2 Years in the anti-phishing group of a large company

Luigi Gallo[a,b,*], Alessandro Maiello[a], Alessio Botta[b] and Giorgio Ventre[b]

[a]Cyber Security Lab TIM S.p.A., Via Reiss Romoli 274, 10148 Turin, Italy
[b]University of Napoli "Federico II", Via Claudio 21, 80125 Naples, Italy

ARTICLE INFO

Keywords:
Cybersecurity
Spam email
Phishing email
Machine Learning
Security Awareness

Abstract

The email threat landscape is constantly evolving and hence difficult to counteract even by carrier-grade spam filters. Dangerous spam emails may thus reach the users and then result in damaging attacks spreading through the corporate network. This paper describes a collaborative approach for early detection of malicious spam emails and its application in the context of large companies. By the joint effort of the employees and the security analysts during the last two years, a large dataset of potentially malicious spam emails has been collected with each email being labeled as critical or irrelevant spam. By analyzing the main distinguishing characteristics of dangerous emails, a set of both traditional and novel features was identified and then tested and optimized by applying common supervised machine learning classifiers. The obtained massive experimental results show that Support Vector Machine and Random Forest classifiers achieve the best performance, with the optimized feature set of only 36 features achieving 91.6% Recall and 95.2% Precision. These results, confirmed by a large empirical experiment conducted on 40,000+ company employees, led to the re-engineering of the email threat management process to ensure a high level of security in the company, as well as an increased security awareness of all company employees.

1. Introduction and motivation

Email is still one of the most used channels for making cyber attacks, thus exposing companies to frequent attempts at security breaches. This collides with the importance and the widespread use emails have in everyday work life. The Internet Security Threat Report by Symantec [40] states that spam level is on the rise, as it has been every year since 2015, with 55 percent of emails received in 2018 being categorized as spam. Within the category of spam emails fall either innocuous attempts to market and sell products, or messages that contain significant threats, such as phishing attempts to steal credentials or malware delivery for espionage and theft of sensitive data. In 2016, the FBI raised the alarm over this important problem as this kind of attacks increased in number and malignance [14], and in 2019 a monetary loss in USA of about U.S. $1.8 billion has been estimated with just a subset of attacks feasible with emails [22]. The European Cybercrime Centre (EUROPOL) claims that email attacks are used as the primary infection vector in 78% of cyber espionage incidents [13]. As a consequence, companies SOCs (Security Operation Center) and CERTs (computer Emergency Response Team) need large teams of security analysts, typically named Anti-Phishing groups[1], monitoring this specific type of threats. Unfortunately, the problem of email security threats is more and more challenging because of the really huge number of spam emails across the network every day, among which malicious emails are mixed. This makes the work of analysts a real "needle in a haystack" search.

Techniques for building effective spam emails vary from using advanced strategies to escape spam filters to sophisticated social engineering techniques to trick people. While spam filters work well as a countermeasure to some troubles caused by spam such as network overload, loss of time and productivity, irritation and discomfort, they still lack to solve the problem of email as an attack vector. The spectrum of email attacks is varied, ranging from the legacy ones concerning purely technical aspects, still feasible due to SMTP protocol and configurations vulnerabilities [8, 38], to the more sophisticated socio-technical methods made possible by modern machine learning and social engineering techniques. The aim of the attackers in these scenarios is usually to spread malware, steal authentication credentials, or commit financial fraud. Depending on the goal, the attacks can be classified as: malware propagation, (spear) phishing, (CEO) fraud, and scam. The most dangerous ones are 'tailored' against specific organizations or groups of people, and differ significantly from generalist attacks. Employees of big companies are normally trained not to be fooled by email attack attempts, but they actually happen to be for various reasons, including the fact that large companies have employees of all age ranges, with various education degrees and different technology expertise and that the lack of concentration of people to recognize phishing attacks can be crucial [31]. Every single employee can represent a point of entry for spammers and attackers. In the context of a company with tens of thousands of employees, millions of emails are received every day, 55% of which are unsolicited. While 95% of these unsolicited mails are blocked by spam filters, the remaining 5% (about 25 thousand every day) is still a potentially dangerous amount of emails, too large to monitor and control.

In this work a spam email is simply an unwanted email, and we are not interested to most of them. We rather want to understand if any of them created or has the potential to

*Corresponding author
luigi.gallo3@unina.it (L. Gallo);
alessandro.maiello@telecomitalia.it (A. Maiello); a.botta@unina.it (A. Botta); giorgio.ventre@unina.it (G. Ventre)
ORCID(s):

[1]Due to a common abuse of notation, "Phishing" means both a certain subclass of attacks and in some cases also the whole set of attacks via email. Hence the name "Anti-phishing group".

create a *security incident*: "a security-relevant system event in which the system's security policy is disobeyed or otherwise breached" [34]. When an employee browses a malicious website or downloads a malicious email attachment (e.g. ransomware, trojan etc.), a security incident can occur. Security incidents can have different impacts, depending on the number and role of the employees involved, the nature of the threat, and how effective the security systems (i.e. corporate antivirus) are against them. We call *critical spam* the emails that caused or have the potential to cause a security incident. Since the number of unsolicited emails received by large companies is indeed huge and constantly increasing, their manual analysis is not feasible. An automatic mechanism to detect critical spam that bypass common antispam filters becomes therefore necessary.

This paper[2] reports on the activities performed during the last 2 years in the anti-phishing group of TIM (the biggest Italian telco) that comprise: construction of a system for real spam emails collection; labeling of this data; study of the characteristics of critical spam emails; design, development, and deployment of an automated system for critical spam detection based on machine learning techniques; conduction of a data-driven awareness campaign based on insights derived from the previous activities. On average, 30 million emails per month reach the 100,000 mailboxes of the company employees and external collaborators, most of which are filtered by the spam filter. The starting idea was to collect over the years spam emails that pass the spam filter and are reported by users as unwanted, also storing the information produced by the SOC analyst about the possible security incident occurred. To this aim, a collaborative framework for reporting and monitoring of such spam emails has been designed with the by-design goal of collecting data (Section 3). This framework supports the work of security analysts, allowing them to annotate the results of their analysis directly on the data, thus obtaining a solid ground truth (Section 3.1). With this approach, a labeled dataset of 22,000+ unique emails reported in the last 2 years has been collected. Several legacy and novel features have been extracted from the samples of the dataset (Section 3.2). Various machine learning algorithms have been used to perform a binary classification: critical or not relevant spam. The main classification algorithms based on machine learning have been tested and compared in order to find the best one, including Gaussian Naive Bayes, Decision Trees, Support Vector Machines, Neural Networks and Random Forest (Section 4).

Results show that Random Forest achieves the best performance, with 95.2% Precision, 91.6% Recall, and 93.3% F-measure. The impact of different feature sets on such performance has been analyzed (Section 4.3). Results show that the best performance can be obtained with a selection of the best 36 features out of 79. Since the extraction cost of a feature is shared among the ones of the same type, they have been grouped into sets referred as *feature fields*. Performance has also been evaluated while varying the number of feature fields: by using 4 out of the 8 feature fields, which results in a significant cost reduction, performance degrades by (only) 5%. The feature ranking work also provides an important explanation on how critical emails are built and can be detected. This knowledge led to the design of a week-long awareness campaign, which involved all 40,000+ employees of the company, including top managers and executives (Section 5). This large social experiment confirms that our system correctly models the phishing phenomenon and, together with well-trained people, represents a global defence ecosystem robust to the majority of email attacks.

[2] Preliminary results within this framework have been presented in [16]

2. Related Work

The research problem of spam, and phishing in particular, has been covered for several years in literature and has led to multiple works. Its importance has been recognized worldwide for many years, but because of its complexity, spam detection and filtering technologies are not yet able to completely solve the problem, especially with regard to the security aspects. A good overview of the topic and the application of machine learning in the field of spam filtering is provided by *Blanzieri et al.* [6]. According to *Blanzieri et al.*, spam filtering, i.e. the automatic classification of messages into spam or legitimate emails, is one of the most popular and effective solutions to the problem of spam. This classification can be performed through machine learning techniques, but, again according to *Blanzieri et al.*, several fundamental problems must be taken into account: the need to find a labeled dataset to perform supervised machine learning; the trade-off between false positives and false negatives; the need to adapt to the continuous evolution of spammer's attack and evasion techniques. The first problem can be reduced by using semi-supervised machine learning techniques as done by *Chan et al.* [7] and *Dai et al.* [12], who show the classification performance of four different algorithms, the best of which is SVM. As for the second problem, having many false negatives can be tedious and dangerous as it forces users to check many spam emails. At the same time, having too many false positives could lead to the loss of important information. *Michelakis et al.* [30] therefore proposes to let the user choose the costs related to the two types of errors, according to the different needs.

Among the open research problems in email spam filtering mentioned by *Dada et al.*[11], we highlight the reactivity of spammers in adapting to new defense techniques. Therefore attempts to predict and prevent spammers' countermeasures are many. These attempts, as far as machine learning is concerned, consist in constantly looking for new features that allow the classification algorithms to identify spam emails. As a consequence, the various approaches in literature differ for the type of features adopted, extracted from the header, body, and text content of the email. *Stringhini et al.* [39] introduce a new approach that filters spam according to the way messages are sent by spammers. It focuses on the delivery mechanism and analyzes the communication at the SMTP protocol level. It does this in two ways: firstly, it studies how different clients implement SMTP communica-

tion and leverages this information to identify botnets; secondly, as spammers can use feedback from the mail server to identify whether an address exists or not, it sends incorrect feedback and prevent spam emails from being sent correctly. *Gansterer et al.* [17] propose three new groups of email features in addition to the more traditional ones: six offline, eight online, and two independent features. The offline features can be extracted locally in an efficient way, and are therefore usable in all those contexts where the email flow is high. Online features have higher extraction costs because they are based on Internet connections and can therefore cause bottlenecks to the performance of the classifier. *Basavaraju et al.* [28] focus exclusively on the text of the message, calculating for each word a metric that estimates its importance. This is to obtain the TF-IDF (*term frequency–inverse document frequency*) model, which works particularly well with deep learning approaches.

All the previously discussed papers attempt to classify and identify spam by focusing only on the characteristics of the attack medium, without taking into account social engineering aspects. According to *Allodi et al.* [2] the main problem lies in the fact that technology is often unable to capture the human dimension, which plays a fundamental role in social engineering attacks. This is mainly due to the lack of a clear formalization of the vulnerabilities, characteristics, and processes of social engineering attacks, which could provide the means to devise more effective countermeasures. Studies on the correlation between the effectiveness of attacks and the application of Cialdini's "persuasion principles" in spam emails [9] have recently been presented. *Van Der Heijden et al.* [19] show that it is possible to predict the effectiveness of phishing attacks through a quantitative estimate of the cognitive vulnerabilities adopted in these attacks. This allows to estimate the dangerousness of the attacks in an automatic way, in order to be able to respond with higher priority to the related security incidents. *Van Der Heijden et al.* also show that there is correlation between different cognitive vulnerabilities and the effectiveness of phishing, but it depends very much on the specific application domain. Organizations can therefore use this information to create effective training campaigns to raise awareness of the dangers of social engineering among their employees. *Cidon et al.* [10], instead, studied one of the most dangerous spear-phishing attacks in the business: business email compromise (BEC). These attacks are very difficult to detect by security systems since they do not contain malicious attachments or links and are very tailored to the recipient. *Cidon et al.* implement a supervised machine learning system capable of identifying BEC attacks with a low false positive rate and high precision.

In contrast with existing literature, this work introduces a framework that focuses on the several possible types of email attacks, incorporating multiple phishing countermeasures [1] coming from different domains (e.g. human aspects, URL blacklisting, protocol analysis etc.). Such framework is used to guide and optimise the work of anti-phishing analysts in enterprise settings. The existing studies, instead, present countermeasures coming from a single domain or focus on a specific type of attack. Our models are trained and evaluated on the information that a security analyst generally has at his disposal when analysing a security incident generated by an email, available when the email is reported as unsolicited. Therefore, the input to our models is not a generic email but the reporting of an unsolicited email. This is a significant aspect because of the inherent difficulty in distinguishing an unwanted email from an actual cyber attack. The novelty lies in this triage task and in the estimating of the probability of a phishing attack to succeed, considering both its technical aspects towards the systems and cognitive aspects towards the victims. Our approach uses traditional supervised machine learning algorithms, but with novel objectives and novel input information. Despite the strengths of supervised learning, it is often impossible to apply due to the absence of a reliable labelled dataset [20]. With the big effort of users and analysts of the company, an extensive and reliable ground truth has been collected in two years. This dataset has been used to build automatic classifiers and to achieve new contributions and significant results about the effectiveness of the various existing countermeasures, the characteristics of successful phishing attacks, and the cognitive vulnerabilities of humans about phishing.

3. Collaborative Framework

This section presents the "life cycle" of a spam email in our company. In the scenario of this work, the defence against attempts at spam fraud follows a collaborative approach: in addition to commercial email filtering systems, there is a system of collection of reports that allows the computer emergency response team (as defined in [34]) to protect the affected users thanks to the recognition of a threat by a more aware/expert user. This distributed approach is important for detecting security incidents that would otherwise go unnoticed. It is just as important, as a prevention strategy, to conduct periodic awareness campaigns to train users to recognize email-based cyber attacks, in order to increase their awareness of risk and to educate them in reporting such attempts to breach the company's security. As explained in [23], in fact, user training plays an important role in reducing their vulnerability to phishing. *Burda et al.* [33], on the other hand, explain how the users' reports constitute an important, and sometimes underestimated, resource that can provide an indicator of the dangerousness of a suspicious email, thus helping to speed up response and mitigation actions. Moreover, according to *Burda et al.* [33], the most experienced users able to identify the attempts of fraud by email, rarely report the email to the appropriate departments, thus denying the analysts an important help to detect an eventual security incident.

Typically, when a security incident occurs, one or more of the following recovery actions are undertaken, in increasing order of relevance:

- Sending notification to all users involved about malicious email detection;

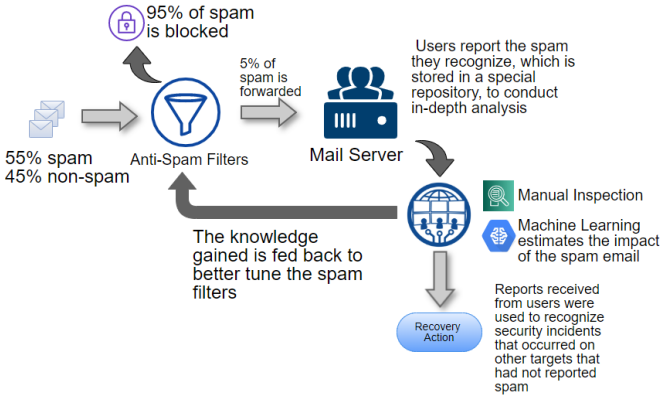


Figure 1: Ecosystem of spam defense

- Adding filters in the navigation proxies to block navigation or downloading from malicious or otherwise unknown sources;
- Rehabilitating of nodes and networks compromised by any malware. Resetting of accounts and credentials that may have been violated;
- Technically analyzing in-depth attachments and links, in order to get a thorough understanding of their risk and adequately protect affected users;
- Investigating on perpetrators and taking possible legal actions.

The purpose of our collaborative framework is both to recognize and resolve security incidents that have occurred, and to intercept them before they occur. The prediction of which spam emails will actually generate a security incident can leverage machine learning techniques. Estimating and assigning an accurate level of risk to each reported email is extremely important. The number of incoming reports is huge. Prioritizing their analysis allows security experts to deepen the investigation exclusively on the most relevant ones. The ecosystem illustrated in Figure 1 has allowed to collect over time the spam messages that reached users and to memorize which of them has led the recipients to download an attachment or to browse a link. This information is recorded directly in the tracking logs of the company navigation proxy, and can be requested by analysts only in the case of a clear possibility of a security incident. The analysis of these messages let us acquire a deep knowledge of the main features of most critical spam emails. In principle, the estimation of the risk score of the email could be made upstream for all emails with unsupervised approaches, even before a user reports it. However the process should not be time consuming, given the huge amount of emails to analyze. For this reason, the impact of feature reduction on classification performance has also been studied (Section 4.3).

3.1. Data collection and ground truth construction

Our data collection system was started in early 2018. Since then, whenever an employee receives an unwanted email and decides to report it to the security department, it is stored in our archive. All emails in the dataset are by definition spam emails. A large amount of additional security-relevant information about each element of the email is automatically computed and stored together with it. This is the typical information that the SOCs of all companies are supplied with in order to correctly manage this type of attack. For this reason the dataset is highly specialized, with information coming directly from the field and promptly made available to analysts. Very often such information is available only through the purchase of third party services such as reputation services, sandboxes, threat intelligence feeds, blacklisting services, etc. Based on this information, a specific group of security analysts composing the anti-phishing group, day by day checks if a security incident is generated by these incoming spam emails. Due to the enormous amount of reports that arrive every day, it is not feasible to perform a thorough security check for each of them, mainly because most of them represent simple noise. This first triage is very important, because it allows an initial filtering that would prioritize spam emails that need to be checked immediately; however, this distinction task cannot be delegated to the simple recipients of the email because it would require a strong security expertise to carry it out. The security check mentioned above is an extensive series of checks, such as the assessment of how widespread this email is in the mailboxes of all the other users, if someone has clicked on a malicious link, if he has downloaded a malicious attachment, if some workstation has been infected, if credentials have been violated, etc. Further on, these checks can also include the log analysis of navigation proxy, user agents installed on workstations, and sometimes also interviews to the recipients of the email. Finally, all the evidence found by analysts is also stored together with the spam report: whether the security incident was detected by analysts or not, and the (possible) remediation actions taken by analysts. This allows a manual classification and labeling of data performed by analysts as a result of their daily work with the data collection system. The email reports in the dataset are categorized in two possible classes:

- **Critical spam - Label 1, Positive**: spam emails that have created a security incident or at least required a defensive action to prevent future infections;
- **Not relevant spam - Label 0, Negative**: spam emails with low or no degree of danger, and did not require any recovery action.

Formally, a dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$ has been built where each sample x_i is characterized by a vector of m feature values $< f_0, ..., f_{m-1} >$ and has an associated class $y_i \in \{0, 1\}$. Several hundreds of reports from our 100,000 users are received every day. Many of these are duplications, because these types of attacks are often executed in large campaigns that target many recipients with the same email. Excluding the duplicates and not considering the many reports not processed due to their huge number, at the time of publication of this work the dataset contains a total of

21,932 distinct samples: 3,931 were labeled as Critical/Positive, 18,001 as Not Relevant/Negative. This dataset has been used to perform supervised machine learning and obtain a classifier that allows immediate recognition of the threats contained in the mails.

3.1.1. ANNOTATION CONSISTENCY EVALUATION

The scarcity of labeled dataset is a known problem in cyber security contexts, amplified by the difficulty that even a human may have in manually labeling a dataset. In other contexts, such as the recognition of a dog or a cat in a photo, for example, the labeling task is much simpler compared to the security analyst labeling job, which requires strong prior knowledge. Despite only hard critical and healthcare environment require an almost perfect level of reliability, we believe that a thorough analysis of manual labeling reliability is always necessary to ensure it does not undermine the correctness of the experiments. Due to the nature of the classification problem set, in fact, the human verdict on the positivity or not of a sample may be ambiguous, or may differ between the various expert analysts involved in the manual labeling. This is why, before starting our studies, we decided to evaluate the annotation consistency and the inter-rater reliability of our manual labeling. These are typical problems of manually labeled datasets, well addressed by M.L. McHugh [29].

The main metrics used to measure the consistency of labeling and the inter-rater reliability are: the percent of agreement and the Kappa statistic. The first, more traditional one is calculated by the number of agreement observations divided by the total number of observations. Its key limitation is that it does not take into account the possibility that raters guessed on the labels; actually it is a remote possibility in our case, given the experience of the raters, but it is known that all humans can make mistakes. The Kappa statistic was designed to take into account the possibility of guessing, nevertheless it has other kinds of limitations. It is calculated through the following:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{1}$$

where $Pr(a)$ is the probability of agreement and $Pr(e)$ is the chance probability of agreement (function of row and column marginals):

$$Pr(e) = \frac{(\frac{cm^1 xrm^1}{n}) + (\frac{cm^2 xrm^2}{n})}{n} \tag{2}$$

Both percent agreement and Kappa statistic have strengths and limitations, but in brief it is possible to assume that the former is an upper bound and the latter is a lower bound of the annotation consistency. To compute these two metrics, two analysts (the most and the less experienced ones) have been asked to work on the same subset of spam reports (composed of $n = 263$ elements) in completely separate sessions, in order to see if they agreed on labeling. The main point to evaluate is the very first look that is given to the report, when the analyst decides whether the email has the potential to create a security incident. The following steps to ascertain whether a security incident has occurred are guided by more standardised procedures, as defined for example by ISO/IEC 27001 standard, based on the evidence found without any discretionary approach.

		Analyst 1		Row Marginals	
		Positive	Negative		
Analyst 2	Positive	42	10	52	rm^1
	Negative	4	207	211	rm^2
Column Marginals		46	217	263	
		cm^1	cm^2		n

Table 1
Data for Kappa calculation

Phishing words				
account	security	user	verify	service
valid	required	credentials	attention	request
suspended	company	bank	deposit	post
Scamming words				
$	€	£	customer	prize
donate	buy	pay	congratulation	death
please	response	dollar	looking	urgent
warning	win	offer	risk	money
transaction	sex	nude		

Table 2
words considered deceiving (for scam and phishing purposes)

The results of this experiment are shown in the Table 1. The second analyst, the less experienced one, considers more spam emails as dangerous compared to the more experienced one; he probably feels less confident and prefers to get false alarms instead of ignoring potential incidents. However, the two analysts are 94.67% in agreement on the responses concerning 263 observations. The Kappa statistic computed with (1), applying the (2), is 83.3%. Therefore, the labeling of our dataset can be considered *strongly reliable* and can be used to train machine learning models to be deployed in this operating environment.

3.2. Feature set design

Starting from the raw information automatically collected when a spam email is reported, the set of features to be extracted and used as input to learning models has been designed. The full set of features extracted from the samples is listed in Table 3 and comprises 79 features. The features are grouped by the nature of the information from which they are extracted or the reason why they were thought to be good at discriminating between the two classes.

Each group of features, referred as *feature field*, is described in depth in the following.

1. **General.** General information, mostly extracted from

[3] Features calculated twice: first on the text extracted from the email content and then on the text extracted with an Optical Character Recognition from the email displayed. Regarding the latter, the feature name used in the paper is the same followed by the "_clean_text" suffix. The field of these features is referred to as "content_view".

[4] Features calculated twice. The alternative version of the features take the "_d1" suffix in the name. Refer to appendix A for further information.

Field	Feature	Description
General	is_html	if it is an html mail
	n_smtp_blackist	the number of smtp servers traversed in the blacklists
	email_size	the size of the email
	n_recipients	the number of recipients
	n_hops	the number of SMTP hops
	is_IT	if the email comes from Italy
	is_EU	if the email comes from Europe
	is_NA	if the email comes from North America
	is_SA	if the email comes from South America
	is_RU	if the email comes from Russia
	is_AS	if the email comes from Asia
	is_AF	if the email comes from Africa
	is_OC	if the email comes from Oceania
Content[3]	language[3]	the language of the mail
	voc_rate[3]	the rate of words of the content in the vocabulary
	vdb_rate[3]	the rate of words of the content within the basic vocabulary
	vdb_agg_rate[3]	the rate of adjectives within the content
	vdb_v_rate[3]	the rate of verbs within the content
	vdb_s_rate[3]	the rate of nouns within the content
	vdb_art_rate[3]	the rate of articles within the content
	gulpease_index[3]	readability index (Italian - Gulpease index [27], English - Flesch formula [15])
	n_words_content[3]	number of words in the content
	n_disguisy[3]	number of disguised words in the entire email (content, subject, address)
	n_phishy[3]	number of deceiving words, related to phishing, in the content and subject
	n_scammy[3]	number of deceiving words, related to scamming, in the content and subject
View	screenshot_width	the width of the email as it is displayed to the recipient
	screenshot_heigth	the heigth of the email as it is displayed to the recipient
	n_images	number of images
	n_images_links	number of images as links
	hidden_text[4]	percentage of text in the content not displayed to the recipient
	hidden_text_words[4]	number of words in the content not displayed to the recipient
	hidden_text_chars[4]	number of characters in the content not displayed to the recipient
Subject	n_words_subject	number of words in the subject
	n_char_subject	number of characters in the subject
	is_non_ASCII_subject	if the object contains non-ASCII characters
	is_re_fwd_subject	if the email is replied or forwarded
Links	n_links	number of links
	n_domains	number of link domains
	vt_l_rate	rate of links considered malicious by at least one engine of VirusTotal
	vt_l_maximum	maximum number of VirusTotal engines that consider a link as malicious
	vt_l_positives	number of links considered malicious by at least one engine of VirusTotal
	vt_l_clean	number of links not considered malicious by all engines VirusTotal
	vt_l_unknown	number of unknown links to VirusTotal
Attachments	n_attachments	number of attachments
	n_image_attachments	number of image type attachments
	n_application_attachments	number of application type attachments
	n_message_attachments	number of message type attachments
	n_text_attachments	number of text type attachments
	n_video_attachments	number of video type attachments
	attachments_size	average size of attachments
	vt_a_rate	rate of attachments considered malicious by at least one engine of VirusTotal
	vt_a_maximum	maximum number of VirusTotal engines that consider an attachment as malicious
	vt_a_positives	number of attachments considered malicious by at least one engine of VirusTotal
	vt_a_clean	number of attachments not considered malicious by all VirusTotal engines
	vt_a_vulnerable	number of attachments considered malicious by VirusTotal engines not including corporate antivirus
	vt_a_partial	number of attachments considered partially malicious by VirusTotal engines not including corporate antivirus
	vt_a_protected	number of attachments considered malicious by VirusTotal engines including corporate antivirus
	vt_a_unknown	number of unknown attachments to VirusTotal
Other	n_tip	number of entities in TIP
	n_tip_a	number of attachments in TIP
	n_tip_l	number of links in TIP
	n_vips	the number of vips among the recipients
	n_medium_vips	the number of managers among the recipients
	n_high_vips	the number of top managers among the recipients

Table 3
Features extracted from the raw data

the smtp headers: if any smtp server is blacklisted, size of the mail, number of recipients etc, plus all those features that give us information about the email's origin and destination.
Rationale: These features are not expected to be very discriminating on their own, but they might be in correlation with others. Moreover, the dangerousness evaluation of an email based on its origin and SMTP path is a typical analysis made by anti-spam filters, and it may be useful in our classification task as well.

2. **Content.** Features extracted from the text in the content of the email: language, number of words, number of deceiving words, number of disguised words, readability indexes, simplicity and correctness of the text etc. As for "deceiving words", previous studies [32] show that the words listed in Table 2 are those most

used to capture the attention of the scammed target. It has been manually verified that this is also true in our dataset. In addition, "disguised word" refers to a word which has an edit distance of 1 from the name of the company, the names of its subsidiaries and the names of its main partners. All the **Content** features have been calculated also on the text extracted with an Optical Character Recognition tool, generating the **Content_View** features (as described in the next feature field).
Rationale: The actual message carried by an email is the content, which is also one of the main elements to analyze in order to detect the presence of an attack and estimate its effectiveness. It is the main means used by attackers to satisfy the first condition we deem necessary for the attack to succeed: the recipient must be subjugated. These features may allow classifiers to distinguish emails that are immediately trashed by the recipients from those that induce a mistake to the attacker's advantage. For example, the search for disguised words is useful because very often addresses or domains similar to those normally used by the company are crafted to deceive employees.

3. **View.** Features extracted from the screenshot of the email as it is displayed to the recipient: height and width of the screenshot, number of images, amount of text within the content but not read by the recipient etc.
Rationale: These features have been selected to include in our analysis also the cognitive visual perception that the recipient has when opening the email. Moreover, very often spammers use html/css-based tricks to inject text into the content of the email, dirtying all the analysis indicators carried out on the text by automatic systems, but avoiding that this is read by the recipient (e.g. text of the same background colour, text with "display: none" option etc.). Several features have been extracted with an Optical Character Recognition (OCR) tool, with a twofold objective: to detect differences between text contained in the email and text actually displayed, as an indicator of malicious behavior, but also to calculate the content features on the text actually read by the recipient (generating the **Content_View** features). The extraction process of these features is described in detail in the appendix A.
4. **Subject.** Features extracted from the subject of the email: number of words, number of characters, if there are non-ASCII characters, if the email is forwarded or answered.
Rationale: The subject line is the first thing the recipient reads of an email, and it is known [37, 42] to have a great importance for the communicative effectiveness of the message carried. For this reason the subject line is also expected to have a great value on how much a recipient can be fooled into believing in a message depending on the characteristics of the subject.
5. **Links.** Features about the links in the email: number, number of link domains, information from URL analysis service, etc.
Rationale: Links can be the carriers of malicious content of an email, they must be carefully analyzed to quantify how much the email meets the second condition we deem necessary for the attack to succeed: the payload of the attack must not be trivial. In this perspective it is very useful to rely on information from online link and domain reputation systems, typically available to the SOCs of companies. VirusTotal has been used for this analysis[5].
6. **Attachments.** Features about the email attachments: number, type, size, information from sandboxes and antivirus, etc.
Rationale: As with links, attachments can be the carriers of the malicious content of an email, they must be carefully analyzed for the same reasons. The information coming from sandbox and antivirus systems can help, especially taking into account the specific systems used by the company.
7. **Other.** Other types of information not in the previous fields: number of malicious entities known thanks to Threat Intelligence activities, role in the company of recipients, etc.
Rationale: Other information closely related to the company also can contribute to the identification of emails more relevant than the others: the strategic importance of the role of recipient or reporter in the company is very useful in assessing the risk that would arise in case of compromise. For example: if a deceived employee answers an email with information about personal agenda or meetings, it may not be considered a security incident. In the case of a manager, because of the sensitivity of the information he/she is dealing with, it certainly is. In addition, information from the Threat Intelligence Platform (TIP), which is an internal platform managed by the company's security department that aims to collect and share IoCs, has been included in this field. [6]

The designed feature set, comprising legacy and novel features, includes information considered in previous works, but now properly turned into ML features, as well as information available to SOCs of companies but never used for these purposes. We believe this set of features represents an important contribution to the field.

[5] VirusTotal is an online malware and url analysis service https://www.virustotal.com/gui/home/upload

[6] Indicator of compromise (IoC): in computer forensics is an artifact (e.g. antiviral signatures, malicious domains or IP Addresses etc.) observed on a network or in an operating system that, with high confidence, indicates a computer intrusion [35]. In this context IoCs are antiviral signatures, malicious IP Addresses, MD5 hashes that uniquely identify a malicious file, URLs and/or domain names from which an attack has been carried or to which a malware connects once activated.

4. Experimental analysis and results

The available dataset has been divided into two parts: training set (85%) and test set (15%). The training set was used in the analyses related to the choice and optimization of model, its hyperparameters, weights, threshold, and features. These results have been validated through 10-fold cross-validation. The test set was used to evaluate the actual performance of the models properly tuned and optimized in the previous phase.

4.1. Selecting supervised Machine Learning models

Several Machine Learning models have been trained to perform the binary classification explained above, with the aim of choosing the best ones and conducting further in-depth experiments on them. Scikit-learn and the following ML-based algorithms have been used: Nearest Neighbors, Linear Support Vector Machine (Linear SVM), Radial Basis Function Support Vector Machine (RBF SVM), Decision Tree, Random Forest, Adaptive Boosting (AdaBoost), Naive Bayes, Quadratic Discriminant Analysis (QDA), Multi-layer Perceptron Neural Network (MLP Neural Net). These ML models have been selected on the basis of the experiments shown by other works concerning the spam detection [6].

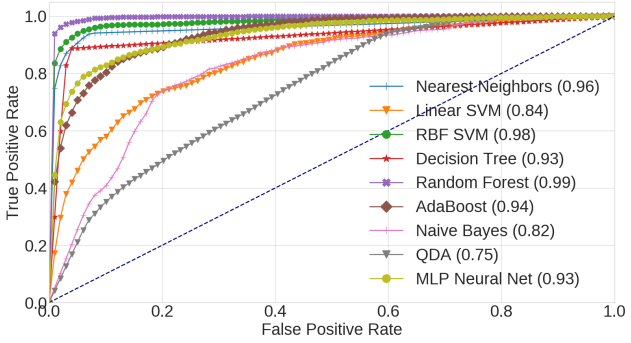


Figure 2: ROC curves of different ML-models (AUC values)

The classification capabilities of these nine supervised approaches have been tested by computing the True and False Positive Rates (TPR/FPR), using as input the full set of features. Figure 2 depicts the Receiver Operating Characteristic (ROC) curves obtained with each model. These results have been obtained with a 10-fold cross validation on the training set. One of the metrics used to evaluate the performance of these approaches is the "Area under Curve (AUC)", which shows that the two best approaches are Random Forest (99%) and RBF SVM (98%). Random Forest has been configured with 140 trees in the forest and 8 variables in the random subset at each node, following the optimization process proposed by *Lee et al.* [25]; RBF SVM has been configured with a gamma coefficient of 0.7 and a penalty parameter C of 5. Since the dataset is unbalanced, the AUC alone cannot properly evaluate the performance [36]. For this reason it has been used only for a preliminary selection of the best models, whereas all the following results are shown in terms of Precision and Recall. The Precision and Recall metrics of the two best-performing approaches are evaluated in details in the following section as a function of the class weights, classification thresholds, and features sets.

4.2. Tuning hyperparameters, class weights, and threshold value

This section explains the approach adopted to properly tune the two models previously selected: Random Forest and RBF SVM. The optimization on the training set of the hyperparameters of such models has been automated using the *RandomizedSearchCV* and *GridSearchCV* functions made available by Scikit-Learn. The functions specify the set of values to be tested for each hyperparameter. In the case of Grid Search, the system is evaluated on all combinations of values of all hyperparameters, while Randomized Search randomly draws values of hyperparameters from the specified distributions, performing a predetermined number of iterations. The best value of the hyperparameters was found by first using *RandomizedSearchCV* to identify the order of magnitude and reduce the range of values to be tested, and then *GridSearchCV* to fine tune the search of the optimal values. According to the tests performed, Random Forest achieved the maximum performance of 98.5% Precision and 89.1% Recall with the following hyperparameters:

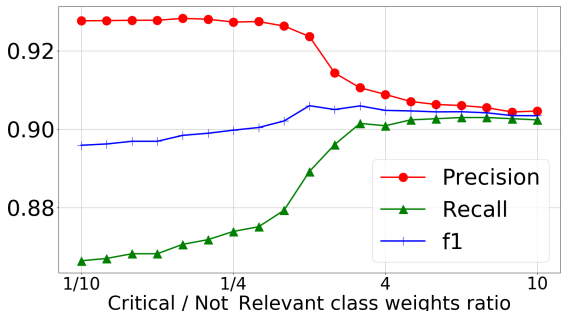
- **n_estimators**= 700
- **max_features**= 'auto'
- **max_depth**= None
- **min_samples_split**= 2
- **min_samples_leaf**= 1
- **bootstrap**= True

RBF SVM achieved the maximum performance of 92.4% Precision and 88.9% Recall with the following hyperparameters:

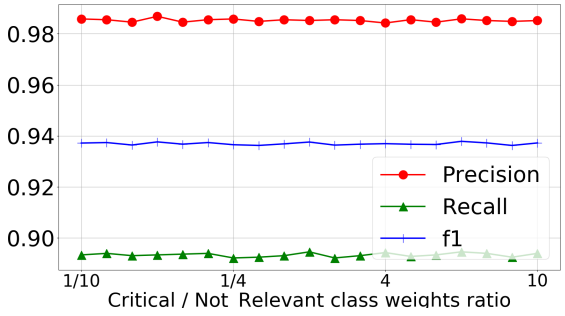
- **C** = 15
- **gamma** = 0.7

Subsequent analyses are then performed using these hyperparameter values for the two models.

Figure 3 shows the Precision, Recall and F-measure of RBF SVM and Random Forest, varying the weights assigned to the two classes. Random Forest has better performance in general (F-measure of 93.8%). On the other hand, RBF SVM obtains higher Recall values (90.3%) at the expense of the Precision (90.6%). In some contexts RBF SVM may be preferred to maximize Recall (up to 90.3%) and minimize risks. In other context, and probably more in general, the tradeoff with Precision (going down to 90.6%) means an excessive amount of false alarms. Random Forest has slightly smaller values of Recall (up to 89.1%), but much higher values of Precision (up to 98.5%). The class weights chosen are therefore:

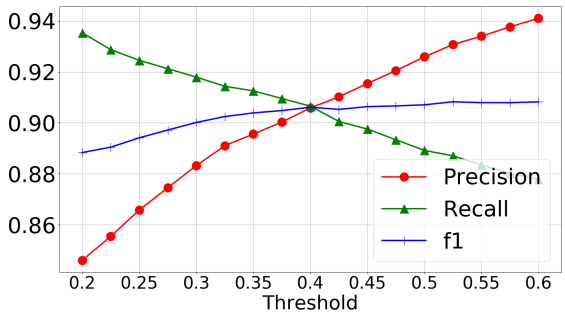


(a) RBF SVM

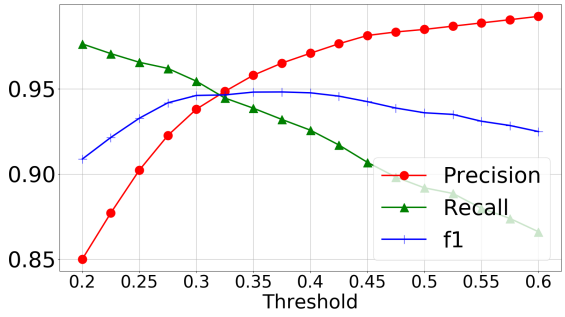


(b) Random Forest

Figure 3: Performance with different class weights



(a) RBF SVM



(b) Random Forest

Figure 4: Performance with different classification threshold values

- **Random Forest** Positive: 1, Negative: 1
- **RBF SVM** Positive: 3, Negative: 1

The classification threshold has been tuned using the class weight reported above. Figure 4 shows the Precision, Recall and F-measure of RBF SVM and Random Forest as a function of the classification threshold value. According to Figure 4, Random Forest shows better performance than RBF SVM even in contexts where the Recall is more important. Analyzing the graphs at peak of the F-Measure curve, Random Forest Recall (93%) achieves higher values than RBF SVM Recall (88.8%), but keeping much higher Precision values (96.5% vs 93%). In addition, even the best RBF SVM Recall value achievable (93.5%) barely beats the previous Random Forest Recall. Random Forest is therefore the best supervised Machine Learning model for our purposes. The best classification threshold values are 0.525 for RBF SVM and 0.375 for Random Forest.

4.3. Feature Ranking

The importance of each feature is analyzed in this section. Two types of analysis have been performed: a first one that considers the individual contribution of each feature, and a second one that considers the contribution of each feature as part of the full feature set, therefore considering the correlations among each other. The first analysis allows to deepen the cognitive phenomenon at the basis of phishing attacks, highlighting the features that have a significant impact in making some spam emails critical compared to others. These results are a fundamental guide to conduct awareness campaigns for the mail recipients, to train them to handle the specific cognitive vulnerabilities they have shown. The second analysis is more concerned with technical aspects: evaluating the real informative contribution of each feature in the context of all the others may lead to identify a subset of features bringing optimal classification performance. Using fewer features however reduces the complexity of the processing to be performed, the execution times and the costs. Moreover, a large number of features does not always correspond to an improvement in performance, due to redundant information, noise in the data and overfitting.

To estimate the individual predictive power of each feature f_i, the mutual information between it and the discrete (binary) target variable y has been computed. The results are shown in Figure 5 and show that the distinguishing characteristics of successful email attacks mainly concern the way in which the content is written. The indexes that estimate the readability of the text evaluating the punctuation, how the message is dispersed in height, the degree of correctness and simplicity of the syntax, and terms used are all very relevant characteristics. Figure 5 also shows that the "Content" features in the "clean text" version, are almost all more important than the "normal" ones. This confirms the need of a method to identify and isolate the hidden text injected in the emails. Interestingly, the origin country of spam is not a discriminating factor to identify critical spam, while features related to reputation systems such as VirusTotal and SMTP blacklists, as well as the number of SMTP hops, provide a quite important contribution.

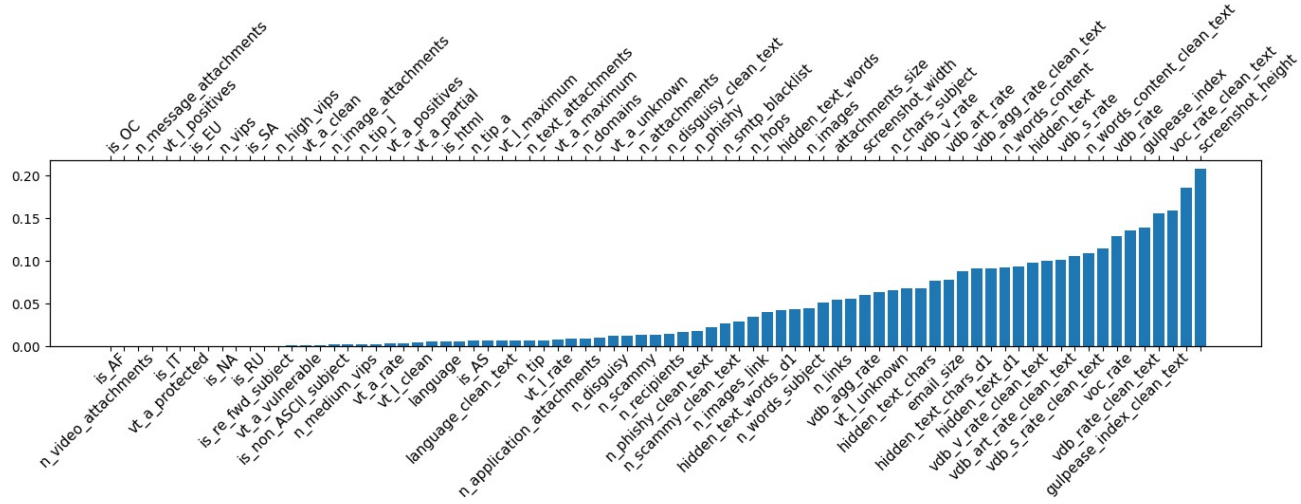


Figure 5: Mutual Information between features and positive class: how much information the feature contributes to the classification

The Wrapper methodology has been used in order to select the best subset of features: it consists in using the prediction performance of a given ML model (named Wrapper) to assess the relative usefulness of subsets of features. In the case of Random Forest the importance of a feature represents how much that feature has contributed to decrease *Gini's impurity*, and this can be easily calculated. As for RBF SVM, instead, computing the actual importance of a feature is a complex procedure as also confirmed by *Liu et al.* [26]. For this reason, SVM with a linear kernel has been used to compute the feature importance. The results of these studies are shown in Figure 6. They confirm what has already been discovered thanks to mutual information analysis, with some interesting additions: the number of recipients and words of the subject are also relevant for the classification. Moreover, information deriving from threat intelligence processes also acquire importance if related with the other features.

The impact of the number of features on the classification performance has been evaluated. The Recursive Feature Elimination procedure has been used for this aim. The results are shown in Figure 7. In both cases, using the entire feature set is counterproductive: the performance of the classifiers slightly degrades while training and classification times increase. The best performances for Random Forest and SVM are achieved with 36 and 51 features respectively, while suboptimal performance can be achieved with 29 (RF) and 38 (SVM) features.

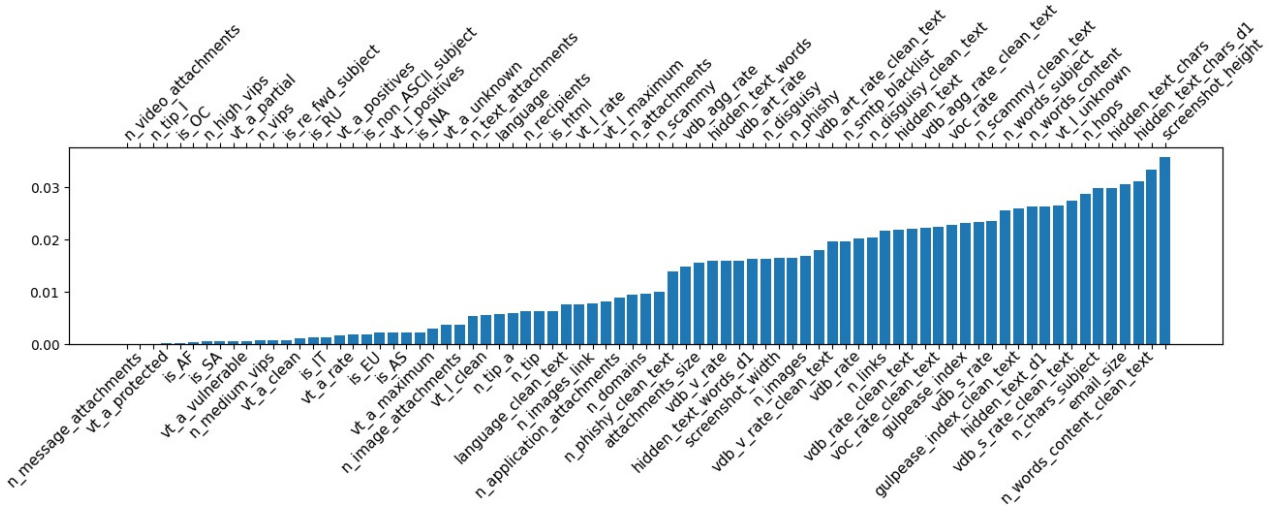
The feature ranking procedure previously performed does not take into account an important factor: the cost of calculation/extraction of the feature. This cost can be both computational (time required to calculate the value of the feature) and monetary (purchase of resources, purchase of licences for third party services). The extraction cost of a feature is to be considered per feature field: if you can obtain the value of a feature then you can obtain all the features of that field. For this reason, the analyses mentioned above were performed also with "feature field" resolution. To this aim, the Wrapper method with Random Forest has been executed with a single feature field at a time. Then, the classification performance has been evaluated increasing the number of feature fields, adding at each step all the features of the best remaining fields, according to the pre-calculated ranking. The results are in Figure 8, and show that:

- as expected the best fields are those concerning the content and the view of the email, thanks to the immediate impact they have on the victims. It also shows that "Content" is redundant when "Content_view" is selected;
- thus not considering "Content", four feature fields are enough to get good performance, avoiding the cost of extracting all the features. However, even just two feature fields could meet the requirements and for this reason an exhaustive research on which was the best pair of fields has been done. Figure 9 shows the F1-score performance of all possible pairs: the best possible performance with two feature fields can be obtained joining the "Content" features with any of the "View", "General", and "Attachments". This is particularly true for "Attachments" and is an unexpected result, since this field alone has bad performance.

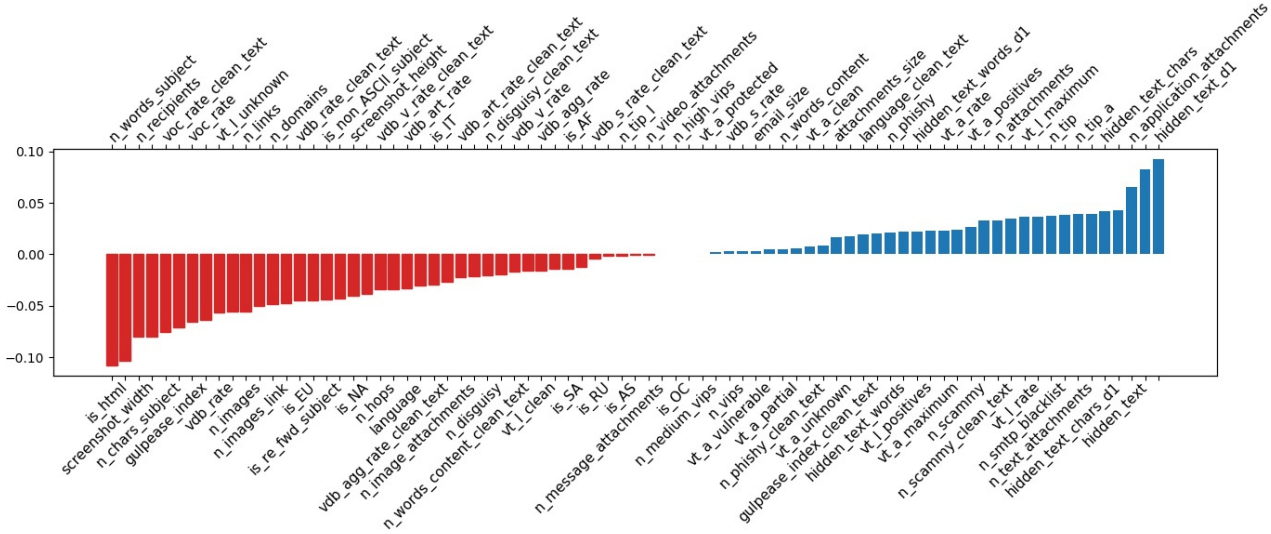
These results are fundamental for deciding which technology and/or service to focus on to develop automated tools for critical spam detection, considering the benefit they bring as a function of their cost.

4.4. Classification performance

The performance evaluation procedure of learning algorithms requires a final test using a set of samples never seen during training and optimization phases. The classification performance of the two models properly tuned were finally tested on the test set (15% of the dataset previously preserved), obtaining the results shown in Table 4. The performance is only 1-2 percentage points lower compared to the



(a) Random Forest as Wrapper



(b) Linear SVM as Wrapper

Figure 6: Feature importance with Wrapper method

Model	Features	Precision	Recall	F1
Random Forest	Full set	0.955	0.909	0.931
	Best 36 Feature	0.952	0.916	0.933
	Best 29 Features	0.933	0.914	0.923
RBF SVM	Full set	0.919	0.871	0.895
	Best 51 Features	0.927	0.880	0.908
	Best 38 Features	0.896	0.885	0.890

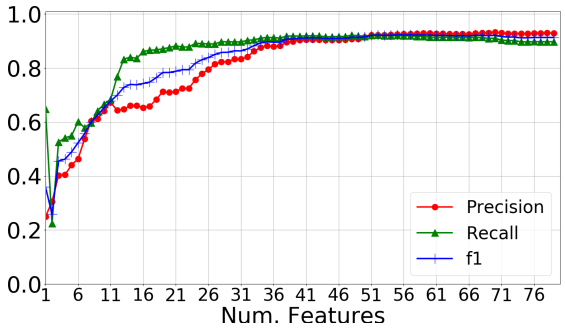
Table 4
Performance on the test set

validation phase. These final results confirm that Random Forest is the best choice for our purposes, with a maximum 95.2% Precision, 91.6% Recall and 93.3% F-Score achieved with 36 features.

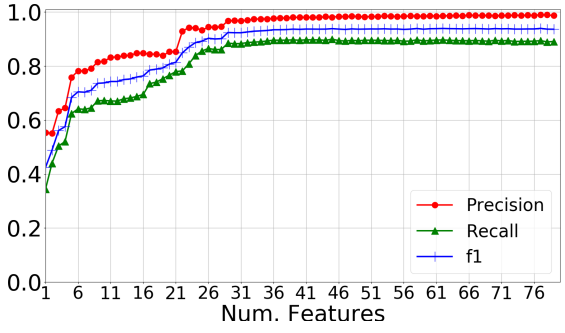
Such impressive performance values allowed to deploy our automated classifier in the infrastructure of the company, integrating it into the actual email threat management process of the company's SOC. In particular, the classifier analyzes all the reports received by the SOC to prioritize them and evidence the most dangerous ones to the analysts who can then make further investigation to prevent possible incidents and mitigate current ones. The integrated system is now enabling the daily detection of several security incidents that would otherwise go undetected.

5. Evading the detector with adversarial samples

The vulnerabilities of machine-learning-based classifiers are well known in the literature [21, 5]. Poisoning attacks are very difficult to perform in our specific scenario because the labeling of samples in the training set is performed manually by a human analyst. Samples that are not manually labeled do not become part of the training set. On the other hand, evasion attacks are much easier to perform, e.g., through a phishing email that has a perturbed value on specific features



(a) RBF SVM



(b) Random Forest

Figure 7: Performance with increasing number of features

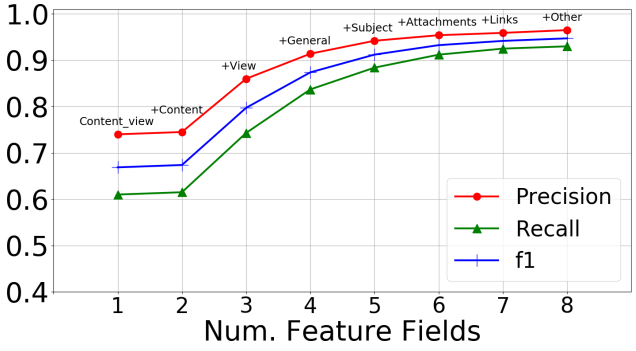


Figure 8: Random Forest performance with increasing number of feature fields

considered important by the classifier. These techniques are very effective against image classifiers, notoriously vulnerable because they over-emphasize a small subset of features (pixels). For example, the image of a panda, with some tampered pixels, is still a panda to the human eye, but not for a classifier based on machine learning. *Apruzzese et al.* and *Biggio et al.* show that this is also possible in contexts such as ours and in particular for Random Forest [3] and SVM [4].

It is therefore important to understand if a similar issue also affects the classifier proposed in this work. In particular, it is important to verify if a highly-effective phishing email (i.e. has very good chances of misleading a human and hurting systems) is still effective when its features are perturbed such that it is no longer relevant for the classifier. To this aim, a huge empirical experiment has been set up: an awareness campaign on almost all employees of the company, including top managers and executives. Adversarial samples increasingly distant from the classifier's positive decision region have been generated as synthetic spam emails and sent to the employees over a one week time span. As reported in details in the following, obtained results show that as soon as samples enter the negative decision region (and are therefore not detected by the classifier), they become slightly effective in succeeding as an attack. The methodology devised for this aim is reported in the following:

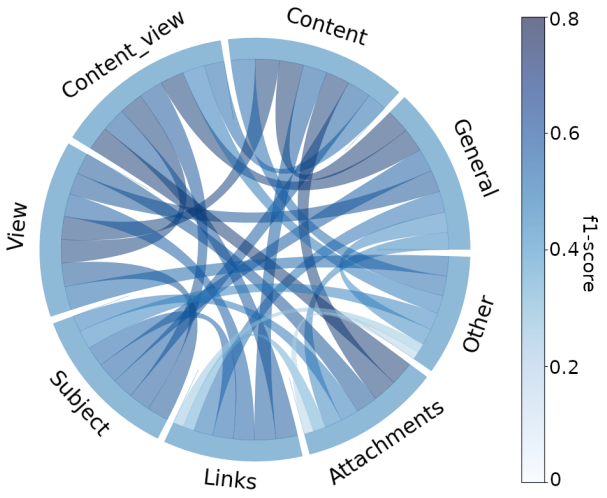


Figure 9: f1-score of all possible pairs of feature fields

1. Clustering of positive samples from the dataset to obtain representative samples of successful attacks. From this procedure about 5 centroids have been obtained, and the most suited one to run the campaign and measure the success rate has been selected: such centroid represents a phishing attack executed with a link. The feature vector from which to generate the adversarial samples has been obtained using this sample

$$f \sim < f_0, ..., f_{m-1} >$$

2. To generate the adversarial samples this feature vector has been manipulated, altering some of the features with a perturbation δ

$$f' = f + \delta \sim < f_0, ..., f_i + \delta_i, ..., f_{m-1} >$$

The intensity of the perturbation was appropriately chosen at each manipulation, depending on which feature it was applied to, in order to preserve the integrity of the attack anyway. It is always about 20% of the value of the feature. The manipulations are 7 (counting also the case of null manipulation) and have been conducted in order to alter the features with more mutual information with the positive class. They are summarized in the Table 5. In order to obtain a set of adversarial samples that are less and less relevant to the classifier, each of them is obtained by adding a

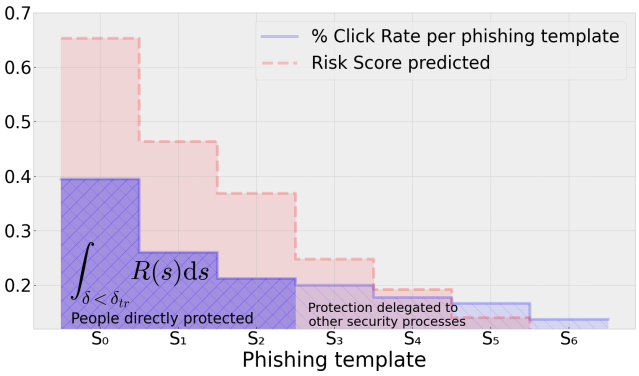


Figure 10: Results of the awareness campaign experiment: impact of feature perturbations δ

new manipulation to the previous sample. All possible combinations of manipulations could not be tested in order to avoid sending too many unsolicited emails to the company's people. Let C be the starting centroid obtained by clustering and let S_i be the adversarial samples

$$S_i = C + \sum_{d=0}^{i} \delta^d$$

3. Seven adversarial samples representing the phishing templates used in our experiment have been obtained. Such synthetic emails have been sent to a total of 41,154 people, of all levels of expertise, education and age. Each phishing template reached 5,879 random people. The purpose is to measure the degree of success of each template.

The experiment designed in this way generated the results shown in Figure 10, which highlights and confirms that:

- As perturbations increase, the probability of belonging to the positive class (risk score) decreases.
- As the risk score decreases, so does the degree of success of the phishing template. This means that the classifier models the phenomenon correctly.
- The alteration of a feature generates a decrease in the click rate proportional to its importance. The first alterations, made on the most important features, degrade the risk score more than subsequent alterations.

Thanks to this analysis it is possible to identify the best risk score threshold (or δ_{tr} perturbation threshold) in order to prioritize the analysis of email spam reports. For each value of this threshold, which represents the amount of effort that the SOC can provide on this task, the number of possible security incidents detected is maximized, minimizing those that remain unnoticed. This is of fundamental importance due to the impossibility to check all such reports and in order not to waste too much effort on those not dangerous.

#	Manipulation	Altered Features
δ^0	No manipulation	None
δ^1	Alteration of the readability of the content by smudging the punctuation	gulpease_index, gulpease_index_clean_text
δ^2	Alteration of the correctness of the content by injecting typing errors	voc_rate(_clean_text), vdb_*_rate(_clean_text)
δ^3	Deletion of hidden text (white text on white background)	hidden_text_*, vdb_*_rate, voc_rate
δ^4	Remotion of deceiving words from the subject	n_scammy, n_phishy, n_*_subject, n_words_subject
δ^5	Dispersion of the deceiving message by adding a long block of text at the bottom of the content and words in the subject line	n_*_subject, n_*_content, screenshot_height
δ^6	Insertion of multiple points where to click by adding clickable images	n_links,vt_l_*, n_images, n_images_links

Table 5
Manipulations performed to generate adversarial samples

6. Discussion and Limitations

This work shows that with our approach it is possible to automatically distinguish whether a received unsolicited email represents an attack attempt and accurately estimate the probability of its success. In this way, the anti-phishing analysts can be assisted to use their limited resources on the most dangerous phishing attacks. The limitations of our approach and evasion strategies that adversaries might pursue are discussed below.

The infeasibility of analyzing all received emails. Since our model is based on complex features, extracted through long computations and usage of licensed third party services, it is not possible to extend this in-depth security analysis to all received emails, which are millions per day, due to monetary and computational constraints. The feature ranking section (Section 4.3) shows the possibility of feature reduction still saving most of predictive power, enabling computation on much more samples. Unfortunately, the construction of the ground truth through the manual labeling of emails by analysts is not practicable because of privacy issues if the emails are not reported as suspicious. Among the future works, there is the extension of the analysis to all emails using unsupervised approaches. Our supervised approach is therefore built on user reports, thus leading to the next point.

The need of virtuous users. Our approach heavily relies on user reporting, which lead to the engagement of the anti-phishing group on the possible attack received. User involvement in identifying phishing suspects is crucial, as it pre-filters the totality of emails received by the company

and selects candidates for a thorough security check. Having a number of users who are aware of these security aspects is therefore an important requirement for achieving a kind of herd immunity that serves to defend themselves and the whole company. With this in mind, it is important to be able to design effective awareness campaigns based on security incidents that affected users in the recent past. The results of feature importance (Section 4.3), from an *Explainable AI* (XAI) perspective, highlight the characteristics of the most impactful email attacks, providing a decisive contribution to tailoring synthetic emails used for awareness campaigns according to the precise vulnerabilities of users, as also demonstrated by the experiment performed, documented above (Section 5).

The lack of protection on single-victim attacks. Our approach lacks visibility of single-target successful attacks, because the anti-phishing analysts are only engaged if at least one user who has received the suspected phishing email reports it. This is actually a rare possibility since these types of attacks, in order to increase the probability of success, are almost always launched with multiple recipients in proper phishing campaigns. However, in the case of single-victim attacks the only possibility is once again to keep users trained to recognise these types of threats, especially for those most targeted by *phishing ad personam* attacks (e.g. top managers, executives and their close collaborators). Client-based tools can also be adopted to support individual users in recognising phishing [41].

Supervised Learning weaknesses. Although the main limitation of supervised approaches has been resolved in this work, namely the need to obtain a labelled dataset, there are other well-known weaknesses to discuss. These include the class imbalance in the dataset; this is one of the problems of the use of supervised approaches [18] in the settings of email security analysis, which exhibit extreme class imbalance (on the order of millions to one). However, our specific approach of processing only reports of phishing suspicions and not all received emails, greatly reduces the class imbalance to the order of 4.5 to 1. In addition, some sub-types of attacks poorly represented in the positive class may be miss-classified, negatively affecting the Recall performance (about 90%, Section 4.4). Finally, supervised approaches cannot detect 0-day methodologies of phishing attacks; therefore, we plan to also experiment with unsupervised approaches to complement our.

7. Concluding Remarks

Email attacks are such a commonly used vehicle for the perpetration of subsequent attacks, representing a major threat that affects all industries and causes significant harm. Anti-spam filters do not solve the problem of cyber attacks by spam emails, which still succeed in spreading malware, stealing confidential data, and generating large illicit profits. For this reason, companies typically rely on teams of security analysts to perform manual inspection on such emails. However, spam emails that evade spam filters, especially in the case of large companies, are too many for such analysis to be effective. In this paper we aimed at providing a contribution to this important problem.

In early 2018, we have built a collaborative framework that collects spam emails and supports the labeling of the actually dangerous ones as critical, through the continuous monitoring of analysts. Using this labeled dataset we have shown that machine learning algorithms can well classify emails as critical, highlighting the threats. We have also identified the main features that make a spam email dangerous and the best techniques and technologies to rely on for the defense. Both legacy and novel features have been used, and their relevance and correlation with the target have been evaluated. Using the best feature set maximizing the f1-score performance, the supervised approaches reaches 95.2% Precision and 91.6% Recall. We have also identified a reduced feature set that greatly reduces costs with a small impact on performance. The possibility to evade these classifiers with adversarial machine learning techniques has also been investigated with a large empirical experiment involving 40,000+ employees of the company, confirming that the used algorithms correctly model the cognitive phenomenon of email dangerousness.

Thanks to the results obtained and lessons learned, we re-engineered the email threat management process around this collaborative approach. It now relies on experienced and aware users who report suspicious emails, an automatic data collection and analysis system, and security analysts who investigate in depth according to the system's suggestions. Some of the IoCs produced as a result of applying this system are also published in our company's Threat Intelligence Platform, in order to improve the overall security defence of the company. In general, the use of this kind of systems, in synergy with existing security processes, greatly helps to mitigate the long-standing problem of email attacks. We believe that our contributions can lead to a greater awareness of the risks faced by companies and, above all, to the automation of the detection of threats in spam emails, in the context of both reporting systems and Managed Security Services.

Acknowledgment

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 830927 (CONCORDIA H2020 Project).

References

[1] Aleroud, A., Zhou, L., 2017. Phishing environments, techniques, and countermeasures: A survey. Computers & Security 68, 160 – 196. URL: http://www.sciencedirect.com/science/article/pii/S0167404817300810, doi:https://doi.org/10.1016/j.cose.2017.04.006.

[2] Allodi, L., Chotza, T., Panina, E., Zannone, N., 2019. On the need for new antphishing measures against spear phishing attacks. IEEE Security & Privacy PP. doi:10.1109/MSEC.2019.2940952.

[3] Apruzzese, G., Colajanni, M., 2018. Evading botnet detectors based on flows and random forest with adversarial samples, in: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), pp. 1–8.

[4] Biggio, B., Corona, I., Nelson, B., Rubinstein, B.I.P., Maiorca, D., Fumera, G., Giacinto, G., Roli, F., 2014. Security Evaluation of Support Vector Machines in Adversarial Environments. Springer International Publishing, Cham. pp. 105–153. URL: https://doi.org/10.1007/978-3-319-02300-7_4, doi:10.1007/978-3-319-02300-7_4.

[5] Biggio, B., Roli, F., 2018. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition 84, 317 – 331. URL: http://www.sciencedirect.com/science/article/pii/S0031320318302565, doi:https://doi.org/10.1016/j.patcog.2018.07.023.

[6] Blanzieri, E., Bryl, A., 2008. A survey of learning-based techniques of email spam filtering. Artificial Intelligence Review 29, 63–92. URL: https://doi.org/10.1007/s10462-009-9109-6, doi:10.1007/s10462-009-9109-6.

[7] Chan, J., Koprinska, I., Poon, J., 2004. Co-training on textual documents with a single natural feature set., pp. 47–54.

[8] Chen, J., Paxson, V., Jiang, J., 2020. Composition kills: A case study of email sender authentication, in: 29th USENIX Security Symposium (USENIX Security 20), USENIX Association, Boston, MA. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/chen-jianjun.

[9] Cialdini, R., 1993. Influence: The psychology of persuasion .

[10] Cidon, A., Gavish, L., Bleier, I., Korshun, N., Schweighauser, M., Tsitkin, A., 2019. High precision detection of business email compromise, in: 28th USENIX Security Symposium (USENIX Security 19), USENIX Association, Santa Clara, CA. pp. 1291–1307.

[11] Dada, E.G., Bassi, J.S., Chiroma, H., Abdulhamid, S.M., Adetunmbi, A.O., Ajibuwa, O.E., 2019. Machine learning for email spam filtering: review, approaches and open research problems. Heliyon 5, e01802. URL: http://www.sciencedirect.com/science/article/pii/S2405844018353404, doi:https://doi.org/10.1016/j.heliyon.2019.e01802.

[12] Dai, Y., Tada, S., Ban, T., Nakazato, J., Shimamura, J., Ozawa, S., 2014. Detecting malicious spam mails: An online machine learning approach, in: Neural Information Processing, Springer International Publishing, Cham. pp. 365–372.

[13] EC3, E., 2019. Spear phishing, a law enforcement and cross-industry perspective. https://www.europol.europa.eu/sites/default/files/documents/report_on_phishing_-_a_law_enforcement_perspective.pdf.

[14] FBI Cleveland, S.A.V.D.A., 2016. Fbi warns of rise in schemes targeting businesses and online fraud of financial officers and individuals. https://www.fbi.gov/contact-us/field-offices/cleveland/news/press-releases/fbi-warns-of-rise-in-schemes-targeting-businesses-and-online/-fraud-of-financial-officers-and-individuals.

[15] Flesch, R., 1948. A new readability yardstick. Journal of Applied Psychology 32(3), 221–233. doi:https://doi.org/10.1037/h0057532.

[16] Gallo, L., Botta, A., Ventre, G., 2019. Identifying threats in a large company's inbox, in: Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, Association for Computing Machinery, New York, NY, USA. p. 1–7. URL: https://doi.org/10.1145/3359992.3366637, doi:10.1145/3359992.3366637.

[17] Gansterer, W., Pölz, D., 2009. E-mail classification for phishing defense, pp. 449–460. doi:10.1007/978-3-642-00958-7_40.

[18] He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284. doi:10.1109/TKDE.2008.239.

[19] van der Heijden, A., Allodi, L., 2019. Cognitive triaging of phishing attacks, in: 28th USENIX Security Symposium (USENIX Security 19), USENIX Association, Santa Clara, CA. pp. 1309–1326.

[20] Ho, G., Sharma, A., Javed, M., Paxson, V., Wagner, D., 2017. Detecting credential spearphishing in enterprise settings, in: 26th USENIX Security Symposium (USENIX Security 17), USENIX Association, Vancouver, BC. pp. 469–485. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/ho.

[21] Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.D., 2011. Adversarial machine learning, in: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Association for Computing Machinery, New York, NY, USA. p. 43–58. URL: https://doi.org/10.1145/2046684.2046692, doi:10.1145/2046684.2046692.

[22] of Investigation Internet Crime Compliant Center, F.B., 2020. 2019 internet crime report. URL: https://pdf.ic3.gov/2019_IC3Report.pdf.

[23] Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L., Hong, J., 2008. Lessons from a real world evaluation of anti-phishing training, pp. 1 – 12. doi:10.1109/ECRIME.2008.4696970.

[24] Lee, M., 2014. Pytesseract https://pypi.org/project/pytesseract. URL: https://pypi.org/project/pytesseract.

[25] Lee, S.M., Kim, D.S., Kim, J.H., Park, J.S., 2010. Spam detection using feature selection and parameters optimization, in: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 883–888. doi:10.1109/CISIS.2010.116.

[26] Liu, Q., Chen, C., ZHANG, Y., Hu, Z., 2011. Feature selection for support vector machines with rbf kernel. Artif. Intell. Rev. 36, 99–115. doi:10.1007/s10462-011-9205-2.

[27] Lucisano, P., Piemontese, M.E., 1988. Gulpease. una formula per la predizione della difficolta dei testi in lingua italiana. In Scuola e Citta (3) , 57–68.

[28] Mallikarjunappa, B., R. Prabhakar, D., 2010. A novel method of spam mail detection using text based clustering approach. International Journal of Computer Applications 5. doi:10.5120/906-1283.

[29] McHugh, M., 2012. Interrater reliability: the kappa statistic. Biochemia Medica 22, 276–282.

[30] Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., Stamatopoulos, P., 2004. Filtron: A learning-based anti-spam filter, in: Proceedings of the 1st conference on email and anti-spam. Mountain.

[31] Ndibwile, J.D., Luhanga, E.T., Fall, D., Miyamoto, D., Blanc, G., Kadobayashi, Y., 2019. An empirical approach to phishing countermeasures through smart glasses and validation agents. IEEE Access 7, 130758–130771.

[32] Nizamani, D.S., Memon, N., Glasdam, M., Duong Nguyen, D., 2014. Detection of fraudulent emails by employing advanced feature abundance. Egyptian Informatics Journal 15. doi:10.1016/j.eij.2014.07.002.

[33] Pavlo Burda, L.A., Zannone, N., . Don't forget the human: a crowdsourced approach to automate response and containment against spear phishing attacks.

[34] R. Shirey, 2007. RFC4949: Internet Security Glossary, Version 2.

[35] RSA, 2012. Understanding Indicators of Compromise (IOC) Part I. URL: https://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/.

[36] Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE 10, 1–21. URL: https://doi.org/10.1371/journal.pone.0118432, doi:10.1371/journal.pone.0118432.

[37] Sappleton, N., Lourenço, F., 2016. Email subject lines and response rates to invitations to participate in a web survey and a face-to-face interview: the sound of silence. International Journal of Social Research Methodology 19, 611–622. URL: https://doi.org/10.1080/13645579.2015.1078596, doi:10.1080/13645579.2015.1078596, arXiv:https://doi.org/10.1080/13645579.2015.1078596.

[38] Sourena Maroofi, Maciej Korczynski, A.D., 2020. From defensive registration to subdomain protection: Evaluation of email anti-spoofing schemes for high-profile domains. URL: http://mkorczynski.com/TMA2020Maroofi.pdf.

[39] Stringhini, G., Egele, M., Zarras, A., Holz, T., Kruegel, C., Vigna, G., 2012. B@bel: Leveraging email delivery for spam mitigation, in: Presented as part of the 21st USENIX Security Symposium (USENIX Security 12), USENIX, Bellevue, WA. pp. 16–32. URL: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/stringhini.

[40] Symantec, 2019. 2019 internet security threat report. https://www.symantec.com/security-center/threat-report.

[41] Volkamer, M., Renaud, K., Reinheimer, B., Kunz, A., 2017. User experiences of torpedo: Tooltip-powered phishing email

detection. Computers & Security 71, 100 – 113. URL: http://www.sciencedirect.com/science/article/pii/S0167404817300275, doi:https://doi.org/10.1016/j.cose.2017.02.004.

[42] Wainer, J., Dabbish, L., Kraut, R., 2011. Should i open this email? inbox-level cues, curiosity and attention to email, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. p. 3439–3448. URL: https://doi.org/10.1145/1978942.1979456, doi:10.1145/1978942.1979456.

A. Clean text extraction with optical character recognition

The process of extracting the clean text from the samples of our dataset is quite complex and consists of several phases, represented in Figure 11. The clean text is the result of the intersection of two text files that can be extracted from a single email: the text obtained by running OCR (Optical Character Recognition) on the screenshot of the email, which we call "OCR Text"; and the text obtained converting the HTML version of the email into a clean and easy to read text, which we call "HTML2Text Text". The reason why we intersect these two text files lies in the main drawback of HTML2Text Text: being derived from the HTML of the mail message, in addition to the text that is shown to the user by the mail client used, it also includes all the text injected into the mail as Hidden Text. This can take the form of text with the same color of the background, for example white text on white background, but it can also be some text not shown by setting the property "display:none" or several other html/css-based tricks. The OCR Text, on the other hand, is the text obtained through OCR performed on the screenshot of the email rendered in the browser, i.e. all the text that the user can actually see and read when he opens the email. This text, however, also includes all the text of any images in the email, i.e. Image Text. It is evident that the text obtained from the intersection of these sets, HTML2Text Text and OCR Text, i.e. the text in both of them, is precisely the clean text we are looking for. The OCR tool, however, can commit some mistakes when recognising words, usually by misreading one character in the word. In order to handle this behavior, we decided to consider equal words with an edit distance of 1, thus generating two types of hidden text features: "hidden_text" and "hidden_text_d1". The "clean_text" version of the features are all obtained by using the "d1" version of the hidden text.

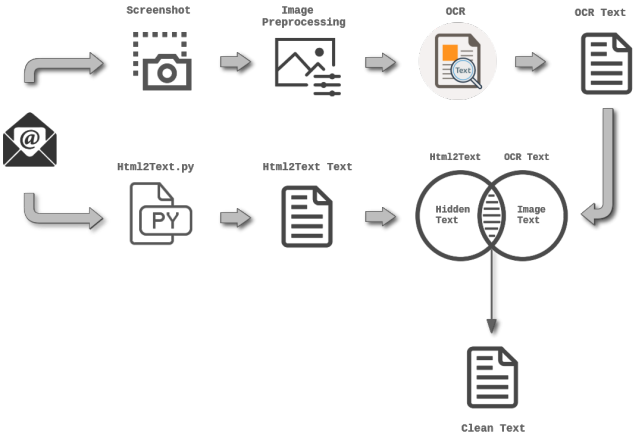


Figure 11: Clean text extraction scheme

The problem therefore shifts to deriving the two main ingredients for the creation of the clean text. The HTML2Text Text can be simply obtained by using of the homonymous python script in order to clean up the HTML code of the email from the various language tags. The operations necessary to obtain the OCR Text are much more complex and require a more detailed analysis and explanation. The whole process of OCR Text extraction, as shown in Figure 11, consists of three main steps, all automated through python scripts:

- Rendering the email in the browser and saving the screenshot
- Post-processing the screenshot
- Text recognition in the screenshot by OCR

The first phase was carried out using Selenium, a set of tools designed to automate browsers, and, more specifically, by using Selenium Webdriver. Through an appropriate Python API you can access all the features offered by Selenium Webdriver in a simple and effective way. In order to work, Selenium needs an appropriate driver to communicate with the chosen browser interface. This driver, which varies from browser to browser, must be downloaded and installed before you can run any python code related to Selenium. In our case, having chosen Chrome as the browser to automate, the driver is made available by Google and is called "chromedriver". Selenium Webdriver allows you to manipulate DOM elements in Web pages and to control the browser through appropriate python commands. It is possible for example to start a new browser instance, make it open the email and capture a screenshot of the screen. In order for all these procedures to work correctly, the webdriver was configured to start the browser in *headless mode* with an opportune window size, a zoom of 450% and with a timeout limit of 2 seconds. This configuration was required to be able to open a web page not limited in size by the display in use and to capture the email, full loaded, with a single high resolution screenshot. The OCR tool used, Python-tesseract, in fact, requires images with a recommended resolution of at least 300 DPI.

As we know, optical character recognition systems are born and are designed to detect the characters contained in a document, i.e. essentially black writings on a white/yellow background. Emails, on the other hand, in addition to having text on a white background, can assume the most varied shapes and colors. This makes the screenshots we have acquired unsuitable to be processed by an OCR tool as they are. It was therefore necessary to edit the images in order to ease the recognition of the characters. The solution we found after countless tests, consists in converting the screenshot into grayscale, in order to reduce the chromatic variability, and

applying a sharpening filter, to make the text stand out more from the background.

As a last step, the screenshot thus obtained and modified, has been processed through the OCR tool Python-tesseract, a wrapper from Google's Tesseract-OCR Engine. For a complete description of the tool and the various possible configurations, please refer to the project page [24]. The configuration setup involves recognizing Italian as the main language in the text, and English as a secondary language. The extracted text has been saved in an appropriate text file, and has been used together with HTML2Text Text to extract the Clean Text.

Luigi Gallo is a Junior Researcher at Telecom Italia LAB (TIM) and a PhD student at the Department of Electrical Engineering and Information Technology, Federico II University of Napoli. Graduated in 2018 (summa cum laude) with a thesis on Big Data Technologies for Anomaly Detection in traffic traces. Currently focused on data-driven research activities for security purposes, dealing with Email Security, Security Awareness, Network Traffic Analysis and 5G Mobile Networks.

Alessandro Maiello is a Cyber Security Analyst at Telecom Italia (TIM) Security Lab since August 2020. He achieved MS degree (summa cum laude) at University of Napoli Federico II in July 2020, with a thesis on a supervised learning approach for preventing security incidents from spam emails. Currently his work focuses on scouting and testing of security solutions.

Alessio Botta received the M.S. degree in telecommunications engineering and the Ph.D. degree in computer engineering and systems from the University of Naples Federico II, Naples, Italy. He is currently an assistant professor at the University of Naples Federico II. He has co-authored over 70 international journal and conference publications. In the research area of networking, he has chaired international conferences and workshops, served and serves several technical program committees of international conferences, is member of the editorial board and reviewer for different international conferences and journals.

Giorgio Ventre is Full Professor of Computer Networks at the Department of Information and Electrical Engineering of the University of Napoli Federico II. He owns a Laurea Degree in Electronic Engineering and a PhD in Computer Engineering, both from the University of Napoli Federico II. He is currently Chair of the Department of Information and Electrical Engineering of the University of Napoli Federico II (DIETI) and Scientific Director of the Apple Developer Academy in Napoli. He is also member of several Scientific Committees and seats in the Board of public and private entities.