Digital Waste Sorting: A Goal-Based, Self-Learning Approach to Label Spam Email Campaigns

Mina Sheikhalishahi[1](✉), Andrea Saracino[2], Mohamed Mejri[1],
Nadia Tawbi[1], and Fabio Martinelli[2]

[1] Department of Computer Science, Université Laval, Québec City, Canada
mina.sheikh-alishahi.1@ulaval.ca
{mohamed.mejri,nadia.tawbi}@ift.ulaval.ca
[2] Istituto di Informatica e Telematica, Consiglio Nazionale delle ricerche, Pisa, Italy
{andrea.saracino,fabio.martinelli}@iit.cnr.it

Abstract. Fast analysis of correlated spam emails may be vital in the effort of finding and prosecuting spammers performing cybercrimes such as phishing and online frauds. This paper presents a self-learning framework to automatically divide and classify large amounts of spam emails in correlated labeled groups. Building on large datasets daily collected through honeypots, the emails are firstly divided into homogeneous groups of similar messages (campaigns), which can be related to a specific spammer. Each campaign is then associated to a class which specifies the goal of the spammer, i.e. phishing, advertisement, etc. The proposed framework exploits a categorical clustering algorithm to group similar emails, and a classifier to subsequently label each email group. The main advantage of the proposed framework is that it can be used on large spam emails datasets, for which no prior knowledge is provided. The approach has been tested on more than 3200 real and recent spam emails, divided in more than 60 campaigns, reporting a classification accuracy of 97 % on the classified data.

1 Introduction

At the end of 2014, emails are still one of the most common form of communication in Internet. Unfortunately, emails are also the main vector for sending unsolicited bulks of messages, generally for commercial purpose, commonly known as *spam*. The research community has investigated the problem for several years, proposing tools and methodologies to mitigate this issue. However, a definitive solution to the problem of spam emails still has to be found. In fact, according to *McAfee Report* [19], unsolicited emails, constitute more than 70 percent

This research has been partially supported by EU Seventh Framework Programme (FP7/2007–2013) under grant no 610853 (COCO Cloud), MIUR-PRIN Security Horizons and Natural Sciences and Engineering Research Council of Canada (NSERC).

S. Foresti (Ed.): STM 2015, LNCS 9331, pp. 3–19, 2015.
DOI: 10.1007/978-3-319-24858-5_1

of total amount of email messages in 2014. Moreover, *Cisco Report* [26] shows that spam volume increased 250 percent from January 2014 to November 2014. Unfortunately, the problem of spam emails is not only related to unsolicited advertisement. Spam emails have become a vector to perform different kinds of cybercrimes including phishing, cyber-frauds and spreading malware.

Motivation: Trying to filter spam emails at the user end, actually is not enough to fight this kind of attacks, which moves the effect of unsolicited spam emails from illicit to real crime. Finding the spammers becomes important not only to tackle at the source the problem of spam emails, but also to legally prosecute the responsible of cybercrimes brought by spam emails different from undesired advertisement. To identify spammers, the early analysis of huge amount of messages to find correlated spam emails with the specific spammer purpose is vital. Several papers in the literature observed that the forensic analysis, which plays a major role in finding and persecuting spammers for cybercrimes, needs a proactive mechanism or tool which is able to perform a fast, multi-staged analysis of emails in a timely fashion [9,10,14,29]. To this end, large amounts of spam emails, generally collected through honeypots should be at first divided in similar groups, which could be related to the same spammer (i.e., spam campaigns). Afterward to each campaign should be assigned a label describing the purpose of spammer. This goal-based labeling facilitates for investigators the analysis of spam campaigns, eventually directed toward a specific cybercrime. However, this analysis generally appears to be a challenging task. In fact, considering the number of produced spam emails and their variance, spam email datasets are huge and very difficult to handle. In particular, human analysis is almost impossible, considering the amount of spam emails daily caught by a spam honeypot [28,29]. On the other hand, an automated and accurate analysis requires the usage of correctly trained computational intelligence tools, i.e. *classifiers*, whose training requires accurately chosen datasets, which presents to the classifier a good reality description in which it will be employed. Moreover, due to the high variance of spam emails, a valid training set may become obsolete in few weeks, and a new up-to-date training set must be generated in a short period of time.

Though previous work largely improved the state of the art in analysis of spam emails for forensic purposes, more improvement is still needed. In particular, previous work either focuses on a specific cybercrime only, especially phishing [11], or exploit in the analysis a small set of features not effective in identifying some cybercrime emails. For example, the analysis of email text words [14], link domains [10] is not effective in identifying emails used to distribute malware, which often do not contain text [20] , or spam emails with dynamic links [5].

Paper Contribution: In this paper we propose Digital Waste Sorter (DWS), a framework which exploits a self learning *goal of the spammer*-based approach for spam email classification. The proposed approach aims at automatically classifying large amount of raw unclassified spam emails dividing them into campaigns and labeling each campaign with its spammer goals. To this end, we propose five class labels to group spammer goals in five macro-groups, namely *Advertisement, Portal Redirection, Advanced Fee Fraud, Malware Distribution* *and Phishing*. Moreover, a set of 21 categorical features representative of email structure is proposed to perform a multi-feature analysis aimed at identifying emails related to a large range of cybercrimes. DWS is based on the cooperation of unsupervised and supervised learning algorithms. Given a set of *classes* describing different spammer goals and a dataset of non classified spam emails, the proposed approach at first automatically creates a valid training set for a classifier exploiting a *categorical* clustering algorithm, named CCTree (Categorical Clustering Tree). In more detail, this clustering algorithm divides the dataset into structurally similar groups of emails, named spam campaigns [7]. DWS is built on the results of CCTree , which is effective in dividing spam emails in homogeneous clusters. Afterward, significant spam campaigns useful in the generation of the training set are selected through similarity with a small set of known emails, representative of each spam class. Hence, a classifier is trained using the selected campaigns as training set, and will be used to classify the remaining unclassified emails of the dataset.

To further meet the needs of forensic investigators, which have limited time and resource to perform email examinations [9], the DWS methodology does not require a prior knowledge of dataset, except the desired classes (i.e. spammer goals) and a small set of emails representative of each class. It is worth noting that this email set cannot be used to train the classifier. In fact, this set contains a small number of emails not belonging to the dataset to be classified, being thus not necessarily descriptive of the reality in which the classifier will operate.

In the following, we will describe in details the DWS framework, explaining the process of division in campaigns, training set generation and campaigns classification. The framework effectiveness has been evaluated against a set of 3200 recent raw spam emails extracted by a honeypot. DWS reported a classification accuracy on this preliminary dataset of 97.8 %. Furthermore, to justify the classifier selection, an analysis of performances on different classifiers is presented.

The rest of the paper is organized as follows. Section 2 reports related work on email classification. Section 3 presents the DWS framework and work-flow in details, also it gives brief background information on the clustering algorithm. Section 4 presents the results of the analysis on a real dataset of spam emails, as well as a comparison on the classification results of four different classifiers. Finally Sect. 5 briefly concludes reporting planned future extensions.

2 Related Work

In the literature, the spam campaigns are usually labeled based on characteristic strings (keywords) representing individual campaign types as in [10,18] and [13]. These approaches are weak against the kind of spam emails which do not contain keywords or that use word obfuscation techniques. Pathak et al. [21] label spam campaigns on the base of URLs, phone number, Skype ID, and Mail ID used as contact information. This methodology is effective only against emails reporting contacts, which are only a subset of all the spam emails found in the wild.

There are several approaches in the literature in which the spammer goal is considered. However, these approaches are mainly focused on detecting phishing emails, not considering other spammer purposes. Fette et al. [11] applied 10 email features to discern phishing emails from ham (good) emails. Bergholz et al. [6] propose a similar methodology with additional features to train a classifier in order to filter phishing emails. Almomani et al. [3] provide a survey on different techniques in filtering phishing emails, while Gansterer et al. [12] compare different machine learning algorithms in phishing detection. Furthermore, the authors propose a technique which refines the previous phishing filtering approaches. In this work, three types of messages, named *ham, spam* and *phishing* are distinguished automatically. Nevertheless, the category of emails containing *spam*, is not precisely characterized. In [8] a methodology to detect phishing emails based on both machine learning and heuristics is proposed. These approaches report accuracy ranging from 92 % to 96 %, where the classifiers have been trained through labeled datasets. On the contrary, DWS generates the training set on the fly, without requiring a pre-trained classifier. Notwithstanding, in the performed experiments DWS shows comparable accuracy.

3 Digital Waste Sorting

DWS is a framework which takes as input datasets of unclassified spam emails. Hence, DWS divides the emails in campaigns by mean of a hierarchical clustering algorithm, then labels each campaign through a classifier. The classifier is trained on the fly, through a training set generated by DWS directly from the unlabeled input dataset, exploiting the knowledge generated by the clustering algorithm.

This section describes in details the DWS framework and methodology. First, we will present the classes used to label each spam campaign. Then, we present the feature extraction process from raw emails, discussing the features relevance in describing structural elements of an email and their relation to each spam class. The framework is then presented, briefly introducing the clustering algorithm and the methodology for the generation of the training set. Finally the classification process is presented.

3.1 Definition of Classes

As anticipated, spam emails can be sent with different intentions, spanning from the common advertisement to vectors of different cybercrimes. We argue that spam emails can be divided in five well-known macro-groups which represent the main target of spammers, and can thus be used to label spam campaigns.

Advertisement: The *advertisement* class contains those emails whose target is convincing a user to buy a specific product [17]. Advertisement emails embody the most typical idea of spam messages, advertising any kind of product which could be of interest of companies or private users. Generally these emails only constitute a hindrance to the users that have to spend time removing them from the inbox. The main requirements for a commercial email to be legal according to

Federal Trade Commission [2], is that it uses no deceptive subject lines, provides correct complete header information, real physical location of the business, offers an opt-out choice, and honors opt-out requests in 10 business days. In this paper, we consider as advertisement emails both the ones which comply with the legal requirements and the ones that does not, given that their purpose is clearly advertising a product.

Portal Redirection: *Portal redirection* spam emails are the enablers of an evolved advertisement methodology. This spam emails are characterized by a minimal structure generally reporting one or more links to one or more websites. Once the user clicks on the link, she is redirected several times to different pages whose address is dynamically generated. The final target page is mostly an advertisement portal with several links divided by categories, generally related to common user needs (e.g., medical insurance). This strategy is useful in reducing the legal responsibility on spam emails of the companies which are advertising a product. The rationale is that the advertised company cannot be sued because another website, i.e. the portal, links to it. As an example, the opt-out clause of advertisement emails [16] does not apply. Moreover, the multi-redirection with dynamic links strategy makes difficult to track the responsible websites. The strategy of portal redirection emails, is also used to redirect users on websites with the intention of defrauding the users, or to distribute malicious code.

Advanced Fee Fraud: An *advanced fee fraud* or *confidence trick* spam email (synonyms include *confidence scheme* or *scam*) attempts to defraud a person after first gaining their confidence, used in the classical sense of trust [15]. Confidential trick spam exploits social engineering to trick the user in paying, by her own will, a certain amount of money to the spammer. Scammers may use several techniques to deceive the user in paying money, generally exploiting sentimental relations or promising a large amount of money in return. The confidential trick emails, mostly are written in a friendly long text, to convince the victim the interactions. These kinds of emails, usually, do not redirect the users to other web pages, mainly contain an email address.

Malware: Emails are an important vector for spreading malicious software or *malware*. Generally the malware is sent as email attachment, while the email structure is very simple, with a small text which encourages the reader to open the attachment or no text at all [20]. Once opened, the malware infects the user device, showing different possible malicious behaviors. Often the malicious file is camouflaged, inserted in a zip file or with a modified extension, which allows to deceive basic anti-virus control implemented by some spam filters.

Phishing: *Phishing* emails attempt to redirect users to websites, which are designed to obtain credentials or financial data such as usernames, passwords, and credit card detail illegally [3]. Generally, these emails pretend to be sent by a banking organization, or coming from a service accessible through username and password, e.g. social networks, instant messaging etc., reporting fake security issues that will require the user to confirm her data to access again the service. To this end, phishing emails are mostly very well presented with a well organized structure, even reporting contact informations such as phone numbers and email. The representative structure of phishing emails we applied in this research, contain short well written text, providing the victim some important news. Mostly there exists one link, which direct the user to a very well designed fake website of a bank, which asks the victim to provide her credit card information.

3.2 Feature Extraction

DWS parses raw spam emails (`eml` files) extracting a set of 21 categorical features building a numerical vector readable by clustering and classification algorithms. The extracted features are reported in Table 1, with a brief description, whilst the values which each feature may assume is reported in [25]. The "number of recipients" which are in the To and Cc fields of the email differentiate between emails which should look strictly personal, e.g. communications from a bank (phishing) and those that pretend to be sent to several recipients, such as some kind of frauds or advertisement. The structure of links in the email text gives several information useful in determining the email goal. Portal redirections emails and advertisement generally show a high "Number of links", in the first case to redirect the user to different portal websites, in the second one to redirect the user to the website where she can buy the products. Generally, fraud emails do not report links except for "IP based links". These links are expressed through IP addresses, without reporting domain names, to reduce the

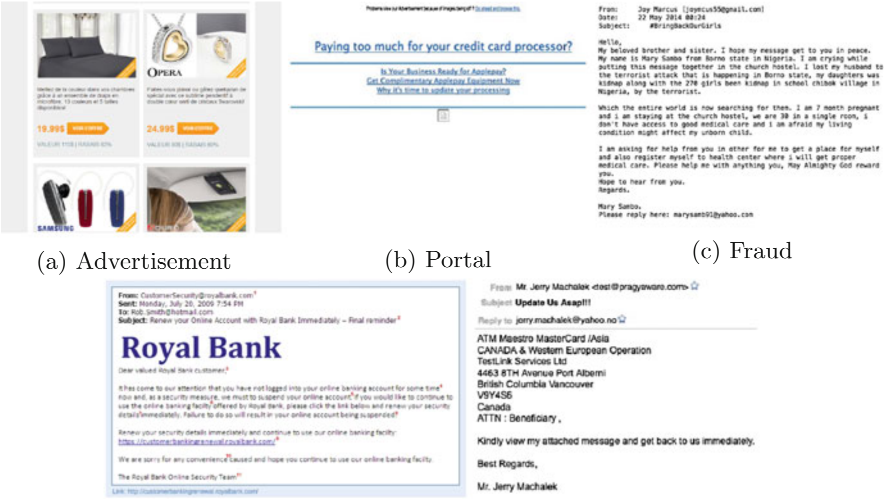
Table 1. Features extracted from each email.

Attribute	Description
`RecipientNumber`	Number of recipients addresses.
`NumberOfLinks`	Total links in email text.
`NumberOfIPBasedLinks`	Links shown as an IP address.
`NumberOfMismatchingLinks`	Links with a text different from the real link.
`NumberOfDomainsInLinks`	Number of domains in links.
`AvgDotsPerLink`	Average number of dots in link in text.
`NumberOfLinksWithAt`	Number of links containing "@".
`NumberOfLinksWithHex`	Number of links containing hex chars.
`NumberOfNonAsciiLinks`	Number of links with non-ASCII chars.
`IsHtml`	True if the mail contains html tags.
`EmailSize`	The email size, including attachments.
`Language`	Email language.
`AttachmentNumber`	Number of attachments.
`AttachmentSize`	Total size of email attachments.
`AttachmentType`	File type of the biggest attachment.
`WordsInSubject`	Number of words in subject.
`CharsInSubject`	Number of chars in subject.
`ReOrFwdInSubject`	True if subject contains "Re" or "Fwd".
`SubjectLanguage`	Language of the subject.
`NonAsciiCharsInSubject`	Number of non ASCII chars in subject.
`ImagesNumber`	Number of images in the email text.

likelihood of being tracked or to make the email text, generally discussing about secret money transaction, more legitimate. The "number of domains in links" represents the number of different domains globally found in all the links in the email text. Phishing and advertisement emails generally have just a single domain respectively of the website where to buy the advertised product and the website of the authority which the message pretends to be sent from. On the other hand portal redirection may contain several domains to redirect the reader to different portal websites. Moreover, links in portal redirection emails generally have a high "average number of dots in links" (i.e. sub-domains) and being dynamically generated are likely to contain hexadecimal or non ASCII - characters. Non ASCII characters in the links are also typical of some advertisement emails redirecting to foreign websites. It is worth noting that all these link-based features consider the real destination address, not the clickable text shown to the user. If the clickable text (hyper-link) shows an address ("click here"-like text is not considered) different from the destination address, the link is considered mismatching and counted through the feature "mismatching links". Phishing and portal redirection emails make extensive use of mismatching links to deceive the user. For a further insight, a sample for each class is shown in Fig. 1. Advertisement and phishing emails may appear like a web-page. In this case, the email contains HTML tags. On the other hand, fraud, malware and portal emails rarely are presented in HTML format. The size of an email is another important structural feature. Confidential trick and portal redirections generally are quite small in size, considering they are raw text. Advertisement, malware and some kind of phishing emails generally have a more complex structure, including images and/or attachments, which makes the message size to noticeably grow. "Attachment Number", "Attachment Size" and "Attachment Type" are structural features mainly used to distinguish between the attachment of malware emails and those of advertisement and phishing emails, which attach to the email images for a correct visualization. The "Number of Images" in an email determines the global look of the message. Images are typical of some advertisement emails and phishing ones. Finally three features are used for the analysis of the subject. For example, some advertisement emails use several one-character words or non ASCII characters in emails to deceive typical spam detection techniques based on keywords [22]. It is worth noting that rarely non ASCII characters are used in phishing emails, to make them look more legit. Moreover, some fraud and phishing emails send deceiving mail subject with the "Re": or "Fwd": keyword to look like part of a conversation triggered by the victim. Furthermore, some fraud emails are characterized by the difference between the email "Language" and the "Subject Language". Many scam emails are, in fact, translated through automatic software which ignore the subject, causing this language duality.

3.3 DWS Classification Workflow

After the email features have been extracted, the resulting feature vectors are given as input to the DWS classification workflow. This process aims at dividing the unclassified spam emails in campaign and label them through a classifier



(a) Advertisement

Paying too much for your credit card processor?

Is Your Business Ready for Applepay?
Get Complimentary Applepay Equipment Now
Why it's time to update your processing

(b) Portal

From: Joy Marcus [joymcus55@gmail.com]
Date: 22 May 2014 00:24
Subject: #BringBackOurGirls

Hello,
My beloved brother and sister. I hope my message get to you in peace. My name is Mary Sambo from Borno state in Nigeria. I am crying while putting this message together in the church hostel. I lost my husband to the terrorist attack that is happening in Borno state, my daughters was kidnap along with the 270 girls been kidnap in school chibok village in Nigeria, by the terrorist.

Which the entire world is now searching for them. I am 7 month pregnant and i am staying at the church hostel, we are 30 in a single room, i don't have access to good medical care and i am afraid my living condition might affect my unborn child.

I am asking for help from you in other for me to get a place for myself and also register myself to health center where i will get proper medical care. Please help me with anything you, May Almighty God reward you.
Hope to hear from you.
Regards.

Mary Sambo.
Please reply here: marysamb91@yahoo.com

(c) Fraud

From: CustomerSecurity@royalbank.com
Sent: Monday, July 20, 2009 7:54 PM
To: Rob.Smith@hotmail.com
Subject: Renew your Online Account with Royal Bank Immediately – Final reminder

Royal Bank

Dear valued Royal Bank customer,

It has come to our attention that you have not logged into your online banking account for some time now and, as a security measure, we must to suspend your online account. If you would like to continue to use the online banking facility offered by Royal Bank, please click the link below and renew your security details immediately. Failure to do so will result in your online account being suspended.

Renew your security details immediately and continue to use our online banking facility:
https://customerbankingrenewal.royalbank.com/

We are sorry for any convenience caused and hope you continue to use our online banking facility.

The Royal Bank Online Security Team

Link: http://customerbankingrenewal.royalbank.com/

(d) Phishing

From Mr. Jerry Machalek <dost@pragyaware.com>
Subject Update Us Asap!!!
Reply to jerry.machalek@yahoo.no

ATM Maestro MasterCard /Asia
CANADA & Western European Operation
TestLink Services Ltd
4463 8TH Avenue Port Alberni
British Columbia Vancouver
V9Y4S6
Canada
ATTN : Beneficiary ,

Kindly view my attached message and get back to us immediately.

Best Regards,

Mr. Jerry Machalek

(e) Malware

Fig. 1. Spam emails representing five categories of spammer goals.

trained on the fly. The workflow is depicted in Fig. 2. The main part of the workflow is aimed at generating a valid training set from the dataset of unclassified emails, applying hierarchical clustering algorithm to divide email in campaigns (step 1 in Fig. 2). The chosen algorithm, named *Categorical Clustering Tree* (CCTree) generates a tree-like structure (step 2) which is exploited to associate a campaign to each email coming from a small dataset of labeled emails. The campaign receives the label of the email associated to it (step 3). Thus, this set of campaigns is used as training set for a classifier (step 4), successively used to label all the remaining campaigns (steps 5 and 6).

In the following the six steps of the DWS workflow are described in detail.

Phase 1: Clustering Spam Emails into Campaigns. The first step performed by the DWS framework is to divide large amounts of unclassified spam emails (constituting the set $\mathcal{D}$) into smaller groups of similar messages (steps 1 and 2 in Fig. 2). Emails are clustered by structural similarity exploiting the CCTree algorithm.

CCTree Algorithm: CCTree is a categorical clustering algorithm, constructed iteratively through a decision tree-like structure. The root of the CCTree contains all the elements to be clustered. Each element is described through a set of *categorical* attributes, such as the *Language* of a message. Being categorical each attribute may assume a finite set of discrete values, constituting its domain. For example the attribute *Language* may have as domain: $\{English, French, Spanish\}$. At each step, a new level of the tree is generated by splitting the nodes of the previous levels, when they are not homogeneous enough. *Shannon Entropy* [23] is used both to define a homogeneity measure called *node purity*, and to select

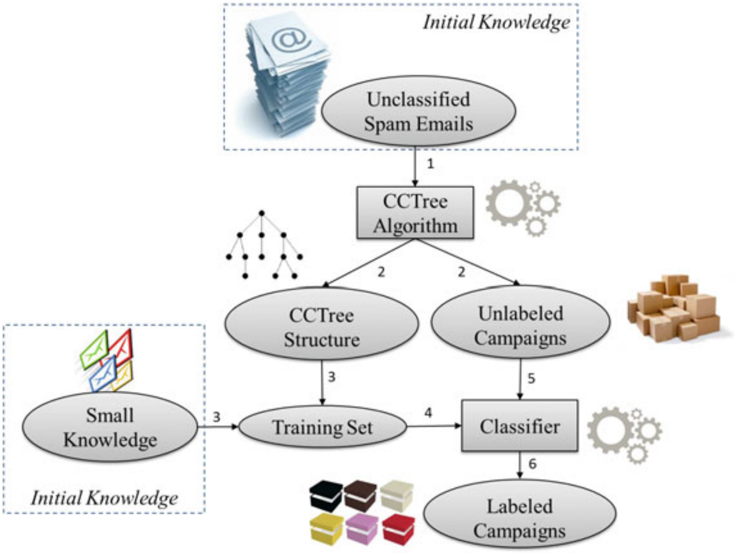


Fig. 2. DWS Workflow.

the attribute used to split a node. In particular non-leaf nodes are divided on the base of the attribute yielding the maximum value for Shannon entropy. The separation is represented through a branch for each possible outcome of the specific attribute. Each branch or edge extracted from parent node is labeled with the selected feature which directs data to the child node. Finally the leaves of the tree are the desired clusters. We refer the reader to [24] for details on the CCTree algorithm.

The CCTree algorithm has already proven to be effective in clustering spam emails into campaigns, as shown in [25]. When used on large dataset with the same set of features presented in Sect. 3.2, the CCTree algorithm generates highly homogeneous clusters, where all emails inside the same cluster belong to the same campaign. As other clustering algorithms which aim at maximizing the cluster homogeneity, the CCTree algorithm is likely to generate some clusters with only one element. Generally these clusters contain outlier emails, i.e. messages not belonging to any specific campaign. DWS discards these clusters not using them in the following steps of algorithm.

Phase 2: Training Set Generation. In order to label the campaigns, it is necessary to train a classifier to recognize emails coming from the five predefined spam classes (steps 3 and 4 in Fig. 2). To this end, it is necessary to provide to the classifier a good training set, which has to be representative of the reality in which the classifier has to operate. For this reason the training set will be extracted from the unclassified emails dataset $\mathcal{D}$ itself. More specifically, the CCTree structure generated in previous step is exploited to label a small number of generated spam campaigns. To this end, small number of campaigns are labeled with the use of a small set of labeled emails $\mathcal{C}$. This set contains a small number of manually selected spam emails, equally distributed in the five classes, all structurally different. These spam emails do not come from the $\mathcal{D}$ set. The emails in the $\mathcal{C}$ dataset have to be accurately chosen on the base of the email that investigator are interested in. For example, Italian police investigators interested in following a phishing case should put in the $\mathcal{C}$ dataset some emails with Italian text and bank names. After extracting the value of the features from the email in $\mathcal{C}$, they are fed one by one to the CCTree generated on $\mathcal{D}$. Following the CCTree structure each email c_i is eventually inserted in the campaign C_j (Fig. 3). Thus the campaign C_j is labeled with the class of c_i and all its emails are added to the training set.

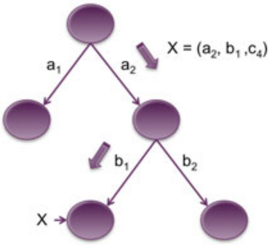


Fig. 3. Insert new instance X in a CCTree

If the same spam campaign is reached by two or more emails of different classes, the campaign is discarded and the emails are re-evaluated to be sent to other campaigns. It is worth noting that such an event is unlikely due to the high homogeneity of the clusters generated through CCTree. Furthermore, in the event that an email in $\mathcal{C}$ does not reach to any campaign, i.e. a specific attribute value of the email is not present in the CCTree, the email is inserted in the more similar campaign. To this end, the node purity of each campaign is calculated before and after the insertion of the email c_i. The email is thus assigned to the campaign in which the difference between the two purities, weighted by the number of elements, is lesser.

Phase 3: Labeling Spam Campaigns. Feeding the training set to the classifier, we are able to classify all remaining campaigns generated by the CCTree (steps 5 and 6 in Fig. 2). To this end, each campaign resulted from CCTree is given to the classifier. The classifier labels each email of received campaign on the base of spammer purpose. Under two conditions DWS considers a spam campaign as non classified. Firstly, it is possible that emails belonging to the same campaign receive different labels, e.g. phishing and portal redirection. In such a case, calling as "majority class" the label with more emails in the cluster, the campaign is considered non classified if the emails of the majority class amount to less than 90 % of all the emails in the campaign. The second condition is instead related to the *prediction error* reported by the classifier on each element of a campaign. The predicted error, computed as $1 - P(e_i \in \Omega_j)$, where $P(e_i \in \Omega_j)$ is the probability that the element e_i belongs to the class Ω_j, i.e. the label assigned to the element e_i. DWS framework considers a campaign as non classified, if the average predicted error is more than 30 %. If the non classified campaigns are a consistent percentage, it is possible to restart the classification process running the CCTree algorithm with tighter criteria for node purity.

4 Results

This section presents the experimental results of the DWS framework. First we discuss the classifier selection process, exploiting two small datasets of manually labeled spam emails. Afterward, we present the results for a real use case of the DWS framework on a recent dataset of spam emails.

4.1 Classifier Selection

In this first set of experiments we compare the performance of three different classifiers. To this end, two sets of real spam emails are provided to be used as training and test sets. These two datasets are extracted from emails collected by the untroubled honeypot [1] in February and January 2015. The emails have been manually analyzed and labeled for standard supervised learning classification and performance evaluation. The manual analysis and labeling process has been performed rigorously analyzing text and images, and following the links in each email. Only the emails for which the discovered class was *certain* have been inserted to the datasets. For a spam email, the label is certain if it matches the label description given in Sect. 3.1 and the label is verified through manual analysis. For example, Portal Redirection emails are *certainly* labeled if the links really redirect to a portal website. The first dataset, used as training set, is made of 160 spam emails, the second one, used as test set, is made of 80 emails.

Experiments have been run on all the classifiers offered by the WEKA library to classify categorical data. For the sake of brevity and clarity we only report the classifier with the better results for each classifier group. More specifically, the chosen classifiers are the K-Star from the *Lazy* group, the Random Tree Forest from the *Tree* group and the Bayes Network from the *NaiveBayes* group. Among these three classifiers, the best one has been used by the DWS framework.

Dataset Dimensioning: The process of manual analysis and labeling is time consuming. However, it is necessary to have a dataset well balanced, without duplicates and representative of the five classes, needed to correctly assess classifier performances. Given the complexity of manual analysis procedure, it is not possible to choose training and testing set of extremely large dimension. Thus, standard dimensioning techniques have been used, for both training and testing set. A general rule to assess the minimum size for a training set is to dimension it as six times the number of used features [27]. It is worth noting that the training set of 160 elements already matches this condition ($6 \times 21 < 160$). However, in multi-class problem, the dimension of data should provide well result in terms

Table 2. Classification results evaluated with K-fold validation on training set.

Algorithm	*K-star*	*RandomForest*	*BayesNet*
True Positive Rate	0.956	0.937	0.95
False Positive Rate	0.01	0.019	0.013
Area Under Curve	0.996	0.992	0.996

of sensitivity and specificity, i.e. true positive rate (TPR) and (1 - false positive rate (FPR)) respectively, when K-fold validation is applied [4]. This must be done keeping balanced the relative frequencies of data in various classes. As shown in the following, the provided testing set returns for K-fold validation a value of Receiver Operating Characteristic's Area Under Curve (ROC-AUC or AUC for short) higher than 90 % for all tested classifiers.

Concerning the test set, it is important the null intersection with the training set and the balanced relative frequencies of the various classes. In [4], the minimum size for a testing set to provide meaningful results, in a problem of classification with five classes, is estimated to be 75, which is smaller than the test set of 80 spam emails provided.

Classification Results: We report now the classification results for the three tested classifiers on the two aforementioned datasets. The first set has been used as training set for the classifiers. According to the methodology in [4], a first performance evaluation has been done through the K-fold (K=5) validation method, classifying the data for K times using each time $K - 1/K$ of the dataset as training set and the remaining elements as testing set. The used evaluation indexes are the True Positive Rate (TPR), False Positive Rate (FPR) and Receiver Operating Characteristic Area Under Curve (ROC-AUC or simply AUC). The AUC is defined in the interval $[0, 1]$ and measures the performance of a classifier at the variation of a threshold parameter T, proper of the classifier itself, according to the following formula:

$$AUC = \int_{-\infty}^{\infty} TPR(T) \cdot FPR'(T) dT$$

where $FPR' = 1 - FPR$. When the value of AUC is equal to 1, the classifier is considered "good" for the classification problem.

Table 2 reports TPR, FPR and AUC of the three classifiers, i.e. the number of correctly classified elements between the five classes for both the K-fold test on the first dataset (160 spam emails). As shown, all classifiers return an accuracy higher than 90 %. Afterward, the whole first dataset has been used to train the three classifiers, whilst the second dataset has been used as test set. Table 3 reports the detailed classification results, where classifiers are trained with training set (160 emails) and evaluated with test set (80 emails). The result is reported on the classes for TPR, FPR and AUC. For a further insight, we report in Fig. 4 the comparison of the ROC curves of the three classifiers for the five classes,

Table 3. Classification results evaluated on test set.

Algorithm	K-star			RandomForest			BayesNet		
Measure	*TPR*	*FPR*	*AUC*	*TPR*	*FPR*	*AUC*	*TPR*	*FPR*	*AUC*
Advertisement	1.000	0.031	0.998	1.000	0.000	1.000	1.000	0.031	0.967
Portal	0.786	0.000	0.996	0.786	0.016	0.985	0.929	0.000	0.998
Fraud	1.000	0.016	0.992	1.000	0.016	0.951	1.000	0.016	0.928
Malware	0.938	0.016	0.995	0.938	0.016	0.908	0.938	0.016	0.957
Phishing	0.947	0.017	0.977	0.947	0.051	0.963	0.842	0.017	0.907
Average	0.9342	0.016	0.9916	0.9342	0.019	0.9614	0.9418	0.016	0.9514

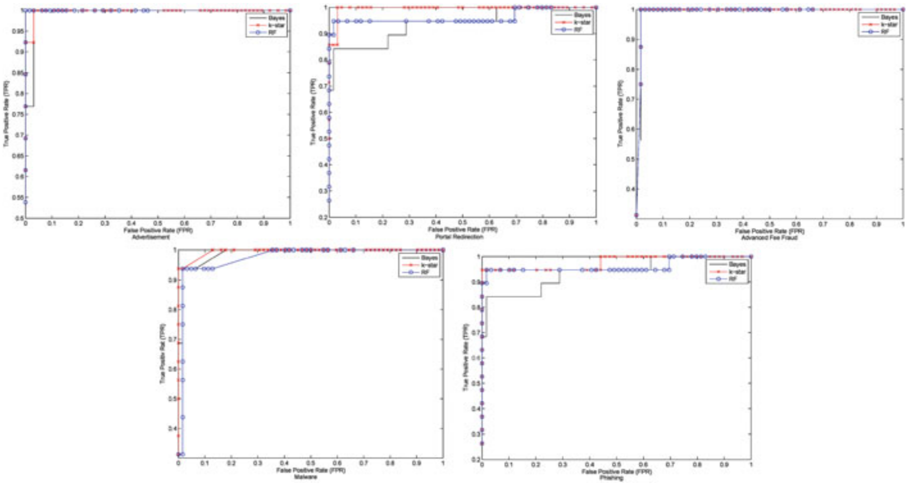


Fig. 4. ROC curves for the five classes labeling on test set.

measured on the test set. It is worth noting that in all cases the area under the ROC curve is close to 1, hence, in general the classifiers show good performances on the testing set for each class.

As can be observed in Table 2, on the average the K-star and Bayes Net classifiers give slightly better K-fold results. However, the K-star classifier yields the better results in terms of AUC in average, evaluated with test set (see Table 3). Therefore, K-star is the classifier used in the DWS framework.

4.2 DWS Application

The second set of experiments aims at assessing the capability of the framework to cluster and label large amounts of spam emails. To this end the DWS framework has been tested on set of 3230 recent spam emails. The spam emails have been extracted from the collection of the honeypot in [1], related to the first week of March 2015. The emails have been manually analyzed and labeled for performance analysis.

Phase 1: Clustering with CCTree: In the first step CCTree has been used to divide the emails in campaigns. The CCTree parameters have been chosen finding the optimal values for number of generated clusters and homogeneity, using the knee method described in [25]. 135 clusters have been generated of which 73 only contains one element. Generated clusters with a single element have not been considered. These emails are, in fact, outliers which do not belong to any spam campaign. The remaining 3149 emails divided in 62 clusters have been used for the following steps.

Phase 2: Training Set Generation: To generate the training set, we used a small dataset made of three representative emails for each of the five classes. These 15 emails have been manually selected from different datasets of real spam emails, including personal spam inbox of the authors. To facilitate the manual analysis of the classified spam emails, the 15 emails of the set $\mathcal{C}$ are written in English language. Each email has been assigned to one of the 62 spam campaigns, following the CCTree structure, as described in Sect. 3.3. The campaigns associated to each email are used as training set.

Table 4. Training set generated from small knowledge.

Class	Number of Emails	Number of Campaigns
Advert.	29	2
Portal	66	3
Fraud	113	3
Malware	27	1
Phishing	17	1
Total	252	10

The generated training set (Table 4) is composed of 252 emails, contained in 10 campaigns. It is worth noting that the 15 emails have not been added to the associated cluster after the CCTree classification, to not alter the decision on the following emails.

Phase 3: Labeling Spam Campaigns: After training the classifier with the generated training set, we label the remaining (52 out of 62) unlabeled spam campaigns of CCTree. The classification results are reported in Table 5. The table reports for each class the amount of campaigns and corresponding email classified correctly or incorrectly. Moreover, we report for the emails the statistics on TPR, FPR and Accuracy (i.e., the ratio of correctly classified elements). The global accuracy, (last row of the table) is of 97,82 %. However, we point out that, due to the conditions on predicted error reported in Subsect. 3.3, 8 campaigns out of 62, containing 344 spam emails are considered unclassified. For the sake of accuracy, considering these 8 campaigns as misclassified, the total accuracy for emails on the dataset is of 87,14 %. The accuracy is in line with previous works on classification emails into phishing and ham [6,8,11].

Table 5. DWS classification results for the labeled spam campaigns.

Class	Campaigns		Emails		TPR	FPR	Accuracy
	Correct	*Wrong*	*Correct*	*Wrong*			
Advert.	5	0	137	0	1	0	1
Portal	26	0	1331	0	1	0.03	0.9935
Fraud	10	2	1032	43	0.96	0.01	0.9788
Malware	3	0	31	0	1	1	1
Phishing	7	1	213	18	0.915	0	0.994
Total	51	3	2744	61	0.975	0.008	0.9782

Concerning the 8 non classified campaigns, 3 campaigns containing 68 spam emails were correctly labeled as portal. However, they are considered unclassified since the average predicted error is higher than 30 % in all the 3 campaigns. 4 campaigns containing 258 spam emails have been classified as phishing. 2 of them with 116 messages, were correctly identified but did not match the predicted error condition. The other 2 campaigns have been incorrectly classified as fraud. However, they are considered as unclassified due to high predicted error. The last campaign with 18 elements is in the advertisement class, but incorrectly classified as fraud, though the predicted error condition again is not matched. It is worth noting how the condition on predicted error is useful in increasing the overall accuracy on classified data.

From Table 5 it is possible to infer what a large portion of spam messages belongs to portal and fraud classes. Even if these preliminary results are related to a relatively small dataset, they are indicative of the current trend of spam emails distribution, which may provide to the spammer the greatest result with the smallest risk.

5 Conclusion and Future Directions

Spam emails constitute a constant threat to both companies and private users. Not only these emails are unwanted, occupy storage space and need time to be deleted, also they have become vectors of security threat and used to perform cybercrimes, such as phishing and malware distribution. In this paper, we have presented a framework, named DWS, for analysis of large amounts of spam emails collected through honeypots. We argue that DWS can provide a helpful tool for police and investigators in forensic analysis of spam emails. In fact DWS automatically clusters and classifies large amount of spam emails in labeled campaigns, to eventually help investigator to focus on campaigns for a specific cybercrime, filtering out the non-interesting spam emails. Moreover DWS is self learning, not requiring any preexistent knowledge of the dataset to analyze.

Preliminary tests performed on a first dataset of more than 3200 emails showed a good accuracy of the framework. More extensive experiments on larger datasets have been planned as future work, including performance analysis and an eventual refinement of the spam campaign labels, to include sub-groups such as *pharmacy-advertisement* or additional classes such as *propaganda*. To improve the effectiveness of DWS, we plan to detect and add more email representative features. Furthermore, application of dataset balancing techniques could be used to increase the quality of the generated training set.

References

1. Spam archive. http://untroubled.org/spam/
2. Federal trade commission (2009). http://www.consumer.ftc.gov
3. Almomani, A., Gupta, B.B., Atawneh, S., Meulenberg, A., Almomani, E.: A survey of phishing email filtering techniques. IEEE Commun. Surv. Tutorials **15**(4), 2070–2090 (2013)
4. Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., Popp, J.: Sample size planning for classification models. Anal. Chim. Acta **760**, 25–33 (2013)
5. Benczur, A.A., Csalogany, K., Sarlos, T., Uher, M.: Spamrank-fully automatic link spam detection work in progress. In: Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (2005)
6. Bergholz, A., PaaB, G., Reichartz, F., Strobel, S., Birlinghoven, S.: Improved phishing detection using model-based features. In: CEAS (2008)
7. Calais, P., Douglas, E.V.P., Dorgival, O.G., Wagner, M., Cristine, H., Klaus, S.: A campaign-based characterization of spamming strategies. In: CEAS (2008)
8. Chen, T.C., Stepan, T., Dick, S., Miller, J.: An anti-phishing system employing diffused information. ACM Trans. Inf. Syst. Secur. **16**(4), 16:1–16:31 (2014)
9. da Cruz Nassif, L., Hruschka, E.: Document clustering for forensic analysis: An approach for improving computer inspection. IEEE Trans. Inf. Forensics Secur. **8**(1), 46–54 (2013)
10. Dinh, S., Azeb, T., Fortin, F., Mouheb, D., Debbabi, M.: Spam campaign detection, analysis, and investigation. Digit. Invest. **12**(1), S12–S21 (2015)
11. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web, pp. 649–656. ACM (2007)
12. Gansterer, W.N., Pölz, D.: E-Mail classification for phishing defense. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 449–460. Springer, Heidelberg (2009)
13. Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., Zhao, B.: Detecting and characterizing social spam campaigns. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC 2010, pp. 35–47. ACM, New York (2010)
14. Hadjidj, R., Debbabi, M., Lounis, H., Iqbal, F., Szporer, A., Benredjem, D.: Towards an integrated e-mail forensic analysis framework. Digit. Invest. **5**(34), 124–137 (2009)
15. Henderson, L.: Crimes of Persuasion: Schemes, Scams, Frauds : how Con Artists Will Steal Your Savings and Inheritance Through Telemarketing Fraud Investment Schemes and Consumer Scams. Coyoto Ridge Press, Azilda (2003)
16. Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G.M., Paxson, V., Savage, S.: Spamalytics: An empirical analysis of spam marketing conversion. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS 2008, pp. 3–14. ACM, New York (2008)

17. Kanich, C., Weavery, N., McCoy, D., Halvorson, T., Kreibichy, C., Levchenko, K., Paxson, V., Voelker, G., Savage, S.: Show me the money: Characterizing spam-advertised revenue. In: Proceedings of the 20th USENIX Conference on Security, SEC 2011. USENIX Association, Berkeley (2011)
18. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., Savage, S.: Spamcraft: An inside look at spam campaign orchestration. In: Proceedings of the 2nd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, LEET 2009. USENIX Association, Berkeley (2009)
19. Labs, M.A.: Mcafee threats report: 2015 (2015). http://mcafee.com
20. Narang, S.: Cryptolocker alert: Millions in the uk targeted in mass spam campaign. (2013). http://www.symantec.com/connect/tr/blogs/cryptolocker-alert-millions-uk-targeted-mass-spam-campaign
21. Pathak, A., Qian, F., Hu, Y.C., Mao, Z.M., Ranjan, S.: Botnet spam campaigns can be long lasting: Evidence, implications, and analysis. SIGMETRICS Perform. Eval. Rev. **37**(1), 13–24 (2009)
22. Seewald, A.K.: An evaluation of naive bayes variants in content-based learning for spam filtering. Intell. Data Anal. **11**(5), 497–524 (2007)
23. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
24. Sheikhalishahi, M., Mejri, M., Tawbi, N.: Clustering spam emails into campaigns. In: 1st International Conference on Information Systems Security and Privacy (2015)
25. Sheikhalishahi, M., Saracino, A., Mejri, M., Tawbi, N., Martinelli, F.: Fast and effective clustering of spam emails based on structural similarity (2015). http://goo.gl/zlzHNl
26. Tillman, K.: How many internet connections are in the world? right. now (2013). http://blogs.cisco.com/news/cisco-connections-counter
27. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I-511–I-518 (2001)
28. Wang, D., Irani, D., Pu, C.: A study on evolution of email spam over fifteen years. In: 2013 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), pp. 1–10, October 2013
29. Wei, C., Sprague, A., Warner, G., Skjellum, A.: Mining spam email to identify common origins for forensic application. In: Proceedings of the 2008 ACM symposium on Applied computing, SAC 2008, pp. 1433–1437. ACM, New York (2008)