Feature Selection for Spam and Phishing Detection

Fergus Toolan & Joe Carthy
UCD Centre for Cybercrime Investigation
School of Computer Science and Informatics
University College Dublin
Ireland
{fergus.toolan,joe.carthy@ucd.ie}

Abstract

Unsolicited Bulk Email (UBE) has become a large problem in recent years. The number of mass mailers in existence is increasing dramatically. Automatically detecting UBE has become a vital area of current research. Many email clients (such as Outlook and Thunderbird) already have junk filters built in. Mass mailers are continually evolving and overcoming some of the junk filters. This means that the need for research in the area is ongoing. Many existing techniques seem to randomly choose the features that will be used for classification. This paper aims to address this issue by investigating the utility of over 40 features that have been used in recent literature. Information gain for these features are calculated over Ham, Spam and Phishing corpora.

1 Introduction

Unsolicited Bulk Email (UBE) has become a huge problem in recent years. The ease of communicating with such a large number of people which the advent of email has caused has led to a huge increase in the number of bulk emails being sent.
UBE can be sub-divided into two related but distinct categories: spam and phishing. Spam is a mass-mailing approach to marketing regularly selling such items as diet supplements, unlicenced medicines and pornography. Phishing is the impersonation of an organisation by the phisher for the purposes of stealing personal data.
According to Message Labs Intelligence report [14] spam now comprises approximately 88%[1] of all email traffic. The effect of this on valid users is many fold.

1. Spammers often advertise products / services that may be harmful or offensive to recipients such as unlicenced medicines or pornography.
2. Recipients productivity is decreased as they read spam emails.
3. Mail server efficiency is decreased due to the extra volume of incoming email.

Similarly the phishing problem is also growing at an alarming rate. According to statistics collected by the Anti-Phishing Work Group [3], the number of phishing web sites are on the increase, 34,758 unique phishing reports were submitted to the APWG in December 2008 [2]. A Gartner Report, released in December 2007, estimated that Phishing related losses cost organisations over $3 billion annually. A Cyveillance report of October 2008 estimated that for every 5,000,000 phishing emails sent to people 2,500 people were successfully scammed. While this represents only 0.05% of recipients the use of bulk mailing programs mean that so many emails are sent that even this small return is a very lucrative source of revenue for phishers.

[0]978-1-4244-7761-6/10/$26.00 2010 IEEE

[1]For comparison purposes in 2006 Spam accounted for approximately 40% of all email traffic.

Many attempts have been made to use machine learning to classify incoming emails as belonging to the spam or phishing categories [4, 6, 9, 11, 19, 22] . One of the most important aspects for the success of any machine learning system is the set of features used to represent each instance [15]. Numerous features have been suggested over the years in order to best represent spam and phishing instances. However, no paper has provided a complete study of the possible features and an evaluation of their potential utility in this task. This paper aims to do just that, from a survey of the literature we have created a list of 40 features that have been used in the past and have evaluated their utility in spam / phishing detection. We then proceed to create a classifier using the best features and evaluate its performance.

This paper is organised as follows. Section 2 presents related work and gives an overview of the techniques used to classify emails. Section 3 provides a list of the features taken from the papers presented in Section 2 that will form the basis of this work. Section 4 introduces the datasets that are used in this study. Section 5 introduces the metrics that we will use in order to examine the utility of each feature. Section 6 presents our evaluation of the feature utility and describes the performance of the classifier created from the best, median and worst five features. We conclude in Section 7 and offer suggestions for future work in this area.

2 Related Work

This section describes the related work. In particular it gives an overview of the techniques used in the systems which we used to generate a list of potential features. Toolan & Carthy [22] used a recall-boosting ensemble approach. Deliberately they only chose a small feature set (five features) in order to speed pre-processing and classification for individual techniques. Their approach used a freely available dataset for evaluation, this being a combination of the Ham email set from the Spam Assassin Project [20] and Nazario's [16] phishing emails. This dataset contained over 8,000 emails in total, of which almost 50% were phishing emails. The recall boosting technique, R-Boost, was based on observations on performance of the C5.0 machine learning algorithm and an ensemble created from instance-based learning techniques. They observed that the precision of C5.0 was very good while the ensemble's recall was much better. The technique involved combining these two ideas, hence when C5.0 classified an email as non-phishing it was reclassified by the ensemble. This technique achieved a recall level of 100% and a precision that was slightly higher than that of C5.0 (although not significantly so).

recall = tp/(tp + fn)

Saberi et al. [18] used a classifier ensemble to detect phishing scams. The ensemble employs a simple consensus technique in order to combine the results of the K-NN, poisson probability distributions and naïve bayes algorithms in order to boost performance of the system. The features used in this work are all textual in nature. A dataset, composed of the Enron spam collection (for spam and ham emails) and the set of phishing emails has a total of more than 6,500 emails. Of these 529 are classified as phishing scams, while the remainder are a mixture of spam (4,500) and ham (1,500) emails. The classifiers were all based on textual features involving the 2,400 most frequently occuring terms in the corpus. The cost of this step makes this approach prohibitve and from the results obtained there are better systems available that use structural features only (which are quicker to extract from an email).

3 Potential Features

This section describes all of the features used in this work. The features used are only those that are internal to the emails themselves, rather than any from external sources. Many authors have utilised features from external sources such as spam assassin scores, domain registry information, or search engine information. For a number of reasons we have decided to avoid all of these external features in the search for the most informative feature set. These reasons include:

- The email message is the only piece of information that is guaranteed to be available to all peo-

ple involved in the spam / phishing detection task.

- Much of the external data changes regularly, for instance DNS information or search results.
- Blacklist / whitelist approaches require too much work on the part of individuals / organisations that they are not a feasible addition to a truly automated spam / phishing detection system.

This study extracted 40 features from email messages. These features were decided upon after a review of the literature in the area. In many cases authors appear to arbitrarily select features without ever studying the potential gain to be had from using particular features.
The features that we identified were roughly divided into five distinct categories. These categories are:

- **Body-based features:** These features are extracted directly from the body text of the email, including such extra information as the email's content-type.
- **URL-based features:** These features are extracted from the anchor tags in HTML emails.
- **Subject-based features:** These features are extracted from the subject line of the email.
- **Script-based features:** These features are related to the presence / absence of scripts in the email and to the effects that scripts might have on the user's experience.
- **Sender-based features:** These features are extracted from the sender email address information in the email.

The body-based features include the following:

- *body_html:* This is a binary feature that represents the presence of HTML in the email body. The *body_html* feature has previously been used in [9, 22].
- *body_forms:* This binary feature represents the presence of forms in HTML email bodies. The *body_forms* feature has been used previously in [4].
- *body_noWords:* This feature measures the total number of words occuring in the email. The *body_noWords* has been used in [6].
- *body_noCharacters:* This features measures the total number of characters occuring in the email body. This has been previously used in [6].
- *body_noDistinctWords:* This feature measures the total number of distinct words occuring in the body of the email. This feature has also been used by [6].
- *body_richness:* Chandrasekaran [6] proposed using this feature. The *richness* is defined as the ratio of the number of words to the number of characters in the document. This is expressed mathematically in equation 1.

$$body\_richness = \frac{body\_noWords}{body\_noCharacters} \quad (1)$$

- *body_noFunctionWords:* Chandrasekaran [6] also listed a set of function words that included: account; access; bank; credit; click; identity; inconvenience; information; limited; log; minutes; password; recently; risk; social; security; service; and suspended. The *body_noFunctionWords* feature measures the total number of occurrences of these function words in the email body.
- *body_suspension:* This binary feature represents the presence of the word *suspension* in the body of the email. This has been used in [19].
- *body_verifyYourAccount:* This binary feature represents the presence of the phrase *verify your account* in the body of the email. This has previously been used in [4, 19].

The subject line features are:

- *subj_reply:* This binary feature records if the email is a reply to a previous email from the sender. This feature has not been used previously.

- *subj_forward:* This binary feature records if the email is forwarded from another account to the recipient. This feature has not been used previously.
- *subj_noWords:* This feature records the total number of words in the subject line of the email.
- *subj_noCharacters:* This feature records the total number of characters in the email's subject line.
- *subj_richness:* This feature records the richness of the subject line using the formula described in equation 1.
- *subj_verify:* This binary feature describes if the email's subject line contains the word *verify*. The *subj_verify* feature was used in [19].
- *subj_debit:* This binary feature describes if the email's subject line contains the word *debit*. The *subj_debit* feature was used in [19].
- *subj_bank:* This binary feature describes if the email's subject line contains the word *bank*. The *subj_bank* feature was used in [19].

The URL features used in this paper are:

- *url_ipAddress:* This binary feature represents the use of IP addresses rather than a qualified domain name. This has been used in prior work such as [4, 9, 22].
- *url_noIpAddresses:* A continuous feature that measures the number of links in an email that contain IP addresses rather than fully qualified domain names. This has previouly been used in [4].
- *url_atSymbol:* This binary feature represents the presence of links that contain an @ symbol. This has been used in [10].
- *url_noLinks:* This continuous numeric feature measures the number of links in the email body. This has been used by [4, 9, 22].
- *url_noIntLinks:* This continuous feature describes the number of links whose target is internal to the email body. This has been used in [4].
- *url_noExtLinks:* This continuous feature describes the number of links whose target is outside the email body. This has also been previously by [4].
- *url_noImgLinks:* This continuous feature measures the number of links where the user needs to click on an image in the email body. This has been used in [4].
- *url_noDomains:* A continuous feature that measures the total number of domains in all URLs in the email. Used by [9].
- *url_maxNoPeriods:* This continuous numeric feature measures the number of periods in the link with the highest number of periods. This has been used previously by [4, 9, 22].
- *url_linkText:* This binary feature is true if the human-readable link text contains one or more of the following terms: click; here; login; or update. This feature is again used in [4].
- *url_nonModalHereLinks:* The modal domain is defined as the domain that is most frequently linked to in an email message [9]. This feature is a binary feature that captures *here* links that link to a domain other than the modal domain. This feature is used by [9].
- *url_ports:* This binary feature indicates whether a URL accesses ports other than 80. This has only been used in [19]. Some papers specify the individual port numbers (for instance 4903 and 87 in [19]).
- *url_noPorts:* This continuous numeric feature represents the the number of links in the email that contain port information in the address. This feature has not been used previously.

The script-based features used included

- *script_scripts:* This binary feature has been used in [4, 22]. It represents the presence of scripts in the email body.

- *script_javaScript:* This feature represents the presence of javascript in the email body. It is a binary feature which has been used in [4, 9].

- *script_statusChange:* A binary feature that is true if the script attempts to overwrite the status bar in the email client. The *script_statusChange* feature has been used in [10, 21].

- *script_popups:* A binary feature that is true if the email contains pop-up window code. This has been used in [10].

- *script_noOnClickEvents:* This continuous feature counts the number of *onClick* events in the email. This feature has been used in [10].

- *script_nonModalJsLoads:* Javascript is not always embedded in the email / webpage. The javascript code can be loaded from an external site using the SRC attribute of the SCRIPT tag. The *script_nonModalJsLoads* feature represents the presence of external javascript forms that come from domains other than the modal domain. This feature has previously been used in [10].

The sender features, newly proposed in this paper, include:

- *send_noWords:* This continuous feature represents the total number of words in the send field. The sender field in an email is not merely an email address, it usually has the form *"Joe Bloggs" <joe@bloggs.com>*.

- *send_noCharacters:* This continuous feature represents the total number of characters in the sender field.

- *send_diffSenderReplyTo:* This binary feature shows if there is a difference between the sender's domain and the reply-to domain.

- *send_nonModalSenderDomain:* This binary feature shows if the sender's domain is different from the email's modal domain.

The combination of these, results in a total of 40 features which were used in this study.

4 Datasets

In this work we used combinations of three freely available datasets for spam and phishing detection. The first of these is the ham (legitimate) email collection available from the Spam Assassin project [20]. The second, from the same source, is a spam email dataset. The third was composed of phishing emails provided by Nazario [16]. The statistics of these datasets are summarised in Table 1.

Dataset	Size	Start	End
Ham (H)	4,202	Jan 2002	Oct 2002
Spam (S)	1,895	Jan 2000	July 2003
Phish (P)	4,563	Nov 2004	Aug 2007

Table 1: Basic dataset statistics.

From these, three datasets were created. The first dataset is used to investigate feature importance in spam detection. This dataset consisted of a combination of the ham and spam datasets from Table 1. The second dataset was for testing feature selection for the phishing detection task and consisted of the ham and phish corpora. These datasets while useful for seeing the relative importance of features for spam and phishing detection respectively are not realistic. The third dataset created was a combination of all three constituents to reflect the fact that real email systems would receive ham, spam and phishing emails simultaneously. The statistics of these datasets are summarised in Table 2.

All experiments on the following sections are performed on the three datasets in Table 2.

Dataset	Components	Size	Classes
1	H, S	6,097	2
2	H, P	8,765	2
3	H, S, P	10,660	3

Table 2: Basic dataset statistics.

5 Information Metrics

In this section we introduce the information theoretic measures that will be used to measure the effectiveness of each feature as a potential instance element for the representation of phishing for the classification task. The two main measures that we will examine are: *entropy* which measures the disorder in a system and the *information gain* which measures the reduction in entropy achieved in classification through use of a particular feature.

5.1 Entropy

Given a collection of instances S, the entropy of S, $E(S)$ is the measure of impurity or disorder in the system [15]. Entropy is defined, in the general case, as shown in Equation 2

$$E(S) = \sum_{i=1}^{N} -p_i log_2 p_i \tag{2}$$

where N is the number of classes (categories) in the entire dataset, and p_i is the probability that a particular instance belongs to class i. In the case of datasets 1 & 2 (see Table 2) there are only two classes, ham and spam / phishing respectively. In the case of dataset 3 there are three classes as spam and phishing are both included in this dataset along with the ham emails. Equations 3 and 4 give the respective formulae for the two and three class problems.

$$E(S) = -p_h log_2 p_h - p_{s/p} log_2 p_{s/p} \tag{3}$$

$$E(S) = -p_h log_2 p_h - p_s log_2 p_s - p_p log_2 p_p \tag{4}$$

where p_p represents the probability that an instance is a phishing email and p_s is the probability that it is a spam email and p_p is the probablity that it is a phishing email. The entropies of the three datasets are shown in Table 3

Dataset	Entropy
1	0.89432
2	0.52394
3	1.49667

Table 3: Entropy values for the three datasets described in Table 2.

Using these entropy values we can now proceed to examine the effects of each feature in Section 3 in reducing the amount of entropy in the system. This is achieved through the information gain metric.

5.2 Information Gain

The most effective attribute for classification is the one that reduces entropy by the largest amount. The metric that measures this is called information gain [15]. The information gain of attribute, A, over the dataset, S, is given in Equation 5.

$$G(S, A) = E(S) - \sum_{v \in values(A)} \frac{S_v}{S} E(S_v) \tag{5}$$

where $E(S)$ is the entropy of the entire dataset as calculated using Equation 3, A is the attibute for which we are measuring information gain, S_v is the number of attributes in S where A has the value v and $E(S_v)$ is the entropy of this subset of the dataset. The next section describes the application of these metrics to the three datasets. The larger values of information gain represent those attributes that give the largest reduction in entropy and as such represent the best attributes for classification.

6 Evaluation

In order to evaluate the attributes for potential effectiveness in a classification system we calculated the information gain of each attribute and then used subsets of these to create classifiers to evaluate our hypotheses. We begin the information gain calculations

with dataset 1 containing ham and spam email (Section 6.2), we continue with dataset 2, ham and phishing, (Section 6.2) and finish with dataset 3, ham, spam and phishing (Section 6.3). We complete the evaluation by testing the initial hypothesis that the attributes with the highest information gain will form the best classifiers. For each dataset we choose subsets of the attributes and evaluate their performance using the C5.0 decision tree learning algorithm [17]. The results from these are shown in Section 6.5.

6.1 Dataset 1

The first dataset contains both ham and spam emails, with a total size of 6,097 emails of which 69% are ham and the remainder are spam. Table 4 shows the information gain values for each of the 40 attributes studied.

From these results it is clear that the 3 best attributes for classification of spam emails are: *body_noFunctionWords*, *body_richness*, and *subj_richness*. The total number of links (*url_noLinks*) and similarly the total number of external links (*url_noExtLinks*) are also very good attributes to use. Conversely the number of internal links (*url_noIntLinks*) in the email is much worse for classification purposes, being the 30^{th} feature in the information gain rankings.

Other good features to use are the number of domains (*url_noDomains*), the presence of HTML in the email (*body_html*) and the maximum number of periods in any URL in the email (*url_maxNoPeriods*). The remaining features in the top 10 are *body_noWords* and *body_noCharacters*. These are less important as the *body_richness* feature is derived from these and hence explains their good resolving power.

The worst features for the recognition of spamming emails are *body_verifyYourAccount* and *subj_debit*. This is explained as both of these features are related to the presence of financial terms in various parts of the email. These features were taken from papers that were focusing solely on the phishing detection problem rather than the more general spam detection problem.

Attribute	**IG**
body_noFuncWords	0.89449
body_richness	0.89285
subj_richness	0.87726
body_noCharacters	0.75251
url_noLinks	0.73466
url_noExtLinks	0.73436
url_noDomains	0.71111
body_html	0.70692
url_maxNoPeriods	0.69789
body_noWords	0.69280
url_ipAddress	0.68157
send_noWords	0.68119
send_noCharacters	0.68107
body_noDistinctWords	0.67426
url_linkText	0.67369
subj_reply	0.66862
url_nonModalHereLinks	0.66050
url_noIpAddresses	0.65661
url_atSymbol	0.65541
subj_noCharacters	0.64778
url_noImgLinks	0.64289
body_forms	0.64181
subj_noWords	0.64137
script_statusChange	0.64023
send_nonModalSenderDomain	0.63951
script_popups	0.63871
url_noPorts	0.63841
url_ports	0.63820
send_diffSenderReplyTo	0.63744
url_noIntLinks	0.63732
subj_verify	0.63727
script_onClickEvents	0.63727
subj_forward	0.63693
script_nonModalJsLoad	0.63679
script_javascript	0.63675
subj_bank	0.63672
body_suspension	0.63672
script_scripts	0.63672
body_verifyYourAccount	0.63670
subj_debit	0.63670

Table 4: Information gain (IG) values for each of the attributes calculated on dataset 1.

6.2 Dataset 2

Dataset 2 is composed of the ham and phishing emails. As seen in Table 3 the entropy for the entire dataset was $E(S) = 0.99872$. Table 5 shows the information gain for each attribute in order from best to worst. From this table it is clear that the utility of the various attributes varies greatly. The most informative attribute is that of *body_richness* which has an information gain of 0.99635 while the least informative attribute is *body_forms* which has an information gain of 0.00011.

The two most informative features, *body_richness* and *subj_richness* would suggest that language modelling approaches to phishing classification may be the most worthwhile. Both of these features give information on the use of language in the email, specifically the ratio of words to characters in the body and subject areas of the email respectively. The third most informative attribute is, not surprisingly, the *body_html* attribute. This feature has been used in many systems over recent years and its use is vindicated by this study.

The next four features are all url-based features. These are *url_noLinks*, *url_noExtLinks*,*url_noDomains*, and *url_maxNoPeriods*. These four features show the importance of link analysis in the detection of phishing emails. The first measure the total number of links and the total number of external links in the email. Conversely the *url_noIntLinks*, number of internal links, feature appears much further down the list, at position 34. This shows that the presence of internal links in an email gives little indication as to the final classification of an email. The least informative attribute was a surprising one, *body_forms*, the presence of forms in the email. Many researchers [4, 19, 9] have felt that the presence of forms inside an email would prove a huge indicator in the phishing classification task. This study has shown that this is not the case and indeed the use of this feature could hinder the overall accuracy of a phishing detection system.

Attribute	**IG**
body_richness	0.99635
subj_richness	0.95022
body_html	0.72444
url_noLinks	0.71221
url_noExtLinks	0.71023
url_noDomains	0.63692
url_maxNoPeriods	0.61339
body_noCharacters	0.54140
subj_reply	0.30345
body_noWords	0.29657
body_noDistinctWords	0.23351
url_ipAddress	0.21655
url_noImgLinks	0.16610
url_noIpAddresses	0.16369
send_noCharacters	0.15985
body_noFunctionWords	0.13908
send_nonModalSenderDomain	0.09807
subj_bank	0.07449
send_noWords	0.07141
subj_noCharacters	0.06392
url_linkText	0.06345
body_suspension	0.05420
send_diffSenderreplyTo	0.04281
script_onClickEvents	0.03394
url_noPorts	0.02640
url_ports	0.02637
body_verifyYourAccount	0.02563
url_nonModalHereLinks	0.02113
subj_verify	0.01665
script_statusChange	0.01423
subj_noWords	0.00944
script_javaScript	0.00768
script_scripts	0.00651
url_noIntLinks	0.00596
script_nonModalJsLoads	0.00537
script_popups	0.00414
subj_debit	0.00269
subj_forward	0.00018
url_atSymbol	0.00013
body_forms	0.00011

Table 5: Information gain (IG) values for each of the attributes calculated on dataset 2.

6.3 Dataset 3

Dataset 3 is composed of all of the emails from the three individual collections i.e. ham, spam and phishing. As it is a three class problem, unlike the previous two datasets, this dataset has the highest entropy value of 1.49667. Table 6 shows the information gain values for the attributes when classifying on this dataset.

Again we see that the best attributes for classification are *body_richness* and *subj_richness*. The total number of links and the number of external links are again very important. The remaining attribute in the top five is *body_noCharacters*. Again the top ten contain *body_html*, *url_maxNoPeriods*, *url_noDomains* and *body_noWords*. The final attribute in the top ten is *body_noDistinctWords*.

In general the script and subject features have very low information gain values in this particular dataset.

Attribute	**IG**
body_richness	1.48804
subj_richness	1.41020
body_noCharacters	0.75611
url_noLinks	0.61340
url_noExtLinks	0.61156
body_html	0.59503
url_maxNoPeriods	0.55664
url_noDomains	0.53154
body_noWords	0.38756
body_noDistinctWords	0.29142
subj_reply	0.28717
send_noCharacters	0.21531
body_noFunctionWords	0.20850
url_IpAddress	0.17751
url_noIpAddresses	0.14254
url_noImgLinks	0.14200
send_noWords	0.13527
send_nonModalSenderDomain	0.08680
subj_noCharacters	0.07820
subj_bank	0.07670
url_linkText	0.07580
body_suspension	0.05751
url_atSymbol	0.04134
url_nonModalHereLinks	0.03936
send_diffSenderReplyTo	0.03482
script_onClickEvents	0.03339
body_verifyYourAccount	0.02658
url_noPorts	0.02303
url_ports	0.02201
subj_verify	0.01515
subj_noWords	0.01493
body_forms	0.01323
script_statusChange	0.01093
script_javascript	0.00803
script_nonModalJsLoads	0.00594
url_noIntLinks	0.00544
script_scripts	0.00541
script_popups	0.00330
subj_debit	0.00209
subj_forward	0.00019

Table 6: Information gain (IG) values for each of the attributes calculated on dataset 3.

6.4 Information Gain Analysis

Overall it is clear that the best two features to use for classification are *body_richness* and *subj_richness*. These features suggest that language modelling approaches may be an effective means of classifying emails for spam / phishing detection.

Over the three datasets 9 features appear in the top ten of all of them. These are listed in Table 7.

From this it is clear that a combination of richness (including number of words and number of characters) with some of the url-based features provides the most informative attributes for classification. These are investigated using a C5.0 classifier in the next section. Some of the more surprising results were the poor information gain for the *body_forms* attrbiute. This has long been suggested as a means of detecting spam / phishing emails but is obviously very ineffective for this task. This is probably due to the changes in spam / phishing since their inception. These scams are now normally partnered with a website on which the form would be accessable to the user through a link in the email body. Another commonly cited important feature is the presence of HTML in an e-mail. This is very informative in all cases, however, it is worth noting that it is less important for spam de-

body_richness	*subj_richness*
url_noLinks	*url_noExtLinks*
url_noDomains	*body_noCharacters*
body_html	*url_maxNoPeriods*
body_noWords	

Table 7: Nine features that appeared in the top ten ranks of information gain across all of the datasets.

tection than phishing detection. This is due to the aim of a phishing email which is to acquire information (or bring the user to a website that attempts to acquire information). In order to do this HTML is necessary so that links can be included in the email body.

6.5 Classifer Construction

In order to show the utility of the results to date in this paper we introduce a hypothesis, namely that classifiers created using the higher ranked information gain attributes will perform better than those using lower ranked attributes. In order to test the validity of this statement we used the C5.0 [17] algorithm as our classifier. This decision tree learning algorithm has been shown to be very effective in many classification tasks and also in the phishing domain itself [22]. For each dataset three groups of five features were used as inputs to C5.0. These were labelled *best*, *median*, and *worst*. Table 8 shows the features used for each dataset.

The experiments used the C5.0 algorithm on ten training / test splits that had been created from each of the datasets. The training set size used was 10% as studies have shown that this algorithm can learn well on small training set sizes [22]. For each dataset the *best*, *median*, and *worst* classifier was trained on each of the ten training sets and then evaluated on the corresponding test set. Table 9 shows the results of these runs and the overall averages.

As can be seen from these results the expected performance is achieved. In each of the ten runs performed on different training / test splits, the group of attributes with the highest information gain values proved to be the most effective at classifying the emails in each dataset. Also it should be noted that this is using the C5.0 algorithm with its most basic parameters. Other studies have shown the benefit in using more advanced variants of this algorithm, for instance by using boosting in conjunction with it, or using ensembles of classifiers to boost the recall achieved by the system [22]. However, our aim was to use this technique to show the importance of information gain for feature selection and to give a list of features that can be used effectively for classification.

7 Conclusions

This paper surveyed currently used features in automated spam / phishing email detection systems. In many of these systems features appear to be chosen based on the author's intuition that they will be effective in classifying an email into the ham or phishing categories. This paper has identified 40 features the majority of which have been used repeatedly in the literature. We extracted these 40 features from a body of over 10,000 emails, which were divided amongst three classes ham, spam and phishing. We then calculated the information gain of all of these features. From this we created C5.0 classifiers using three groups of features, those with the best IG values, the median IG values, and finally the worst IG values. As expected, in each case, the classifier trained on the best features outperformed all of the others.

References

[1] Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. *A Comparison of Machine Learning Techniques for Phishing Detection*, in Proceedings of the APWG eCrime Researchers Summit, Pittsburgh, USA, 2007.

[2] The Anti-Phishing Work Group, http://www.apwg.org/, Last accessed June 2009.

Dataset	*Best*	*Median*	*Worst*
1	*body_noFunctionWords*	*url_noIpAddreses*	*subj_bank*
	body_richness	*url_atSymbol*	*body_suspension*
	subj_richness	*send_noCharacters*	*script_scripts*
	body_noCharacters	*url_noImgLinks*	*body_verifyYourAccount*
	url_noLinks	*body_forms*	*subj_debit*
2	*body_richness*	*subj_bank*	*script_popups*
	subj_richness	*send_noWords*	*subj_debit*
	body_html	*subj_noCharacters*	*subj_forward*
	url_noLinks	*url_linkText*	*url_atSymbol*
	url_noExtLinks	*body_suspension*	*body_forms*
3	*body_richness*	*send_nonModalSenderDomain*	*url_noIntLinks*
	subj_richness	*subj_noCharacters*	*script_scripts*
	body_noCharacters	*subj_bank*	*script_popups*
	url_noLinks	*url_linkText*	*subj_debit*
	url_noExtLinks	*body_suspension*	*subj_forward*

Table 8: Features used in each of the nine classifiers.

[3] The Anti-Phishing Work Group, *Phishing Activity Trends Report 2nd Half 2008*, Available at http://www.apwg.org/reports/apwg_report_H2_2008.pdf, 2009.

[4] Bergholz, A., Paaß, G., Reichartz, F., Strobel, S., & Chung, J. H. *Improved Phishing Detection using Model-Based Features*, in Proceedings of the International Conference on E-mail and Anti-Spam, 2008.

[5] Bergholz, A., De Beer, J., Glahn, S., Moens, M-F., Paaß, G. & Strobel, S. *New Filtering Approaches for Phishing Email*, in Journal of Computer Security, Vol 18, No 1, pp 7-35, 2010.

[6] Chandrasekaran, M., Narayanan, K. & Upadhyaya, S. *Phishing E-mail Detection Based on Structural Properties*, in Proceedings of the 9th Annual NYS Cybercrime Security Conference, Symposium on Information Assurance, 2006.

[7] Chen, K-T., Huang, C-R., Chen, C-S., & Chen, J-Y. *Fighting Phishing with Discriminative Features*, in IEEE Internet Computing, Volume 13 Number 3, 2009.

[8] Cohen, W. W. *Learning Rules that Classify E-Mail*, AAAI Technical Report, SS-96-05, 1996.

[9] Fette, I., Sadeh, N., & Tomasic, A. *Learning to Detect Phishing Emails*, Technical Report, Institute for Software Research International, School of Computer Science, Carneige Mellon University, 2006.

[10] Gansterer, W. N. & Polz, D. *E-Mail Classification for Phishing Defense*, in LNCS Advances in Information Retrieval, Vol 5478, pp 449-460, 2009.

[11] Garera, S., Provos, N., Chew, M., & Rubin, A. D. *A Framework for Detection and Measurement of Phishing Attacks*, in Proceedings of the 2007 ACM Workshop on Recurring Malcode, 2007.

[12] Jakobsen, M., & Ratkiewicz, J., *Designing Ethical Phishing Experiments: A Study of (ROT13) rOnl Query Features* in Proceedings of the 15th International Conference on the World Wide Web, Scotland, 2006.

[13] Martin, S., Sewani, A., Nelson, B., Chen, K., & Joseph, A. D., *Analyzing Behavioral Features*

	Dataset 1			Dataset 2			Dataset 3		
Run	***Best***	***Median***	***Worst***	***Best***	***Median***	***Worst***	***Best***	***Median***	***Worst***
1	97.1%	70.8%	49.0%	84.6%	75.0%	68.9%	79.9%	61.5%	43.0%
2	96.6%	71.7%	49.5%	83.7%	74.3%	68.6%	79.9%	61.4%	42.6%
3	97.2%	71.4%	52.2%	83.3%	73.6%	68.7%	79.5%	61.3%	42.9%
4	97.4%	71.7%	51.9%	84.1%	74.4%	68.7%	79.2%	61.7%	42.6%
5	96.7%	71.5%	52.0%	83.1%	74.5%	69.1%	78.4%	60.3%	42.4%
6	97.0%	72.0%	51.7%	84.6%	74.8%	68.7%	78.3%	61.7%	42.7%
7	97.3%	69.6%	52.0%	82.2%	74.0%	68.9%	78.5%	62.3%	42.7%
8	97.4%	73.3%	51.8%	84.6%	73.0%	68.9%	78.3%	61.5%	42.8%
9	97.4%	72.8%	52.0%	84.2%	74.4%	69.1%	79.2%	62.4%	42.6%
10	97.3%	71.5%	52.0%	84.1%	74.4%	68.8%	79.7%	61.4%	42.7%
Avg.	97.1%	71.6%	51.4%	84.1%	74.2%	68.8%	79.1%	61.6%	42.7%

Table 9: Results of the *best*, *median* and *worst* classifiers as predicted by information gain over the three datasets.

for Email Classification, in Proceedings of the 2^{nd} Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, Stanford University, 2005.

[14] Message Labs Intelligence, *Message Labs Intelligence Q3 / September 2009*, available at http://www.messagelabs.co.uk/intelligence.aspx, accessed 27 Nov 2009.

[15] Mitchell, T. *Machine Learning*, McGraw-Hill, 1997.

[16] Nazario, J. Phishing Corpus, From http://www.monkey.org/ jose/wiki/ doku.php?id=phishingcorpus. Last accessed June 2009.

[17] Quinlan, J. R. *Is See5/C5.0 better than C4.5*, available at http://www.rulequest.com/see5-comparison.html. Last accessed November 2009.

[18] Saberi, A., Vahidi, M., & Bidgoli, B. M. *Learn to Detect Phishing Scams Using Learning and Ensemble Methods*, in Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, pp 311-314, USA, 2007.

[19] SonicWall, *Bayesian Spam Classification Applied to Phishing E-Mail*, Sonicwall White Paper, 2008.

[20] Spam Assassin Project, Ham Email Corpus, http://spamassassin.apache.org/publiccorpus/. Last Accessed June 2009.

[21] Strickroth, S. *Phishing Detection*, 2008

[22] Toolan, F. & Carthy, J. *Phishing Detection using Classifier Ensembles*, in Proceedings of the 4^{th} E-Crime Researchers Summit, Tacoma, WA, 2009.