Efficient spam filtering based on informative features extracted from the header fields and the urls in the message

Article *in* Computer Systems Science and Engineering · January 2014

CITATIONS
0

READS
256

3 authors:

Aziz Qaroush
Birzeit University
28 PUBLICATIONS **299** CITATIONS
SEE PROFILE

Mahdi Washha
Institut de Recherche en Informatique de Toulouse
33 PUBLICATIONS **269** CITATIONS
SEE PROFILE

Ismail Khater
Simon Fraser University
23 PUBLICATIONS **336** CITATIONS
SEE PROFILE

Efficient Spam Filtering based on Informative Features Extracted from the Header Fields and the URLs in the Message

Aziz Qaroush, Mahdi Washaha, and Ismail Khater
Department of Computer Systems Engineering
Birzeit University, Birzeit, West Bank, Palestine
{aqaroush, mwashaha, ikhater}@Birzeit.edu

ABSTRACT

The dramatic increase in spam is regarded as one of the major problems afflicting internet email service, as spammers endeavor to defeat spam filters by modifying and developing new techniques to raise the effectiveness of their campaigns for advertising or phishing websites. In this paper we present the results of our analysis of message header fields and the URLs in the message body, and propose informative and discriminative email spam detection features based on recent public email data sets. Furthermore, the Web of Trust (WOT) service was used to measure the reputation of the sender and the URLs included in the message. Subsequently, several machine learning-based classifiers were applied to evaluate the performance of these features, including the reputation feature. The results demonstrate the power of the extracted features, and also establish that the Random Forest (RF) classifier has the best performance of all the classifiers used in terms of accuracy, precision, recall, F-measure, and total cost ratio of 99.69%, 99.70%, 99.90%, 99.8%, and 65 respectively.

Keywords - Spam, Ham, Spam Filtering, Classification, Machine Learning, URLs.

1. INTRODUCTION

Email is one of the most pervasive and efficient communication methods on the internet. Unfortunately, the dramatic increase in misuse of emails led to serious problems for both individuals and organizations. Spam, also known as unsolicited bulk mail, is an example of such misuse. The initiator, or "spammer," sends emails of this type in order to achieve a variety of goals, including profit. There is also another flavor of this activity, known as web spam which is used artificially to inflate the ranking of certain web pages in order to degrade a search engine's results to user queries (1). The main goal of web spamming is to drive traffic to certain pages in order to generate profits, much the same as much of the email spam. In spite of this commonality in their ultimate motivation, the used techniques by practitioners of both types of spam differ. Whereas web spammers focus on techniques to achieve higher rankings for web pages(1), email spammers develop complex methods to craft custom messages to defeat spam filters.

Email spam is a global challenge because spammers have the ability to launch huge campaigns to attack any group or organizations, at a cost that is negligible compared to the cost of managing these messages on the

receiving end. These costs include reduced employee productivity, waste of network resources, and server storage costs (2). Although significant efforts have been made to address this challenge, existing techniques are still found wanting in accuracy and false positive rates. Recent statistics published in the Symantec Intelligence Report (3) indicate that the global spam rate was 64.1% at the end of January 2013, and the most spammed industry sector was non-profits, at a rate of 65.5%. The preferred targets were large enterprises (1501-2500 employees), at a spam rate of 64.4%. And according to Nucleus Research Inc (4), spam management costs U.S businesses more than $71 billion annually in lost productivity.

Typically, an email message is composed of two parts: header and body. These parts include many fields, which may be either mandatory or optional. Sender address and recipient(s) address(es) are mandatory fields; subject and body are examples of optional fields (5). RFC 821/822, RFC 2822, and RFC 5321 (6, 7) define the structure of email messages and applicable constraints.

Several complex methods and techniques are used by spammers to generate and send spam emails that can bypass spam detection filters. In these approaches, the spammers generate thousands of spam emails using different templates to produce emails with different attributes in the header and body fields, such that the lack of obvious structural similarities across these messages effectively cloaks the spamming behavior. The message header portion may contain a large number of fields beyond the common ones such as "To:", "From:", and "Message-ID," but they are largely optional. Generally speaking, the spammers select the most common fields in the header part for manipulation, since they are the focus of most header-based email spam filters. Techniques such as spoofing the sender address, and randomization are applied and use on common header fields to evade detection by header-based spam filters. However, the existence of the MIME protocol in fact makes the message body more significant than the header when it comes to spam. Filtering content-based spam is complicated by the possible appearance of HTML tags, images, and attachments in the message body. This added flexibility enables content-based spamming techniques such as Token Breaking, MIME Attacks, Text Chaff, URL Obscuring, and Character Encoding (8).

Researchers and anti-spammers expend vast resources in developing methods to fight spammer's tricks. These methods can be categorized into three approaches (9): *Pre-send* methods act on the sender side to reduce or stop the flow of spam messages over the network, which means that the problem is prevented before it occurs. *Post-send* methods act on the receiver end, after network resources have been expended on message transfers. The last approach is to define new protocols based on modifying and organizing the transfer process of emails (10). Post-send methods can be further categorized into two types: machine learning and non-machine learning techniques (9). The non-machine learning techniques use a set of created or predefined rules to classify the message as spam or ham. These techniques include heuristics (rule based), signature, and blacklisting techniques. Machine learning techniques, on the other hand, do not require rules to be defined explicitly, but instead require training data or samples to learn the classifiers to be used in the filtering process.

Post-send methods can be developed by analyzing the header and body sections of a spam message, and defining a feature vector space to describe the spam message for classification. Since the feature vector may

contain attributes drawn from either header or body, it is important to mention that header-based filtering is lightweight in comparison with content-based options, which rely heavily on statistical analysis. Headers are strictly defined by RFC 821/822, RFC 2822, and RFC 5321 and do not allow for as much variability (6, 7). However, the header session can be defeated easily by spoofing his fields. On the other hand, filtering using the message body introduces many challenges due to the flexibility in the content of the body. The email body could carry several types of data such as a plaintext, hypertext markup language (HTML), images, and attachments. These types are utilized effectively and efficiently by spammers to improve a variety of tricks to confuse and evade the content based spam filters.

In this paper we identify features, taken from both parts of the message, that are effective in filtering spam. This paper is an extension of another one presented at the Cube 2012 conference, entitled "Identifying Spam E-mail Based-on Statistical Header Features and Sender Behavior" (11) which presented a group of features extracted from the header part without taking the body in the consideration. The current work includes the message body fields by extracting a set of features which reflect the confidence and reputation of the URLs inside the body of the message. It also presents two new filtering performance measures, and a variety of new experiments to evaluate the performance of the URL-based features using three recent public data sets.

The rest of this paper is organized as follows: Section 2 reviews the related work for email header and body spam filters. Section 3 illustrates the proposed work and features selection. Section 4 shows the performance analysis for different experiments using several machine learning classifiers. Finally, Section 5 summarizes the authors' conclusions from this research.

2. RELATED WORK

The exponential growth in email spamming during the recent years has drawn much attention to this problem from IT industry researchers and the media. Although considerable efforts have been dedicated to developing and creating solutions and methods to reduce and eliminate email spam, there is no definitive solution has been invented to uproot the problem. Many methods have been introduced to address spam email, with header-based methods being the most widely used. These techniques focus on studying and analyzing specific information in the header. Wu (12) suggested a solution based on developing a hybrid method of rule-based and back-propagation neural networks (BPNNs) to classify spam messages based on email header information. The rule-based approach was used to capture the spamming behaviors observed in email message headers and syslogs, by comparing header and syslog fields. This work is based on common header fields, which appeared frequently in spam and ham messages taken from publicly available data sets (10022 spam, 22809 ham). The selected header fields are: 'Received', 'Return-Path', 'From', 'Delivered-To', 'To', and 'Date'. Since there are no publicly available data sets for syslogs, the email messages were analyzed on the server, specifically the following fields: 'from', 'to', 'nrcpts', and 'date'. Then, the enhanced BPNN with weighted learning was applied as a classifier to filter the messages as spam or ham based on the extracted header and syslogs features. The performance achieved was 99.6% accuracy, 0.6% false positives, and 0.17% false negatives.

Ye (13) built a model based on the Support Vector Machine (SVM) to discriminate spam messages based on header features. They extracted at least one feature from each of the header session fields 'Return-path', 'Received', 'Message-ID', 'From', 'To', 'Date', and 'X-Mailer'. For example, the number of recipients was extracted as a feature from the 'To' field. The performance has been evaluated on CCERT data sets which contain Chinese emails where 10,000 messages were used to test the proposed model, derived from training sets of size 1,000, 2,000, 4,000, 8,000, and 16,000. They achieved an accuracy of 98.40%, 99.30% precision, and 97.50% recall when the training set was 16,000 messages.

Hu (14) presented an Intelligent Hybrid Spam-Filtering Framework (IHSFF) that detects spam by analyzing email headers only. Because this framework is efficient and scalable, it is suitable for massive servers (such as Hotmail, Yahoo, and Gmail) that deal with millions of emails daily. The researchers extracted five features from the email header: originator field, destination field, X-Mailer field, sender server IP address field, and mail subject field. The subject field was encoded using an 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 data set contained 33,209 labeled emails, and the second data set contained 21,725 labeled emails. The experimental results showed that the Random Forests was the best classifier with accuracy, precision, recall, and F-measure of 96.70%, 92.99%, 92.99%, and 93.30% respectively.

Wang (15) presented the idea of filtering junk mail by utilizing the header session fields. Since most anti-spam techniques focus 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 analysis was applied to over 10,024 Junk e-mails collected by Spam Archive, and the results show that 92.5% of e-mails were classified as Junk e-mail by using the selected header fields.

Sheu (16) presented a method to classify spam emails by analyzing header attributes. First, the e-mails were classified into the following categories: sexual, finance and jobs, marketing and advertising, and total. Then, the basic header fields were analyzed, namely e-mail title, sender's name, sender's email address, and sending date. The final step was to apply a decision tree generating algorithm in order to find association rules for classifying spam emails. The proposed method achieved excellent performance: 96.50% accuracy, 96.67% precision, and 96.30% recall.

Al-Jarrah (17) identified 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'. Many different machine learning-based techniques were used in the classification phase: C4.5 Decision Tree, Support Vector Machine, Multilayer Perception, Nave Bays, Bayesian Network, and Random Forest. The experimental results show that the Random Forest (RF) classifier obtained the best performance. with average accuracy, precision, recall, F-measure, ROC area of 98.5%, 98.4%, 98.5%, and 98.5%, respectively.

Spoofing and falsification of header information motivated the researchers to study and analyze the content of the message in order to develop and improve solutions to fight spammers. Al-Duwairi (18) suggested an

image spam filtering technique called Image Texture Analysis-Based Image Spam Filtering (ITA-ISF) that makes use of low-level image features for image characterization. Ahmed (19) focused on analyzing the plain text of the message and used a simple rule-based word-stemming algorithm that can extract the base or stem the misspelled or modified word.

Albercht (20) introduced an extendable and open spam filter framework called *Spamato.* In Spamato, three different URL-based techniques are implemented in the filtration process, namely Razor Filter, Earl Grey Filter and Dominator Filter. Razor Filter is a collaborative filter that extracts all URLs from a message and then checks if their domains are blacklisted. Each domain is evaluated by Razor networks, and the message is classified as spam if any of the included domains is labeled as spam. This scheme is called a *single-URL* technique because the spam evaluation depends on each single domain. Earl Grey Filter works like the Razor filter, where the probability of the URL being spam is taken from global rating of linked domains. This is called a *multi-URL* technique, which means that all URLs inside the message will be evaluated as a single entity. The result is a message fingerprint which is subsequently used in the Earl Grey network to obtain the spam probability. The Domainator filter removes known ham and fake domains before the Razor and Earl Grey filters are applied. This filter works by querying Google's databases to determine the number of web pages that contain the domain associated with a key word like “spam” or “blacklist.” The experimental results for the Domainator filter show that most spam domains are clustered in an area, but healthy domains are rarely found, meaning that the Google search results are useful for classifying spam and ham emails. A study by Eleni (21) focused on the distributions and on the properties of the URLs, in addition to the characteristics of HTML included in spam emails. These researchers collected emails from two different sources in order to derive the properties and the characteristics of URLs included in spam messages. The size of data sets used was 234,000 spam messages. Experimental results showed that the spam URL addresses are non-repetitive, short-lived, and elusive, all attributes that complicate the detection and filtration process. Reputable URLs were also observed in spam messages, where they were used as decoys to evade detection by spam filters.

In summary, most header-based spam email detection techniques used the common appearance and mandatory fields such as “To,” complemented by some optional fields. Consequently, it can be concluded that these fields are used massively in generating spam emails by spammers, which provides powerful motivation for studying and analyzing header fields in depth. However, using the same fields does not necessarily mean that the extracted features will also be the same, which explains the variance in experimental results. It is not a simple task to determine which methodology is best, because the features are extracted in different ways. The relative performance of those features was evaluated by the authors using several machine learning-based classifiers, and is presented in Section 4.4. A message's body has a different structure, and using it to address spam is, in general, very time consuming compared to using the header. However, selecting very specific elements of the body to drive the filtration process can be a useful strategy to make the filtration process more lightweight. In particular, extracting URLs from the message body is one such approach to detecting spam emails. Studying the characteristics of the URLs (21) can help in identifying spam emails by introducing useful results about spam URLs. Checking URLs using blacklisting domains is other way to identify email

spam; however, there are many URLs that may not be blacklisted and nevertheless lead to decreased performance.

3. PROPOSED WORK AND SELECTION OF FEATURES

The proposed work is based on studying the information that is available in the message header part to extract and select useful features. In addition, the URLs in the message body are extracted to analyze the reputation and the confidence of URLs' domains. The process starts, as shown in Figure 1, by preparing a set of emails form available data-sets 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 each email header as well as the URL addresses, are analyzed so as to convert them into message features that are used to build a features vector that represents the message. The features vector space representing all of the email messages is then built for use in classifying incoming messages as spam or ham.

Figure 1. Proposed Work Process Step by Step

3.1 Mail Header Fields

According to RFC 821/822, RFC 2822, and RFC 5321 (6, 7) the header part of the email contains two types of fields: mandatory and optional. Although most of the fields are optional, most of the mandatory header fields were selected to extract features from them:

From: Represents the sender of the email and it is one of the mandatory fields that should be appear in each email. The absence of this field can be treated as spamming behavior.

To and Cc: It shows the recipient(s) of the email message. The message should have at least one recipient address in the 'To' or in the 'Cc' field.

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

Return-Path: It is added by the final transport system that delivers the message to the recipients. It contains 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 submits the message.

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

Errors-To: It has the address to which error notifications are to be sent, and includes 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 the 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 it 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, a mismatch between domains in 'Message-ID' and 'From' can be treated as spamming behavior.

3.2 Mail Header Features Selection and Extraction

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

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

NO.	Header Field	Extracted Features	Range
1	From:	From field exists or not	0 or 1
2		Invalid address	0 or 1
3		Partial match 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 match 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 match between “Message-ID” and “From” domains.	[0 to 1]
18		Partial match between “Message-ID” and “From” address domains first received field.	[0 to 1] or Null
19		Partial match between “Message-ID” and “Return-Path” domains.	[0 to 1] or Null
20		Partial match between “Message-ID” and “Sender” domains.	[0 to 1] or Null

21		Partial match 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 match between “Return-Path” and “From” domains.	[0 to 1] or Null
25		Partial match between “Return-Path” and “From” address domains first received field.	[0 to 1]
26		Partial match 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 match between domains addresses in “ReplyTo” and “To” addresses.	[0 to 1] or Null
30		Partial match 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 match between “To” and “InReplyTo” domains.	[0 to 1] or Null
34		Partial match 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 match between “ErrorTo” and “MessageID” server domains.	[0 to 1] or Null
38		Partial match between “ErrorTo” and “From” domains.	[0 to 1] or Null
39		Partial match 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 match between “Sender” and “From” address domains	[0 to 1] or Null
43		Partial match between “Sender” 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 match between domains in Reference field and “ReplyTo” domain.	[0 to 1] or Null
47		Partial match between domains in Reference field and “InReplyTo” domain.	[0 to 1] or Null
48		Partial match between domains in Reference field and “To” domains.	[0 to 1] or Null

As shown in Table 1, there are 48 features that can be extracted from the most common fields in email headers. The selected features are not extracted from spam messages just to use them for spam detection, but also for classifying ham messages. For example, feature number 13 represents the number of relay servers that were used in the transport of the email message. The statistics for this feature that emerged in the classification phase show that when the number of relay servers is greater than three, the probability of the message being ham is high. Further, not all of these features discriminate between spam and ham messages.

Regarding the features values, it is important to mention that the partial match between domains or addresses used in some features is implemented by using an n-gram algorithm. N-gram (22) 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 values indicate the probability of a partial match, while "NULL" value appears frequently when one of the compared fields is optional, indicating that the field did not appear in the email header. The range "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 has occurred or not - where "0" means "false" and "1" means "true".

3.3 Behavioral Mail Header Feature

The selected features mentioned in the previous subsection do not hold any historical information about the sender of the message. Nevertheless, it is possible to use the sender's reputation as a discriminant. To this end, we introduce a new feature called a "Trust," computed using the value of the "From" field. This feature can be in one the following states: "Strongly Ham", "Weakly Ham", "Weakly Spam", and "Strongly Spam". The value of the trust feature depends on whether the sender of the message is new or old.

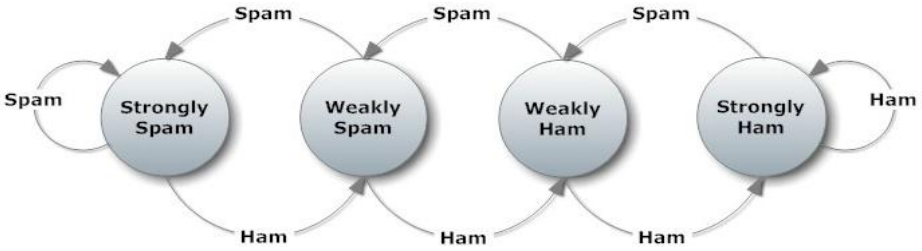


Figure 2. Trust Value Predication

For a new sender, the domain is taken from 'From' field and used as input to the Web of Trust (WOT) service (23), which provides the reputation and confidence attributed to the sender domain. If WOT does not have any information about the sender domain, the trust value will be "Weakly Ham," based on an assumption of good will.

For a known sender, the trust value is taken from the sender history that is stored in the system, which is the sender's trust rating until the current message was received. The new trust value will be computed based on

the previous value, and the output of the classifier on the new message. For example, 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”.

It is worthwhile to go into some detail about the semantics of the various possible values of the trust ratings, or states. “Strongly Spam” means that the message sender has a very bad reputation and inspires low confidence, or that has sent a significant amount of spam messages. Conversely, “Strongly Ham” indicates that the sender has a very good reputation and confidence, or sent mostly ham messages. “Weakly Spam” indicates that the sender has acceptable confidence and reputation, or that he has a history of sending both spam and ham messages, but predominantly spam. However, “Weakly Ham” has more than acceptable confidence and reputation, or the ham messages ratio is more than spam messages ratio for sender.

Four states were used instead of two in order to reduce classifier errors. For example, if the trust feature value was “Strongly Spam” and the message classified as ham, when in fact it is a spam message, the trust value would be changed to “Weakly Spam.” This preserves a relatively low trust value for the sender, which can assist the classifier with future spam messages from that sender, and possibly lead to a future return of the trust value to “Strongly Spam.” The added value of the trust feature is shown by the experimental results in Subsection 4.3.

3.4 URL Addresses Features

Sometimes classifying messages using only header information can lead to a high rate of misclassification, due to the flexibility available to spammers in manipulating that information. Analyzing message content is more complex than analyzing message header information due to the multiplicity of data types that may appear in the body, such as plain text, hypertext markup language (HTML), images, and attachments which in most cases require semantic analysis. On the other hand, URL addresses constitute one of the content elements used most widely by spammers in their campaigns due to the negligible costs of sending and advertising. Spammers use URLs in the content of the message to avoid detection by spam filters and to hide their identities. Furthermore, recent research shows that 81.5% of spam messages contain at least one URL in their body (21). This observation was confirmed by our experiments based on three publicly available data sets, in which 78% of all emails contain at least one URL in the body. For these reasons, the authors selected URL addresses embedded in the email body as additional features to be used in conjunction with those extracted from email headers.

The URL features extraction process passed through three stages: parsing URL addresses, retrieving URL domain information, and building the features vector. The parsing of URL addresses is accomplished by using a predefined regular expression to extract all URLs inside the message, including HTML tags such as URL, SRC, ACTION, and HREF, because HTML is used widely in spam messages to improve the representation of the message content with fonts, colors, tables. Once the URLs are extracted, the domain for each URL is stored to be used in the next processing stage. For example, suppose the following URL

"http://en.wikipedia.org/wiki/Total_expense_ratio" is extracted from the email body, then its domain is "en.wikipedia.org".

The second stage is to retrieve information about the URL's domain using a free web service called Web of Trust (WOT) (24). The WOT rates the domains based on feedback from users or clients around the world, by installing a plug-in on their browsers. The domains are rated using two different aspects: the domain's confidence and its *reputation*. Each aspect is evaluated by four different criteria's, which are: Trustworthiness, Vendor reliability, Privacy, and Child Safety. Table 2 describes the criteria used in evaluating the reputation and the confidence of domains.

Table 2: Description of WOT Criteria (24)

Criteria	Description
Trustworthiness	Do you trust this website? Is it safe to use? Does it deliver what it promises?
Vendor reliability	Is the site safe for buying and selling or for business transactions in general?
Privacy	Can you trust the site owner, safely supply personal information, and download files?
Child safety	Does the site contain age-inappropriate material?

For each of the criteria, the WOT service provides a numeric value between 0 and 100. The service also provides a mapping table to assign semantics to various sub-ranges, for easier understanding by humans. For example, if the trustworthiness criterion for domain reputation is 90, the WOT evaluates it as an excellent reputation in the trustworthiness aspect. Table 3 shows the mapping between numeric values and their semantic equivalents.

Table 3 Reputation and Confidence Numeric Values Mapping Table (24)

Reputation	Confidence	Description
≥ 80	≥ 45	Excellent
≥ 60	≥ 34	Good
≥ 40	≥ 23	Unsatisfactory
≥ 20	≥ 12	Poor
≥ 0	≥ 0	Very poor

In our work, for each extracted and processed URL, the reputation and confidence are retrieved with all mentioned criteria's values using WOT's request API (24). Once the numeric values for each aspect or criteria are retrieved, they are mapped to their meaning or description based on table 3. However, it is important to mention that the returned values for some or all criteria maybe ”Null”, which means that there is no available information about the requested domain.

Once URLs domains had been evaluated and mapped, the next stage is extracting features to build the feature vector space. Table 4 shows the list of all features with their names, descriptions, and the potential values for each one. The content of the message may include more than one URL; in that case, each URL is evaluated separately. Each URL is ranked numerically according to two aspects: *Reputation* and *Confidence*. For the reputation aspect, the criteria that have excellent or good values are scored as a *Positive Rank*. Conversely, criteria that have unsatisfactory, poor or very poor values are scored as a *Negative Rank*. It is important to

mention that the NULL value is not scored either way. The final rank of the URL is computed by subtracting the positive scores from negative scores and then dividing the result by four, which is the number of criteria used, in order to normalize the number between [-1,1]. The same process is followed for the confidence aspect, using the corresponding criteria values.

Table 4 List of the Extracted Features from URLs with their Descriptions and Potential Values

NO.	Feature Name	Feature Description	Potential value
1	Reputation of Child Safety	Evaluate the domain reputation by checking if the contained material is appropriate to child ages.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
2	Reputation of Privacy	Measure the reputation of trustworthiness of the website owner and supplying personal information.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
3	Reputation of Trustworthiness	Reflects the reputation of the site in terms of safety use.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
4	Reputation of Vendor Reliability	Evaluate the reputation of the website in terms of safety of business transactions in general .	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
5	Confidence of Child Safety	Evaluate the domain confidence by checking if the contained material is appropriate to child ages.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
6	Confidence of Privacy	Measure the confidence of the website owner and supplying personal information.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
7	Confidence of Trustworthiness	Reflects the confidence of the site in terms of safety use.	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
8	Confidence of Vendor Reliability	Evaluate the confidence of the website in terms of safety of business transactions in general .	Excellent , Good , Unsatisfactory, Poor, Very poor, NULL
9	Worst URL	Score of the worst URL	[-1 to 1]
10	Confidence of All URLs	Average confidence score for all URLs	[-1 to 1]
11	Reputation of All URLs	Average reputation score for all URLs	[-1 to 1]

As shown in Table 4, there are 11 features extracted from the domain information embedded in the message body. The first eight features reflect criteria values for the ranked URL. The main reason from adding the ninth feature to the URLs features space, is to account for the spammer's tricks of misdirection to hide their bad intentions among several reputable URLs. The Worst URL feature has the worst score of the first eight

features. The potential score value should be a decimal value between -1 and 1, where a value around -1 indicates the reputation and the confidence are very bad, and a value around 1 indicates the reputation and confidence are very good. To evaluate all URLs as a single entity to reflect the overall value, The average confidence and reputation across all of the URLs in the message body are also computed, in order to provide a value that can be used to classify the message as a whole.

4. PERFORMANCE EVALUATION

In this section, the performance of the extracted features mentioned in section 3 is evaluated by applying the machine learning classifiers most commonly used in spam filtering (25). These classifiers include: Random Forest (RF), C4.5 Decision Tree (J48), Voting Feature Intervals (VFI), Random Tree (RT), Bayesian Network (BN), and Naïve Bayes (NB). All of these classifiers are available in the Weka tool (26). After that, the classifiers are compared based on the performance metrics used widely in spam classification analysis.

4.1 Description of the Data Sets and Email Parser

The features extraction phase and the testing phase are based on three publicly available data sets:

- CEAS2008 Data set (27): CEAS2008 live spam challenge laboratory corpus data sets contains 140,000 labeled emails. However, 40,000 emails were selected randomly. There are 11,410 tagged as ham and 28,590 tagged as spam.
- CSDMC2010 Data set (28): CSDMC2010 SPAM corpus data sets contains 4327 labeled emails where 2949 emails tagged as ham and 1378 emails tagged as spam.
- Spam Archive (1-2012, 2-2012, 3-2012, and 4-2012) (29): contains up to 141,456 emails where tagged as spam. These data sets were shuffled and combined to produce a combined data set with a total of 242,438 emails messages. Then, 100,000 emails were selected randomly for our testing and evaluation, of which 88,865 had been tagged as spam, and 11,135 as ham. All of these emails contain at least one URL in their body. This new data set was divided into training and testing sets by using a 10-fold cross validation algorithm (26).

The email parser and feature extraction process was implemented using VB.NET framework in order to generate and build the feature vector space as comma separated values (CSV) files. These files were 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, as defined by the following equations (17):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{----------} \quad 1$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{----------} \quad 2$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{----------} \quad 3$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \quad \text{----------} \quad 4$$

$$\text{False Negative Rate} = \frac{FN}{FN + TP} \quad \text{----------} \quad 5$$

$$\text{F - Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{----------} \quad 6$$

Where:

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

False positive rate shows the rate of ham messages tagged by spam in relative to all ham messages. Conversely, the false negative rate describes the rate of spam messages tagged by ham, relative to all spam messages. The misclassification of ham emails as spam is more costly than misclassification of spam messages as ham. A weighted accuracy metric (Waccuracy) was used to capture this cost (30). Waccuracy is defined by (30) as:

$$Waccuracy = \frac{TP + \lambda * TN}{Ms + \lambda * Mh} \quad \text{----------} \quad 7$$

Where:

- Ms: indicates the number of spam messages
- Mh: indicates the number of ham messages
- λ: indicates the cost of the false positive compared to false negative cost. More λ indicates more penalties on false positive in relative to false negative.

In (30) three values for λ were introduced :1, 9, and 999. For example, when λ equal 9 it indicates that the cost of the false positive is nine times the cost of the false negative. When choosing high value to λ (e.g. λ = 999), this leads to misinterpretation using Waccuracy. In (30) they avoided this problem by comparing weighted accuracy and weighted error to the baseline in which ham messages are never blocked and spam messages pass the filter. The weighted accuracy and the weighed error using baseline are defined by (30) as:

$$Waccuracy\_b = \frac{\lambda * Mh}{\lambda * Mh + Ms} \quad \text{----------} \quad 8$$

$$Werror\_b = \frac{Ms}{\lambda * Mh + Ms} \quad \text{----------} \quad 9$$

In addition, we used the Total Cost Ratio measure (30) to facilitate the comparison using the baseline and to reflect the effect of the spam filter as a single measure.

$$TCR = \frac{Werror\_b}{Werror} = \frac{Werror\_b}{1 - Waccuracy} = \frac{Ms}{\lambda * FP + FN} \quad \text{----------} \quad 10$$

If TCR is equal 1 then using the baseline is better. Therefore, in order to have better performance, the TCR must be greater than 1. 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: using header features only (including trust feature), using URLs features only, using header and body features including trust features, and finally after features selection for header and URLs features including trust features. The main reason for doing these experiments is to show the effects of header and URLs features separately, and in conjunction, in addition to the time complexity before and after feature selection process. The experiments were conducted 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 Using Header Features Only (Including Trust Feature)

Figure 3 shows the results for several machine learning-based algorithms using header features only including trust feature. It can be seen that the J48 classifiers outperformed all other classifiers in terms of accuracy (when $\lambda = 9$) and precision with 99.21% and 100% respectively, but the RF classifier outperforms all others in terms of accuracy (when $\lambda = 1$), recall and F-measure with 99.69%, 99.90%, and 99.80% respectively. Also, figure 4 shows that J48 outperforms all other classifiers in terms of total cost ratio, with a value of 60.

Table 5 shows the false positive rate, false negative rate and time complexity measures. The J48 classifier outperformed all other classifiers in terms of false positive rate with 1.00%, but in time complexity the VFI classifier is ahead, taking 5 seconds to classify the combined data set. The RF classifier scored best on false negatives, with a rate of 0.90%.

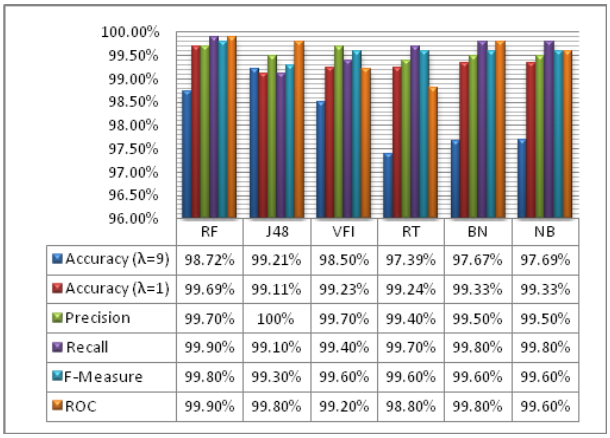


Figure 3. The performance of different machine learning-based techniques using header features including trust feature in terms of accuracy, precision, recall, and F-measure.

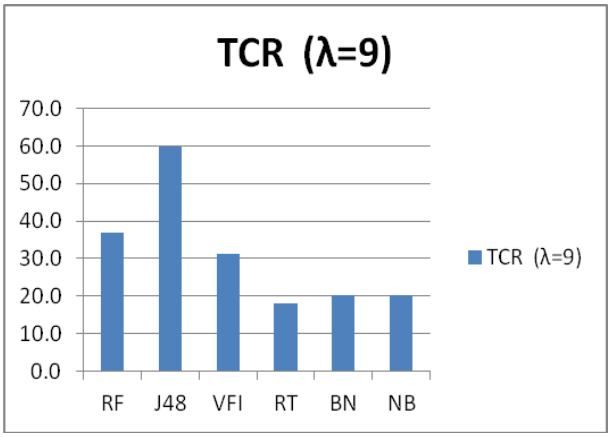


Figure 4. TCR Measure over all classifier using header features only

Table 5. The performance of different machine learning-based techniques using header features including trust feature in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in second.

	RF	J48	VFI	RT	BN	NB
FPR	2.40%	1.00%	2.30%	4.70%	4.20%	4.10%
FNR	0.10%	0.90%	0.60%	0.30%	0.20%	0.20%
Time	50	63	5	13	15	8

Table 6. The performance of different machine learning-based techniques using URLs features in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	BN	NB
FPR	11.00%	11.10%	5.70%	11.80%	6.20%	6.90%
FNR	1.60%	1.80%	9.60%	1.60%	9.00%	8.90%
Time	44	21	3	6	6	2

4.3.2 Results Using URLs Features Only

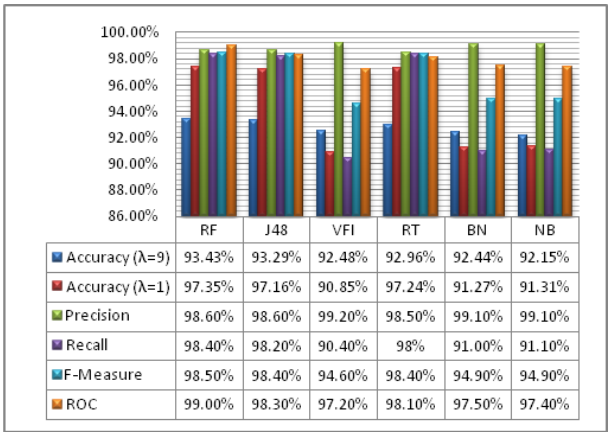
Figure 5 shows the performance of the various classifiers in term of accuracy, precision, recall, and F-measure using only body features. It can be seen that the RF classifier outperformed all other classifiers in terms of accuracy, recall and F-Measures with 93.43%, 98.40%, and 98.50% respectively. But in terms of precision the VFI is best at 99.20%. Figure 6 shows that RF outperformed all classifiers in terms of total cost ratio with 7.2.

Furthermore, Table 3 shows that VFI outperformed all other classifier in terms of false positive rate and time complexity, with 5.70%, and 3. But RF outperformed all other algorithms in terms of false negative rate with 1.60%. Comparing these results to those in Figures 3 and 4 and in Tables 5 and 6, we can see that using header features outperformed the URLs features in terms of accuracy (when λ=9), precision, recall, F-Measure, false positive rate, false negative rate, and total cost ratio. On the other hand, the results indicate that adding URLs to the feature vector space is also a good way to classify messages as spam or ham.

4.3.3 Results Using Header Features Combined with URLs Features

Figures 7 and 8 show the results for several machine learning-based algorithms using header features combined with body features. The results show that RF classifiers outperform all other classifiers in terms of accuracy, F-measure, and total cost ratio with 99.80%, 99.90%, and 65 respectively. On the other hand, VFI

classifier outperforms all other classifiers in terms of precision with 99.90%. However, the RT classifier is the best in terms of recall, with 100%.



	RF	J48	VFI	RT	BN	NB
Accuracy (λ=9)	93.43%	93.29%	92.48%	92.96%	92.44%	92.15%
Accuracy (λ=1)	97.35%	97.16%	90.85%	97.24%	91.27%	91.31%
Precision	98.60%	98.60%	99.20%	98.50%	99.10%	99.10%
Recall	98.40%	98.20%	90.40%	98%	91.00%	91.10%
F-Measure	98.50%	98.40%	94.60%	98.40%	94.90%	94.90%
ROC	99.00%	98.30%	97.20%	98.10%	97.50%	97.40%

Figure 5. The performance of different machine learning-based techniques using URLs features only, in terms of accuracy, precision, recall, and F-measure.

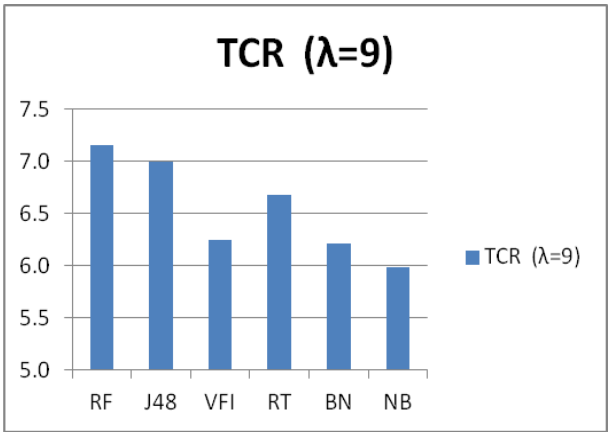
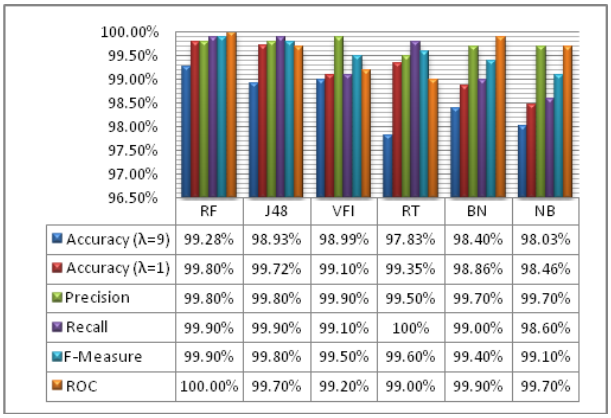


Figure 6. TCR Measure over all classifier using URLs features only.



	RF	J48	VFI	RT	BN	NB
Accuracy (λ=9)	99.28%	98.93%	98.99%	97.83%	98.40%	98.03%
Accuracy (λ=1)	99.80%	99.72%	99.10%	99.35%	98.86%	98.46%
Precision	99.80%	99.80%	99.90%	99.50%	99.70%	99.70%
Recall	99.90%	99.90%	99.10%	100%	99.00%	98.60%
F-Measure	99.90%	99.80%	99.50%	99.60%	99.40%	99.10%
ROC	100.00%	99.70%	99.20%	99.00%	99.90%	99.70%

Figure 7. The performance of different machine learning-based techniques using header and URLs features in terms of accuracy, precision, recall, and F-measure.

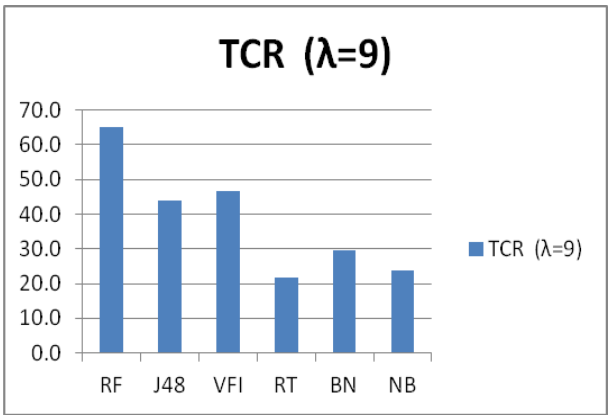


Figure 8. TCR Measure over all classifier using both the header features and the URLs features.

Table 7 shows that the VFI classifier outperforms all other classifiers in terms of false positive rate with 1.10%. However, the RF classifier outperforms all other classifiers in terms of false negative rates with 0.10%. By comparing these results to those in figures 3 and 4, and in tables 5 and 6 we can conclude that the performance of classifiers was improved – with the exception of J48 - and that RF was the best classifier was RF in terms of accuracy (when λ=9), precision, recall, F-Measure, false negative rate, and total cost ratio with 99.28%, 99.80%, 99.80%, 99.90%, 99.90%, 0.10% and 65 respectively.

Table 7. The performance of different machine learning-based techniques using header features combined with URLs features in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

Table 8. The performance of different machine learning-based techniques after features selection process in terms of false positive rate (FPR), false negative rate (FNR), and time complexity in seconds.

	RF	J48	VFI	RT	BN	NB
FPR	1.30%	1.90%	1.10%	3.90%	2.10%	2.50%
FNR	0.10%	0.10%	0.90%	0.20%	0.10%	1.40%
Time	49	62	7	7	18	4

	RF	J48	VFI	RT	BN	NB
FPR	3.00%	3.60%	2.3%	1.70%	6.2%	3.10%
FNR	0.1%	0.10%	0.90%	0.10%	9.00%	0.20%
Time	22	8	3	4	5	2

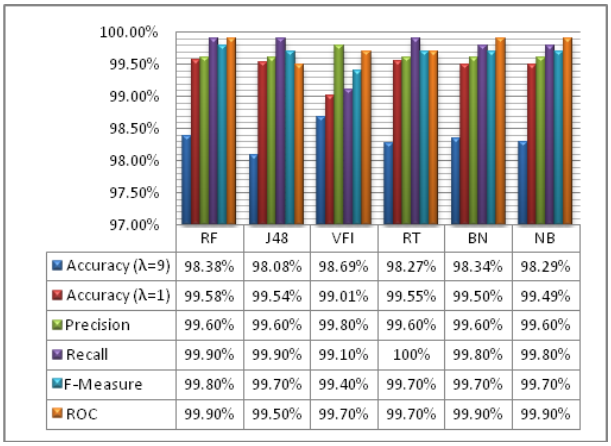
4.3.4 Results Using Header Features Together with URLs Features after Features Selection Process

Time complexity analysis is one of the major factors in spam filtering, particularly when a huge volume of emails is received on the server side. Improving classification time is required in some situations, but it could decrease the performance of other metrics, such as accuracy. Consequently, the feature vector space of 48 header features plus 11 URLs features was pruned using the Correlation-based Feature Subset Selection (CFS) algorithm implemented in the Weka tool (26). The new subset of features after this selection process includes 16, 24, 30, 31, 39, and 44 from Table 1, and feature 1 from Table 4, plus the Trust feature. This subset of the original feature space was determined to possess the greatest discriminative value.

Figures 9 and 10 show the performance of several machine learning-based techniques in terms of accuracy, precision, recall, F-measure and total cost ratio after minimizing features vector space. The results show that RF classifier outperformed all other classifiers in terms of accuracy, and F-measure with 99.58%, and 99.80% respectively. But VFI outperformed all classifiers in terms of precision and total cost ratio with 99.80% and 36 respectively. There is also a small decrease in the performance of the classifiers compared with the results in figures 7 and 8; this is due to the elimination of many features from the vector space.

Minimizing the feature space slightly decreased the performance of classifiers; for example, the accuracy (λ=1), precision, recall and F-measure decreased from 99.80%, 99.80%, 99.90%, and 99.90% to 99.58%, 99.60%, 99.90%, and 99.80% respectively for RF classifier. On the other hand, time complexity of classification was improved considerably.

Table 8 shows the performance for the proposed classifiers in terms of false positive rate, false negative rate, and time complexity. Comparing these results with those in Table 7, it can be seen that the false positive and the false negative rates have been increased from 1.30% to 3.00% and from 1.90% to 3.60% for the RF and J48 classifiers, respectively. However, the time complexity decreased significantly for those same classifiers. The RF time dropped from 49s to 22s, and for J48, from 62s to 8s.



	RF	J48	VFI	RT	BN	NB
Accuracy (λ=9)	98.38%	98.08%	98.69%	98.27%	98.34%	98.29%
Accuracy (λ=1)	99.58%	99.54%	99.01%	99.55%	99.50%	99.49%
Precision	99.60%	99.60%	99.80%	99.60%	99.60%	99.60%
Recall	99.90%	99.90%	99.10%	100%	99.80%	99.80%
F-Measure	99.80%	99.70%	99.40%	99.70%	99.70%	99.70%
ROC	99.90%	99.50%	99.70%	99.70%	99.90%	99.90%

Figure 9. The performance of different machine learning-based techniques using header features together with URLs features after feature selection process in terms of accuracy, precision, recall, and F-measure.

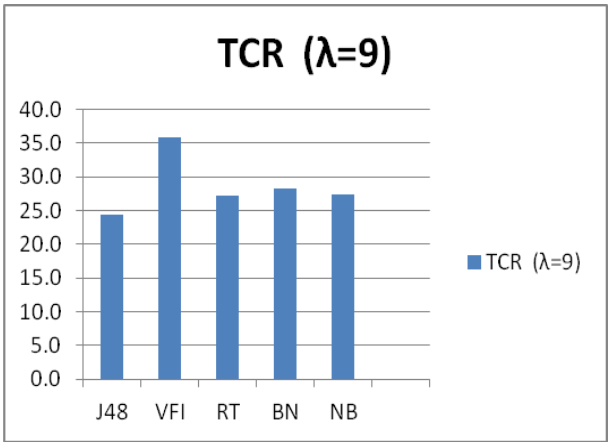


Figure 10. TCR Measure over all classifier using header features together with URLs features after feature selection process

4.4 Comparison with Previous Works

Table 9 summarizes the results of our work compared with the results of the previous related work. The results show that our approach outperformed the others, which can be attributed to the discriminative power of the features extracted from both the message header and the URLs in the message body.

Table 9. The performance of the proposed work compared to other related work. A: Accuracy, P: Precision, R: Recall, F: F-Measure, and TCR= Total Cost Ratio

Spam Filter	Ye (13) et al, 2008	Hu (14) et al, 2010	Al-Jarrah (17) , 2012	Al-Duwairi (17), 2012	Ahmed (17), 2004	Wang (17) & Chen, 2007	Our Approach
Classifiers	SVM	RF,DT,NB,BN,SVM	DT, SVM, MP, NB, BN, RF	DT, SVM, MP, NB, BN, RF	Rule-Based Word Stemming Algorithm	Statistical Analysis	RF,J48,VFI, RT, ,BN, NB
Best Performance	A=98.1 % P=99.28 % R=96.9%	RF (A=96.7%, P=93.5%, R=92.3%, F=93.3%)	RF (A=98.5% , P=98.9%, R=99.2%, F=99%)	RF (A=98.9% , P=99%, R=99.5%, F=99.2%)	94% of spam emails are classified	92.5% of junk emails are filtered out	RF (A=99.69%, P=99.90%, R=99.80%, F=99.8%, TCR=65)

5. CONCLUSION

Email spam filtering continues to be a challenging problem, as spammers persist at inventing and developing new spamming methods. Spammers also work hard at generating different templates for spam messages in order to evade detection by spam filters. This renders solutions based on filtering on features extracted from header fields only, or from body fields only, extremely inefficient. Header-based filters can be defeated easily by spoofing header fields, and those based on the body are vulnerable to the manipulation of data types such as plain text, HTML, images, and attachments to confuse and evade spam filters.

In this paper we presented 59 features that were extracted from both header and body parts, with a focus on the URL content in the body, since URL addresses are used most extensively by spammers in their campaigns

due to the negligible cost of sending them. In addition, we proposed a new "Trust" feature based on the behavior of the sender.

The features were evaluated using different machine learning–based classifiers including Random Forest (RF), C4.5 Decision Tree (J48), Voting Feature Intervals (VFI), Random Tree (RT), Bayesian Network (BN), and Naïve Bayes (NB). The feature extraction and testing were carried out using 100,000 emails selected from a mixed data set prepared from three publicly available sources. The experimental results show that classifying message using header and URL features together improves the performance of the filters. For example, the RF classifier has an accuracy (when λ=9), precision, recall, F-Measure, false positive rate, false negative rate, and total cost ratio with 99.28%, 99.80%, 99.80%, 99.90%, 99.90%, 1.30%, 0.10% and 65 respectively. Moreover, our results outperform the previous related work results (14) in terms of accuracy, precision, recall, and F-measure.

REFERENCES

1. Alexandros N, Marc N, Mark M, Dennis F. Detecting spam web pages through content analysis. Proceedings of the 15th international conference on World Wide Web. Edinburgh, Scotland: ACM; 2006.
2. Christian K, Chris K, Kirill L, Brandon E, Geoffrey MV, Vern P, et al. Spamcraft: an inside look at spam campaign orchestration. Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more. Boston, MA: USENIX Association; 2009.
3. Intelligence S. Symantec Intelligence Report: January 2013 2013 [cited February, 2013]; Available from: http://www.symantec.com/content/en/us/enterprise/other_resources/b-intelligence_report_01-2013.en-us.pdf
4. The Real Cost of Spam. 2007 [cited March, 2012]; Available from: http://www.itsecurity.com/features/real-cost-of-spam-121007/
5. Reading and Understanding Email Headers. [cited March, 2012]; Available from: http://www.by-users.co.uk/faqs/email/headers/
6. J. K Network Working Group. Simple Mail Transfer Protocol. [cited; Available from: http://tools.ietf.org/html/rfc5321
7. P. R. Network Working Group E. Request for Comments RFC 2822,. [cited March, 2012]; Available from: http://tools.ietf.org/html/rfc2822.html
8. Chhabra S. Fighting Spam, Phishing and Email Fraud: UNIVERSITY OF CALIFORNIA RIVERSIDE; 2005.
9. Gansterer W, Ilger M, Straurgen, Lechner P, Neumayer R. Anti-spam methods - state-of-the-art. Technical Report; 2005.
10. Gansterer WN, Janecek AGK, Neumayer R, Berry MW, Castellanos M. Spam Filtering Based on Latent Semantic Indexing Survey of Text Mining II. Springer London; 2008. p. 165-83.
11. Aziz Q, Ismail MK, Mahdi W. Identifying spam e-mail based-on statistical header features and sender behavior. Proceedings of the CUBE International Information Technology Conference. Pune, India: ACM.
12. Chih-Hung W. Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert Syst Appl. 2009;36(3):4321-30.

13. Miao Y, Tang T, Fan-Jin M, Xiao-Hui C. A Spam Discrimination Based on Mail Header Feature and SVM. Wireless Communications, Networking and Mobile Computing, 2008 WiCOM '08 4th International Conference on; 2008 12-14 Oct. 2008; 2008. p. 1-4.
14. Hu Y, Guo C, Ngai EWT, Liu M, Chen S. A scalable intelligent non-content-based spam-filtering framework. Expert Systems with Applications.37(12):8557-65.
15. Wang C-C, Chen S-Y. Using header session messages to anti-spamming. Computers & Security. 2007;26(5):381-90.
16. J. S. An Efficient Two-phase Spam Filtering Method Based on E-mails Categorization. I J Network Security. July 2009;9:34-43.
17. Al-Jarrah O, Khater I, Al-Duwairi B. Identifying Potentially Useful Email Header Features for Email Spam Filtering. The Sixth International Conference on Digital Society (ICDS), 2012 30 Jan. 2012; Valencia, Spain. p. 140 to 5.
18. Al-Duwairi B, Khater I, Al-Jarrah O. Texture Analysis-Based Image Spam Filtering. Internet Technology and Secured Transactions (ICITST), 2011 International Conference for 11-14 Dec. 2011. p. 288-93.
19. Ahmed S, Mithun F. Word Stemming to Enhance Spam Filtering. Proceedings of the First Conference on Email and Anti-Spam (CEAS). 2004.
20. Albrecht K, Burri N, Wattenhofer R. Spamato -- An Extendable Spam Filter System. Second Conference on Email and Anti-Spam. 2005.
21. Eleni G, Marios DD, Athena S. On the properties of spam-advertised URL addresses. J Netw Comput Appl. 2008;31(4):966-85.
22. n-gram. [cited March, 2012]; Available from: http://en.wikipedia.org/wiki/N-gram
23. Group JKNW. Simple Mail Transfer Protocol. [cited March, 2012]; Available from: http://tools.ietf.org/html/rfc5321
24. Web of Trust. [cited March, 2012]; Available from: http://www.mywot.com/
25. Aman Sharma SS. A Comparative Study of Classification Algorithms for Spam Email Data Analysis. International Journal on Computer Science and Engineering (IJCSE). 2011;3(5):6.
26. Mark Hall EF, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. The WEKA Data Mining Software: An Update. SIGKDD Explorations. 2009.
27. Corpus CLSCL. [cited March, 2012]; Available from: http://plg1.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/fooceas
28. C. GROUP. (2010 Sed, CSDMC2010, corpus) S. [cited March, 2012]; Available from: http://csmining.org/index.php/spam-email-datasets-.html
29. SPAM Archive. [cited May, 2012]; Available from: http://untroubled.org/spam/
30. Le Z, Jingbo Z, Tianshun Y. An evaluation of statistical spam filtering techniques. 2004;3(4):243-69.