks legitimate mail, which causes legitimate SMTP clients to try resending mail and mail delivery delay occurs. S25R (selective SMTP rejection) [3] is a method that determines whether an SMTP client seems to be an end-host that belongs to an Internet service provider, and rejects mail accordingly. This method makes a decision on spam sender clients by using the character string features of the client's hostname, which leads to many false positives. Hence, this method should be used with the greylisting method and manage whitelists.

In this paper, we address the above problems that legitimate SMTP clients are refused to send mail; we propose a method of detecting spam mail with very few false positives before receiving the message body. We focus on information that is efficient to detect spam mail by collaborating one another and propose the method which functions as a lightweight primary filter. In this method, we derive the tendency of spam sender clients by focusing on information such as the HELO/EHLO and IP address of the SMTP client, the DNS query data of the IP address, and the envelope FROM address that are obtained before receiving the message body, and make classifiers based on those tendencies. With this method, we achieved efficient spam mail detection with few false positives before receiving the DATA command that intends to send the message body, which indicates that we can reduce the task of legitimate SMTP clients of resending mail.

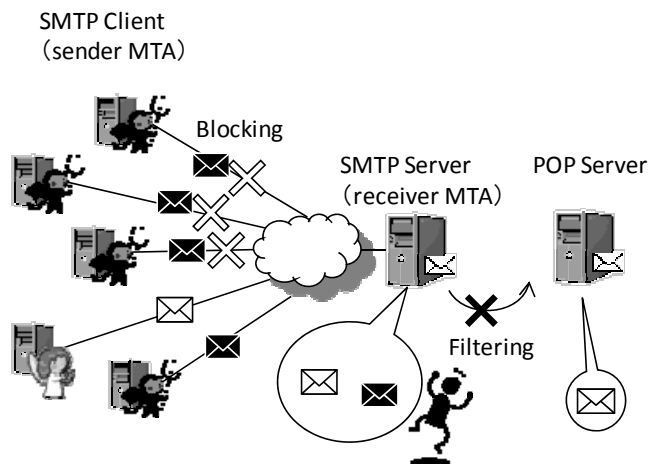
## 2 Related Works

Before we discuss related works, we explain the sequence of detecting spam mail. Figure 1 shows a summary of detecting spam mail. The SMTP client such as the sender MTA accesses the SMTP server and starts the SMTP session. After starting the SMTP session, information such as commands and responses for them and mail message bodies are sent between the SMTP client and the server. The common approach to anti-spam is that SMTP servers detect spam mail after receiving the message body and make a decision according to the host information or contents of the mail message body such as [7]. This method is friendly to legitimate SMTP clients because all mail is accepted by SMTP servers. However, the servers are becoming frustrated with the task of spam mail delivery processing. So recently, spam mail-detecting methods that determine whether the SMTP client is likely to send spam mail or not have been proposed [9] [10] [11] and frequently employed.

The following section shows examples of the spam mail detection method by determining SMTP clients that send spam mail using host information or features based on their behavior.



(1) An SMTP server detects after receiving message body



(1) An SMTP server detects before receiving message body

**Fig. 1 Summary of spam mail detection**

## 2.1 DNSBL

A common approach to detecting spam mail by determining spam sender SMTP clients is the use of DNSBL such as [4]. DNSBLs operate a list of spam senders' IP addresses and SMTP servers block the mail according to this list. Although this method is efficient for well-known spam sender clients, it is not efficient for bot-infected hosts. To make matters worse, as DNSBLs often register a wide range of IP addresses, legitimate IP addresses may be determined to be malicious IP addresses in that case.

## 2.2 Greylisting

There are some methods of detecting SMTP clients that send spam mail by using features based on the behavior of the clients. Greylisting [2] is a method whereby the SMTP server sends reply code that makes the SMTP client resend mail. Well-configured SMTP clients will retry to send the mail later, but currently, most spamming software and mail-based viruses do not attempt retries. SMTP clients that attempt to resend mail are registered in the whitelist and thus the legitimate SMTP client will be able to send mail normally. This method can block spam mail at a high rate while it also blocks legitimate mail, which causes legitimate SMTP clients to try resending mail and mail delivery delay occurs. Moreover, a weakness of this approach is that if spammers implement the retry function, this technique will become less effective.

## 2.3 S25R: Selective SMTP Rejection

S25R [3] is a method whereby SMTP servers determine whether the SMTP client seems to be an end-user of the Internet service provider network.

This method is based on the fact that spammers send spam mail directly from end-users' computers, which are infected by bots, while legitimate mail senders use the MTA that is operated by, for example, ISP.

To make a decision regarding the end-user's computer, they use the character string features of the hostname obtained by querying the DNS reverse lookup of IP addresses. For example, if the hostnames of SMTP clients include "dhcp", "dialup", or "ppp", or have a lot of numbers, which are often seen in end-host names, they are determined to be spam sender clients. It also determines clients that have no hostname of the IP address DNS reverse lookup to be spam senders. As too many false positives occur by using this method, it is usually used with greylisting and manages whitelists or blacklists.

## 3 Proposed Spam Detection Method

In this section, we propose a method whereby the SMTP server determines whether mail is a spam or not before receiving the DATA command from the client. As mentioned in Section 2, although the conventional methods detect spam mail at a high rate and stop message bodies being received, they also detect legitimate mail as spam and refuse to receive it. Thus, our requirement of the proposal is to create a lightweight primary filter where almost all legitimate mail is accepted and a few false negatives, which are actually spam mail but detected as legitimate, are acceptable. This method prevents SMTP servers from wasting resources for processing unnecessary spam mail, and it does not request legitimate SMTP clients to resend

mail. In our approach, we focus on information that is efficient to detect spam mail by collaborating one another and can be retrieved before receiving the message body.

### 3.1 Basic Structure

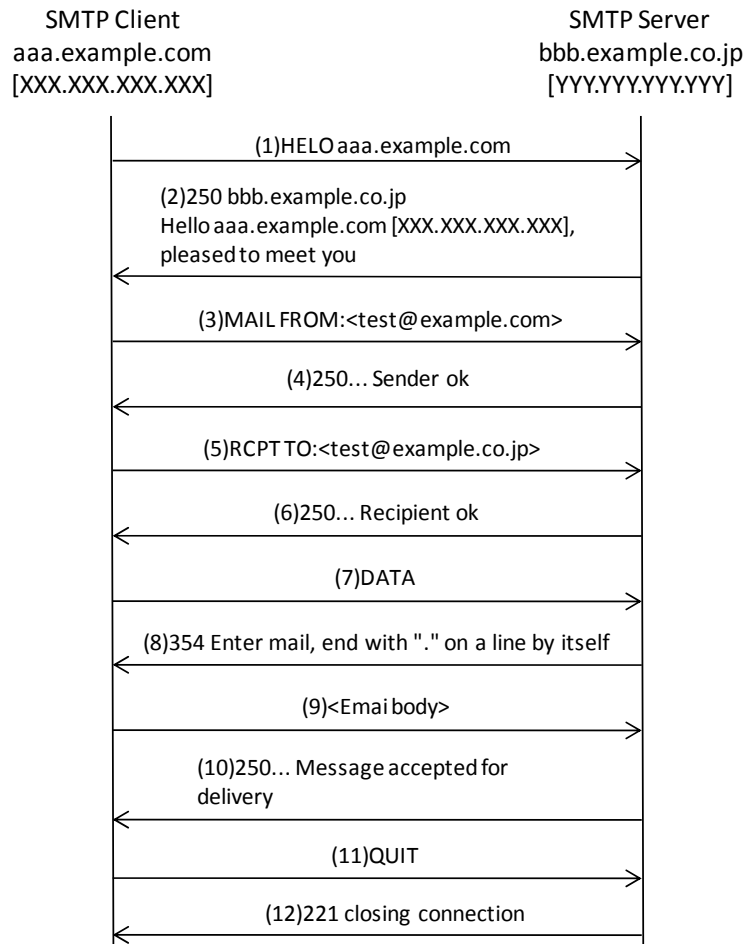
Figure 2 shows the basic sequence of the SMTP connection between the SMTP server and the client based on RFC5321. In the SMTP connection, the HELO/EHLO, MAIL, RCPT, DATA, and QUIT commands are sent from the client to the server. After the DATA command is sent, the client sends the message body. We propose a method that detects SMTP clients that send spam mail by using the information sent from the SMTP client before the DATA command is sent, so that we achieve a spam mail-blocking system before receiving the message body.

Here, we draw up a list of sender information that can be obtained from the sequence as follows:

1. The IP address of the SMTP client: This information is obtained by the TCP/IP connection and it cannot be forged.
2. The HELO/EHLO hostname: This information is included in line (1) and it can be configured at the sender's discretion and is easy to forge.
3. The envelope MAIL FROM address: This information is sent at line (3) and can be easily forged.

From information 1, we can also obtain the following information:

4. The hostname obtained by DNS reverse lookup of IP address: This information is the Answer Section of the IP address DNS reverse lookup, and basically, it is not forgeable except in the case that the authoritative name server is controlled by spammers.
5. The hostname of the authoritative name server: This information is the Authority Section of the IP address DNS reverse lookup.
6. The country of the IP address: This information is obtained by the mapping table offered by [5] and translates the IP address to a geographical location.



**Fig.2 Basic sequence of the SMTP connection between the SMTP server and client**

### 3.2 Classification Method

As mentioned in the previous section, certain information obtained before receiving the DATA command in the SMTP session is forgeable. Spam sender clients often forge it, while legitimate SMTP clients rarely do so. If they forge the information, there should be some contradiction between forgeable information and unforgeable information. For example, if a spam sender forges the HELO/EHLO hostname, it would be different from the hostname obtained by the DNS reverse lookup of the IP address. To disclose whether the SMTP client is lying or not, comparing the domain

parts extracted from one SMTP session is effective. The following domain names are obtained in one session.

$D_n$ : domain part of the authoritative name server that replies to the DNS reverse lookup of the IP address

$D_m$ : domain part of the envelope MAIL FROM address

$D_h$ : domain part of the HELO/EHLO hostname that is sent in the HELO/EHLO command

Here, we define the 2nd and 3rd level domain names that are registered in the WHOIS database (e.g., example.com, example.co.jp) as the domain part. We prepared 3 feature vector elements by comparing the domains above and determined whether they are the same or not.

- $X_1$ : If  $D_n = D_m$ , then 1, otherwise 0
- $X_2$ : If  $D_n = D_h$ , then 1, otherwise 0
- $X_3$ : If  $D_m = D_h$ , then 1, otherwise 0

The S25R method assumes that spammers send spam mail directly from end-users' computers, which are infected by bots, and this method detects SMTP clients that seem to be end-users. In our approach, we determined the end-user's hostname and made feature vector elements. As users' computers often have hostnames including a lot of numbers (e.g., 12.34.56.78.example.com) or they have no hostnames, i.e. DNS reverse lookups of IP address are "unknown", we established the following features vector elements for detection of SMTP clients as end-users.

- $X_4$ : If the hostname obtained by the DNS reverse lookup of the IP address has more than 6 numbers or more than the numbers used in the IP address (the example of former condition is that DNS reverse lookup of IP address 192.168.0.1 has hostname 192168000001.example.com, and the example of the latter condition is that DNS reverse lookup of IP address 10.0.0.1 has hostname 10-0-0-1.example.com) then 1, otherwise 0.
- $X_5$ : If the HELO/EHLO hostname of the SMTP client has more than 6 numbers or more than the numbers used in the IP address, then 1, otherwise 0.
- $X_6$ : If the Answer section of the IP address DNS reverse lookup is unknown, then 1, otherwise 0.

If the condition  $X_6 = 1$ , then inevitably  $X_4$  will be 0.

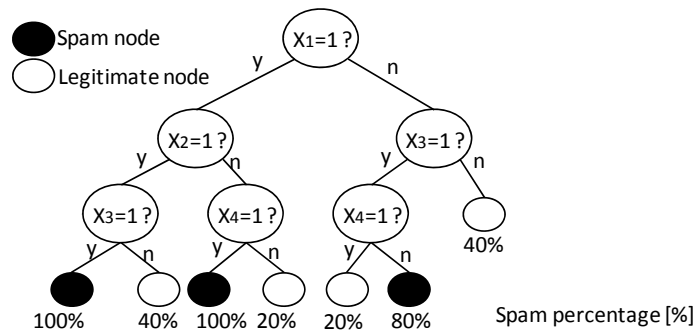
In addition, we found heuristically that the following 3 feature elements appear frequently in spam sender clients.

- $X_7$ : If the client sends HELO/EHLO hostname in IP address format (e.g., not xxx.example.com but 123.45.67.89) and it is not the same as the actual IP address, then 1, otherwise 0.
- $X_8$ : If the country of the SMTP server is not the same as the client, then 1, otherwise 0.

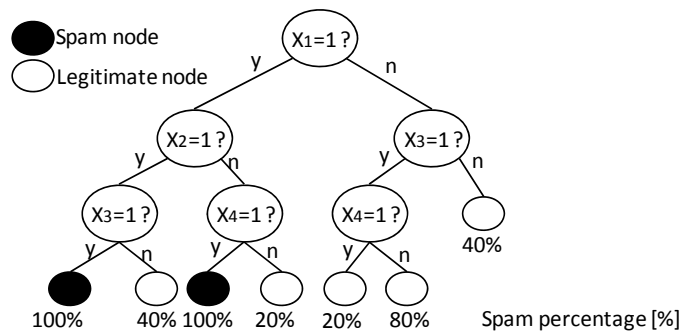
- $X_9$ : If the the HELO/EHLO hostname of the SMTP client is same as the hostname part of the envelope RCPT TO address (e.g., xxx.example.com in abcde@xxx.example.com), then 1, otherwise 0.

By using the above 9 vector elements, we designed feature vector  $X=(X_1, X_2, \dots, X_9)$  and used this vector for making the decision on spam sender clients. We subsequently applied a classification method to decide spam sender clients.

As we need fast classification for it, we used a binary decision tree method, which is a supervised form of learning and a fast learning method. We selected the ID3 [8] method to make a decision tree and obtained the classifier with the training dataset described in Fig.3 (1). In general, as the accuracy of classification of the binary decision tree is inferior to other learning methods, we considered the nodes that include no legitimate mail as the spam mail nodes described in (2) to reduce the number of false positives.



(1) Original binary decision tree



(2) Proposed binary decision tree

**Fig. 3 Binary decision tree classifier**

## 4 Evaluation

In this section, we evaluated the method discussed in the previous section and considered whether this method worked with actual data. We made a dataset composed of the above 9 feature vector elements, i.e., vector  $X=(X_1, X_2, \dots, X_9)$  from spam mail and legitimate mail collected in a certain period and used it for evaluation.

### 4.1 Classification Result

To evaluate the efficiency of the proposed method, we used the S25R [3] as a comparison that can detect high spam sender clients. The dataset used in the experiment was collected from 1 mail account during the period 11/1/2007-11/30/2007. This account received various kinds of mail, such as business (many countries), friends, several kinds of mailing lists, spam mail, etc, and there were 7,221 spam messages and 819 legitimate messages. Here, as we could not collect the envelope MAIL FROM address from mail, we used the "From:" field in the mail header instead.

#### 4.1.1 Evaluation for the number of training data

First, for our evaluation, we determined the number of training data that offer efficient spam classification.

We randomly selected  $N$  mail from the mail collected during 11/1/2007-11/28/2007 for training, i.e. making the classifier, and we used the 493 spam mail and 75 legitimate mail that were collected during the period 11/29/2007-11/30/2007 for evaluation. In general, the ideal training dataset should include the same number of legitimate mail and spam mail. According to this nature, we should take a sample of the spam training dataset to make the number of data the same as that of the legitimate training data. But as we used the extended decision tree mentioned in the previous section to reduce the number of false positives, we did not apply such a scheme.

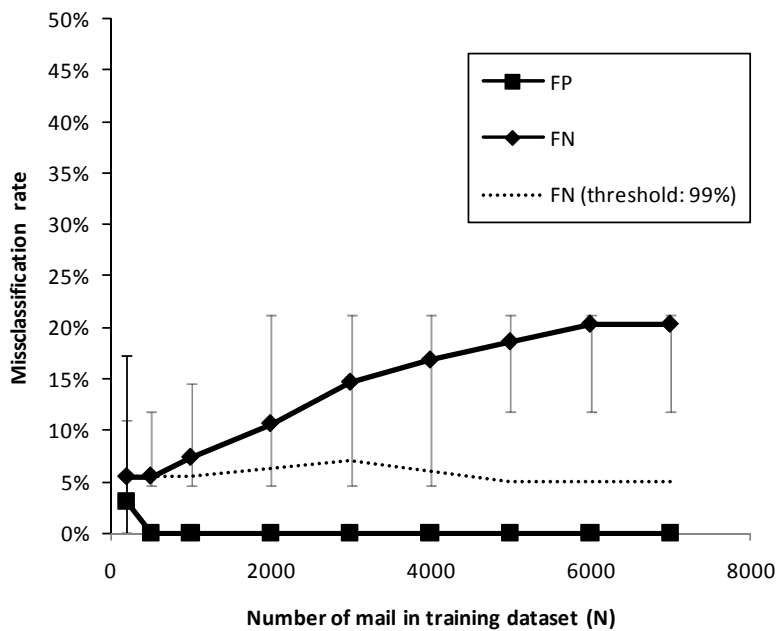
Figure 4 shows the 10-trial average misclassification rate as a function of the number of training data by the binary decision tree. False positive (FP) indicates legitimate mail that is detected as spam mail. False negative (FN) indicates spam mail that is not detected as spam.

The result shows that the false positive rate becomes 0 at the point  $N = 1,000$ , while the false negative rate increases as the number of training data increases and saturates around the point  $N = 6,000$ . In general, both FP and FN decreases and saturates as the number of the training data increases. In this case, although FP decreases and saturates, the FN does not. The reason is that we decided the nodes that consist of 100% spam in the decision tree as the spam mail node and all of the other nodes as legitimate mail nodes. Legitimate mail, which has many similar characteristics to many spam mail, appears as the number of the training dataset increases. As a result, the nodes in the decision tree that include such legitimate mail are decided as legitimate mail and FN increases. If we decide the nodes that consist of 99% spam in

the decision tree as the spam mail node, the FN is described as the dotted line in Fig.5. It saturates at the point  $N = 200$ . Here, we checked the legitimate mail in the training dataset which were contained in the over 99% spam node to make sure what tendency they had. As a result, 1.5 legitimate mail in average were contained when the number of training dataset was 7000. The examples of such mail were that they were advertisement mail or came from mailing list server and its DNS reverse lookup was “unknown”. On the other hand, when the number of mail in training dataset was smaller than 2000, the classification result of the training dataset was same as the condition that the threshold was set to 100%.

Based on these results, we obtained two findings: (1)  $N = 6,000 - 7,000$  is a sufficient number of training data not allowing occurrence of all false positives. (2) If the requirement of the classifier is to reduce the number of false negatives, allowing the possibility of few false positives originated by misconfiguring of SMTP clients, then the number of training dataset can be set to around 1,000.

Incidentally, the FP rate of the training dataset classified by the classifier made by itself is 16% on average under the condition of  $N = 7,000$ , and 5% under the condition of  $N = 1,000$ .



**Fig. 4** Average misclassification rate as a function of the number of training data

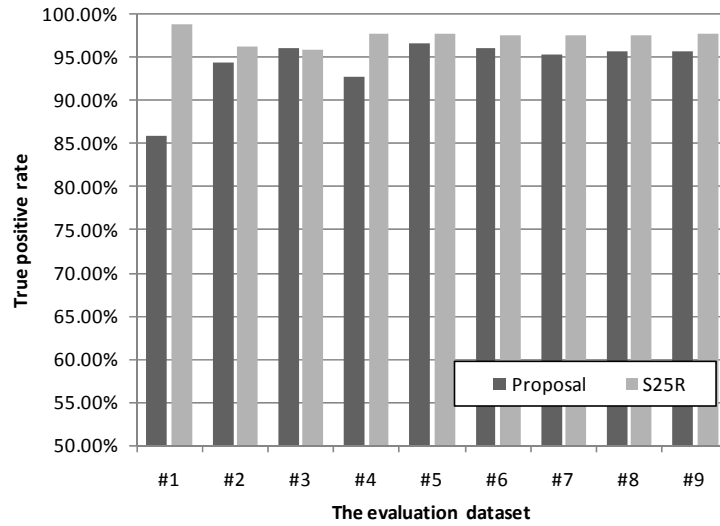
#### 4.1.2 Evaluation for effectiveness by comparing with other method

In case when the proposed method is applied in real environment, i.e., SMTP server, the decision tree classifier should be updated according to time passage. Then we set apart the data that was collected during the period 11/1/2007-11/30/2007 in  $n$  days and set the past  $n$ -day data as the training dataset for making the binary decision tree. We set 3 as  $n$  as shown in Table 1, because 3-day data include 800 mail on average, which is included in one of the ranges mentioned in the findings of Section 4.1.1.

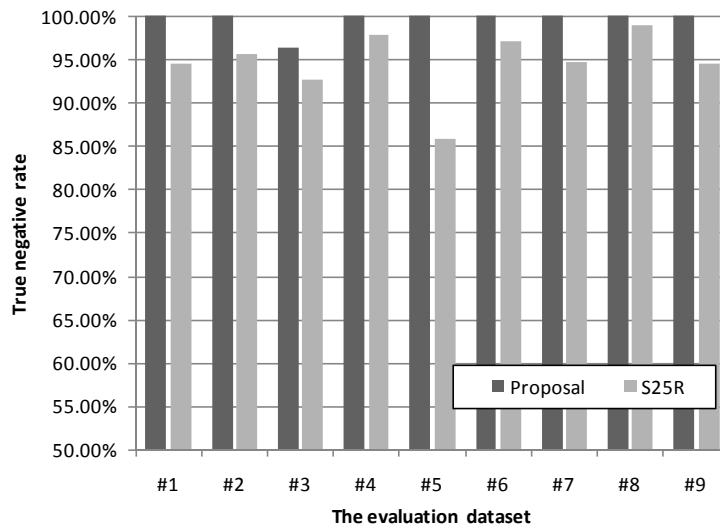
Figures 5 and 6 show the results classified by the binary decision tree and S25R. Here, the true positive rate ( $TP = 100 - FN$  [%]) indicates spam mail that is detected as spam, and the true negative rate ( $TN = 100 - FP$  [%]) indicates legitimate mail that is not detected as spam respectively. This result indicates that the classification method using the decision tree without increase in the false positive rate. Moreover, our method achieved the classification more precisely than the S25R, which causes 5% false negatives and 11% false positives on average.

**Table 1 Mapping table between the training dataset and evaluation dataset**

	<i>Evaluation dataset</i>	<i>Dataset to make a classifier</i>
#1	11/4/2007-11/6/2007 (spam: 701, legitimate: 54)	11/1/2007-11/3/2007 (spam: 755, legitimate: 74)
#2	11/7/2007-11/9/2007 (spam: 691, legitimate: 112)	11/4/2007-11/6/2007 (spam: 701, legitimate: 54)
#3	11/10/2007-11/12/2007 (spam: 683, legitimate: 27)	11/7/2007-11/9/2007 (spam: 691, legitimate: 112)
#4	11/13/2007-11/15/2007 (spam: 684, legitimate: 92)	11/10/2007-11/12/2007 (spam: 683, legitimate: 27)
#5	11/16/2007-11/18/2007 (spam: 759, legitimate: 49)	11/13/2007-11/15/2007 (spam: 684, legitimate: 92)
#6	11/19/2007-11/21/2007 (spam: 778, legitimate: 104)	11/16/2007-11/18/2007 (spam: 759, legitimate: 49)
#7	11/22/2007-11/24/2007 (spam: 717, legitimate: 113)	11/19/2007-11/21/2007 (spam: 778, legitimate: 104)
#8	11/25/2007-11/27/2007 (spam: 740, legitimate: 85)	11/22/2007-11/24/2007 (spam: 717, legitimate: 113)
#9	11/28/2007-11/30/2007 (spam: 713, legitimate: 109)	11/25/2007-11/27/2007 (spam: 740, legitimate: 85)



**Fig. 5 True positive rate of the classification result**



**Fig. 6 True negative rate of the classification result**

## 4.2 Other Evaluation

To evaluate whether this method can be applied to other conditions, we tested this method for the other datasets. We randomly selected 1,000 mails from the dataset

used in Sections 4.1.1 and 4.1.2 for training, which includes 7,221 spam mail data and 819 legitimate mail data for training and made a classifier with them.

Then we prepared 2 datasets to evaluate the accuracy of the classifier. One is a spam mail dataset that includes 100 mail data randomly selected from a spam archive site [6] during the period 2008/1/1-2008/5/30. The other is a legitimate mail dataset that includes 126 mail collected at another mail account belonging to the same network as the training dataset during the period 2007/10/1-2008/8/19.

As a result, 85.5% of spam mail were detected as spam mail by the classifier on ten-trial average varying the training dataset. Although this value is inferior to the results in Section 4.1 and 4.2, it maintains good performance. In addition, 100% of the 126 legitimate mail were detected as legitimate mail. This result indicates that the nature of legitimate SMTP clients and spam sender clients are generalized, and even if the training dataset and evaluation data are collected in different places, the proposed method can be applied anywhere.

## 5 Discussion

In Section 4.1.1, we obtained two values about the number of the training dataset depending on the intended use. However, the ideal condition is that the classifier can detect spam mail at a high rate without any false positives. To realize this condition, we assume the collaborative use of the proposed classifier that reduce the number of false negatives described in finding (2) and greylisting for mail which are classified as spam by the proposed method to give opportunity to be delivered. By collaborating them, most of the legitimate SMTP clients but very few legitimate SMTP clients are requested to resend mail compared with the use of only greylisting. Moreover, the smaller number of training data is sufficient compared with the finding (1). Here, to make sure the influence of collaborating with greylisting, we roughly estimated the false negatives by using data described in [12]. As they say that 95% of spam mail is reduced by using greylisting and our method also reduce 95% of spam mail, we can reduce  $0.95 \times 0.95 = 0.90$  (90%) of all spam mail with 0% false positives in condition that all of legitimate mail is resent. This result shows that we can achieve higher performance than the condition (1) described in 4.1.1, and collaboration with greylisting is considered effective.

In Section 4.1.2, there is 1 false positive mail in dataset #3. This mail has the features that appear frequently in spam mail as follows:  $X_6 = 1$  (the hostname of the IP address is unknown), and  $X_8 = 1$  (the country of the SMTP server and client is not the same). We observed this mail in detail, and we found that this mail was a kind of advertisement mail related to academic activities. Moreover, the IP address of this mail was listed in Spamhaus DNSBL later, which indicates that this mail message was grey-zone mail that seemed to be spam mail according to the person who judges it. In this condition, if we judge this mail as spam, we achieved the classifier with 0 % false positives.

The result of Section 4.2 showed that the proposed method would work even if it is applied in a different environment from that where the training is performed, indicating that we can achieve construction of a highly generalized spam mail-

detecting system. As mentioned above, our method, which derives from the tendency of spam sender clients, focuses on information that conventional methods do not use for detecting spam sender clients and that is retrieved during the period from the start of the SMTP session to when the DATA command is sent. It is indicated that our approach can detect spam mail efficiently with little misdetection of legitimate mail, and can reduce the task of legitimate SMTP clients having to resend mail.

We evaluated the performance effect of applying the proposed method into the real environment. In general, it is said that spam mail occupies 85% of all mail traffic [1] and our method detects spam mail at the rate of 94% which is described in section 4.1.2. Then  $0.85 \times 0.94 = 0.80$  (80 %) of traffic consumed by message body is reduced, while the traffic of legitimate and necessary mail is not obstructed. It is indicated that we can reduce the load to the SMTP server and we can cut down the cost to operate mail servers.

Lastly, as we could not always judge the spam mail which comes from mailing lists as spam because they are sent from the SMTP clients which also send legitimate mail. This problem is not solved even in greylisting or other schemes. The future work is how to separate mail that come from mailing lists server into spam and legitimate ones.

## 6 Conclusion

In this paper, we propose a method whereby the SMTP server determines whether the incoming mail is spam or not before receiving the DATA command from the client and blocks the receipt of spam mail. In our approach, we focus on information that can be retrieved before receiving the message body. We achieved efficient blocking of spam mail with little misdetection of legitimate mail.

As the proposed method works as a primary filter of spam mail, the number of false negatives was increased compared with those of the conventional approach, while we can reduce the task whereby legitimate SMTP clients must resend mail.

In future, we should increase spam mail detection accuracy by using features that are obtained heuristically or with the help of a database. We should also test in the actual environment and evaluate the performance of the system.

## References

1. Symantec Messaging and Web Security: The State of Spam A Monthly Report April 2009, [http://www.symantec.com/connect/sites/default/files/b-state\\_of\\_spam\\_report\\_04-2009.en-us.pdf](http://www.symantec.com/connect/sites/default/files/b-state_of_spam_report_04-2009.en-us.pdf) /
2. Evan, H: The Next Step in the Spam Control War: Greylisting, <http://www.greylisting.org/articles/whitepaper.shtml>
3. Asami, H.: Study Report of an Anti-spam System with a 99% Block Rate --The Selective SMTP Rejection (S25R) System --, <http://www.gabacho-net.jp/en/anti-spam/paper.html>
4. Spamhaus.org, [http://www.spamhaus.org/dnsbl\\_function.html](http://www.spamhaus.org/dnsbl_function.html)
5. IP2Location.com, <http://www.ip2location.com/>

6. Untroubled.org, <http://untroubled.org/spam/>
7. SpamAssassin, <http://spamassassin.apache.org/index.html>
8. Quinlan J.R.: Introduction of Decision Trees, Machine Learning, Vol.1, No.1 pp. 81—106, Springer Netherlands (1986)
9. Xie M., Yin H., Wang H.: An Effective Defense against Email Spam Laundering, 13th ACM Conference on Computer and Communications Security (2006)
10. Gomes, L.H., Cazita, C., Almeida, J.M., Almeida, V., Wagner .M. Jr.: Characterizing a Spam Traffic, Internet Measurement Conference (2004)
11. Twining, R.D., Williamson, M. M., Mowbray, M., Rahmouni, M.: Email Prioritization: reducing delays on legitimate mail caused by junk mail, Proceedings of Usenix (2004)
12. Greylisting performance, <http://users.aber.ac.uk/auj/spam/greyperf.shtml>