Identifying Spam E-mail Based-on Statistical Header Features and Sender Behavior

Conference Paper · September 2012

DOI: 10.1145/2381716.2381863

Mahdi Washha
Institut de Recherche en Informatique de Toulouse

Ismail Khater
Simon Fraser University

Aziz Qaroush
Birzeit University

Identifying Spam E-mail Based-on Statistical Header Features and Sender Behavior

Aziz Qaroush
Department of Computer Systems Engineering
Birzeit University, Birzeit, West Bank, Palestine
aqaroush@birzeit.edu

Ismail M. Khater
Department of Computer Systems Engineering
Birzeit University, Birzeit, West Bank, Palestine
ikhater@birzeit.edu

Mahdi Washaha
Department of Computer Systems Engineering
Birzeit University, Birzeit, West Bank, Palestine
mahdi.washaha@gmail.com

ABSTRACT

Email Spam filtering still a sophisticated and challenging problem as long as spammers continue developing new methods and techniques that are being used in their campaigns to defeat and confuse email spam filtering process. Moreover, utilizing email header information imposing additional challenges in classifying emails because the header information can be easily spoofed by spammers. Also, in recent years, spam has become a major problem at social, economical, political, and organizational levels because it decreases the employee productivity and causes traffic congestions in networks. In this paper, we present a powerful and useful email header features by utilizing the header session messages based on publicly datasets. Then, we apply many machine learning-based classifiers on the extracted header features to show the power of the extracted header features in filtering spam and ham messages by evaluating and comparing classifiers performance. In experiment stage, we apply the following classifiers: Random Forest (RF), C4.5 Decision Tree (J48), Voting Feature Intervals (VFI), Random Tree (RT), REPTree (REPT), Bayesian Network (BN), and Naïve Bayes (NB). The experimental results show that the RF classifier has the best performance with an accuracy, precision, recall, F-measure of 99.27%, 99.40%, 99.50%, and 99.50% when all mentioned features are used included the trust feature.

Categories and Subject Descriptors

K.4.4 [Electronic Commerce]: Security.

General Terms

Algorithms, Performance, Design, Experimentation, Security, Theory, Legal Aspects.

Keywords

Spam, Ham, Spam Filtering, Classification, Machine Learning.

1. INTRODUCTION

Email is one of the most significant, efficient, and popular communication techniques that are used through internet. Unfortunately, the dramatic increase in misusing of emails led to serious problems for both individuals and organizations. Spam is an example of misusing emails, which is also commonly known as unsolicited bulk email, where the spammer sends this type of emails to achieve many goals such as economical goals. Now, Spammers have the ability to launch huge spam campaigns to attack groups or organizations. The cost of sending spam messages is cheap compared with managing spam emails because it wastes the employee productivity and consume the network traffic resources [1]. Modern statistics in Symantec intelligence report [2] stated that the global spam rate was 70.5% at the end of November 2011, and the most spammed industry sector was automotive industry with spam rate of 73.0%, moreover, the large enterprises that contain (1001-1500) employees are the desired target for spammers with a spam rate of 70.1%. Also, according to Nucleus Research Inc [3], spam management costs U.S businesses more than $71 billion annually in lost productivity.

Typically, the email composed of two parts header and body. These parts include many fields which are categorized into two types: *mandatory* and *optional* fields. The mandatory fields are required to send email for recipient(s) such as sender address, and recipient(s) addresses, however the optional fields aren't necessary to compose the email such as subject, and body [4]. RFC 821/822, RFC 2822, and RFC 5321 [5, 6] define the structure of email and constrains that were added on composing email.

Several complicated methods and techniques are used by spammers to generate and send spam emails in order to bypass spam detection. In these techniques, the spammers try to generate thousands of spam emails using different templates to produce emails with different attributes in order to guarantee no clear similarities between spam emails, and to hide any spamming behavior that could be noticed. So, defeating spam filters is the main goal for spammers. For example, spammers employ different sophisticated methods on message header such as spoofing the email address of the sender to evade the detection by spam filters. Also, to obtain high chance to win in spamming war, the body field is considered one of the important fields for spammers to make spam filtering process more complicated by applying methods that are invented especially for body content. Since the appearance of MIME protocol, the processing of body has become more difficult for filters because the body doesn't have plaintext only, but it could contain HTML tags, images, etc.

Anti-spam methods can be categorized into three approaches: *Pre-send* methods act at the sender side to reduce or prevent transporting the e-mail over the network which means the problem should be prevented before it occurs. *Post-send* methods act at the receiver side after consuming the networks resources because the email has been transferred to the receiver side which means in this case, the problem should be addressed after it occurred. The last approach is to define new protocols which are

CUBE 2012, September 3–5, 2012, Pune, Maharashtra, India.
Copyright 2012 ACM 978-1-4503-1185-4/12/09…$10.00.

based on modifying and organizing the transfer process of emails [7]. Also, Post-send methods can be categorized into two types: machine learning and non-machine learning techniques. The non-machine learning techniques uses a set of created or predefined rules to classify the message as spam or ham such as heuristics (rule based), signature, and blacklisting techniques, whereas the machine learning techniques don't need to define rules explicitly, but they need training data or samples to learn the classifier in order to use them in classification process.

Email spam filtering problem could be addressed in many ways. Filtering emails based on the content of the header part is one of many solutions to address spam problem because the header of the email contains information that could be used as features and then use them in the filtering process to classify message as spam or ham. Other approaches are based on the features extracted from the header part and the features extracted from the body part of the message. In this paper, a useful features extracted from email header were proposed based on two publicly datasets. Moreover, several machine learning classifiers are applied on the header features to evaluate the performance of classifying emails.

The rest of this paper is organized as follows: Section 2 reviews the related work for email filter based on header features. Section 3 illustrates the proposed work and feature selection. Section 4 shows the performance analysis for several machine learning classifiers in filtering emails based on header information. Finally, Section 5 concludes the paper.

2. RELATED WORK

Email spam filtering based on header session is efficient and lightweight approach because it utilizes the information in header session to classify email as spam or ham (legitimate). Several methods or algorithms were proposed to handle the spam filtering problem based on the features extracted from the header part. Wu [8] proposed a hybrid method of rule based and back-propagation neural networks (BPNNs) to classify spam mails based on email header information. The rule based was used to digitize and utilize the spamming behaviors which are observed from the headers and syslogs of emails, and then comparing between headers and syslogs fields. The frequently and meaningfully header fields have been selected since they appeared in spam and ham messages which were taken from publicly datasets (10022 spam, 22809 ham). The selected header fields are: 'Received', 'Return-Path', 'From', 'Delivered-To', 'To', and 'Date'. For syslogs fields, the emails were analyzed on mail server because there are no public syslogs dataset available and the syslogs fields are: from, to, nrcpts, and date. Then the enhanced BPNN with a weighted learning was applied as a classifier to filter the messages as spam or ham based on the extracted header and syslogs features. The achieved performance was with accuracy, false positive and false negative of 99.6%, 0.6%, and 0.17%, respectively.

Ye[9] proposed a model based on Support Vector Machine (SVM) to discriminate spam messages depends on mail header features. They utilized a certain fields in header session to extract features. The fields are 'Return-path', 'Received', 'Message-ID', 'From', 'To', 'Date', and 'X-Mailer'. For each mentioned field, there is at least one feature was extracted. For example, the number of recipients has been extracted as a feature from 'To' field. The performance has been evaluated on CCERT data sets which contain Chinese emails where 10000 of emails were used to test the proposed model and different sizes (1000, 2000, 4000, 8000, 16000) of emails were used to train his model to observe the output performance. They achieved an accuracy of 98.40%, 99.30% precision, and 97.50% recall when the training set was 16000 emails.

Hu [10] presented an Intelligent Hybrid Spam-Filtering Framework (IHSFF) to detect spam by analyzing email headers only. Because this framework is efficient and scalable, it is suitable for giant servers (e.g., Hotmail, Yahoo, and Gmail) that deal with millions of emails daily. They extracted five features from email header: originator field, destination field, X-Mailer field, sender server IP address field, and mail subject field. The subject field has been digitalized by using n-gram algorithm to obtain better performance. Five machine learning classifiers were applied on the extracted header features: Random Forests, C4.5 Decision Tree, Naïve Bayes , Bayesian Network , and SVM. They used two data sets in testing and training where the first dataset contains 33,209 labeled emails and the second data set contains 21,725 labeled emails. The experimental results show that the Random Forests was the best classifier with accuracy, precision, recall, and F-measure of 96.7%, 92.99%, 92.99%, 93.3%, respectively.

Wang [11] presented an idea to filter junk mail by utilizing the header session messages. Since the most anti-spam techniques focused on the subject and the content fields to distinguish between spam and ham mails, they extracted features from the most popular header fields. The fields are message-ID, mail user agent, sender and receiver addresses. Content was applied analysis over 10024 Junk e-mails collected by Spam Archive and the output result shows that 92.5% of e-mails have been classified as Junk e-mail by using the selected header fields.

Sheu [12] proposed a method to classify spam emails by analyzing header attributes. Firstly, the e-mails have been classified into several canonizations as follows: sexual, finance and job, marketing and advertising, and total. Secondly, the basic header fields were analyzed, which were the following fields: e-mail title, sender's name, sender's email address, and sending date. Then, decision tree algorithm was used in order to find association rules to use them in classifying spam emails. The proposed method obtained the following excellent performance: 96.5% of accuracy, 96.67% of precision, and 96.3% of recall.

Al-Jarrah [13] identifies potentially useful email header features for email spam filtering. The following fields were used in feature extraction: 'Received', 'To', 'Date', 'Cc', 'Bcc', 'X-Mailer', 'Message-ID', and 'Subject' fields. Many different machine learning-based techniques were used in classification phase. The classifiers are: C4.5 Decision Tree (DT), Support Vector Machine (SVM), Multilayer Perception (MP), Nave Bays (NB), Bayesian Network (BN), and Random Forest (RF). The experimental results show that the Random Forest (RF) classifier obtained the best performance with an average accuracy, precision, recall, F-measure, ROC area of 98.5%, 98.4%, 98.5%, and 98.5%, respectively.

3. PROPOSED WORK AND FEATURE SELECTION

The proposed work is based on studying the information that is available in header part to extract and select useful features to be used in the classification phase. The process starts as shown in Figure 1 by preparing a set of emails form available datasets that will be used as an input for email parser which was implemented according to the specification in RFC. Then, the parsed fields (mandatory and optional) in email header are analyzed to extract useful features in order to build features vector. Finally, the features vector is used to build the feature space for all emails that will be used in classification process.

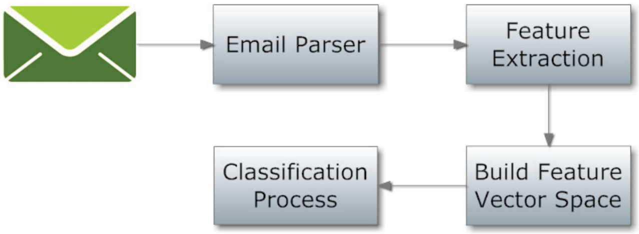


Figure 1. Proposed Work Process Step by Step

3.1 Mail Header Fields

According to RFC 821/822, RFC 2822, and RFC 5321 [5, 6] the header part of the email contains two types of fields: mandatory and optional. Some of these fields are:

From: represents the sender of the email and it is one of the mandatory fields that should be appeared in each email. The missing of 'From' field can be considered as spamming behavior.

To and Cc: it shows the recipient(s) of email; where the email can be sent to one or many recipient. The message should have at least one recipient address in 'To' or 'Cc' fields.

Received: this field contains information about servers that received and sent message during message journey.

Return-Path: It is added by the final transport system that delivers the message to the recipients. Also, it has information about the address and the route back to the message originator.

Date: The date and time at which the message was sent including time zone. It is added once the user submitted the message.

Reply-To: defines the email address that is automatically inserted into the 'To' field when a user replies to an email message.

Error-To: It has the address to which notifications are to be sent and a request to get delivery notifications.

Sender: It is inserted by some systems if the actual sender is different from the text in the 'From' field, where the contained address in sender field represents authenticated user or system.

References and In-Reply-To: They have identifications for other correspondence. These fields hold the message identifier of the original and other messages when creating a reply to a message.

Message-ID (Optional): It is a unique Id that is generated by the system for each message when the message is first created. It can sometimes be useful in fault tracing if multiple copies of a message have been received. In general the domain of 'Message-ID' should be the same domain in 'From' field. Therefore, the mismatching between domains in 'Message-ID' and "From" can be considered as spamming behavior.

3.2 Features Selection and Extraction

The second step for building spam filter is feature extraction. Table 1 provides a summary of the features extracted from the header fields.

Table 1. Selected Email Header Fields with Features Descriptions and Feature Value

NO.	Header Field	Extracted Features	Potential Value
1	From:	From field exists or not	0 or 1
2		Invalid address	0 or 1
3		Partial matching between domains in "From" address and "from" address in first received field.	[0 to 1] or Null
4	To and Cc:	Invalid email address in To field.	0 or 1
5		Exists To field or not.	0 or 1
6		Number of recipients in To field.	0,1,2,…n
7		Invalid email address in Cc field.	0 or 1
8		Number of recipients in Cc field.	0,1,2,…n
9		Exists Cc field or not.	0 or 1
10		Similarity between addresses in Cc field.	[0 to 1]
11		Similarity between To field addresses and Cc field addresses	[0 to 1]
12		Partial matching between "To" field domains and "For" domain in last added received field.	[0 to 1]
13	Received:	Number of relay servers which were used in email transporting from sender to destination address.	0,1,2,…n
14		Invalid IP address.	0 or 1
15	Message-ID:	Domain address is valid or not.	0 or 1
16		Exists field or not.	0 or 1
17		Partial matching between "Message-ID" and "From" domains.	[0 to 1]
18		Partial matching between "Message-ID" and "From" address domains first received field.	[0 to 1] or Null
19		Partial matching between "Message-ID" and "Return-Path" domains.	[0 to 1] or Null
20		Partial matching between "Message-ID" and "Sender" domains.	[0 to 1] or Null
21		Partial matching between "Message-ID" and "ReplyTo" domains.	[0 to 1] or Null

22	Return-Path:	Invalid address.	0 or 1
23		Exists or not.	0 or 1
24		Partial matching between “Return-Path” and “From” domains.	[0 to 1] or Null
25		Partial matching between “Return-Path” and “From” address domains first received field.	[0 to 1]
26		Partial matching between “Return-path” and “RelpyTo” domains.	[0 to 1] or Null
27	Reply-To:	Invalid address.	0 or 1
28		Exists or not.	0 or 1
29		Partial matching between domains addresses in “ReplyTo” and “To” addresses.	[0 to 1] or Null
30		Partial matching between “ReplyTo” field domain and “For” domain in last added received field.	[0 to 1] or Null
31	InReply-To:	Exists or not.	0 or 1
32		Invalid address.	0 or 1
33		Partial matching between “To” and “InReplyTo” domains.	[0 to 1] or Null
34		Partial matching between “InReplyTo” field domain and “For” domain in last added received field.	[0 to 1] or Null
35	Error-To:	Exists or not.	0 or 1
36		Invalid address.	0 or 1
37		Partial matching between “ErrorTo” and “MessageID” server domains.	[0 to 1] or Null
38		Partial matching between “ErrorTo” and “From” domains.	[0 to 1] or Null
39		Partial matching between “ErrorTo” and “Sender” domains.	[0 to 1] or Null
40	Sender:	Exists or not.	0 or 1
41		Invalid address.	0 or 1
42		Partial matching between “Sender” and “From” address domains	[0 to 1] or Null
43		Partial matching between “Senders” and “from” domains in first received field.	[0 to 1] or Null
44	Reference:	Exists or not.	0 or 1
45		Invalid address.	0 or 1
46		Partial matching between domains in Reference field and “ReplyTo” domain.	[0 to 1] or Null
47		Partial matching between domains in Reference field and “InReplyTo” domain.	[0 to 1] or Null
48		Partial matching between domains in Reference field and “To” domains.	[0 to 1] or Null

As shown in Table 1, there are 48 features that could be extracted from the most appearance fields in email header. The selected features aren’t extracted from spam messages only to use them in classifying phase, but also the features are extracted to help in classifying ham messages. For example, feature number 13 represents the number of relay servers that are used in email transporting. The statistics for this feature that were noticed in classification phase shows that when the number of relay servers is more than three, the probability of the message to be ham is high. Moreover, not all of these features have a distinct behavior on spam and ham messages.

Regarding the features values, it is important to mention that the partial matching between domains or addresses have used in some features by using n-gram algorithm. N-gram [14] is a contiguous sequence of *n* items for a given text. It is used to compare between two sequences of items by converting each sequence to a set of n-grams. Since the n-gram is used to calculate matching between two domains, the result of these features could be a decimal value “[0 to 1]” or “NULL”. Decimal value indicates for the probability of partial matching while “NULL” value appears frequently when one of comparing fields is optional which means the optional field didn’t appear in email header. The value “0, 1, 2…n” appears in some features to count the number of recipients or the number of hops where *n* represents any non-negative number. “0” and “1” are nominal values which are used in most selected features in order to set if the feature is occurred or not. Where “0” means “false” and “1” means “true”.

3.3 Behavioral Mail Header Feature

The selected features that were mentioned in the previous subsection didn’t hold any historical information toward the sender of the message, because the selected features focusing on the techniques used by the spammer when launching their campaigns. So, the reputation of the sender can be utilized to know and to build a profile about him. For this end, we introduce a new feature which is called a “Trust” feature by utilizing “From” field. This feature can be in one the following state: “Strongly Ham”, “Weakly Ham”, “Weakly Spam”, and “Strongly Spam”. The value of trust feature depends on whether the sender of the message is new or old.

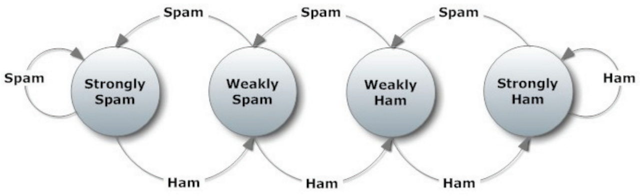


Figure 2. Trust Value Predication

For new sender, the domain for sender is taken from ‘From’ field and then by using web of trust (WOT) service [15], the reputation and the confidence can be known about sender domain. From retrieved information about sender, the value of trust value should be set. However, if there is no information about sender domain, the trust value will be “Weakly Ham” because goodwill about sender should be assumed.

For old sender, the trust value is taken from sender history that is stored in the system. Sender history means when the sender sends one or many messages, the value of trust will change depending on the output of the classifier and current trust value as shown in Figure 2. For instance, if the stored current value of trust is “Weakly Spam” and the output of the classifier is spam, then the new trust value is updated to “Strongly Spam”.

Regarding the trust values or states, it is important to interpret them because each one gives different indication about state of the message. For “Strongly Spam” state, this means that the message sender has very bad confidence and reputation or he sent significant amount of spam messages. In contrast, “Strongly Ham” shows that he has a high rank of confidence and reputation or most of the sent messages were ham. “Weakly Spam” state indicates that the sender confidence and reputation is acceptable or he could be sent spam and ham messages, but with high ratio of spam messages. However, “Weakly Ham” has more than acceptable confidence and reputation or the ham messages ratio is more than spam messages ratio for sender.

In this feature, four states were used instead of two states value in order to reduce classifier mistakes or misclassifications. For example, if the trust feature was “Strongly Spam” and the message classified as ham where the actual type of message is spam, then the trust value will be changed to “Weakly Spam” which means the trust value is affected and still help classifier to classify incoming spam messages and to return for the original trust state or value. The added value for trust feature is shown in experimental results subsection 4.3.

4. PERFORMANCE EVALUATION

In this section, the performance of the extracted features mentioned in section 3 is evaluated by applying several machine learning based classifiers. These classifiers are: Random Forest (RF), C4.5 Decision Tree (J48), Voting Feature Intervals (VFI), Random Tree (RT), REPTree (REPT), Bayesian Network (BN), and Naïve Bayes (NB). All of these classifiers are available in the Weka tool[16] . After that, the classifiers are compared by using the most widely performance metrics used in spam classification analysis.

4.1 Datasets Description and Email Parser

The features extraction phase and the testing phase are based on two publicly datasets:

- CEAS2008 Dataset [17]: CEAS2008 live spam challenge laboratory corpus datasets contains 140000 labeled emails. However, 40000 emails were selected randomly. There are 11410 tagged as ham and 28590 tagged as spam.
- CSDMC2010 Dataset [18]: CSDMC2010 SPAM corpus datasets contains 4327 labeled emails where 2949 emails tagged as ham and 1378 emails tagged as spam.

Regarding the datasets, mixed dataset was produced which contains a total of 44327 emails, 14359 are tagged ham and 29968 are tagged spam. Also, the dataset have been divided into a training and testing sets by using 10- fold cross validation algorithm [16].

The email parser and feature extraction process was implemented using VB.NET framework in order to generate and build the feature vector space as a comma separated values (CSV) files. These files are used as inputs for the Weka tool to classify the given emails as spam or ham.

4.2 Performance Metrics

In spam filtering performance evaluation, the following metrics are used to measure classifier performance: accuracy, recall, precision, F-measure, false positive rate, and false negative rate which are defined by the following equations [13]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Fale Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{Fale Negative Rate} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

$$\text{F - Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- True Positive (TP): The number of spam messages that are classified correctly.
- False Positive (FP): The number of misclassified ham messages.
- False Negative (FN): The number of misclassified spam messages.
- True Negative (TN): The number of ham messages that are classified correctly.

Accuracy is the fraction of all messages (ham and spam) classified by the classifier. Recall represents the performance that spam is successfully discriminated, while precision shows the probability of misclassifying a ham message. Since no relation between recall and precision, F-measure combines them by weighting average of them. False positive rate shows the rate of ham messages tagged by spam in relative for all ham messages, in contrast, false negative rate describe the rate of spam messages tagged by ham in relative for all spam messages. Moreover, it is important to mention that the time complexity is measured in seconds for each proposed classifier.

4.3 Experimental Results

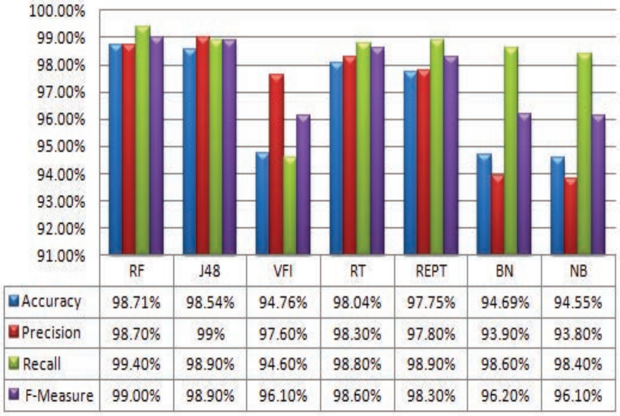
In this subsection, the experimental results are shown for selected classifiers based on the extracted features. The experimental results have been divided into four experiments: before features selection excluded trust feature, before features selection included trust feature, after features selection excluded trust feature, and after features selection included trust feature. The main reason for dividing results is to show the effect of trust feature and time complexity before and after feature selection process. Moreover, the experiments are done under the following environment specifications:

- Processor: Intel Core i7 CPU 860@2.5GHz 2.93Hz.
- Memory (RAM): 8.00 GB.
- Operating System: Windows 7 Ultimate 64-bit.
- 500GB hard disk size.

4.3.1 Results before features selection excluding trust feature

Figure 3 shows the results for several machines learning-based before including trust feature and applying features selection algorithm on the extracted features. It can be noticed that RF and J48 classifiers outperform all other classifiers in terms of accuracy, precision, recall, and F-measure, but RF classifier outperforms J48 in some metrics such as accuracy, and F-measure. However, precision was better in J48 classifier compared with RT classifier. The classifier RT has the best accuracy among other classifiers, J48, RT, REPT, VFI, BN, and NB classifiers are ordered by accuracy after RT classifier.

Table 2 shows the false positive rate, false negative rate, and time complexity measures. J48 classifier outperforms all other classifiers with 2.10% false positive rate, but in time complexity VFI classifier takes 5 seconds to classify the combined datasets. Regarding the false negative rate, RF classifier was the best one with 0.60%



	RF	J48	VFI	RT	REPT	BN	NB
🟦 Accuracy	98.71%	98.54%	94.76%	98.04%	97.75%	94.69%	94.55%
🟥 Precision	98.70%	99%	97.60%	98.30%	97.80%	93.90%	93.80%
🟩 Recall	99.40%	98.90%	94.60%	98.80%	98.90%	98.60%	98.40%
🟪 F-Measure	99.00%	98.90%	96.10%	98.60%	98.30%	96.20%	96.10%

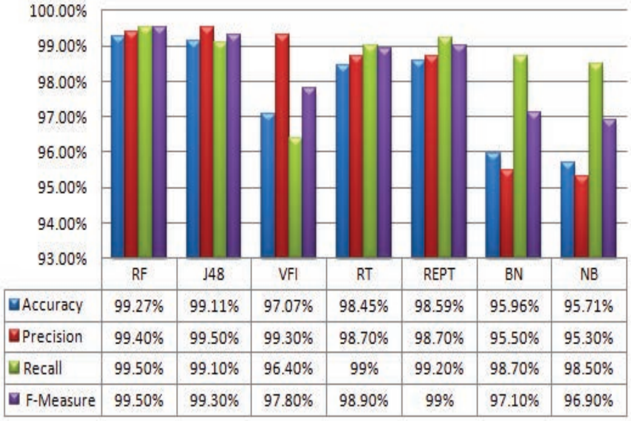
Figure 3. The performance of different machine learning-based techniques before features selection excluding trust feature in terms of accuracy, precision, recall, and F-measure.

Table 2. The performance of different machine learning-based techniques before features selection excluding trust feature in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	REPT	BN	NB
FPR	2.7%	2.1%	4.9%	3.6%	4.7%	13.4%	13.5%
FNR	0.6%	1.1%	5.4%	1.2%	1.1%	1.4%	1.6%
Time (sec)	162	25	5	7	65	300	8

4.3.2 Results before features selection included trust feature

Figure 4 shows the performance for different classifiers in term of accuracy, precision, recall, and F-measure after adding trust feature. It can be noticed that the RF classifier outperforms all other classifiers with an accuracy, precision, recall, and F-measure of 99.27%, 99.40%, 99.5%, and 99.5%, respectively. By comparing the performance results in Figure 3 and Figure 4, the results have been improved by adding trust feature. This means that the pre-knowledge about the sender state leads to improve the performance of the classifier in classifying emails correctly.



	RF	J48	VFI	RT	REPT	BN	NB
🟦 Accuracy	99.27%	99.11%	97.07%	98.45%	98.59%	95.96%	95.71%
🟥 Precision	99.40%	99.50%	99.30%	98.70%	98.70%	95.50%	95.30%
🟩 Recall	99.50%	99.10%	96.40%	99%	99.20%	98.70%	98.50%
🟪 F-Measure	99.50%	99.30%	97.80%	98.90%	99%	97.10%	96.90%

Figure 4. The performance of different machine learning-based techniques before features selection including trust feature in terms of accuracy, precision, recall, and F-measure.

In addition, including the trust feature decreasing the required time for some classifiers; in contrast, a little increase in time for some classifiers has been noted in Table 3. However, the effect of adding trust feature is clear by comparing the results of Table 2 with the results obtained in Table 3. The false positive rate has been decreased from 2.1% to 1% for J48 classifier which outperforms all the other classifiers and also false negative rate decreased from 1.1% to 0.9%. As a result, the trust feature beside the rest features improves the performance for all classifiers.

Table 3. The performance of different machine learning-based techniques before features selection including trust feature in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	REPT	BN	NB
FPR	1.3%	1%	1.4%	2.8%	2.6%	9.8%	10.2%
FNR	0.5%	0.9%	3.6%	1%	0.8%	1.3%	1.5%
Time (sec)	94	27	5	4	38	305	4

4.3.3 Results after features selection excluding trust feature

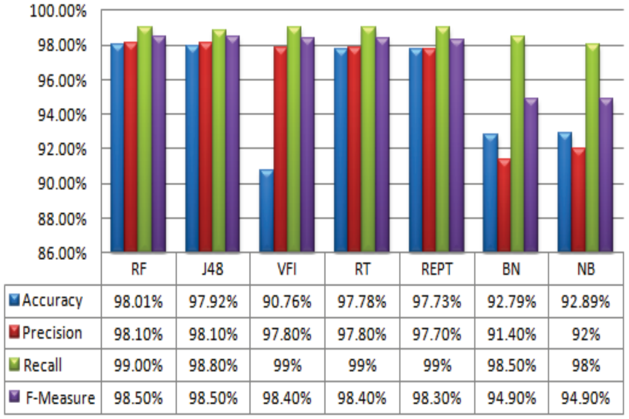
Time complexity analysis is one of the major factors in spam filtering, specially, when a huge number of emails are received at the server side. Improving classification time is required in some situations, but it could decrease the performance of other metrics such as accuracy. So, the mentioned 48 features in section 3 have been minimized by using the genetic features selection algorithm implemented in Weka tool. The new subset of features after the selection process has the numbers: 4, 13, 14, 17, 19, 21, 25, 27, 28, 32, 40, 41, 43, 45, 47, and 48 according to Table 1. The remaining features in the feature space have the most informative and discriminative features.

Figure 5 shows the performance of several machine learning-based techniques in terms of accuracy, precision, recall, and F-measure after minimizing feature vector and without trust feature. The results show that RF classifier outperforms all other classifiers with an accuracy, precision, recall, and F-measure of 98.01%, 98.10%, 99.00%, and 98.50%, respectively. In addition, the performance of the classifiers has been decreased compared

Table 6. The performance of the proposed work compared to other header based spam filters. A: Accuracy, P: Precision, R: Recall, F: F-measure

Spam Filter	Sheu [12], 2009	Ye [9] et al, 2008	Wu [8], 2009	Hu [10] et al, 2010	Wang [11] & Chen, 2007	Al-Jarrah [13] , 2012	Our Approach
Classifier(s)	DT	SVM	Rule-based & back propagation NN	RF,DT,NB, BN,SVM	Statistical Analysis	DT, SVM, MP, NB, BN, RF	RF,J48,VFI, RT,REPT,BN, NB
Best Performance	A=96.5% P=96.67% R=96.3%	A=98.1% P=99.28% R=96.9%	A=99.6%, 0.63% of Ham misclassification	RF (A=96.7%, P=93.5%, R=92.3%, F=93.3%)	92.5% of junk emails are filtered out	RF (A=98.5% , P=98.9%, R=99.2%, F=99%)	RF (A=99.2%, P=99.40%, R=99.50%, F=99.50%)

with the results of Figure 3, due to reducing the number of features in the features space.



	RF	J48	VFI	RT	REPT	BN	NB
Accuracy	98.01%	97.92%	90.76%	97.78%	97.73%	92.79%	92.89%
Precision	98.10%	98.10%	97.80%	97.80%	97.70%	91.40%	92%
Recall	99.00%	98.80%	99%	99%	99%	98.50%	98%
F-Measure	98.50%	98.50%	98.40%	98.40%	98.30%	94.90%	94.90%

Figure 5. The performance of different machine learning-based techniques after features selection excluding trust feature in terms of accuracy, precision, recall, and F-measure.

Although minimizing of features decreased the performance of classifiers, but in the other hand it improved time complexity. Table 4 depicts the performance for the proposed classifiers in terms of false positive rate, false negative rate, and time complexity. By comparing the results with Table 2, the false positive and false negative rates have been increased; in contrast, the time complexity decreased significantly in some classifiers such as RF, and REPT. In RF the time decreased from 162 to 58 and in REPT the time decreased from 65 to 18.

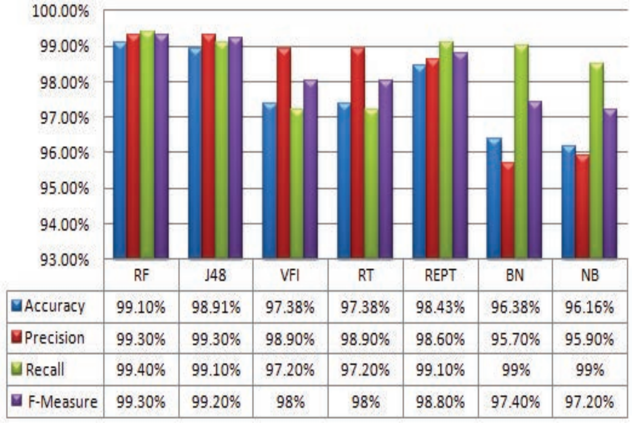
Table 4. The performance of different machine learning-based techniques after features selection excluding trust feature in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	REPT	BN	NB
FPR	4%	4%	10.1%	4.7%	4.9%	19.2%	17.9%
FNR	1%	1.2%	8.8%	1%	1%	1.5%	2%
Time (sec)	58	10	2	3	18	62	2

4.3.4 Results after features selection including trust feature

Figure 6 shows the performance of different machine learning-based techniques in terms of accuracy, precision, recall, and F-measure after features selection including trust feature. It can be seen that RF classifier outperforms all other classifiers with accuracy, precision, recall, and F-measure of 99.10%, 99.30%, 99.40%, and 99.30%, respectively. These results are almost the same with the result shown in Figure 4.

Moreover, Table 5 shows the performance of the classifiers in terms of false positive rate, false negative rate, and time complexity. It can be noticed that adding trust feature improved false positive and false negative rates. For RF classifier the false positive and the false negative rates improved from 4.00% to 1.5% and from 1.00% to 0.60%, respectively.



	RF	J48	VFI	RT	REPT	BN	NB
Accuracy	99.10%	98.91%	97.38%	97.38%	98.43%	96.38%	96.16%
Precision	99.30%	99.30%	98.90%	98.90%	98.60%	95.70%	95.90%
Recall	99.40%	99.10%	97.20%	97.20%	99.10%	99%	99%
F-Measure	99.30%	99.20%	98%	98%	98.80%	97.40%	97.20%

Figure 6. The performance of different machine learning-based techniques after features selection including trust feature in terms of accuracy, precision, recall, and F-measure.

Table 5. The performance of different machine learning-based techniques after features selection includind trust feature in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	REPT	BN	NB
FPR	1.5%	1.5%	2.3%	2.3%	3%	9.2%	8.75%
FNR	0.6%	0.9%	2.8%	2.8%	0.9%	1%	1.48%
Time (sec)	59	9	2	4	12	65	2

4.4 Comparison with Previous Works

In this section, a comparison with other proposed solutions based on extracting features from header of email are done. Table 6 summarizes the results of our work compared with the results of the previous related work. The results shows that our work outperform the other and this is due to the large numbers of features extracted and also the introducing of behavioral mail header feature (trust feature).

5. CONCLUSION

Email spam filtering is still a challenging problem while spammers continue employing and inventing new methods. Moreover, classifying spam email based on header session imposed additional problems and challenges because spammers can easily spoof header information. Also, they can generate different templates for spam messages in order to evade detection by spam filters. In this paper, we presented 48 features which were extracted from header part of email. In addition, we proposed a new feature called trust feature based on the behavior of the sender. Then the extracted features had been evaluated using different machine learning–based classifiers including Random Forest (RF), C4.5 Decision Tree (J48), Voting Feature Intervals (VFI), Random Tree (RT), REPTree (REPT), Bayesian Network (BN), and Naïve Bayes (NB). The feature extraction phase and the testing phase are accomplished using a mixed dataset prepared from two public available datasets. The experimental results shows that RF classifier outperforms all the other classifiers with an accuracy, precision, recall, F-measure of 99.27%, 99.40%, 99.50%, and 99.50% when all mentioned features are utilized including trust feature. Moreover, our results outperform the previous related work results [8, 9, 10, 11, 12, and 13] in terms of accuracy, precision, recall, and F-measure.

6. REFERENCES

1. Christian, K., et al., *Spamcraft: an inside look at spam campaign orchestration*, in *Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. 2009, USENIX Association: Boston, MA.
2. Intelligence, S. *Symantec Intelligence Report: November 2011* 2011 [cited January, 2012]; Available from: http://www.symantec.com/content/en/us/enterprise/other_resources/b-intelligence_report_11-2011.en-us.pdf.
3. *The Real Cost of Spam*. 2007 [cited January, 2012]; Available from: http://www.itsecurity.com/features/real-cost-of-spam-121007/.
4. *Reading and Understanding Email Headers*. [cited March, 2012]; Available from: http://www.by-users.co.uk/faqs/email/headers/.
5. J. K Network Working Group. *Simple Mail Transfer Protocol*. [cited; Available from: http://tools.ietf.org/html/rfc5321.
6. P. R. Network Working Group, E. *Request for Comments RFC 2822,*. [cited March, 2012]; Available from: http://tools.ietf.org/html/rfc2822.html.
7. Gansterer, W.N., et al., *Spam Filtering Based on Latent Semantic Indexing Survey of Text Mining II*. 2008, Springer London. p. 165-183.
8. Chih-Hung, W., *Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks*. Expert Syst. Appl., 2009. **36**(3): p. 4321-4330.
9. Miao, Y., et al. *A Spam Discrimination Based on Mail Header Feature and SVM*. in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*. 2008.
10. Hu, Y., et al., *A scalable intelligent non-content-based spam-filtering framework*. Expert Systems with Applications. **37**(12): p. 8557-8565.
11. Wang, C.-C. and S.-Y. Chen, *Using header session messages to anti-spamming*. Computers &amp; Security, 2007. **26**(5): p. 381-390.
12. J., S., *An Efficient Two-phase Spam Filtering Method Based on E-mails Categorization*. I. J. Network Security. **9**: p. 34-43.
13. Al-Jarrah, O., I. Khater, and B. Al-Duwairi. *Identifying Potentially Useful Email Header Features for Email Spam Filtering*. in *The Sixth International Conference on Digital Society (ICDS), 2012*. Valencia, Spain.
14. *n-gram*. [cited March, 2012]; Available from: http://en.wikipedia.org/wiki/N-gram.
15. *Web of Trust*. [cited March, 2012]; Available from: http://www.mywot.com/.
16. Mark Hall, E.F., Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. *The WEKA Data Mining Software: An Update. SIGKDD Explorations*.
17. corpus, C.L.S.C.L. [cited March, 2012]; Available from: http://plg1.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/fooceas.
18. C. GROUP. (2010, S.e.d., CSDMC2010 and S. corpus). [cited March, 2012]; Available from: http://csmining.org/index.php/spam-email-datasets-.html.