Knowledge-Based Systems 310 (2025) 112939

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Spam email classification based on cybersecurity potential risk using natural language processing

Francisco Jáñez-Martino *, Rocío Alaiz-Rodríguez, Víctor González-Castro, Eduardo Fidalgo, Enrique Alegre

Department of Electrical, Systems and Automation Engineering, Universidad de León, Spain
INCIBE (Spanish National Institute of Cybersecurity), León, Spain

ARTICLE INFO

Keywords:
Malicious spam email detection
Feature extraction
NLP
Security awareness
Cybersecurity

ABSTRACT

Spam emails go beyond being merely annoying messages, they have become one of the most used vectors for cyberattacks such as stealing personal information or spreading malware. These breaches in cybersecurity lead to financial and data loss for individuals and organisations. Thus, the ability to differentiate potentially risky emails is crucial to launch earlier warnings and gain relevant information for cybersecurity experts. Recent works have proposed models to detect phishing, fraudulent or critical spam emails. However, their focus is often restricted to a particular email type or only evaluated on spam emails received by organisations. In this work, we propose a new set of 56 features extracted using Natural Language Processing (NLP) techniques and grouped into five categories: *Headers*, *Text*, *Attachments*, *URLs*, and *Protocols*. We build a dataset, Spam Email Risk Classification (SERC), divided into two sub-datasets: one collected from a private source and another from Bruce Guenter's public corpus. To assess the potential risk of spam emails for users, we follow two strategies: a binary classification using low and high risk and a regression approach to predict the level of risk from 1 to 10. We evaluated three Machine Learning classifiers and three regression models. Random Forest obtains the highest classification performance with 0.914 of F1-Score on SERC and Random Forest Regressor achieves the lowest Mean Square Error (MSE) of 0.781 for regression. We also conduct an analysis of the feature importance in terms of each feature and group where those from the *Headers* and *Text* groups become more relevant.

1. Introduction

Spam has become a popular way for criminals to conduct illegal online activities for the last few years. Such operations include stealing sensitive information, selling counterfeit goods, and distributing malware, among others. Nowadays, there is a rise in the number of cyberattacks originating from this kind of potentially risky spam emails. As a result, these emails often contain valuable cybersecurity information [1].

Risk in spam emails refers to those emails whose content can expose the integrity, security, and privacy of users by containing scams to obtain money, personal data or spread malware. We can define the risk in spam emails as follows:

Definition 1. The risk associated with spam emails refers to the severity of potential cybersecurity incidents that may affect recipients if the email successfully bypasses anti-spam filters and achieves its malicious intent.

The behaviour of users when they receive a potentially risky email is crucial. Among these behaviours, clicking on links in phishing emails or downloading documents exposes the user's device to be used as an entry point, compromising their computer system or, in some cases, organisational devices [2]. Such cybersecurity risks can result in financial and data loss and have a negative impact on an organisation reputation. According to [3] and the IBM report "Cost of a Data Breach Report 2023",[1] the average cost of a data breach has increased to $4.45 million. Additionally, cybercriminals propagate spam campaigns that involve the distribution of malware, phishing and other kind of scams [4], which compromise the privacy, integrity and security of users.

* Correspondence to: Department of Electrical, Systems and Automation Engineering, University of León, Spain
E-mail address: francisco.janez@unileon.es (F. Jáñez-Martino).

[1] https://cutt.ly/fwczL6Z2, retrieved November 2024.

https://doi.org/10.1016/j.knosys.2024.112939
Received 6 March 2024; Received in revised form 6 November 2024; Accepted 28 December 2024
Available online 4 January 2025

Overconfidence in information security can lead to incident misbehaviour [5], thereby, so it is worth detecting and alerting as much as possible. Sturman et al. [3] conducted an empirical study on participants ability to detect phishing emails with or without cues. From an applied perspective, their results indicated that the user may benefit from cue-based training in addition to knowledge-based training, although the decision of the participants showed bias. The study identified the sender's address, subject line, greeting, spelling/grammar, URL link, and corporate logo as clues, i.e., phishing features. In their study of a phishing simulation followed by a survey of 590 employees in a large financial organisation, Buckley et al. [6] also highlighted that training employees to pay attention in the address of the senders, URL or file extension would enable them to differentiate phishing elements.

Organisations dedicated to cybersecurity, such as the Spanish National Cybersecurity Institute (INCIBE), employ semi-automated analysis of thousands of emails daily to identify potential risks to users. Given the vast volume of emails, human resources may not be enough for this task. The development of a system capable of efficiently filtering these emails serves to enhance and expedite these efforts. In addition, the alerts launched by INCIBE[2] provide cybersecurity information on the reason of the threat of the targeted email, which is also interesting to be extracted automatically. Cybersecurity organisations aim not only to filter traditional spam containing unsolicited and bulk content but also, and more importantly, to identify the most potentially dangerous spam emails, such as phishing and spoofing. They prioritise the detection of these threats to protect organisations and citizens from potential cybersecurity incidents.

Some previous works have identified sets of features that can help to detect emails created by sophisticated spammer strategies to bypass anti-spam filters used by companies and enough social engineering techniques to confuse employees alike [7–9]. Understanding human behaviour and spammer strategies plays an essential role in extracting features to distinguish this type of harmful spam email. These studies primarily focused on employees within companies and their responses to malicious emails, which often aim to harm the reputation of a company, resources, or benefits or obtain an economic reward using different threats such as ransomware. However, individual users also encounter malicious emails with similar aims, such as stealing personal information, extortion, giving fake rewards or exploiting health, sexual or pharmacy concerns [4]. Spam emails commonly exploit the topic of the black economy and employ tactics such as online money-making schemes or fake work offers to deceive and mislead individuals.

In this paper, we present a new set of 56 features grouped into five subsets — headers, text, attachments, URLs, and protocols — and extracted using Natural Language Processing (NLP) techniques. We study which are the key patterns within potentially risky emails, employing a methodology aligned with the practices of cybersecurity experts. INCIBE provides us with a comprehensive list of vulnerabilities and critical elements found in spam emails, including headers, body content, URLs, and attachments. We pay particular attention to sender addresses [10], cybersecurity topics [4] and spammer tricks [11].

The aim of this work is not only to create an intelligent system for filtering potentially risky spam emails but also to provide relevant information to cybersecurity experts to gain insight into the workings of cybercriminals. Additionally, it seeks to identify and extract crucial attributes that facilitate the detection of potentially harmful emails, thereby enhancing the information provided to both citizens and companies. We reviewed some of the findings from similar studies to assess the possibility of incorporating, modifying, or enhancing features from their proposed sets [7–9].

These previous studies do not have public access to their code and datasets, or their objectives are slightly different from ours. As a result, we have generated two new datasets: one private, using the resources of INCIBE, and another publicly available, from the Spam Archive of Bruce Guenter. The latter is made available to the scientific community, enabling comparisons with future research and facilitating the analysis of the information using our feature set. We follow two methodologies, each accompanied by their respective annotations, to address this issue: (a) classification, and (b) regression for predicting the risk level on a scale from 1 to 10. Finally, during the evaluation, we also consider the combination of both datasets to train the Machine Learning models. We name our dataset as Spam Email Risk Classification (SERC). While the classification approach enables potentially risky emails to be filtered, the inclusion of a risk level scale can be essential for cybersecurity experts. This scale serves as a valuable tool, allowing experts to set different thresholds tailored to their analyses. To the best of our knowledge, this is the first work to develop a system that applies both regression and classification to spam emails to identify the most harmful ones. However, similar research, including studies on spam email classification [4,12], provides a foundation for the feasibility of this approach and supports the use of both traditional and deep learning models. In addition, we have addressed the challenge of the lack of annotated datasets by manually annotating two available sets of data (see Section 4.1). The features proposed in this work are derived from a combination of previous studies, manual inspection, and the insights of cybersecurity experts, who regularly analyse these features to detect high-risk emails. Furthermore, the proposed system is designed to be scalable and is tailored to meet the computational requirements of INCIBE-CERT, and therefore any other similar security incident response centre, making it suitable for real-world deployment.

We consider Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF) as classifiers, and we predict the risk using Linear Regression (LinR), Support Vector Regression (SVR), and Random Forest Regression (RFR). The wide use of Transformers models [13] in NLP applications and their high performance, motivates us to benchmark our approach against them. We apply transfer learning approach to make this comparison. Additionally, for the classification approach, we assess the importance of each feature individually, and we also examine the relative contribution of each group of features. A flowchart which illustrates the comprehensive process conducted in this work is shown in Fig. 1.

The rest of the paper is organised as follows. Section 2 reviews the most recent works extracting valuable features to identify potentially risky spam emails. Section 3 presents our five sets of features and how each is extracted. Section 4 explains the dataset creation and its statistical attributes, as well as the classifiers and estimators used. Section 5 carries out an empirical evaluation of both the features and the models. Next, in Section 6, we discuss the experimental results to highlight the most relevant findings and offer explanations for these outcomes. Finally, Section 7 sums up the contributions of our work and identifies the limitations and future work.

2. Background

There is a rise of scams in spam emails and their use across different social networking systems, but while spam occurs in various forms, it mainly serves two purposes: advertising and fraud [14]. Due to this fact, some works have focused on a specific form of spam, being phishing emails the most popular one [15]. While other works have found those mails with fraud content and potential danger to cause a security incident [7,9]. To prevent users from these sophisticated frauds, researchers have developed intelligent email software [16].

2.1. Phishing email detection

A phishing email is a deceptive message that uses social engineering to appear to come from a trustworthy source, with the aim of tricking recipients into revealing sensitive information such as passwords

[2] https://www.incibe.es/ciudadania/avisos?field_tax_etiquetas_target_id=All retrieved November 2024.

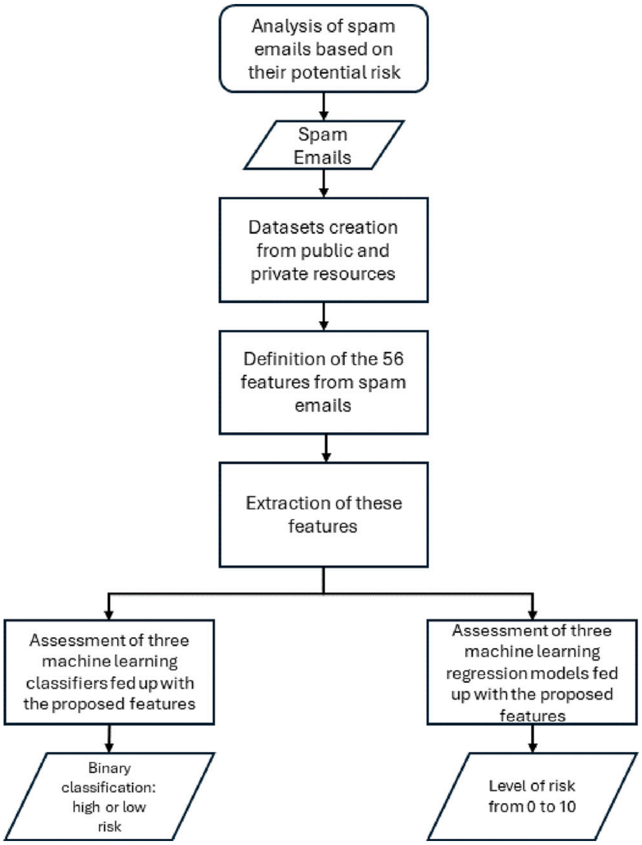

Analysis of spam emails based on their potential risk
Spam Emails
Datasets creation from public and private resources
Definition of the 56 features from spam emails
Extraction of these features
Assessment of three machine learning classifiers fed up with the proposed features
Assessment of three machine learning regression models fed up with the proposed features
Binary classification: high or low risk
Level of risk from 0 to 10

Fig. 1. Flowchart illustrating the comprehensive process implemented in this study, from the creation of the spam email datasets to the identification of potentially risky emails by means of classification and regression, through feature retrieval.

or credit card details. These emails often contain malicious links or attachments to exploit individuals for identity theft, financial fraud, or unauthorised access to personal data. The models to spot phishing emails are based on anti-spam filters but were specifically designed to spot phishing [17].

Volkamer et al. [18] developed a tool called TOoltip-poweREd Phish Email DetectiOn (TORPEDO) capable of disabling links for a short period, detecting re-directions and tiny URLs and returning diagrams with phishing detection advice. They highlighted URLs as the most significant reason for people falling victim to phishing, and their tool focused on analysing them. Sankhwar et al. [19] also directed their work towards detecting phishing by spotting malicious URLs.

Some filters are based on NLP techniques along with Machine Learning and Deep Learning [17,20–23], which followed a text classification pipeline. Smadi et al. [20] proposed a filter that joined Neural Network models and Reinforcement Learning. Halga et al. [21] presented a Recurrent Neural Networks system to detect phishing attacks. Alhogail and Alsabih [23] designed a classifier employing a Graph Convolutional Network and emphasised the significance of examining the text features of an email body. These filters achieved high performance, but as with spam detection, they are assessed on fraud dataset [24] from 2008. Lee and Verma [22] suggested a model based on Recurrent Convolutional Neural Networks using the textual message and evaluated the addition of headers. They also identified and solved four spammer strategies for poisoning text tricks. However, their proposals may exhibit weaknesses in other spammer strategies, such as hidden text or image-based messages. Additionally, it is worth considering the formal and legitimate writing style spammers use to create phishing emails.

Machine Learning models need a significant investigation in feature engineering to face new phishing scams [25]. Following this approach, some studies have focused on discriminating features from both emails and users to fix how spammers achieve their objectives [26]. Aassal et al. [27] introduced a novel taxonomy of features for phishing emails, websites, and URLs and also proposed a framework, namely PhishBench, as a benchmark to compare new models. For emails, the taxonomy contains header-based text and links clustered in body-based and attachment features. Gangavarapu et al. [28] conducted an extensive survey on diverse textual feature engineering in detecting spam and phishing emails.

Bountakas and Xenakis [8] proposed a tool to spot phishing emails through a feature extraction based on NLP and two Machine Learning methods. They used four groups of features: body, syntactic, headers, and URLs. Body included features related to the structure of the email, if it had any HTML code, if it contained scripts, and the number of attachments and image links. The syntactic group comprised diverse linguistic characteristics found in phishing emails, which differed from benign ones; they counted the frequent words, distinct words and email parts of both classes using the subject and body and calculated the richness. Regarding the header group, they only analysed email encoding and the number of recipients. Finally, URLs contained the IP of URLs, number of hyperlinks, if text and HREF differed, number of dots, port URLs and domains. Although the results of their models achieved a very high performance, reaching 0.994 of F1-Score, they evaluated them on 2000s decade datasets plus a recent phishing email dataset (the Nazario Phishing Corpus,[3]). Moreover, they designed a set of features that relied on volatile characteristics and focused on phishing emails, not including other scams. As Jáñez-Martino et al. [11] empirically shown, filters should consider dataset shifts and the presence of adversarial elements.

2.2. Fraudulent email detection

Although phishing emails may be considered as the most prevalent form of fraud within spam emails, other types of scams, such as spreading malware and ransomware, are also present [29]. Bera et al. [9] expanded the analysis to include various scams beyond phishing emails and identified eight different intentions among spammers. These intentions encompassed tactics such as emotion arousing, personalisation, and impersonation. Their dataset included a set of spam and ham emails from the 2000s — Ling Spam, SpamAssassin and Enron — along with phishing emails from the repository of Nazario and spear phishing emails from Kaggle.

Gallo et al. [7] divided spam email into *not relevant* and *critical* spam, which had enough danger to create a cybersecurity incident. They reported on the activities performed during two years (2018–2020) in an anti-phishing group belonging to the biggest Italian technology company. They collected spam emails and annotated them according to employee reports. They targeted two classes, with low or no degree of danger (*not relevant spam*) and potentially risky, capable of creating a security incident (*critical spam*). This last class was mainly focused on cybersecurity issues encountered by company employees, such as malware propagation, (spear) phishing, CEO fraud and scams.

Their dataset contained 3931 critical spam cases and 18,001 non-relevant emails. They designed seven sets of features — *General, Content, View, Subject, Links, Attachments* and *Others* — comprising a total of 79 features. *General* extracted information from SMTP headers, if any server was blacklisted, email size, number of recipients or country/continent of email origin. *Content* contained NLP features from the email message, such as language, the rate of words and word types, readability and scammy and phishing predefined vocabulary. *View* obtained cognitive visual perception, including screenshot width and height and hidden text recognition. *Subject* grouped the number

[3] https://monkey.org/~jose/phishing/README.txt Retrieved November 2024.

of words and characters of the subject, “FWD” inclusion and if it contained non-ASCII characters. *Links* encompassed features about the number of links, domains, and VirusTotal analysis to find malicious and unknown links. *Attachments* included the number of attachments and their types and malware analysis using VirusTotal.

Finally, they clustered other information related to the company and Threat Intelligence activities into a set called *Others*. Using these features, they constructed a pipeline to feed Random Forest and SVM classifiers. After analysing the importance of each feature, they achieved — with 36 features and Random Forest — the highest performance 0.933 of F1-Score. They found out that *Content* set and features as *hidden text, number of links, email size* and *links analysis* were the most important features. In terms of limitations, they highlighted the challenges associated with relying on third-party sources, which included financial and computational drawbacks due to licensing restrictions. Additionally, they acknowledged the lack of protection for external users, such as single victims, outside the scope of a company.

3. Feature extraction

In our work, we followed the recommendations and critical aspects indicated by cybersecurity experts to identify potentially risky emails. Based on them, we extracted 56 features, grouped into five sets, depending on the analysed email elements, namely *headers, content, attachments, URLs*, and *protocols*. Using this set of features we can feed data into our Machine Learning classifiers and estimators, providing also relevant information for cybersecurity experts. Table 1 summarises our feature sets.

3.1. Headers

Emails are accompanied by metadata, commonly known as headers, which provide comprehensive information. Some headers are mandatory, such as “To”, “From” or “Date”, and others are optional but highly recommended and used, like “Subject” or “Cc”. Users can provide an extensive list of headers to add security and detailed information, such as “Received” and “Reply-to”. However, most headers are easily editable by spammers, who take advantage of them to mislead users and anti-spam filters [30]. While other headers such as “Received” or protocols (see Section 3.5) provide more robust assurance, they are also susceptible to being duplicated and slightly modified [30].

Spammers take advantage of the ease of modifying the “From” header and employ various strategies to hide their identity [28]. Cybersecurity experts point out that the most common techniques are (i) not including an address, (ii) including multiple addresses and (iii) using the recipient’s address. Moreover, due to the importance of the sender address, we have followed an extensive investigation about it on [10]. Our goal was to develop an automated method for measuring the quality of the sender’s address since it plays a vital role in shaping the initial impression of the recipients, ultimately influencing their level of trust or mistrust. While the “To” header is mutable, the “Received” header is intentionally designed to be immutable and cybersecurity experts recommend its use. Hence, spammers can attempt to bypass this advantage by adding multiple entries, which complicates header identification.

We process the features of mandatory headers, i.e., “From” and “To”. Apart from the mandatory headers, we focus on “Reply-to” and “Received” due to their widespread use by scammers. We decide to group protocol features separately, even though they are indicated in the email headers, as they have a lower prevalence and distinct properties (cf. Fig. 3). We also analyse the optional header “Subject” along with the remaining textual content of the email message. Consequently, we include the headers “From”, “To”, “Reply-to”, and “Received” in this group as they encompass the sender, recipient/s nicknames, domains, and addresses. Spammers often leverage this information in other email parts like introducing the recipient’s name in the body. We also include these interactions within features belonging to this group.

3.2. Text

As we have already mentioned in Section 2, previous studies have developed their models by starting from textual information, crawling semantic and contextual features, and applying a text classification approach. The textual content is the essential part that allows spammers to interact with users and persuade them to perform an action with the rest of the email, e.g., attachments or links. Therefore, they adapt the writing style and topic according to their target. Saidini et al. [12] demonstrated the relevance of semantic features and categorisation in effectively filtering spam emails. The textual information in an email is a combination of the subject header, body content, and any attached or hyperlinked image. We extract this textual information following the methodology proposed by Jáñez-Martino et al. [4].

Spammers apply some of their strategies to these parts [11], i.e., the “Subject” or the body of the email. Finding a spammer’s strategy often indicates the spam email may contain some risky and harmful elements or, at least, something that should be examined. Moreover, we can find brand imitations in spoofing, phishing, and other scams, and cryptocurrency wallets for the black economy, e.g., in extortion or emails with fake job offers [4]. It is common to see references to amounts of money in emails with fake rewards, gifts, or discounts [4]. We try to capture all this information in this set of features.

According to Gallo et al. [7], readability is essential in malicious spam emails. In contrast with them, who used just one readability measure, we selected four of the most popular readability scores in English and Spanish. For English, we selected the Flesch Reading Ease score [31], the Flesch-Kincaid Grade Level [32], the Simple Measure of Gobbledygook (SMOG) index [33] and Gunning Fog [34]. For Spanish, we chose the Fernandez-Huerta score [35], the Gutierrez score [36], Szigriszt-Pazos [37] and readability μ [38].

Spam emails contain various topics, such as health, advertisements or communications, and different writing styles [14]. We can effectively encapsulate this range of options by assessing readability using a set of four indexes. For example, the SMOG index is mainly used in the healthcare sector, while Flesch-Kincaid is ideal for advertisements.

To detect the presence of hidden text in spam emails, we follow Eq. (1), which computes the ratio between text extracted from HTML code, which may include hidden text, and text extracted using OCR (Optical Character Recognition), which only contains the text visible in the email. This ratio indicates the proportion of each type of text. Then, if the ratio exceeds a value of 2, the email is marked as containing hidden text. This threshold was established through experimental analysis, involving a manual inspection of emails to identify those containing hidden text. The minimum value for effective discrimination was then determined based on these observations.

$$\text{TextRate} = \frac{\text{Text from HTML code}}{\text{Text from OCR recognition}} \tag{1}$$

Regarding the cybersecurity topics, we decided to assign a specific feature to each class based on the output of the best models per language among the ones evaluated in [4]. Cybersecurity topics refer to the classification of spam emails that are of interest to cybersecurity experts to distinguish those emails associated with certain suspicious activities, such as fake rewards, phishing, malware, black economy, or illicit pharmacies. To avoid noise, we discarded the class “Other”, leaving ten classes as features. This allows us to examine the importance of each class and its relationship to malicious emails.

Furthermore, we compiled a list of the top 100 valuable brands worldwide and the top 100 valuable brands in Spain,[4] as our dataset contains emails from Spain (see Section 4.1). Additionally, we created a list of currency symbols that include unique currency symbols and their common variants, e.g., the symbol of euro (€) and its “EUR” variant.

[4] We consulted the most valuable brands in 2022 both worldwide and only Spain on Statista (https://en.statista.com).

Table 1
The 56 features extracted from the spam email grouped in five categories: Headers, Text, Attachments, URLs and Protocols. Cybersecurity topics follows [4]. In value B stands for binary response — 0 is a negative value, 1 is positive —, R refers to real numbers and Z encompasses the number zero, a positive or negative integer.

Group	Feature	Description	Value
Headers	sender_same_receptor	Sender address has been replaced using same address as receptor	B
	address_quality	Machine Learning model scores the quality of an address to be trustworthy for users [10]	0–100
	fake_sender	Sender address has been modified to mislead the user	B
	many_received_headers	Email contains many headers of “Received”. It can be legitimate, but usually spammers add them	B
	received_from_match	Both “Received” and “From” headers match the domains	B
	has_replyto	Email contains “Reply-to” head	B
	replyto_match_from	Both “Reply-to” and “From” headers match the domains	B
	replyto_match_received	Both “Reply-to” and “Received” headers match the domains	B
	sender_company_name	Nickname or address of sender contain a company brand	B
	hide_sender_address	The “From” header is empty	B
	equal_domain_url_sender	The domain of sender is in email URLs	B
	equal_domain_url_sender_match	The domain of sender is the same as email URLs	B
	receptor_in_text	Nickname or address of the sender is in the email body.	B
	receptor_in_url	Nickname or address of sender is in the email URLs.	B
Text	readability_1	Readability score based on Flesh Reading Ease and Fernández-Huerta in English and Spanish, respectively	1–100/0–100
	readability_2	Readability score based on Flesh Kincaid Grade Level and Gutiérrez in English and Spanish, respectively	0–18/Z
	readability_3	Readability score based on SMOG index and Szigriszt-Pazos in English and Spanish, respectively	0–500/0–100
	readability_4	Readability score based on Gunning Fog Index and Legibilidad μ in English and Spanish, respectively	0–20/0–100
	has_hidden_text	Email contains text hidden in the background	B
	is_academic_media	The cybersecurity topic: academic media	B
	is_extortion_hacking	The cybersecurity topic: extortion hacking	B
	is_fake_reward	The cybersecurity topic: fake reward	B
	is_health	The cybersecurity topic: health	B
	is_identity_fraud	The cybersecurity topic: identity fraud	B
	is_money_making	The cybersecurity topic: money making	B
	is_pharmacy	The cybersecurity topic: pharmacy	B
	is_service	The cybersecurity topic: service	B
	is_sexual_content_dating	The cybersecurity topic: sexual content dating	B
	is_work_offer	The cybersecurity topic: work offer	B
	has_brand_in_text	Textual information of the email contains brand/s	B
	counter_brand_occurrences	Number of appearances of brand names inside textual information	R
	has_currency	Textual information of the email contains currency symbols	B
	counter_currency_occurrences	Number of appearances of currency symbols inside textual information	R
	has_cryptocurrency	Textual information of the email contains cryptocurrency wallets	B
	counter_cryptocurrency_occurrences	Number of appearances of cryptocurrency wallets inside textual information	R
Attachments	has_attachment	Email contains one or more attachments	B
	is_attached	One attachment at least is attached, not inline form	B
	many_formats	One attachment at least contains several formats	B
	has_disk_image_format	One attachment at least is disk image format	B
	has_executable_format	One attachment at least is executable format	B
	has_compressed_format	One attachment at least is compressed format	B
	has_office_macro_format	One attachment at least is macro office format	B
	att_word_in_message	Textual information of email contains vocabulary related to opening attached files	B
URLs	has_urls_repeated	One or more URLs in the email are duplicated	B
	has_urls_unique	Email does not have any URLs duplicated	B
	count_urls_occurrences	Number of URLs inside the email, even if they are duplicated	R
	has_vocab_url	Textual information of email contains vocabulary related to clicking on URLs links	B
	has_phishing_malware	One or more URLs are suspicious to contain malware or phishing attacks	B
	url_infected_counter	Number of URLs inside the email, even if they are duplicated, suspicious to contain malware or phishing attacks	R
Protocols	has_dkim_signature	Email headers contain DKIM protocol	B
	dkim_contain_all_parameters	DKIM protocol contains the nine parameters	B
	dkim_domain_match	Domain of DKIM protocol and from domain match	B
	has_spf_signature	Email headers contains SPF protocol	B
	has_dmarc_authentication	Email headers contains DMARC authentication	B

We prepared regular expressions to match three popular and widely used cryptocurrencies mentioned in these emails: Bitcoin, Ethereum, and Monero, along with their wallet format.

3.3. Attachments

Cybersecurity experts point out that attachments and URLs in spam emails are common entry points for attacks. We detect whether an email has attachments and determine the number of attachments present. Spammers employ tactics to obfuscate the true format of attached files by using multiple file extensions, such as *docx.htm* [39]. We identify emails with attachments that exhibit this property. Attached files can serve as carriers for hidden ransomware and malware, often disguised as macros within office packages like Word or Excel, as well as executable or image-disk files. Building upon the research by [7], we have expanded the range of attachment-type features and based our analysis on verifying the file format of the attachments. We detected executable, image-disk, compressed, and office macro formats. We

Table 2
List of the most frequent words, both in English and in Spanish, that call for actions related to attachments in spam emails.

Attachment words				
attached	attach	attachment	attachments	
adjunto	adjuntos	adjuntar	adjunta	adjuntas

compiled the lists in base of FileInfo,[5] in December 2022. We added *txt*, *html*, *htm*, and *pdf* formats to the macro office list. Finally, we analysed the email message to identify vocabulary that calls for opening attachments (Table 2).

3.4. URLs

Links within emails have proved to be the most common harmful element for users, as they establish connections with external sources and can be easily hidden inside other elements [8]. Spammers actively entice users to click on URLs, exposing them to sophisticated scams such as phishing attempts, fraudulent websites, or fake dating pages. Other links can connect to email accounts and WhatsApp numbers. Based on our analysis, there are two main strategies used by spammers: the use of the same link inserted in many elements — images, hyperlinks, or direct URLs — or the use of different links associated with these elements. We capture both approaches and the number of links inside the email, even if they are duplicated.

As in the case of attachments, we created a common vocabulary used by spammers to refer to clicking the URLs (Table 3). To obtain this list, we analysed our dataset Spam Email Risk Classification (SERC) to identify the most frequently occurring words with hyperlinks (above 100 occurrences).

Finally, we incorporated a phishing detector based on Machine Learning [40] to spot the links with suspicious content. We converted the classifier output into a binary feature and counted the number of links with suspicious content. This allows us to avoid the dependency of requiring third-party services like VirusTotal[6] to look for malicious links.

3.5. Protocols

There are some authentication methods, such as Sender Policy Framework (SPF), Domain Keys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting, and Conformance (DMARC), which provide proof that an email is legitimate [30].

SPF is an authentication protocol that uses a DNS TXT record to register IP addresses that are authorised to send emails on behalf of specific domains. SPF is susceptible to being broken when a message is forwarded, and it does not provide sufficient protection for brands against malicious actors who may spoof the display name or Friendly-From address. These shortcomings were overcome by the creation of the DKIM protocol.

DKIM is like a passport that allows verification of the email server from which it was sent using an encrypted key pair (one public in the DNS and one private). The DKIM protocol validates the authenticity of the sender and identifies if the message was altered or tampered during transit. DKIM includes seven mandatory tags. The DKIM protocol has been identified as the most commonly used protocol header by spammers (cf. Fig. 3). Based on this observation, our initial verification process involves checking whether a spam email contains DKIM and ensuring the presence of all mandatory tags.[7]

DMARC serves as an email authentication, policy, and reporting protocol. Its purpose is to assist domains in combating domain spoofing and phishing attacks by mitigating the unauthorised use of the domain in the Friendly-From address of email messages (nickname plus the address of the emails).

4. Methodology

After defining and designing the features that we used, this Section describes the creation of the corpora, explaining both the classification and regression approaches.

4.1. Dataset for spam risk evaluation

To the best of our knowledge, there are no publicly available datasets containing spam emails labelled based on their risk. Assuming that an email contains spam with “risk”, we want to measure how dangerous or harmful that email could be to an end user. Hence, we have created our custom dataset Spam Email Risk Classification (SERC). This dataset is obtained by merging two sub-datasets distinguished by variations in email sources and their public accessibility, Spam Email Risk Classification - Bruce Guenter (SERC-BG) and Spam Email Risk Classification - Internal (SERC-I).

A set of 2500 spam emails randomly collected from each dataset, i.e., SERC-I and SERC-BG, was annotated based on the knowledge of cybersecurity experts. Our regression approach predicts the level of risk of an incoming spam email on a scale of 1 to 10, with 1 being the lowest risk, and 10 being the highest. Our classification approach is a binary model containing *low* and *high risk* classes. The *low* class includes those spam emails with minimum risk, whereas the *high* class contains potentially harmful spam that could target both organisations and individuals. We started with the annotation of the regression approach by defining a range of values from 1 to 10. Using the range of risk levels as a baseline, the *low* class includes spam emails annotated from 1 to 6 and the *high* class those labelled from 7 to 10. The threshold was determined by manually inspecting a combined total of 50 emails from both datasets — 25 from SERC-I and 25 from SERC-BG — across each of the conflicting risk levels, specifically, 5, 6, and 7.

Finally, our dataset from INCIBE, namely Spam Email Risk Classification - Internal (SERC-I), contains 297 and 1447 instances in the low and high class, respectively, for 1744 spam emails. SERC-I includes 712 email written in English (40.83%) and 1032 in Spanish (59.17%). We refer to the dataset collected from the Bruce Guenter repository as Spam Email Risk Classification - Bruce Guenter (SERC-BG), which contains 1044 and 861 emails of *low* and *high* class, respectively, i.e., 1905 spam emails in total. SERG-BG are 1886 emails written in English (99.99%) and 19 in Spanish (0.01%). We refer to the combination of both datasets as Spam Email Risk Classification 4007 (SERC). Fig. 2 depicts the statistical information on both datasets from a regression perspective.

During our collaboration with INCIBE they provided us with batches of spam emails. Their honeypots have collected emails like those found on individuals’ mailboxes. Most of these emails are mainly written in both English or Spanish. We have therefore included both languages for our models. We have selected spam emails dated in June 2022 to cover the most recent threats.

Gallo et al. [7] focused their work on spam emails reported by company employees, while we aim to include individuals as targets as well, thus, making our dataset more general-purpose and encompassing a wider range of email users. Consequently, we prepared two datasets from different sources: one provided by INCIBE, which cannot be disclosed publicly due to the lack of authorisation to reveal personal data, and another publicly accessible to everyone. This approach enables the replication of our experiments by the scientific community and extends the applicability of our research, while also optimising our model and respecting the INCIBE requirements.

[5] https://fileinfo.com/ retrieved November 2024.

[6] https://www.virustotal.com/gui/home/upload retrieved November 2024.

[7] https://datatracker.ietf.org/doc/html/rfc6376 retrieved in November 2024.

Table 3
List of the most frequent words that call for actions related to URLs in spam emails from our dataset SERC. The frequency of words decreases by rows. The words are divided based on the language, common URLs words refer to those words containing social media platforms or popular brands.

English URLs words				
click here	advertisement	unsubcribe	have a good cashless	
Common URLs words				
amazon.com	instagram	facebook	twitter	dusbes
visa	amazon	emmail	securitas direct	
Spanish URLs words				
aquí	haga click	para desuscribirse hacer click en este enlace	detalles del producto	ver mensaje en la web
reclama ya	elige tarifa y ahorra	inscribirse	programa calcula online	
detalles del producto				

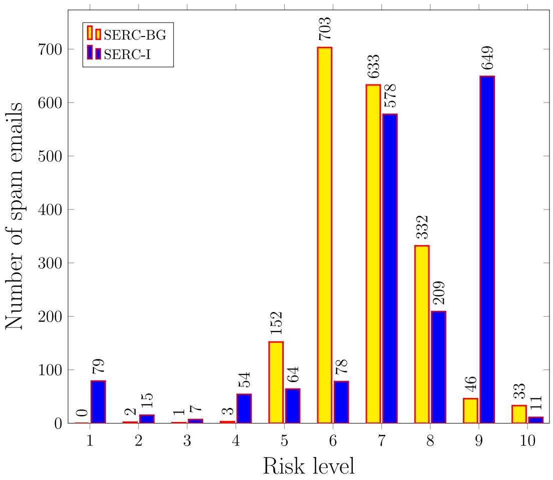


Fig. 2. Statistical analysis of the number of spam emails in each level of the risk range 1–10 for the SERC-I and SERC-BG datasets (regression approach). The impact of data sources on the distribution of risk levels is evident, with SERC-BG provided by a personal honeypot and SERC-I by a cybersecurity organisation. The SERC-BG distribution is more generic, with most emails concentrated in the middle of the risk levels. While the SERC-I distribution shows a clear positive bias.

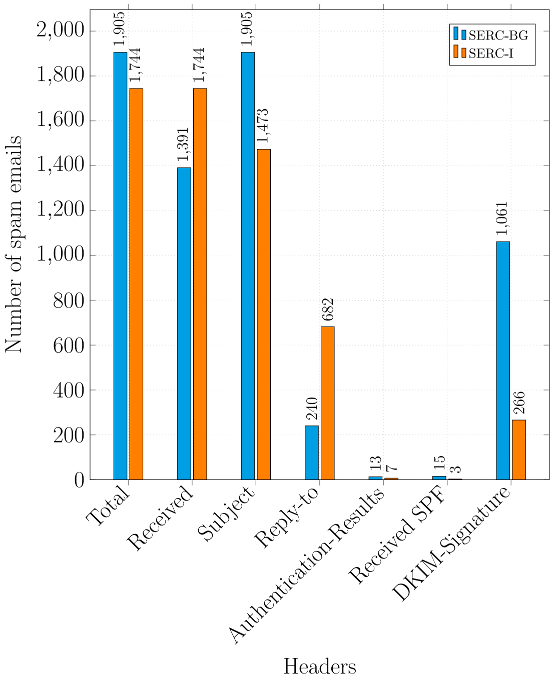


Fig. 3. Number of occurrences of the headers used in our feature set for SERC-BG and SERC-I. It is worth noting that all emails from SERC-I have a "Received" header and, similarly, for SERC-BG, the "Subject" header is consistently present. The "Reply-to" header is present in almost nearly three times the number of emails, compared to SERC-BG. Regarding security protocols, SPF and DMARC ("Authentication-Results" header) are hardly used in both datasets, while the DKIM protocol is more common in SERC-BG than in SERC-I.

In order to provide a publicly available dataset, we collected spam emails from the Bruce Guenter repository, also known as Spam Archive.[8] This repository has published spam emails collected in the personal honeypots of Bruce Guenter since 1999. Several previous works [11,41,42] have performed their experiments using subsets of this source. Moreover, the collection environment closely resembles individual user mailboxes, potentially enabling us to discover emails that resemble those encountered by users daily. In 2018, Bruce Guenter let his main domain licence expire, significantly reducing the number of monthly emails recollected and their characteristics [11]. While the majority of emails are written in English, there is a significant number of messages written in other languages such as Spanish, French, or Japanese [11]. We discarded all languages except English and Spanish. Our selection of spam emails dates from April 2022 to May 2023 to encompass an entire year, rather than focusing solely on the most recent ones, as we did with the INCIBE dataset. During this process, we specifically included only emails with attachments to ensure the presence of this type of content in the final dataset.

In terms of header prevalence, we analysed both datasets to determine the frequency of spam emails containing each header used for creating our feature set. We do not include the mandatory headers "To" and "From" in our analysis. We have summarised this information in our datasets SERC-BG and SERC-I in Fig. 3. Based on the observations from the data analysis, In SERC-BG, 73.01% of emails have a "Received" header. In contrast, every email in SERC-I contains this header. The "Subject" header is present in all emails from SERC-BG (100%) and almost all emails from SERC-I (99.93%). The "Reply-to" header is found in 12.60% of SERC-BG emails and 39.11% of SERC-I emails. The presence of SPF and DMARC protocols is extremely low in both datasets, being less than 0.01%. Notably, the DKIM protocol is used in a significantly higher proportion of emails from SERC-BG (55.70%) compared to SERC-I (15.25%)

4.2. Classification and regression models

For the classification task, we evaluated the following models: Random Forest (RF) [43], Support Vector Machine (SVM) [44] and the Logistic Regression (LR) model [45]. The first two models have been evaluated and recommended by [7]. The LR model was also selected for its simplicity and the good performance shown in other natural language applications [4,46,47]. Due to the negative values

[8] http://untroubled.org/spam/, retrieved November 2024.

of Gutiérrez readability score in one of the features, we discarded the Naïve Bayes classifier [48].

Our regression approach presents a novel solution for assessing the risk associated with spam emails. We selected well-known estimators such as the Support Vector Regressor (SVR) [44], Random Forest Regressor (RFR) [43] and Linear Regression (LinR) [49].

Algorithm 1 Email Spam Risk Regression and Classification

```
Inputs: Path of the spam emails
Extract headers from the emails
Extract content from the body and attachments
if Headers group is used then
    Extract 14 features from the headers set
end if
if Text group is used then
    Extract 21 features from the text set
end if
if Attachments group is used then
    Extract 10 features from the attachments set
end if
if URLs group is used then
    Extract six features from the URL set
end if
if Protocols group is used then
    Extract five features from the protocols set
end if
return: Regression output: risk level (1 to 10) and class (low-risk
    or high-risk)
```

4.3. Evaluation metrics

Regarding the binary classification approach, we have selected four widely used metrics in the literature [11,46]: Accuracy, Precision, Recall and F1-Score. In addition, we used Cross-Entropy metric to determine the variability of probabilities returned by the classifiers. To define these metrics, we first establish the following variables: **True Positive (TP)** occurs when the model correctly predicts the positive class. **True Negative (TN)** occurs when the model correctly predicts the negative class.**False Positive (FP)** occurs when the model incorrectly predicts the positive class. **False Negative (FN)** occurs when the model incorrectly predicts the negative class.

Accuracy is calculated as the ratio of correct predictions (TP and TN) to the total number of samples (Eq. (2)). Accuracy measures how often the model makes correct predictions, as it is calculated as the ratio of correctly predicted instances (true positives and true negatives) to the total number of instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

Precision measures the ratio of positive predictions that are actually correct (Eq. (3)). This metric is crucial to detect when false positives (FP) have a significant cost.

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

Recall measures the ratio of actual positives that were correctly identified by the model (Eq. (4)). It is particularly important in scenarios where the cost of missing positives cases (FN) is high.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

F1-Score is the harmonic mean of precision and recall, which provides information about the balance of FP and FN (Eq. (5)).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Cross Entropy measures the difference between two probability distributions: the true distribution y_i, ground truth labels, and the predicted distribution $\hat{y}_i$, model's predicted probabilities (Eq. (6)).

$$\text{Cross-Entropy} = \left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \right] \quad (6)$$

Regarding the regression approach, we used Mean Absolute Error (MAE), Mean Square Error (MSE) and Root Mean Square Error (RMSE) to evaluate our models.

Mean Absolute Error (MAE) measures the average absolute difference between the actual values y_i and the predicted values $\hat{y}_i$, providing the average prediction error in the same units as the original data (Eq. (7)).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (7)$$

Mean Squared Error (MSE) measures the average of the squared differences between the actual and predicted values, making it more sensitive to outliers by highlighting larger errors (Eq. (8)).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (8)$$

Root Mean Square Error(RMSE) is the square root of the Mean Squared Error (MSE) and is commonly used to detect large errors (Eq. (9)).

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (9)$$

5. Experimentation

5.1. Experimentation setup

We carried out the experiments on a computer with 128 GB of RAM, two processors Intel Xeon E5-2630v3 of 2,4 GHz and two Nvidia Titan Xp.

We evaluated our complete set of features on two datasets, SERC-BG and SERC-I, which contain spam emails in English and Spanish (see Section 4). Additionally, we also used the entire dataset (SERC).

Although all the datasets in the classification problem are nearly balanced, we adjusted the class imbalance parameter to compensate for any slight difference. We used the Scikit-learn[9] Python package to implement the pipelines. To determine the optimal combination of model parameters, we conducted a parameter evaluation using the GridSearch function from the Scikit-learn for both classification and regression.

Next, we indicate the values that were adjusted in each of the models used in the binary classification of spam, leaving the rest of the parameters of each model with their default values. We chose a C value of 100 and *sag* solver for the Logistic Regression model. We opted for a polynomial kernel for the SVM model and tuned the value of C to 1000. Regarding RF, we used 250 tree estimators with a maximum depth of 25, as well as the squared root, to calculate the number of features to look for the best split. Additionally, we selected 3 as the minimum number of samples required to split an internal node and 1 as the minimum number of samples required to be at a leaf node. We evaluated the performance with stratified-10-fold cross-validation and reported the accuracy, precision, recall, and F1-Score as performance metrics. In addition, we reported the widely used loss function, cross-entropy. When training a model by minimising cross-entropy, the resulting model provides estimations of posterior probabilities.

In the case of the regression approach, we kept the remaining model parameters for the Linear Regression estimator. For SVR, we

[9] https://scikit-learn.org/stable/ retrieved November 2024.

Table 4
Performance of every model on our three datasets (SERC-BG, SERC-I and SERC) in terms of Precision, Recall, F1-Score, Accuracy and Cross Entropy. The classifiers LR, SVM and RF have been fed up using the feature sets.

Classifiers/Metrics	SERC-BG			SERC-I			SERC		
	LR	SVM	RF	LR	SVM	RF	LR	SVM	RF
Precision	0.796 ±0.041	0.852 ±0.025	0.875 ±0.031	0.964 ±0.032	0.949 ±0.026	0.954 ±0.025	0.807 ±0.064	0.883 ±0.032	0.912 ±0.031
Recall	0.611 ±0.066	0.822 ±0.026	0.839 ±0.054	0.882 ±0.054	0.965 ±0.022	0.970 ±0.021	0.655 ±0.172	0.905 ±0.084	0.918 ±0.060
F1-Score	0.690 ±0.052	0.836 ±0.040	**0.855** **±0.040**	0.921 ±0.035	0.957 ±0.023	**0.962** **±0.029**	0.712 ±0.126	0.893 ±0.040	**0.914** **±0.040**
Accuracy	0.702 ±0.042	0.824 ±0.033	0.846 ±0.048	0.875 ±0.040	0.928 ±0.037	0.935 ±0.026	0.659 ±0.118	0.854 ±0.057	0.884 ±0.062
Cross Entropy	0.602 ±0.026	0.396 ±0.025	0.587 ±0.048	0.611 ±0.035	0.258 ±0.020	0.448 ±0.048	0.372 ±0.099	0.423 ±0.049	0.385 ±0.062

Table 5
Extraction runtime in seconds per email for each set of features and dataset (SERC-BG and SERC-I).

	Extraction runtime (s/email)				
	Headers	Text	Attachments	URLs	Protocols
SERC-BG	8.00	32.74	2.31	3.06	2.30
SERC-I	7.84	39.76	2.30	2.96	2.30

opted for a Radial Basis Function (RBF) as kernel and a C value of 1000. Regarding the RFR model, we configured 1000 estimators with a maximum depth of 25. We again used the squared root to calculate the maximum number of features and 1 as the minimum number of samples required to be at a leaf node. In this approach, we set 10 as the minimum number to split an internal node. We also assessed the model performance through stratified-10-fold cross-validation in terms of Mean Square Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). We also carried out a feature analysis to assess the impact of each feature both independently and within their respective groups on the model.

5.2. Classification results

We show the performance of our Machine Learning models using the entire set of features in Table 4. It is worth highlighting that the ground truth class labels were assigned by applying a threshold to a score assigned by the annotators, i.e., ranging from 1 to 10 (see Section 4.1). Since we consider this property plays an essential role in understanding the behaviour of the models. Moreover, we intend to conduct an analysis of the impact of features using sets and to assess the significance of each feature. Fig. 4 illustrates the performance of Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF) when removing one feature set (see Table 1) on SERC. Additionally, Fig. 5 shows the results obtained on the SERC by keeping only one group. This analysis of group importance becomes more relevant when examining feature extraction times (Table 5).

Fig. 6 shows a graph where the abscissa axis shows the number of features used in the feature set to feed the model and the ordinate axis shows the F1-Score value obtained. We added each feature sequentially, starting from the most relevant to the least relevant. The model was established when the 25th feature was added. However, it is worth highlighting that the model achieves an F1-Score of more than 0.900 from the inclusion of the 10th feature. Fig. 7 displays a sorted list illustrating the impurity-based importance of each feature, calculated using Gini importance [50].

5.3. Regression results

The results of the regression estimators are shown in Table 6. In contrast to the classification model, the estimators calibrated with

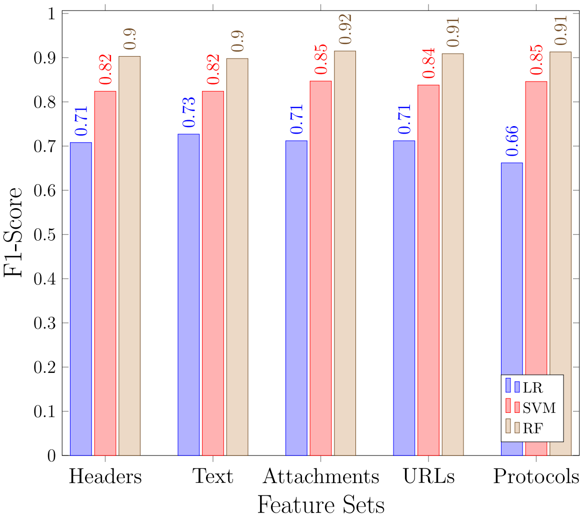


Fig. 4. Performance of Logistic Regression, Support Vector Machine and Random Forest classifiers when one group is removed from the feature set. The values in the abscissa axis indicate the removed group. There is no an evident reduction in the F1-score when compared to using all sets. A slightly noticeable reduction is when removing *Headers* or *Text* groups.

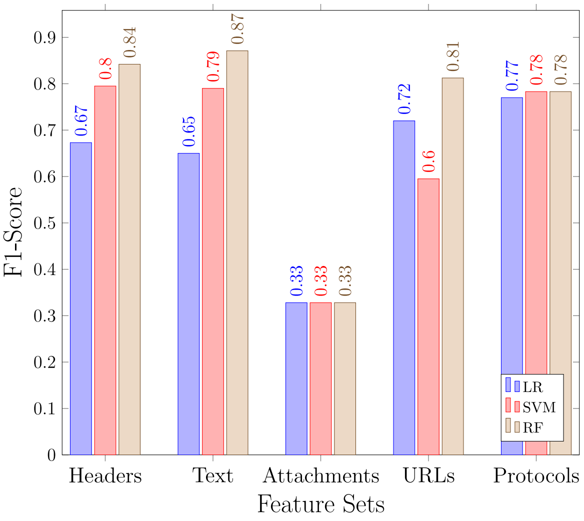


Fig. 5. Performance of Logistic Regression, Support Vector Machine, and Random Forest classifiers when only one group remains from the feature set. The values in the abscissa axis indicate the remaining group. It is worth noting the significant reduction when the *Attachments* group is used exclusively. There is also a noticeable reduction in the F1 score for *URLs* and *Protocols*, although much less than for attachments.

Table 6
Performance of every model on our three datasets (SERC-BG, SERC-I and SERC) in terms of Mean Absolute Error (MAE), Mean Square Error (MSE), and Coefficient of determination (R^2).

Estimators/Metrics	SERC-BG			SERC-I			SERC		
	LinR	SVR	RFR	LinR	SVR	RFR	LinR	SVR	RFR
MAE	0.659 ±0.039	0.617 ±0.047	0.526 ±0.049	0.768 ±0.038	0.537 ±0.099	0.516 ±0.047	0.807 ±0.098	0.598 ±0.118	0.537 ±0.119
MSE	0.808 ±0.137	0.858 ±0.080	**0.579** **±0.073**	1.455 ±0.347	1.393 ±0.332	**0.939** **±0.256**	1.301 ±0.205	1.096 ±0.225	**0.781** **±0.196**
R^2	0.262 ±0.132	0.216 ±0.066	0.472 ±0.061	0.627 ±0.071	0.641 ±0.138	0.758 ±0.133	0.488 ±0.102	0.568 ±0.139	0.692 ±0.077

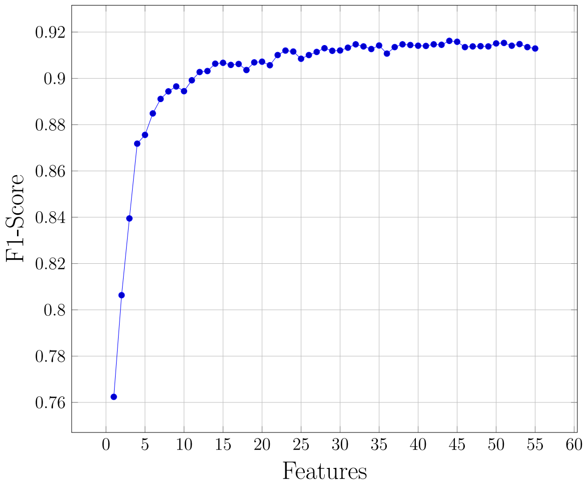


Fig. 6. F1-Score values of Random Forest model calibrated on SERC when adding up each feature one by one, starting from the most important to the least important. The Abscissa axis shows the number of features. The model was established when the 25th feature was added. However, it is interesting to observe the model achieves an F1-Score exceeding 0.900 starting from the 10th feature.

SERC-I exhibit lower performance than with SERC-BG in terms of MAE and MSE. This may be due to how the risk-level score is distributed in this dataset in which SERC-I was slightly biased to high levels of risk.

The MAE metric provides a direct measure of the error between the actual and predicted values. Upon observation, the range of MAE remains relatively consistent across all models, achieving lower values in the RFR estimators for all datasets.

The MSE metric provides insights into the model's behaviour concerning outliers. The results align with the distribution of both datasets, where SERC-BG contains more generic data, while SERC-I exhibits a bias towards high levels. Analysing this metric plays an essential role in real-world spam scenarios where the model can face spam campaigns with similar content or sophisticated and isolated cases targeting specific objectives.

The R^2 value may support this hypothesis, as estimators trained on SERC-BG exhibit a poorer fit to the observed data. Once again, the SERC dataset provides enough balance to train more robust models despite the increase in MSE. Among all the datasets, RFR achieved the highest performance in terms of MSE, with values of 0.579, 0.939, and 0.781 (for SERC-BG, SERC-I and SERC, respectively).

5.4. Comparison with state-of-the-art

Finally, we have conducted a comparative analysis between the pipeline that achieved the highest performance, i.e., our proposed features with RF, and a selection of widely recognised models prevalent in the literature. From a similar perspective that has been used in spam filtering [4,12], the detection of potentially risky spam emails can be addressed using a text classification approach. The growing adoption of deep learning has led to a significant increase in its use for analysing email content [51]. Across the evaluation of different text classification tasks, the RoBERTa-large model and its variants have consistently proved to be one of the most suitable models in various natural language processing tasks [52]. Given the multilingual nature of our SERC-I dataset, which predominantly comprises Spanish emails, we also opted for a cross-lingual and widely used Transformer, the large XLM-RoBERTa [53,54], which has achieved high performance in its evaluations [55]. To address the language issue with RoBERTa, we used the model trained in the BERTIN project [56] for Spanish emails and RoBERTa-large for English ones. However, this Spanish adaptation of RoBERTa is only available in the base version.

While attention-based Transformer models often dominate in the realm of text classification tasks [57], we also explored the performance of traditional pipelines with high performance metrics in some text classification tasks [11,52,58]. Kawintiranon et al. [59] conducted a comparison among different models, from traditional machine learning pipelines to Transformer models to detect spam in Twitter. We followed this methodology and evaluated the combination of two traditional text representation models, Term Frequency –Inverse Document Frequency (TF-IDF) and Bag of Words (BoW), along with the previous machine learning classifiers: LR, SVM and RF. Specifically, we evaluated combinations of TF-IDF and BoW text vectorisation techniques along with the previous assessed models (LR, SVM and RF).

We used the Python package simpleTransformers[10] to implement the models, as well as the Transformers library[11] to adapt the models for features input. We left their parameters as default, except the maximum sequence length, which we set to 512 in order to process the complete email text of the majority of the emails. We also modified the train batch size to the maximum value for our computational resources, i.e., 24, the number of iterations up to 6 epochs and a learning rate of 2e-6 to permit the models to undergo continual learning, following the configuration's recommendations of the original papers [60,61] and following works [52,54]. We followed the same configuration for both approaches, i.e., classification and regression. While Machine Learning classifiers and regression models take the feature set as input, for Transformer models we have followed two approaches: (a) to feed them directly with the text (T) and (b) to use both the text and our feature (T+F) set as input to the models. Regarding traditional techniques, we used the default parameter setup provided by the scikit-learn library[12] for text vectorisers and the same parameters as when we fed the model with our proposed set of features for Machine Learning models.

Finally, we evaluated the models TF-IDF and BoW along with LR, SVM, and RF for classification, as well as LinR, SVR, and RFR for regression. We also assessed RoBERTa base (R_b), RoBERTa large (R_l), XLMRoBERTa base ($XLMR_b$), and XLMRoBERTa large ($XLMR_l$). The results are shown in Table 7 for classification and Table 8 for regression. We reported the average and standard deviation of the performance metrics used in Table 4 for classification and Table 6 for regression models after a 10-fold cross-validation evaluation.

[10] https://simpleTransformers.ai/ retrieved November 2024.

[11] https://pypi.org/project/transformers/ retrieved November 2024.

[12] https://scikit-learn.org/stable/ November 2024.

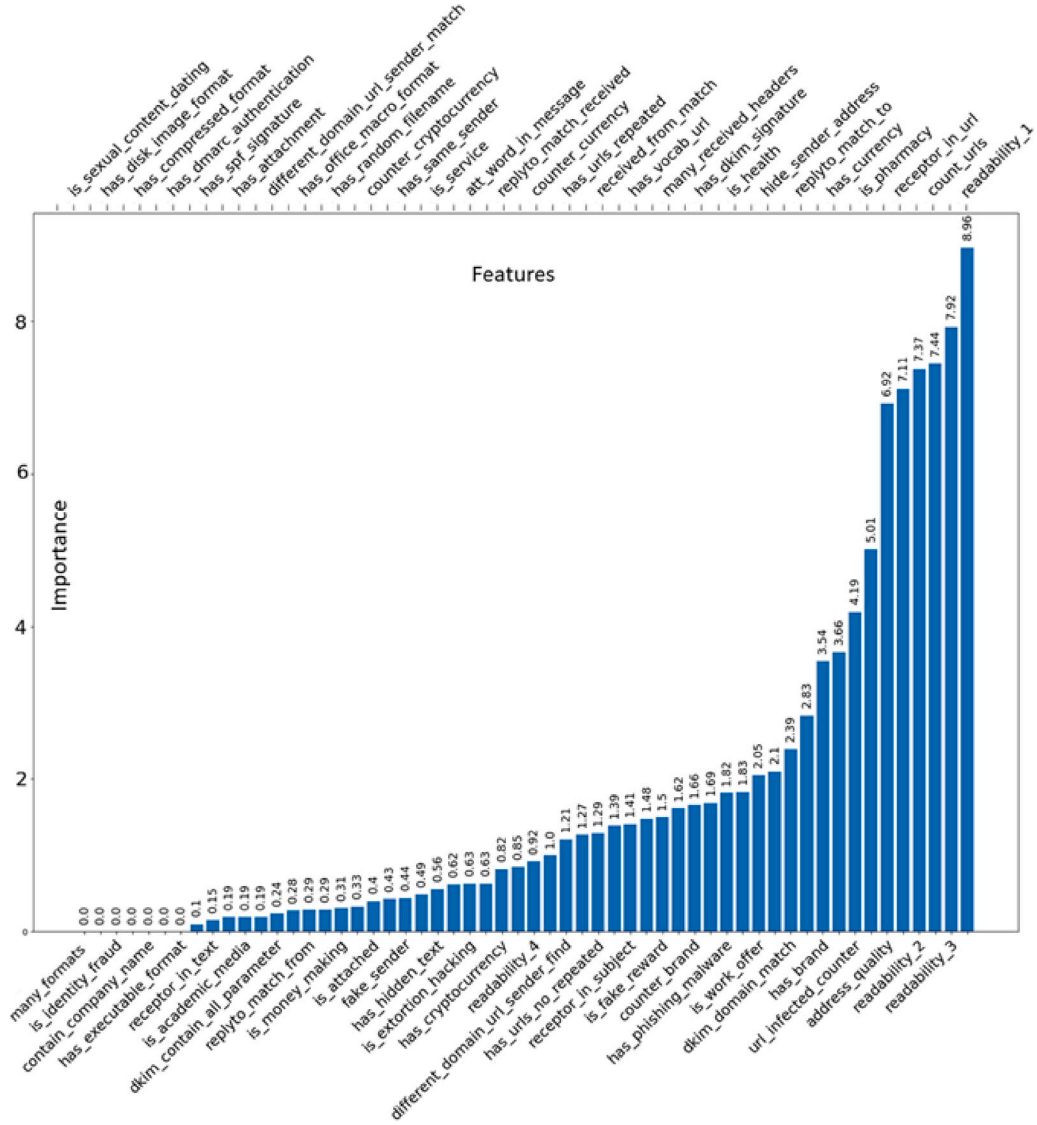


Fig. 7. Feature importance based on the Gini metric for classification approach using Random Forest. The values in the abscissas axis are displayed alternately on the inferior and superior axes to improve readiness. All values add up to 100. We can observe that six out of the 10 top values belong to the *Text* group, with three of them specifically related to readability. It is also worth noting that the address classification feature is in this top group.

6. Discussion

Regarding the classification, it is worth mentioning that the models show a decrease in performance on SERC-BG compared to the results obtained with the SERC-I dataset. This can be attributed to the significant percentage of emails of the Bruce Guenter repository data on risk levels 6 and 7 (see Fig. 2), which represent a borderline between the low and high-risk classes. Additionally, the SERC-BG dataset contains a relatively small number of extremely low-level emails (levels 1 and 2).

While the SERC-I dataset exhibits a distribution focused on medium-high levels with a notably higher number of elements in each level. All models achieved their highest F1-Score performance when being trained on the SERC-I dataset, with RF achieving an F1-Score of 0.962. The highest score for the SERC dataset was also obtained by RF, with an F1-Score of 0.914. The LR model trained on the SERC dataset showed a significant decrease in performance, while the SVM model only showed a minor decline. Hence, we may claim that the LR model is more sensitive to the training data distribution.

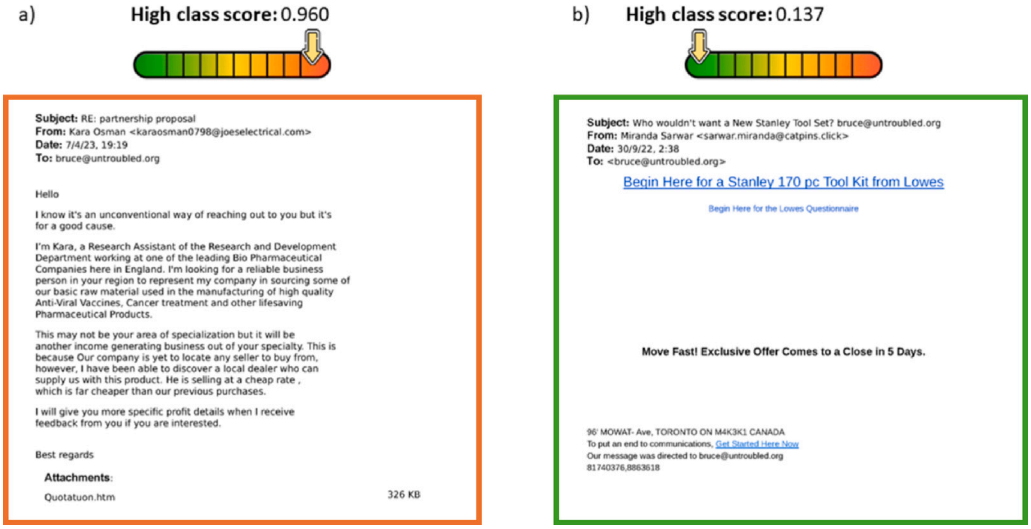
When analysing the precision and recall metrics, we observe that recall is the more sensitive metric because the cost of missing high-risk emails may be high. As a result, we find that the models trained on the SERC-BG dataset have lower recall in accurately detecting positive cases. This can be attributed to the fact that the high-risk class in the SERC-BG dataset primarily consists of borderline risk levels (level 7), with fewer instances of extreme high-risk cases. However, when considering cross-entropy, LR and RF models showed more robustness when being calibrated on the combination of both datasets (SERC). RF obtained the highest F1-Score in every training set assessed. As a result, we mainly conducted the feature evaluation on this classifier. This choice is made because the SERC dataset combines both SERC-BG and SERC-I, aligning with our primary research goal.

In comparison with state-of-the-art models, both the Transformer models and the traditional combinations yield lower F1-Scores across all datasets compared to the top-performing Random Forest (RF) model. This difference is especially remarkable when trained with the combined approach (SERC). Although these text classification pipelines can achieve results close to our proposed model, their focus only on textual data, crucial for classification or regression tasks, contrasts with our model's ability to offer additional insights for cybersecurity experts. Beyond performance metrics, our model extracts vital information encapsulated within the proposed 56 features.

Concerning the influence of the different groups of features on the performance of the models, the results indicate that removing one

Table 7
Performance of every model on our three datasets (SERC-BG, SERC-I and SERC) in terms of Precision, Recall, F1-Score, Accuracy and Cross Entropy. Evaluation of the models TF-IDF and BoW along with LR, SVM, and RF and RoBERTa base (R_b), RoBERTa large (R_l), XLMRoBERTa base ($XLMR_b$), and XLMRoBERTa large ($XLMR_l$) fed up with only text (T) or both textual and our proposed features (T+F) against our best model.

	SERC-BG					SERC-I					SERC				
	Acc	P	R	F1	CE	Acc	P	R	F1	CE	Acc	P	R	F1	CE
TF-IDF LR	0.834 ±0.065	0.843 ±0.059	0.837 ±0.064	0.833 ±0.066	0.445 ±0.100	0.930 ±0.039	0.893 ±0.075	0.850 ±0.082	0.867 ±0.078	0.295 ±0.343	0.834 ±0.061	0.846 ±0.054	0.790 ±0.087	0.791 ±0.090	0.426 ±0.079
TF-IDF SVM	0.824 ±0.084	0.834 ±0.079	0.822 ±0.083	0.820 ±0.086	0.507 ±0.218	0.936 ±0.041	0.894 ±0.076	0.874 ±0.084	0.882 ±0.079	0.239 ±0.206	0.846 ±0.070	0.852 ±0.065	0.824 ±0.050	0.823 ±0.066	0.480 ±0.215
TF-IDF RF	0.820 ±0.077	0.833 ±0.070	0.822 ±0.076	0.817 ±0.079	0.510 ±0.176	0.930 ±0.036	0.889 ±0.067	0.853 ±0.077	0.868 ±0.073	0.233 ±0.168	0.841 ±0.066	0.848 ±0.061	0.812 ±0.070	0.812 ±0.072	0.431 ±0.174
BoW LR	0.829 ±0.056	0.839 ±0.048	0.833 ±0.054	0.828 ±0.057	0.443 ±0.094	0.924 ±0.036	0.889 ±0.073	0.830 ±0.070	0.855 ±0.071	0.279 ±0.357	0.843 ±0.057	0.850 ±0.056	0.812 ±0.065	0.813 ±0.063	0.401 ±0.093
BoW SVM	0.757 ±0.068	0.779 ±0.074	0.747 ±0.068	0.746 ±0.071	0.602 ±0.091	0.935 ±0.039	0.890 ±0.073	0.873 ±0.080	0.880 ±0.076	0.226 ±0.192	0.776 ±0.049	0.793 ±0.062	0.695 ±0.084	0.695 ±0.092	0.539 ±0.078
BoW RF	0.689 ±0.066	0.761 ±0.054	0.707 ±0.062	0.675 ±0.076	0.754 ±0.197	0.913 ±0.036	0.859 ±0.069	0.822 ±0.070	0.837 ±0.068	0.336 ±0.102	0.722 ±0.061	0.720 ±0.109	0.589 ±0.086	0.570 ±0.107	0.735 ±0.258
R_b T+F	0.709 ±0.115	0.720 ±0.109	0.711 ±0.112	0.702 ±0.122	0.639 ±0.227	0.851 ±0.038	0.722 ±0.163	0.650 ±0.084	0.660 ±0.104	0.456 ±0.144	0.802 ±0.113	0.778 ±0.122	0.747 ±0.119	0.737 ±0.127	0.499 ±0.211
R_b T	0.833 ±0.018	0.834 ±0.017	0.835 ±0.018	0.833 ±0.018	0.388 ±0.037	0.839 ±0.007	0.494 ±0.094	0.512 ±0.017	0.476 ±0.029	0.252 ±0.018	0.878 ±0.022	0.841 ±0.066	0.802 ±0.052	0.800 ±0.053	0.272 ±0.022
$XLMR_b$ T+F	0.633 ±0.111	0.638 ±0.117	0.630 ±0.117	0.619 ±0.121	0.777 ±0.151	0.841 ±0.046	0.718 ±0.142	0.632 ±0.098	0.646 ±0.111	0.494 ±0.147	0.718 ±0.087	0.686 ±0.099	0.658 ±0.109	0.650 ±0.110	0.622 ±0.128
$XLMR_b$ T	0.826 ±0.026	0.827 ±0.026	0.826 ±0.026	0.825 ±0.026	0.420 ±0.026	0.835 ±0.015	0.497 ±0.166	0.521 ±0.050	0.488 ±0.079	0.308 ±0.023	0.855 ±0.014	0.842 ±0.015	0.818 ±0.026	0.827 ±0.021	0.353 ±0.020
R_l T+F	0.629 ±0.125	0.649 ±0.135	0.631 ±0.126	0.611 ±0.135	0.834 ±0.281	0.851 ±0.080	0.777 ±0.128	0.688 ±0.080	0.693 ±0.096	0.438 ±0.158	0.647 ±0.103	0.616 ±0.115	0.582 ±0.078	0.559 ±0.103	0.734 ±0.188
R_l T	0.841 ±0.030	0.842 ±0.029	0.841 ±0.032	0.840 ±0.031	0.371 ±0.057	0.931 ±0.034	0.889 ±0.114	0.839 ±0.103	0.856 ±0.106	0.215 ±0.047	0.900 ±0.021	0.892 ±0.022	0.874 ±0.042	0.879 ±0.036	0.301 ±0.054
$XLMR_l$ T+F	0.534 ±0.051	0.542 ±0.054	0.535 ±0.049	0.515 ±0.058	0.863 ±0.156	0.812 ±0.046	0.618 ±0.132	0.553 ±0.053	0.551 ±0.076	0.593 ±0.123	0.607 ±0.055	0.522 ±0.062	0.518 ±0.050	0.495 ±0.062	0.812 ±0.135
$XLMR_l$ T	0.841 ±0.025	0.842 ±0.025	0.842 ±0.026	0.840 ±0.025	0.400 ±0.051	0.876 ±0.034	0.781 ±0.187	0.656 ±0.108	0.677 ±0.137	0.261 ±0.023	0.885 ±0.012	0.866 ±0.014	0.871 ±0.018	0.868 ±0.015	0.304 ±0.022
Our model	0.875 ±0.031	0.839 ±0.054	0.855 ±0.040	**0.846** ±**0.048**	0.587 ±0.048	0.954 ±0.025	0.970 ±0.021	0.962 ±0.029	**0.935** ±**0.026**	0.448 ±0.048	0.912 ±0.031	0.918 ±0.060	0.914 ±0.040	**0.884** ±**0.062**	0.385 ±0.062



Subject: RE: partnership proposal
From: Kara Osman <karaosman0798@joeselectrical.com>
Date: 7/4/23, 19:19
To: bruce@untroubled.org

Hello

I know it's an unconventional way of reaching out to you but it's for a good cause.

I'm Kara, a Research Assistant of the Research and Development Department working at one of the leading Bio Pharmaceutical Companies here in England. I'm looking for a reliable business person in your region to represent my company in sourcing some of our basic raw material used in the manufacturing of high quality Anti-Viral Vaccines, Cancer treatment and other lifesaving Pharmaceutical Products.

This may not be your area of specialization but it will be another income generating business out of your specialty. This is because Our company is yet to locate any seller to buy from, however, I have been able to discover a local dealer who can supply us with this product. He is selling at a cheap rate , which is far cheaper than our previous purchases.

I will give you more specific profit details when I receive feedback from you if you are interested.

Best regards

Attachments:
Quotatuon.htm 326 KB

Subject: Who wouldn't want a New Stanley Tool Set? bruce@untroubled.org
From: Miranda Sarwar <sarwar.miranda@catpins.click>
Date: 30/9/22, 2:38
To: <bruce@untroubled.org>

Begin Here for a Stanley 170 pc Tool Kit from Lowes

Begin Here for the Lowes Questionnaire

Move Fast! Exclusive Offer Comes to a Close in 5 Days.

96' MOWAT- Ave, TORONTO ON M4K3K1 CANADA
To put an end to communications, Get Started Here Now
Our message was directed to bruce@untroubled.org
81740376,8863618

Fig. 8. Examples of the output from the *high* risk class score: (a) high score and (b) low score. Both examples have been collected from SPEC-BG. We can observe that the email (a) is more informative, well-constructed, and may resemble a legitimate email, which makes it more likely to deceive a user. The quality and authenticity of the message, as well as the attachment of a file with a suspicious format, explain the high score.

Table 8
Performance of every model on our three datasets (SERC-BG, SERC-I and SERC) in terms of Mean Absolute Error (MAE), Mean Square Error (MSE), and Coefficient of determination (R^2). Evaluation of the models TF-IDF and BoW along with LinR, SVR, and RFR and RoBERTa base (R_b), RoBERTa large (R_l), XLMRoBERTa base ($XLMR_b$), and XLMRoBERTa large ($XLMR_l$) fed up with only text (T) or both textual and our proposed features (T+F) against our best model.

	SERC-BG			SERC-I			SERC		
	MAE	MSE	R2	MAE	MSE	R2	MAE	MSE	R2
TF-IDF LinR	2.516 ±3.330	35.313 ±70.565	−37.557 ±76.859	0.975 ±0.417	4.319 ±5.387	−0.857 ±2.991	0.952 ±0.192	2.823 ±1.137	−0.842 ±1.104
TF-IDF SVR	0.573 ±0.095	0.684 ±0.221	0.350 ±0.117	0.525 ±0.202	1.056 ±0.932	0.687 ±0.171	0.545 ±0.110	0.832 ±0.506	0.546 ±0.197
TF-IDF RFR	0.611 ±0.093	0.734 ±0.232	0.306 ±0.101	0.630 ±0.209	1.135 ±0.929	0.656 ±0.174	0.700 ±0.120	1.072 ±0.546	0.393 ±0.269
BoW LinR	1.181 ±0.363	7.778 ±12.913	−5.568 ±9.230	0.743 ±0.210	2.275 ±1.384	0.237 ±0.526	1.022 ±0.509	4.264 ±6.240	−2.133 ±5.040
BoW SVR	0.693 ±0.109	0.968 ±0.333	0.080 ±0.212	0.526 ±0.205	1.115 ±0.914	0.655 ±0.194	0.628 ±0.170	1.067 ±0.561	0.363 ±0.359
BoW RFR	0.616 ±0.090	0.747 ±0.227	0.292 ±0.099	0.532 ±0.214	1.018 ±1.009	0.708 ±0.173	0.678 ±0.125	1.036 ±0.544	0.398 ±0.296
R_b T+F	1.058 ±0.180	2.253 ±1.094	−1.014 ±0.977	1.342 ±0.501	3.367 ±2.573	0.226 ±0.729	1.215 ±0.371	3.486 ±3.191	−0.042 ±0.668
R_b T	0.860 ±0.013	1.149 ±0.041	−0.028 ±0.028	2.431 ±0.124	8.592 ±0.557	−0.782 ±0.114	1.617 ±0.073	4.787 ±0.495	−0.178 ±0.130
$XLMR_b$ T+F	0.879 ±0.072	1.148 ±0.159	−0.024 ±0.120	1.222 ±0.098	2.624 ±0.410	0.333 ±0.095	0.876 ±0.068	1.427 ±0.189	0.450 ±0.074
$XLMR_b$ T	1.402 ±0.243	4.231 ±2.338	−2.803 ±2.115	1.806 ±0.132	5.734 ±0.767	−0.458 ±0.173	1.456 ±0.118	3.884 ±0.430	−0.496 ±0.167
R_l T	0.927 ±0.174	1.872 ±1.036	−0.663 ±0.908	1.275 ±1.275	3.163 ±3.163	0.366 ±0.366	1.101 ±1.101	2.461 ±2.461	0.161 ±0.161
R_l T	0.554 ±0.045	0.579 ±0.096	0.482 ±0.085	0.589 ±0.049	1.123 ±0.256	0.714 ±0.068	0.557 ±0.035	0.797 ±0.172	0.693 ±0.064
$XLMR_l$ T+F	1.104 ±0.120	2.439 ±0.999	−1.203 ±0.955	1.568 ±0.079	4.624 ±0.313	−0.178 ±0.079	1.331 ±0.089	3.289 ±0.524	−0.265 ±0.191
$XLMR_l$ T	1.175 ±0.108	2.229 ±0.287	−0.988 ±0.218	1.143 ±0.098	2.413 ±0.380	0.387 ±0.087	0.836 ±0.069	1.340 ±0.185	0.484 ±0.068
Our model	0.526 ±0.049	**0.579** **±0.073**	0.472 ±0.061	0.516 ±0.047	**0.939** **±0.256**	0.758 ±0.133	0.537 ±0.119	**0.781** **±0.196**	0.692 ±0.077

group rarely has a negative impact on the models. In particular, the removal of *Headers* and *Text* set exhibit more noticeable changes. It is also worth noting that the LR model experiences a significant decrease in performance when the *Protocols* set is removed. In this scenario, the impact of the *Header* and *Text* set is particularly noticeable, along with a substantial decrease in performance when using only the *Attachments* set. This observation can be attributed to the fact that both datasets contain some emails with attachments, leading to a diminished representation of the captured information by this feature subset in the models.

This analysis of group importance becomes more relevant when examining feature extraction times (Table 5). Because it is necessary to extract text for *Text* features, this process takes a long time as it first analyses whether there is hidden text to extract or not the visual content using an OCR. We will address this limitation in the future since it becomes significant for a model that prioritises detection speed. However, once the features are analysed, it can be observed that despite removing text, the performance of the model does not decrease below 0.9 of F1-Score, which allows us to use four sets of features to analyse the emails with greater speed.

Based on that, it is worth highlighting that text features, especially those related to readability, play an essential role, as well as features associated with URLs. Our novel approach to measuring address quality is one of the most distinctive aspects. This encourages us to continue our investigation and strengthens our hypothesis regarding the significance of sender addresses in misleading individuals.

It is surprising to observe, in the same Fig. 7, that the most relevant cybersecurity topic is *Pharmacy*, sixth bar from the end, despite encompassing other critical topics such as *Extortion Hacking* or *Identity Fraud*. This finding could be attributed to the fact that the borderline risk levels (6 and 7) are constituted by a significant portion of this particular class.

Features containing references to brands or currencies also hold importance in distinguishing patterns within the data. The DKIM protocol gains value to discriminating classes since it is the most used protocol among spammers.

While features related to the format of attachments do not provide sufficient information to contribute significantly to the classifier's decision-making process. Despite exploring different approaches to capture this information, we reached similar conclusions as [7]. This may be attributed to the limited number of emails with attachments in our datasets. This observation is consistent with the notion that spammers increasingly rely on URLs rather than attachments in their malicious activities.

In addition to classifying instances as either *low* or *high* risk categories, we aim at providing an estimate of the probability that a spam email belongs to the *high* risk class. This score provides additional information about the confidence of the classifier in its prediction. Fig. 8 illustrates two examples: (a) with a high score and (b) with a low score, both associated with the *high* risk class.

It means that the email text is more informative, well-constructed, and may resemble a genuine email rather than spam. This makes it more likely to deceive an end user, owing to the quality and authentic appearance of the message, thus explaining the high score.

To observe the evolution of the F1-Score when the features are incorporated one by one, starting with the most informative one, we evaluated our model using Random Forest on SERC following this strategy as can be seen in (Fig. 6). We can observe that the F1-Score goes below 0.900 when the feature set only includes the ten most important features. This implies that the model experiences a minimal reduction in F1-Score while the feature dimensionality is significantly reduced. This reduction in the number of features results in a shorter execution time and demonstrates the robustness of the model in maintaining a high F1-Score despite using a small number of features.

Regarding datasets, SERC-I offers more distributed information, allowing more robust models to be generated. On the other hand, when training on the SERC-BG dataset, we encountered lower performance due to the predominance of emails in two risk levels (6 and 7), corresponding to the *low* and *high* classes, respectively. SERC, the combination of both, serves as a balance between advantages and disadvantages.

We observed a similar trend in the regression perspective. Although models trained on SERC-BG leveraged its generic distribution by achieving a lower MSE, those trained on SERC-I obtained the highest robustness against outlines, which can improve the model performance against real-scenario data. RFR achieved the highest performance as well.

Given the complexity of the task and the wide range of models evaluated, future work could focus on developing ensemble methodologies to further enhance performance [62]. Since the top-performing models for both tasks were based on the Random Forest (RF) algorithm, it is worth exploring the use of bagging techniques, which may further boost model performance. Moreover, considering the consistently high overall performance of the models, another promising direction is the development of an ensemble model based on stacked generalisation [63]. This approach could exploit on the strengths of each individual model. Incorporating dimensionality reduction techniques, such as Principal Component Analysis (PCA), into the ensemble could also improve performance by selecting the most relevant features for each model within the ensemble [64]

7. Conclusions

Our objective in this paper was to develop an intelligent system to identify high-risk spam emails targeting individuals and organisations. We extended the scope of Gallo et al. [7] beyond a company environment and also incorporated spam emails targeting individuals. To achieve this, we annotated two datasets: a private dataset provided by the Spanish National Cybersecurity Institute (INCIBE) and a publicly available dataset collected from the Bruce Guenter repository. These datasets were named SERC-I and SERC-BG, respectively, and were merged to create a larger dataset, SERC.

We approached the problem from two machine-learning perspectives: classification and regression. We proposed a feature extraction approach based on five NLP features that depend on the specific email component: *Headers*, *Text*, *Attachments*, *URLs*, and *Protocols*. Our approach introduced novel characteristics, such as the detection of brands, currency and cryptocurrency inside the textual information of emails. In addition, to mitigate third-party requests, we integrated a Machine Learning model [65] designed to detect malicious URLs. Furthermore, we introduced new features to extract information from attachments without opening them. Our analysis also covered the evaluation of DKIM, SPF, and DMARC protocols.

We evaluated three well-known classifiers for the risk assessment model: Logistic Regression, Support Vector Machine and Random Forest. After assessing them, we can recommend Random Forest as the most suitable model for this task. We also performed a thorough feature analysis that highlighted the significance of the *Text* and *URL* features and the *Headers* set. Additionally, we observed a relatively lower importance of the *Attachments* feature set.

The outcomes of our study motivate us to continue developing and enhancing anti-spam filters that prioritise the assessment of spam content risks for both individuals and organisations. We aim to explore more distinctive features from the five feature sets. Given the best-performing models for both approaches were based on the Random Forest algorithm, we are motivated to explore enhanced versions of Random Forest. For instance, the lazy variant using nearest neighbours [66] as well as models that incorporate classification accuracy and correlation measurements between decision trees [67]. The emergence of Transformers, especially the Large Language Models (LLMs), such as GPT family, can provide different ways to collect information or develop text classification pipelines. A challenge inherent in our study is the availability of labelled data. The annotation process requires significant resources, including human effort and time of experts. Accurately assessing the potential risk associated with each spam email is a time-consuming task. Therefore, obtaining a larger amount of labelled data becomes essential for the continual improvement of our models. Given the relevance of the 56 proposed features and the insights derived from this work, annotators can manually use these features to assess the risk level of spam emails, either from their own honeypots or from publicly available platforms like Spam Archive. This can also be part of a semi-automated annotation process, where the proposed models in this work are first used to label the samples, and then an expert validates the results.

CRediT authorship contribution statement

Francisco Jáñez-Martino: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Rocío Alaiz-Rodríguez:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis. **Víctor González-Castro:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Eduardo Fidalgo:** Writing – review & editing, Resources, Project administration. **Enrique Alegre:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been funded by the Recovery, Transformation, and Resilience Plan, financed by the European Union (Next Generation) thanks to the LUCIA project (Fight against Cybercrime by applying Artificial Intelligence) granted by INCIBE to the University of León.

Data availability

Data will be made available on request.

References

[1] Sreelaja, Ant colony optimization based light weight binary search for efficient signature matching to filter ransomware, Appl. Soft Comput. 111 (2021) 107635.

[2] H. Jones, J. Towse, Examinations of email fraud susceptibility: Perspectives from academic research and industry practice, in: Psychological and Behavioral Examinations in Cyber Security, IGI Global, 2018, pp. 1–18.

[3] D. Sturman, C. Valenzuela, O. Plate, T. Tanvir, J.C. Auton, P. Bayl-Smith, M.W. Wiggins, The role of cue utilization in the detection of phishing emails, Appl. Ergon. 106 (2023) 103887.

[4] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, E. Alegre, Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach, Appl. Soft Comput. 139 (2023) 110226, http://dx.doi.org/10.1016/j.asoc.2023.110226.

[5] M. Frank, L. Jaeger, L. Manuel Ranft, Using contextual factors to predict information security overconfidence: A machine learning approach, Comput. Secur. 125 (2023) 103046.

[6] J. Buckley, D. Lottridge, J. Murphy, P. Corballis, Indicators of employee phishing email behaviours: Intuition, elaboration, attention, and email typology, Int. J. Hum.-Comput. Stud. 172 (2023) 102996.

[7] L. Gallo, A. Maiello, A. Botta, G. Ventre, 2 years in the anti-phishing group of a large company, Comput. Secur. 105 (2021) 102259.

[8] P. Bountakas, C. Xenakis, HELPHED: Hybrid ensemble learning phishing email detection, J. Netw. Comput. Appl. 210 (2023) 103545.

[9] D. Bera, O. Ogbanufe, D.J. Kim, Towards a thematic dimensional framework of online fraud: An exploration of fraudulent email attack tactics and intentions, Decis. Support Syst. (2023) 113977.

[10] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, Trustworthiness of spam email addresses using machine learning, in: Proceedings of the 21st ACM Symposium on Document Engineering, DocEng '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 4, http://dx.doi.org/10.1145/3469096.3475060.

[11] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, E. Alegre, A review of spam email detection: analysis of spammer strategies and the dataset shift problem, Artif. Intell. Rev. 56 (2022) 1145–1173, http://dx.doi.org/10.1007/s10462-022-10195-4.

[12] N. Saidani, K. Adi, M.S. Allili, A semantic-based classification approach for an enhanced spam detection, Comput. Secur. 94 (2020) 101716, http://dx.doi.org/10.1016/j.cose.2020.101716.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017, CoRR abs/1706.03762.

[14] E. Ferrara, The history of digital spam, Commun. ACM 62 (8) (2019) 82–91.

[15] M. Nguyen, T. Nguyen, T.H. Nguyen, A deep learning model with hierarchical LSTMs and supervised attention for anti-phishing, 2018, arXiv:1805.01554.

[16] R. Kaur, D. Gabrijelčič, T. Klobučar, Artificial intelligence for cybersecurity: Literature review and future research directions, Inf. Fusion 97 (2023) 101804.

[17] S. Magdy, Y. Abouelseoud, M. Mikhail, Efficient spam and phishing emails filtering based on deep learning, Comput. Netw. 206 (2022) 108826.

[18] M. Volkamer, K. Renaud, B. Reinheimer, A. Kunz, User experiences of TORPEDO: Tooltip-powered phishing email detection, Comput. Secur. 71 (2017) 100–113.

[19] S. Sankhwar, D. Pandey, P.R. Khan, Email phishing: An enhanced classification model to detect malicious URLs, ICST Trans. Scalable Inf. Syst. 6 (2018) 158529.

[20] S. Smadi, N. Aslam, L. Zhang, Detection of online phishing email using dynamic evolving neural network based on reinforcement learning, Decis. Support Syst. 107 (2018) 88–102.

[21] L. Halgaš, I. Agrafiotis, J.R.C. Nurse, Catching the phish: Detecting phishing attacks using recurrent neural networks (RNNs), in: I. You (Ed.), Information Security Applications, Springer International Publishing, Cham, 2020, pp. 219–233.

[22] D. Lee, R.M. Verma, Adversarial machine learning in text: A case study of phishing email detection with RCNN model, in: Adversary-Aware Learning Techniques and Trends in Cybersecurity, Springer International Publishing, Cham, 2021, pp. 61–83.

[23] A. Alhogail, A. Alsabih, Applying machine learning and natural language processing to detect phishing email, Comput. Secur. 110 (2021) 102414.

[24] D. Radev, CLAIR collection of fraud email (repository)-ACL wiki, 2008.

[25] S. Salloum, T. Gaber, S. Vadera, K. Shaalan, Phishing email detection using natural language processing techniques: A literature survey, Procedia Comput. Sci. 189 (2021) 19–28, AI in Computational Linguistics.

[26] K. Singh, P. Aggarwal, P. Rajivan, C. Gonzalez, Cognitive elements of learning and discriminability in anti-phishing training, Comput. Secur. 127 (2023) 103105.

[27] A. El Aassal, S. Baki, A. Das, R. Verma, An in-depth benchmarking and evaluation of phishing detection research for security needs, IEEE Access 8 (2020) 1–1.

[28] T. Gangavarapu, C. Jaidhar, B. Chanduka, Applicability of machine learning in spam and phishing email filtering: review and approaches, Artif. Intell. Rev. 53 (2020) 64.

[29] C. Beaman, A. Barkworth, T.D. Akande, S. Hakak, M.K. Khan, Ransomware: Recent advances, analysis, challenges and future research directions, Comput. Secur. 111 (2021) 102490.

[30] S. Chakkaravarthy, D. Priya, T. Reddi, M.S.T. Reddy, M.K. Khan, A comprehensive examination of email spoofing: Issues and prospects for email security, Comput. Secur. (2023) 103600.

[31] R.F. Flesch, A new readability yardstick, J. Appl. Psychol. 32 (3) (1948) 221–233.

[32] J.P. Kincaid, R.P. Fishburne, R.L. Rogers, B.S. Chissom, Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel, 1975.

[33] G.H. Mclaughlin, SMOG grading - A new readability formula, J. Read. (1969).

[34] R. Gunning, The Technique of Clear Writing, McGraw-Hill, Toronto, 1652, p. 329.

[35] J. Fernández Huerta, Medidas sencillas de lecturabilidad, Consigna 214 (1959) 29–32.

[36] F. Szigriszt Pazos, Investigación sobre lectura en Venezuela, 1972, Ponencia presentada ante las Jornadas de Educación Primaria. Caracas, Ministerio de Educación, mimeografiado.

[37] F. Szigriszt Pazos, Sistemas predictivos de legilibilidad del mensaje escrito: fórmula de perspicuidad, Univ. Complut. Madrid Serv. Publ. (1992).

[38] Miguel Misael Muñoz Baquedano, José Miguel Muñoz Urra, Legibilidad μ, 2023, https://legibilidadmu.cl/, Accessed: XXX 2023.

[39] W. Ge, J. Wang, T. Lin, B. Tang, X. Li, Explainable cyber threat behavior identification based on self-adversarial topic generation, Comput. Secur. 132 (2023) 103369.

[40] M. Sánchez-Paniagua, E.F. Fernández, E. Alegre, W. Al-Nabki, V. González-Castro, Phishing URL detection: A real-case scenario through login URLs, IEEE Access 10 (2022) 42949–42960, http://dx.doi.org/10.1109/ACCESS.2022.3168681.

[41] J.R. Méndez, T.R. Cotos-Yañez, D. Ruano-Ordás, A new semantic-based feature selection method for spam filtering, Appl. Soft Comput. 76 (2019) 89–104.

[42] D. Ruano-Ordás, F. Fdez-Riverola, J. R. Méndez, Using evolutionary computation for discovering spam patterns from e-mail samples, Inf. Process. Manage. 54 (2) (2018b) 303–317.

[43] T.K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, 1995, pp. 278–282, http://dx.doi.org/10.1109/ICDAR.1995.598994, vol.1.

[44] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[45] D.R. Cox, The regression analysis of binary sequences, J. R. Stat. Soc. Ser. B Stat. Methodol. 20 (2) (1958) 215–232.

[46] E.G. Dada, J.S. Bassi, H. Chiroma, S.M. Abdulhamid, A.O. Adetunmbi, O.E. Ajibuwa, Machine learning for email spam filtering: review, approaches and open research problems, Heliyon 5 (6) (2019) e01802.

[47] L.Á. Redondo-Gutierrez, F. Jáñez-Martino, E. Fidalgo, E. Alegre, V. González-Castro, R. Alaiz-Rodríguez, Detecting malware using text documents extracted from spam email through machine learning, in: Proceedings of the 22nd ACM Symposium on Document Engineering, DocEng '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 4, http://dx.doi.org/10.1145/3558100.3563854.

[48] A. Mccallum, K. Nigam, A comparison of event models for naive Bayes text classification, Work Learn. Text Categ. 752 (2001).

[49] F. Galton, Regression towards mediocrity in hereditary stature, J. Anthropol. Inst. Great Brit. Ireland 15 (1886) 246–263, URL: http://www.jstor.org/stable/2841583.

[50] B. Menze, B. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, F. Hamprecht, A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data, BMC Bioinform. 10 (2009) 213.

[51] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, Comp. Sci. Rev. 40 (2021) 100379, http://dx.doi.org/10.1016/j.cosrev.2021.100379.

[52] M. Reusens, A. Stevens, J. Tonglet, J. De Smedt, W. Verbeke, S. vanden Broucke, B. Baesens, Evaluating text classification: A benchmark study, Expert Syst. Appl. 254 (2024) 124302, http://dx.doi.org/10.1016/j.eswa.2024.124302.

[53] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2019, CoRR abs/1911.02116. arXiv:1911.02116. URL: http://arxiv.org/abs/1911.02116.

[54] P. Přibáň, J. Šmíd, J. Steinberger, A. Mištera, A comparative study of cross-lingual sentiment analysis, Expert Syst. Appl. 247 (2024) 123247, http://dx.doi.org/10.1016/j.eswa.2024.123247.

[55] J.A. García-Díaz, G. Beydoun, R. Valencia-García, Evaluating transformers and linguistic features integration for author profiling tasks in Spanish, Data Knowl. Eng. 151 (2024) 102307, http://dx.doi.org/10.1016/j.datak.2024.102307.

[56] J. de la Rosa, E.G. Ponferrada, P. Villegas, P.G. de Prado Salas, M. Romero, M. Grandury, BERTIN: Efficient pre-training of a spanish language model using perplexity sampling, 2022, arXiv:2207.06814.

[57] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P.S. Yu, L. He, A survey on text classification: From traditional to deep learning, ACM Trans. Intell. Syst. Technol. 13 (2) (2022) http://dx.doi.org/10.1145/3495162.

[58] M.W. Al Nabki, E. Fidalgo, E. Alegre, I. de Paz Centeno, Classifying illegal activities on Tor network based on web textual contents, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 35–43.

[59] K. Kawintiranon, L. Singh, C. Budak, Traditional and context-specific spam detection in low resource settings, Mach. Learn. 111 (7) (2022) 2515–2536, http://dx.doi.org/10.1007/s10994-022-06176-x.

[60] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv:1907.11692.

[61] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020, arXiv:1911.02116.

[62] Y. Zhang, J. Liu, W. Shen, A review of ensemble learning algorithms used in remote sensing applications, Appl. Sci. 12 (17) (2022) http://dx.doi.org/10.3390/app12178654.

[63] R.S. Wilkho, S. Chang, N.G. Gharaibeh, FF-BERT: A BERT-based ensemble for automated classification of web-based text on flash flood events, Adv. Eng. Inform. 59 (2024) 102293, http://dx.doi.org/10.1016/j.aei.2023.102293.

[64] E. Abdali, M.J. Valadan Zoej, A. Taheri Dehkordi, E. Ghaderpour, A parallel-cascaded ensemble of machine learning models for crop type classification in google earth engine using multi-temporal sentinel-1/2 and landsat-8/9 remote sensing data, Remote Sens. 16 (1) (2024) http://dx.doi.org/10.3390/rs16010127.

[65] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro, E. Alegre, Impact of current phishing strategies in machine learning models for phishing detection, in: A. Herrero, C. Cambra, D. Urda, J. Sedano, H. Quintián, E. Corchado (Eds.), 13th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2020, Springer International Publishing, Cham, 2021, pp. 87–96.

[66] T. Salles, M. Gonçalves, V. Rodrigues, L. Rocha, Improving random forests by neighborhood projection for effective text classification, Inf. Syst. 77 (2018) 1–21, http://dx.doi.org/10.1016/j.is.2018.05.006.

[67] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, X. Liang, An improved random forest based on the classification accuracy and correlation measurement of decision trees, Expert Syst. Appl. 237 (2024) 121549, http://dx.doi.org/10.1016/j.eswa.2023.121549.