IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 19, 2020, accepted December 4, 2020, date of publication December 9, 2020,
date of current version December 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043396

The Answer Is in the Text: Multi-Stage Methods for Phishing Detection Based on Feature Engineering

EDER SOUZA GUALBERTO[1,2], RAFAEL TIMOTEO DE SOUSA, JR.[1], (Senior Member, IEEE), THIAGO PEREIRA DE BRITO VIEIRA[2], JOÃO PAULO CARVALHO LUSTOSA DA COSTA[1,3], (Senior Member, IEEE), AND CLÁUDIO GOTTSCHALG DUQUE[4]

[1]Department of Electrical Engineering, University of Brasília, Brasília 70910-900, Brazil
[2]Superintendence of Information Management, National Agency of Telecommunications, Brasília, Brazil
[3]Department 2 in Lippstadt, Hamm-Lippstadt University of Applied Sciences, 59063 Hamm, Germany
[4]Faculty of Information Science, University of Brasília, Brasília 70910-900, Brazil

Corresponding author: Eder Souza Gualberto (eder.gualberto@redes.unb.br)

This work was supported in part by the CAPES–Brazilian Higher Education Personnel Improvement Coordination under Grant PROAP/UnB/PPGEE, Grant 23038.007604/2014-69 FORTE, and Grant 88887.144009/2017-00 PROBRAL; in part by the CNPq–Brazilian National Research Council under Grant 312180/2019-5 PQ-2, Grant 303343/2017-6 PQ-2, Grant BRICS 2017-591 LargEWiN, and Grant 465741/2014-2 INCT in cybersecurity; in part by the FAP-DF–Brazilian Federal District Research Support Foundation under Grant 0193.001366/2016 UIoT and Grant 0193.001365/2016 SSDDC, in part by the Brazilian Ministry of the Economy under Grant 005/2016 DIPLA and Grant 083/2016 ENAP, in part by the Institutional Security Office of the Presidency of Brazil under Grant ABIN 002/2017, in part by the Administrative Council for Economic Defense under Grant CADE 08700.000047/2019-14, and in part by the General Attorney of the Union under Grant AGU 697.935/2019.

ABSTRACT A phishing attack is a threat based on fraudulent communication, usually by e-mail, where the cybercriminals, impersonating a trusted person or organization, try to lure and coax a target. Phishing detection approaches that obtain highly representational features from the text of these e-mails are a suitable strategy to counter these threats since these features can be used to train machine learning algorithms, thus generating models able to classify mail samples as phishing or legitimate messages. This paper proposes a multi-stage approach to detect phishing e-mail attacks using natural language processing and machine learning. The proposed multi-stage approach consists of feature engineering within natural language processing, lemmatization, feature selection, feature extraction, improved learning techniques for resampling and cross-validation, and the configuration of hyperparameters. We present two methods of the proposed approach, the first one exploiting the Chi-Square statistics and the Mutual Information to improve the dimensionality reduction, while the second method associates Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA). Both methods handle the problems of the ‘‘curse of dimensionality", the sparsity, and the amount of information that must be obtained from the context in the Vector Space Model (VSM) representation. These methods yield reduced feature sets that, combined with the XGBoost and Random Forest machine learning algorithms, lead to an F1-measure of 100% success rate, for validation tests with the SpamAssassin Public Corpus and the Nazario Phishing Corpus datasets. Even considering just the text in e-mail bodies, the proposed multi-stage phishing detection approach outperforms state-of-the-art schemes for an accredited data set, requiring a much smaller number of features and presenting lower computational cost.

INDEX TERMS Cybersecurity, phishing detection, natural language processing, feature engineering, feature selection, feature extraction.

I. INTRODUCTION

The Internet plays a crucial role on the industries and societies worldwide by providing a wide variety of services. According to [1], [2], the number of Internet users corresponds to 62% of the world population, and this percentage will increase to 66% in 2023. Related cybercrimes are growing proportionally and evolving, thus becoming more crafty and refined, as is the case of phishing. For instance, according to [3], in the first quarter of 2020, 75% of all phishing sites use secure sockets

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

layer (SSL) and since mid-March of this year phishing attacks have been launched using the coronavirus disease of 2019 (COVID-19) as their theme. To convince their targets, these phishing attacks contain textual compositions including several matters such as the Internet and security technologies, and information related to the COVID-19 pandemics.

Traditionally, a phishing campaign starts with an e-mail [4]–[7]. Therefore, the detection of this type of e-mail is critical to counter these attacks. Currently, as pointed in [8]–[10] and [7], phishing detecting mechanisms based on Natural Language Processing (NLP) and Machine Learning (ML) techniques, such as [11], [12] and [13], are an effective way to defend against this type of threat, since such approaches exploit the morphology and semantics of the text.

This phishing e-mail detection paradigm is based on obtaining, from the text of the e-mails (body or/and subject), the features that feed machine learning classification algorithms. These algorithms, in turn, determine whether each e-mail message is phishing or a legitimate one (ham e-mail). In this way, each e-mail is represented as an item in a Vector Space Model (VSM) [14], where each term in each text in the whole corpus[1] is a dimension in which each e-mail is denoted by its term ranking (through a Document-Term Matrix - DTM).

As discussed in [16], since the VSM has as many dimensions as the number of terms in a used corpus, and the fact that not all terms are present in each of the e-mails, the feature engineering step of the phishing e-mail detection process has to deal and address questions related to the ‘‘Curse of Dimensionality" and the sparsity [17]–[19]. Additional crucial aspects refer to the context portion embedded in the VSM, i.e., considering the issues of how to improve its utilization and how to explicit its latent features [20].

In this paper, we propose two methods based on combined techniques to obtain more distinguishing features, and related attributes, for phishing detection. These methods comprehend feature selection and feature extraction of the VSM generated from the e-mails texts.

Our approach is centered in a structured process, from the wrought of the features to the classification algorithms learning. First, in a common stage to both proposed methods, the text of e-mail bodies goes through a pre-processing step. The output is submitted to tokenization, part-of-speech (POS) tagging, and lemmatization, using the Stanza database and toolkit. Next, in a second stage, a Document-Term Matrix (DTM) is extracted from this processed text, with this matrix ranking represented by the Term Frequency-Inverse Document Frequency (TF-IDF). This matrix is employed in two different exclusive ways: to select a subset of the DTM dimensions, using the Chi-Squared and Mutual Information statistical measures (Method 1), and to extract new features from the initial dimensions with Principal Components Analysis and Latent Semantic Analysis (Method 2). In the final stage, the different sets of features obtained through both methods feed the same machine learning classification algorithms, using improved learning techniques to conclude the operation of each method.

This work aims to propose strategies to correctly detect phishing e-mails for preventing them from reaching the target user. In this sense, the main contribution of this research is a feature engineering process and an overall approach for phishing detection, based on NLP and ML. This approach integrates strategies that improve the threat identification predictions of the adopted algorithms and address the problems related to the VSM representation derived from the DTM, i.e., the ‘‘Curse of Dimensionality", the sparsity, and the information that must be obtained from the context.

The proposed approach brings answers to the listed problems while demonstrating an optimal representation capacity since it uses a smaller number of features compared to previous approaches but still presents better performance figures. The chosen features provide an enhanced distinction between phishing messages and legitimate e-mails for the selected mail datasets.

Compared to previous works our approach representation potential uses an amount of new extracted features that is approximately 0.004% of the initial volume of features, but still achieving measures of 100% success rate, using only twenty-five features, in the validation scenario. To the best of our knowledge, it is the best result in phishing detection research for an accredited data set based only on the body text of e-mails.

In this paper, the following notations are used: lowercase boldface letters denote vectors ($\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$), whereas uppercase boldface letters describe matrices ($\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$), and lowercase letters with index denote their elements (the element of the matrix $\boldsymbol{A}$, located at line i column j, is indicated by $a_{i,j}$).

The remainder of this paper is structured into five sections. In Section II, the baseline study is described, presenting the related works based on natural language processing and machine learning techniques for phishing detection. In Section III, the data modeling, the architecture of the proposed approach and the adopted methodology are explained. In Section IV, the proposed methods are evaluated using real data and compared with the baseline study, whereas, in Section V, conclusions are drawn, and future works are outlined.

II. RELATED WORKS

Machine learning and data-driven approaches have been increasingly employed to solve cybersecurity-related problems [9], [21], [22] [23]. The phishing detection research landscape shows that, through natural language processing techniques, robust results have been obtained. Most of this research is centered on how to extract, from the text and the metadata of the e-mail, highly distinctive features that allow it to identify differences and similarities among these messages, in order to separate them in phishing or legitimate e-mails.

One of the first approaches to phishing e-mail detection based on machine learning was proposed by Fette *et al.* [24].

[1] Corpus is a computer-readable collection of text or speech [15].

It generated features based on e-mail texts and properties, such as if these e-mails contain javascript code, the number of links in the e-mail, or the number of dots in the present Uniform Resource Locators (URLs). It detected over 96% of the phishing e-mails when submitting the best ten features they found to the Random Forest classification algorithm. Also proposing to select a set of content-based and behavior-based features, Hamid *et al.* [25] achieved a 94% accuracy rate through the use of Bayes Net Algorithm, which was fed with eight features.

Similarly in [26], from forty-eight features selected from the specialized literature (related to the e-mail body and header, Javascript and URLs), Daeef *et al.* proposed a phishing e-mail classification based on two stages, extracting features and submiting them to three ML classification algorithms. They attained an accuracy rate of 99.40%. Also, using hand-crafted features extracted from the e-mail body and header (twenty-one features), Islam and Abawajy [27] proposed a 3-tier model classification, based on well-known ML algorithms. They obtained an accuracy rate of 97%, when distinguishing phishing and legitimate e-mails.

In [28], Toolan and Carthy proposed an analysis, based on entropy and information gain measures, with 40 features extracted from the body, subject, and sender e-mail fields, and from the presence/absence of any script and URL. The authors utilized the C5.0 decision tree algorithm for classification, reaching an 84.6% success rate when classifying phishing against ham e-mails. Also, based on information gain to select features for phishing detection, Yasin and Abuhasan [29] used text stemming and WordNet database to pre-process and enrich their e-mails representation. Through this approach, they obtained an accuracy mark of 99.1%, whereas the proposal PhishNet-NLP, presented by Verma *et al.* [30], achieved a phishing e-mail detection rate of 97%. The approach in [30] is based on NLP techniques to check if e-mails are informative or actionable and other features extracted from the body and header of the e-mails. The object in [31] was to develop an improved phishing e-mail classifier with better prediction accuracy and fewer numbers of features. In this sense, Akinyelu and Adewumi used a set of 15 phishing e-mail features, identified from the literature, and fed the random forest machine learning algorithm. An accuracy of 99.7% was achieved.

An analysis of techniques to promote feature reduction to phishing detection is detailed in [32]. Four techniques (Chi-Square, Information Gain, Latent Semantic Analysis - LSA, and Principal Component Analysis - PCA) were compared. In the approach in [32], the use of these techniques was preceded for stemming, and the features were based on header contents and eventual URLs, besides the body of the e-mails. This proposal reached an accuracy rate of almost 98%. L'Huillier *et al.* [33] proposed an approach whose features are extracted from three ways: structural features extracted directly from the text, features based on keyword, and features obtained through the application of LSA and Latent Dirichlet Allocation (LDA) techniques over the TF-IDF Matrix, which is generated from the corpus texts. The approach in [33] attained an F1-score mark of 99.58%, using 1017 features to feed the SVM classification algorithm. Likewise, using NLP methods, Ramanathan and Wechsler presented phishGILLNET [34]. The phishGILLNET is a 3-layer approach based on topics model. Through the use of techniques such as Probabilistic Latent Semantic Analysis (PLSA) and Co-training, the phishGILLNET obtained an F1-measure of 100% for 200 topics, and 98.3% for 25 and 10 topics (these topics were employed to express the features input to AdaBoost classification algorithm). In [35], Ramanathan and Wechsler proposed another phishing detection approach. In this method, they also try to discover the entity/organization that the attackers impersonate during phishing attacks. This proposal employed Conditional Random Field (CRF), Latent Dirichlet Allocation (LDA), and the AdaBoost in its best variation, identifying the impersonated entity from messages classified as phishing with a discovery rate of 88.1%.

In [36], using the text as its primary features source, and also incorporating the domain knowledge and lexical features, an approach that reaches an F1-measure of 98% is presented. This approach is based on DTM and TF-IDF, and its best mark was obtained through the use of Logistic Regression classification algorithm, whereas [37] and [38] proposed methods to phishing detection based on Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), also considering DTM and TF-IDF. The obtained features were submitted as input for several classifications algorithms, achieving its respectively best mark of 94.6% (using k-Nearest Neighbor - KNN classification algorithm) and 95.3% (using SVM with 30 features).

Unnithan *et al.* [12] compared the employment of the TF-IDF matrix and the Doc2Vec representation to phishing e-mail detection. They used seven different classification algorithms to assess these two approaches, achieving their best mark (an 88.95% accuracy rate) through the use of the SVM classifier fed by the Doc2Vec representation. Also, a word embedding approach, the proposal presented in [39] was based on the FastText technique. Through the syntactic and semantic similarity of e-mails extracted by the techniques employed, their approach attained an accuracy rate of 99%. The same authors proposed another approach [40], based on Word2Vec and Neural Bag-of-Ngrams, for phishing e-mails detection. The obtained representations fed some classifiers such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP), reaching in its best variation a 99.1% F1 Score (using Word2Vec and LTSM).

The strategy employed in [41] was based on content and behavior-based features and also in word embedding (Word2Vec) techniques. The obtained features were imputed to a Neural Network classification algorithm, achieving an accuracy of 92.2%. In [42], using 240 features, 200 from Doc2Vec representation (to capture the syntax and semantics of the e-mails) and 40 content- and behavior-based features,

Gangavarapu and Jaidhar introduced a hybrid metaheuristic to obtain a discriminative and informative feature subset crucial to Unsolicited Bulk E-mails (UBE). When classifying e-mails in phishing or legitimate e-mail, this research achieved an accuracy of 99.4% employing the Multi-Layer Perceptron (MLP) for the classification task. Also proposing an approach to select the most discriminative feature set among 40 extracted features, [43] presents a structured procedure to extract and select content and behavior-based features to detecting UBE. Employing 27 of these 40 features (selected through a low variance filter) and the Randon Forest algorithm, it obtained an F1-measure of 99,2 separating phishing and legitimate e-mails.

Chin *et al.* presented a deep packet inspection (DPI) in [44]. Their approach was based on two components: phishing signature classification and real-time DPI, and the best mark achieved was 98.39% of accuracy (using an Artificial Neural Network - ANN) when detecting and mitigating phishing attacks. Based on Recurrent Neural Network (RNN), the approach proposed in [45] takes sequences of integer values as input for this classification algorithm. These values are obtained abstracting the computer-native copy of an e-mail as a sequence of bytes into the high-level representation (unigrams), represented as unique integers. This approach attained an F1-measure of 98.63% and an accuracy rate of 98.91%. Through the use of Recurrent Convolutional Neural Networks (RCNN), Fang *et al.* [46] proposed THEMIS, that employing Word2Vec models e-mails at four levels simultaneously (header, body, character, and word). Its best mark was an F1-Score of 99.31% and an accuracy of 99.84%.

Gualberto *et al.* [47] had the goal of obtaining highly distinctive features for phishing detection from the text of the e-mails. They proposed an approach based on machine learning performed through a feature engineering process based on natural language processing, lemmatization, topics modeling (using LDA), improved learning techniques for resampling and cross-validation, and hyperparameters configuration. Our approch handled the problems of ‘‘the curse of dimensionality", the sparsity, and the text context portion included in the obtained representation for e-mails. This proposal reached marks with an F1-measure of 99.95% success rate using the XGBoost algorithm, with just ten features. It outperforms state-of-the-art of phishing detection researches (with a reduced set of features) for an accredited data set, in applications based only on the body of the e-mails, without using other e-mail features such as its header, IP information or number of links in the text.

Unlike some previous work, we did not intend to use detection techniques based on blacklist and filter rules, but on focusing on discovering the purpose of the e-mail (if malicious or legitimate) from its text content. In this regard, we aspired to design a process in which each of the proposed phases would contribute to the textual representation and the classification of the analyzed e-mails. For instance, differently from some previous work, the proposed data cleansing and the use of lemmatization instead of stemming, as well as the criteria to choose the number of features to work in each perspective, proved to support the semantic and similarity enrichment and boost the performance of the proposed methods.

Considering our previous proposal [47], the approach presented in this paper are results of an improvement over it, since it employs the new Stanza NLP toolkit as the database in the feature engineering process to promote tokenization, POS tagging, and lemmatization (instead of WordNet, as done in [47]); and proposes methods to select feature based on statistical measures, and methods to extract new features, from that initially presented in DTM, based on PCA and LSA (instead of LDA, as done in [47]).

The datasets of most of the works listed in this section are obtained from the PhishingCorpus [48] and from the Spamassassin PublicCorpus [49], which are the same datasets adopted to evaluate the proposed approach. As exceptions, some of them, such as [34], [35], [39], [40], [46] and [44], used a clustered dataset in which PhishingCorpus and Spamassassin were part of their sources.

III. PROPOSED MULTI-STAGE APPROACH FOR PHISHING DETECTION

In this section, the proposed multi-stage proposal is exposed in detail, approaching its architecture and methodology, as well as addressing the strategies and the techniques employed in its implementation. In Subsection III-A, the dataset is presented. In Subsection III-B, the data modeling is addressed. In Subsection III-C, an overview of the entire architecture is portrayed. In Subsection III-D, it is displayed how the texts were parsed from the e-mail files to labeled data structures indicating their respective e-mail classes. In Subsection III-E, the pre-processing stage is explained, mainly the lemmatization step. In Subsection III-F, the scheme for terms vectorization, based on Bag of Words (BoW), DTM and TF-IDF, is presented. In Subsections III-H and III-I, our two proposed methods are detailed. Method 1 is established on feature selection through the use of statistical measures, and Method 2 is established on feature extraction through the use of PCA and LSA. Lastly, in Subsection III-K, the strategy to train and to test the employed ML classification algorithms is available.

A. DATASET

Two sources of raw data were considered for the propose of this research: the SpamAssassin Public Corpus [49] and the Nazario Phishing Corpus [48]. Respectively, they are assumed as the Ham Dataset and as the Phishing Dataset. These two compilations of e-mails are both public datasets and their employment to evaluate phishing detection approaches is a widespread practice [45], [50]. The approaches presented in [42], [45], [47], [51], [29], [31], [33], [52], [24], [26], [27] and [25] are validated using the

SpamAssassin Public Corpus and the Nazario Phishing Corpus as datasets sources.

This dataset has 6,429 e-mails. From these, 4,150 labeled as ham e-mails from SpamAssassin Public Corpus (specifically from easy and hard ham, that respond for legitimate e-mails in this collection of e-mail), and 2,279 labeled as phishing e-mails from Nazario Phishing Corpus (specifically from phishing3.mbox).

B. DATA MODELING

In this proposed approach, the data modeling starts from the two collections of e-mail described in Subsection III-A (the raw data), selecting the e-mail files of the subsets of interest. From these files, the text of the e-mail bodies is extracted. Then, starting our feature engineering process, these texts pass through a pre-processing step and then are folded in two sets, training and test sets. Afterward, from a terms vectorization step, a DTM is generated, where each remaining term stands for a feature of the e-mails. A dimensionality reduction over the DTM representation is performed, where we select the best features or extract new ones. The training set feeds the machine learning algorithms, and the test set feeds their resulting models, classifying those e-mails as phishing or ham e-mails. Figure 1 presents this workflow.

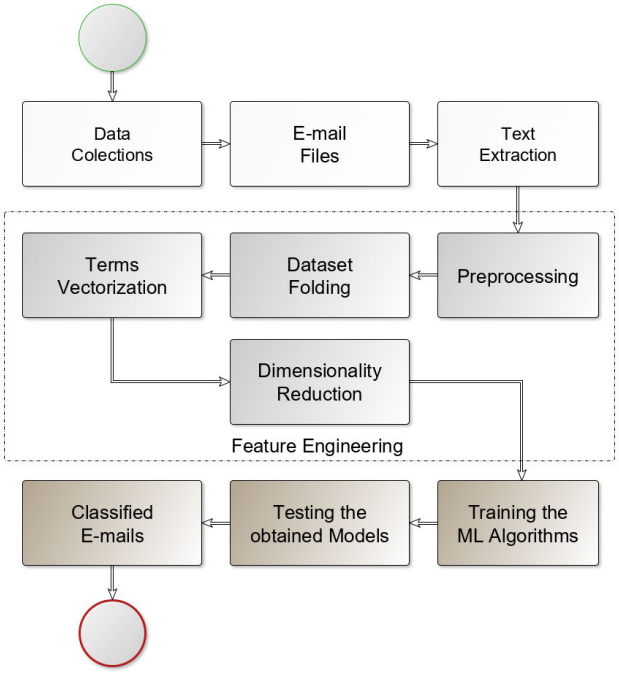


FIGURE 1. The data modeling workflow of the proposed approach, from the data collections to classified e-mails.

The modeling data assumes varied vector and matrix shapes along the proposed approach. These shapes, the methods, and the architecture of the proposed approach are presented in the next section.

C. PROPOSAL OVERVIEW

Through the use of natural language processing and machine learning techniques, the proposed approach aims to achieve refined predictions in phishing e-mail classification using the smallest possible number of features. The proposal is based on a multi-stage methodology, as expressed in Fig. 2, where the central purpose orbits around deriving more informative features, and feeds the chosen ML algorithms, training them using established strategies of folding.

The process begins from parsing the text from the e-mail bodies to a vector structure. For each e-mail text, the corresponding label is assigned according to the e-mails collection of the dataset from which it originates.

Starting the feature engineering phase, there is a pre-processing step (first stage), where operations such as lowercase the text, removing specific characters and words categories, and grouping different inflected forms of words are made. Next, the terms are vectorized, that is, a DTM is generated, in which the documents are represented in function of all remaining terms present in all e-mails of the corpus. This representation is yet balanced by the TF-IDF statistical measure (second stage). Then, as the last step of the feature engineering phase, there is a feature reduction step. These stages are what differentiates the two proposed methods. Method 1 performs its feature reduction selecting a subset of the features in two approaches, using the Chi-Square and the Mutual information statistical measures. Method 2 plays its dimensionality reduction extracting a new reduced set of features also in two approaches, through the use of Principal Component Analysis and of Latent Semantic Analysis. These techniques were chosen to decrease the number of features for certain, but also in the perspective of reducing the DTM sparsity, and embed the highest possible amount of contextual information from the text in this e-mail corpus representation. They are exclusive stages.

In the last stage, the resulting features sets from these methods are employed separately, feeding various ML algorithms in order to detect phishing e-mails, predicting whether each e-mail belongs to the phishing e-mail class or the legitimate e-mails class. These algorithms are trained and evaluated, considering consolidated strategies for dataset division and folding, and for cross-validation.

In the following subsections, all of these activities are highlighted.

D. PARSING

From all the e-mails files, the text of the e-mail bodies is extracted and arrayed in a string vector structure. That is, from 6,429 e-mail files, it is generated a string vector $\boldsymbol{e}$ with the same number of rows, where d is 6,429:

$$\boldsymbol{e}^{\mathbf{T}} = \begin{bmatrix} e_1 & e_2 & e_3 & \dots & e_d \end{bmatrix}. \tag{1}$$

The corresponding labels of these e-mails are also vectorized. They are set according to the collection from which

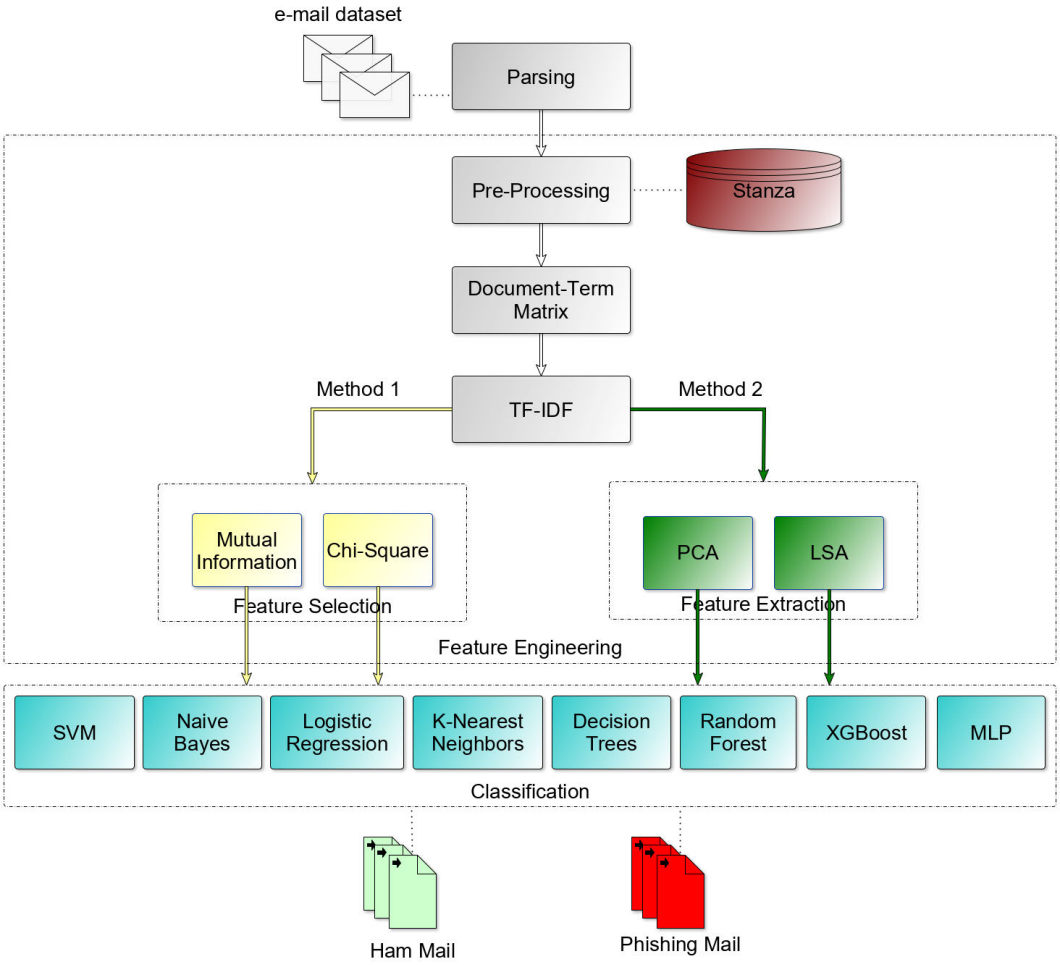


FIGURE 2. The main architecture of the proposed Multi-Stage approach for Phishing Detection.

e-mail originates (from phishing e-mail set or legitimate e-mail set). This vector is delineated as $\boldsymbol{l}$, in this way:

$$\boldsymbol{l}^{\mathbf{T}} = \begin{bmatrix} l_1 & l_2 & l_3 & \dots & l_d \end{bmatrix}. \tag{2}$$

where d is 6,429.

E. PRE-PROCESSING

The pre-processing stage uses $\boldsymbol{e}$ as input. All the letters of the texts in $\boldsymbol{e}$ are converted into lowercase. Then, punctuation marks,[2] special characters and any accents are excluded.

A tokenization process is performed over the remaining texts, in a term level, i.e., using white spaces[3] as delimiters, each text is divided into terms (tokens). The tokenization step is critical, not only to create a vocabulary for the corpus under analysis [15], [53], but also to perform other required NLP actions to the proposed approach, such as removing stopwords[4] (which is performed next).

The last step of the pre-processing stage is the lemmatization process. Its objective is to transform a word into its common base form. To reduce the inflectional forms and the derivationally related forms of a word, lemmatization normally involves the use of a vocabulary and a morphological parsing. A word is analyzed in a morpheme[5] level, in order to separate its root morpheme (stem[6]) from its accessory morphemes (affixes[7]), returning it in a lemma shape,[8] i.e., obtaining this stem in a dictionary form.

[2]Punctuation marks encompass sentence endings; comma, semicolon, and colon; dash and hyphen; brackets, braces and parentheses; and apostrophe, quotation marks and ellipsis

[3]White spaces refer to space, tab, and newline

[4]Stopwords refer to a class of words that usually has little lexical content or does not contribute much to the meaning of a sentence. Although there is no universal list representing all the stopwords, most cases take prepositions and articles as such.

[5]Morphemes refers to the smaller meaning-bearing units that build a word [15].

[6]Stem refers to the morpheme that concentrates the main meaning of the word [15].

[7]Affixes refers to the morphemes that offers additional meanings of various types to a word [15].

[8]Lemma is a word or expression, a particular form, that is chosen to represent a lexeme (the basic meaning of a stem) [53].

Most of the works presented in Section II does not perform lemmatization or performs stemming[9] instead.

Lemmatization is performed through the use of Stanza (as well as the tokenization step). Stanza is a NLP toolkit that supports 66 human languages [54]. From a raw text as input, Stanza delivers useful annotations such as tokenization, Multi-Word Token (MWT) expansion,[10] Part-of-Speech (POS)[11] and morphological feature[12] tagging, and lemmatization (tokenization, MWT expansion, and POS and morphological feature tagging are required to perform lemmatization using Stanza). It also produces annotations related to dependency parsing and named entity recognition. This latter resource was tested, but it did not improve the prediction results, but in fact, it showed a worsening.

The remaining number of terms, our potential number of features, is 47,107 after prep-processing stage. This is the number of features used in all the methods perspectives in this paper before the implementation of dimensionality reduction techniques.

At this point, each element of the vector $\boldsymbol{e}$ was pre-processed, and the data in it has yet the same shape, 6,429 rows.

If stanza had been used just for tokenization, i.e., lemmatization, MWT expansion and POS and morphological feature tagging had not been performed, that number would be 54,680. Even more, if punctuation marks, special characters and any accents had not been removed, that number would be 80,311. Thus, the pre-processing steps also provides a potential dimensionality reduction.

F. THE BAG OF WORDS MODEL AND THE DOCUMENT-TERM MATRIX

The pre-processed texts present in $\boldsymbol{e}$ are divided in two sets, training and test sets. This split is done here, to avoid any information leakage from test set, that would lead to biased results. From this stage onwards, the proposed operations are fitted over the training set, and the proposed transformations carried out over the training set and the test set.

Through the use of these two sets of $\boldsymbol{e}$ in a Bag of Words (BoW) model, a Document-Term Matrix (DTM) is constructed. BoW is a model to represent text in terms of words occurring in it, a list of them and their respective multiplicity [55]. DTM is a representation of the BoW model, where each row is a text of the corpus, each column is a unique term present in the corpus, and each element is an indication of how many times a term occurs in a text. Given that after the pre-processing stage, we are basing ourselves on 47,107 unique terms.

Since, after the pre-processing phase, the approach is basing on 47,107 unique terms, from the training set fit, our data now has 6,429 rows (number of e-mails) and 47,107 columns. DTM is represented by $\boldsymbol{M}$:

$$\boldsymbol{M} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & \cdots & m_{1,t} \\ m_{2,1} & m_{2,2} & m_{2,3} & \cdots & m_{2,t} \\ m_{3,1} & m_{3,2} & m_{3,3} & \cdots & m_{3,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{d,1} & m_{d,2} & m_{d,3} & \cdots & m_{d,t} \end{bmatrix} \tag{3}$$

where d is 6,429, and t is 47,107.

The DTM basic ranking is the occurrence count. In this proposal, to promote a feature weighting based on term frequency, the DTM is submitted to the TF-IDF measure.

G. TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure that assigns weights to the importance of a term for a text, which is inserted in a corpus [56]. The TF-IDF is given by:

$$w_{i,j} = tf_{i,j} \cdot \log\left(\frac{N}{df_i}\right) \tag{4}$$

In Eq. (4), $tf_{i,j}$ refers to the number of occurrences of the term i in document j, the total number of documents is N, and df_i refers to the number of documents containing i.

Thus, when the term i occurs many times in a small number of df_i documents, given the equation (4), a high weight is assigned to the term i in the df_i documents in which it occurs. Likewise, a lower weight is assigned to the term i, if this term occurs a few times in a document or if it occurs in many documents (a smaller weight is still assigned to a term i if it occurs in all N documents) [53].

At this point, the matrix $\boldsymbol{M}$ is expressed according to the TF-IDF measure by $\boldsymbol{F}$, yet with 6,429 rows and 47,107 columns.

$$\boldsymbol{F} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,t} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,t} \\ f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{d,1} & f_{d,2} & f_{d,3} & \cdots & f_{d,t} \end{bmatrix} \tag{5}$$

where d is 6,429, and t is 47,107.

$\boldsymbol{F}$ is used in both proposed methods, namely, Method 1 - based on feature selection and Method 2 - based on feature extraction. These two perspectives are discussed in the next subsections.

H. METHOD 1 - FEATURE SELECTION

The main objective of the feature selection techniques is to choose a subset of original features [57]. For this purpose, the features are ordered by a utility measure, whereby those

[9]Stemming also intends to transform a word into its common base form, but perform this just cutting the beginning or the end of a word (based on a list of prefixes and suffixes that are usually found in inflected words)

[10]MWT expansion, according to [54], refers to expand a raw token into multiple syntactic words.

[11]POS tagging process annotates the grammatical class of each token (in Stanza case, is possible tagging a term as belong to any class of Universal POS tags)

[12]Morphological Feature tagging works as an extension of POS tagging, through which it is possible to annotates words with features that distinguish their additional lexical and grammatical properties [54].

with the highest values will be selected according to the desired number of features. Unselected features, those with values less than the threshold value, are removed and no longer used in the following activities.

For the purposes of this work, the Chi-Square and the Mutual Information measures are used, whose input data is the DTM based on TF-IDF measure ($\boldsymbol{F}$). They are both uni-variate feature selection methods. The first one, Chi-Square, aims to measure the linear dependency between two random variables (a input feature and the target), whereas the second one, Mutual Information, also captures nonlinear relationships between the input feature under analysis and the target.

This perspective is denominated Method 1. These two perspectives are explained below.

1) CHI-SQUARE

Chi-square is a statistical test used to calculate the relationship degree between the feature variables and the target variable in a dataset (in this proposed approach, calculate how dependent on each of the incoming features the e-mail class (phishing mail or ham mail) is [58].

This test score is given by the equation (6):

$$\tilde{\chi}^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}, \tag{6}$$

where O_k is the number of observations of class (observed frequency) and E_k the number of expected observations of class if there is no relationship between the feature variable and the target variable (expected frequency). If the chi-square test is 0 (the null hypothesis), there is no association between both variables. They are independent. On the other hand, the higher is the chi-square value, the greater is the relationship between the two variables (the alternative hypothesis).

2) MUTUAL INFORMATION

Mutual information is a measure for quantifying the mutual dependence between two variables, based on the entropy (from the information theory) of a random variable. Mutual information calculates which amount of information is reached in a random variable from another random variable. In the context of this work proposal, identify how much information each feature provides to determine if an e-mail is a phishing or a legitimate mail [59].

The mutual information of two jointly continuous random variables is given by the double integral expressed in the Equation (7):

$$I(X, Y) = \int_Y \int_X p_{(X,Y)}(x, y) \log \left(\frac{p_{(X,Y)}(x, y)}{p_X(x) p_Y(y)} \right) dxdy, \tag{7}$$

where $p_X(x)$ is the probability density of x, $p_Y(y)$ is the probability density of y, and $p_{(X,Y)}(x,y)$ is the probability joint density, with X being the feature variable and Y being the target variable, or vice versa [60].

I. METHOD 2 - FEATURE EXTRACTION

The main objective of these techniques is to extract new features from the original features set. It is expected these new features bring more distinctive information about the texts, with less noise, gathering and making latent information from the underlying data explicit [61].

For the purposes of this work, the PCA and LSA techniques are used, whose input data is the DTM based on TF-IDF measure ($\boldsymbol{F}$). This approach is denominated Method 2. These two perspectives are elucidated below.

1) PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis is a technique that, according to [62], focuses on finding a mapping from the inputs in the original dimensional space to a new smaller dimensional space, always seeking for the minimum loss of information.

The principal components can be understood as the underlying structure in the data. They are found by the search for eigenvectors and eigenvalues that maximize the variance of projected data and make them more spread out in the new dimensional space. The basic idea of this technique consists of convert variables, potentially correlated, into linearly uncorrelated variables, the principal components, by an orthogonal transformation, as [59] exposes. Each of them is represented by a pair of eigenvectors and eigenvalues. The eigenvector represents the direction of a principal component, and the eigenvalue represents how much variance that direction contains. Then, the first principal component contains more variance from the original data than the second; the second principal component contains more variance than the third; and so on. All the principal components are orthogonal to each other.

PCA is based on covariance matrix [63], [64]. Although the code implementation of this technique uses different calculations in order to be more computationally efficient, the first step of the PCA is to normalize the input variables transforming them into unitary variance zero mean variables. Next step is to calculate the covariance matrix, which is done to compute the relationships between the data and also to reduce the size of the data, since usually the amount of samples is much greater than the amount variables. Then, from the covariance matrix, by an eigendecomposition, the eigenvectors and the respective eigenvalues, also known as principal components, are obtained. Thereon, a further step is to choose the significant components, also known as principal components, and to discard the irrelevant components. The data is then projected onto the principal components [65]. More details about PCA can be found in [66].

PCA can also be calculated through the use of SVD, explained in Subsubsection III-I2.

2) LATENT SEMANTIC ANALYSIS (LSA)

LSA refers to a mathematical technique in natural language processing, whose purpose is to make the topics embedded in the input data (the documents) explicit, from

the analysis of the relationships between these documents and the terms contained therein. Documents and terms are expressed as vectors of elements that correspond to these topics. Thus, the elements in these vectors indicate the degree of participation of a document or term in the represented topic [67], [68].

This technique is based on a factorization through Singular Value Decomposition (SVD) [53], [69] [70]. Using $\boldsymbol{F}_{d,t}$, which expresses the DTM based on TF-IDF measure, the SVD settings for this work are used as follows:

$$F_{d,t} = B_{d,m}\Sigma_{m,m}C^T_{t,m}, \tag{8}$$

where: $\boldsymbol{B}_{d,m}$ is the eigenvectors matrix of $\boldsymbol{D}_{t,t} = \boldsymbol{F}^T\boldsymbol{F}$, $\boldsymbol{C}_{t,m}$ is the eigenvectors matrix of $\boldsymbol{T}_{d,d} = \boldsymbol{F}\boldsymbol{F}^T$. $\boldsymbol{\Sigma_{m,m}}$ is the diagonal matrix of the singular values σ_i of $\boldsymbol{F}$ (for $i = 1, \ldots,$ min (6,429 $\times$ 47,107)), which are the square roots of the nonzero eigenvalues of $\boldsymbol{B}$ and $\boldsymbol{C}$.

In the particular case of the problem discussed in this research, matrix $\boldsymbol{D}$ is a matrix that expresses relation between the texts of the e-mail bodies (our documents), so if e-mail j and k have x terms in common, then $d_{j,k} = x$, while in matrix $\boldsymbol{T}$, which expresses relation between the terms, if the terms l and m occur together in y e-mail bodies, then $c^T{}_{l,m} = y$ [53]. Similarly, as explained in [69], $\boldsymbol{B}_{d,m}$ maps terms to topics ($b_{i,j}$ is the weight of term i in the topic j) and $\boldsymbol{\Sigma_{m,m}C}_{t,m} = S$ maps topics to documents ($s_{i,j}$ is the weight of the topic i in the documents j). In this way, LSA would be used in a similar way as Latent Dirichlet Allocation (LDA) is used in [47].

To work with k singular values, this decomposition of $\boldsymbol{F}$ is truncated with k elements, as expressed in (9), as shown at the bottom of the page.

Truncated SVD maintains only the first k columns of $\boldsymbol{B}_{d,m}$, the first k lines and the first k columns of $\boldsymbol{\Sigma_{m,m}}$ and the first k lines of $\boldsymbol{C}^T{}_{t,m}$. That is, the coefficients of these matrices perform a projection onto a k-dimensional Space.

Some details about PCA and LSA are particularly noteworthy. The main difference between PCA, when using SVD, and LSA is the feature-wise normalization. PCA performs it over the DTM before executing SVD, whereas LSA executes SVD directly, without this normalization. Thus while PCA tries to reproduce the highest amount of variance of the original data, LSA does not to scale up the weight of rarely occurring terms. Both techniques aim to remove some of the noise of the data, provide improved similarity measures among the instances (documents), and reduce the dimensionality [66], [68].

Another relevant point refers to the DTM transformation proposed by the SVD. It considers the underlying process as a process defined by a normal distribution. While the word occurrence count in a text, as well as phishing in a set of incoming e-mails, may be better explained by being conceived as a process governed by a Poisson distribution, what depending on the followed paradigm would be an inconsistency [71]. The point here is although the elements of DTM are derived from the term occurrence count in the texts, they are not used as such, but rather as the weights of its discriminative features in documents similarities [16], and this weight can be ranked based on the term occurrence count, its frequency, TF-IDF or other measures. The other perspectives based on VSM (such as the Chi-Squared used in Method 1) follow the same conception since their methods are designed for normally-distributed data.

J. FEATURES ATTRIBUTES

The feature amount choices for the two methods and their perspectives are defined as follows.

For Method 1, the techniques used in its two perspectives are based on a ranking of the highest values that each feature provides according to a certain measure of utility (chi-squared or mutual information), that is, a measure that captures non-linear relationships between variables [57]. In this sense, they also followed the feature quantity settings employed in Method 2 perspectives.

For Method 2, in the case of PCA, one of the most commonly used criteria for selecting a quantity of the obtained principal components is to select a Cumulative Percentage of Total Variation (CPTV) [72], [73] [66], that is a percentage of variance, given by the sum of the variances of the first n principal components. For LSA, the definition of the number of dimensions to select was presented as a decision based on empiricism since the objective is to find an optimal dimensionality, that brings similar or better results for the process using it (through the correct induction of underlying similarity relations) [74].

Thus, since in PCA, it is typically indicated working with around an amount between eighty and ninety percent (depending on the practical details of the dataset under analysis) of the initial variance [66], amounts of principal components between 2 and 100 were chosen as the number of features, which represent about 86.29% and 98.70% of the variance respectively. The results for these feature quantities were better than those shown when selecting 160 features (99.00% of the initial variance). Possibly because this last variance percentage also captures, along with the tendency of the underlying process, some amount of noise. For LSA, we also tested several quantities of singular values between

$$\begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,t} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,t} \\ f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{d,1} & f_{d,2} & f_{d,3} & \cdots & f_{d,t} \end{bmatrix} = \begin{bmatrix} b_{1,1} & \cdots & b_{1,k} & \cdots & b_{1,n} \\ b_{2,1} & \cdots & b_{2,k} & \cdots & b_{2,n} \\ b_{3,1} & \cdots & b_{3,k} & \cdots & b_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{d,1} & \cdots & b_{d,k} & \cdots & b_{d,n} \end{bmatrix} \cdot \begin{bmatrix} \sigma_{1,1} & \cdots & \sigma_{1,k} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \cdots & \sigma_{2,k} & \cdots & \sigma_{2,n} \\ \sigma_{k,1} & \cdots & \sigma_{k,k} & \cdots & \sigma_{k,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sigma_{d,1} & \cdots & \sigma_{d,k} & \cdots & \sigma_{d,n} \end{bmatrix} \cdot \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \cdots & c_{1,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{k,1} & c_{k,2} & c_{k,3} & \cdots & c_{k,t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & c_{n,3} & \cdots & c_{n,t} \end{bmatrix} \tag{9}$$

2 and 100, trying to achieve the best classification predictions with the least amount of features.

It is important to note that most of the research described in Section II does not address this critical point of how many features to select in each of the employed techniques. This item has a relevant influence on the obtained models' performance, as can be inferred from the results expressed in Section IV.

These four techniques of the proposed methods are used in this work as follows: after passing it through the pre-processing steps explained above, generating the DTM through the BoW representation based on word unigrams and performing TF-IDF over it, choose the number of features that we want to work on and perform the Method 1 and Method 2 perspectives. In this way, from the selected or the extracted features, respectively, the e-mails can be represented in a lower-dimensional space.

This low-dimensional space has the same rows quantity (6,429), and options from 2 to 100 columns according to the number of chosen features. Our best setting among the four perspectives is found using LSA in Method 2, employing twenty-five singular values as features, and its matrix has 6,429 rows and 25 columns. The matrix $\boldsymbol{S}$ represents this setting:

$$\boldsymbol{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3} & \cdots & s_{1,25} \\ s_{2,1} & s_{2,2} & s_{2,3} & \cdots & s_{2,25} \\ s_{3,1} & s_{3,2} & s_{3,3} & \cdots & s_{3,25} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{d,1} & s_{d,2} & s_{d,3} & \cdots & s_{d,25} \end{bmatrix} \tag{10}$$

K. CLASSIFICATION

At this stage, from the features attributes provided by the feature engineering phase, models are trained to properly fit the e-mail characteristics, achieving a discriminating function to classify[13] these e-mails instances as legitimate or phishing e-mail.

The obtained features sets are derived from two methods, Method 1 and Method 2, each with two perspectives (using a different technique in each of them), as stated earlier. Before implementing the proposed dimensionality reduction in these four perspectives, the proposed approach pre-processes the text of the e-mail bodies, represents them in a DTM shape, and calculates TF-IDF over it. After that, while Method 1 uses Chi-Squared or Mutual information to promote its feature selection, Method 2, through the use of PCA or LSA, extracts reduced features sets. Both methods try to feed the classification ML algorithms with the most distinctive representation from their respective feature sets.

The proposal also draws a course of action to the classification, in order to provide a holistic approach to phishing detection, extracting the most suitable configuration from each of the proposed stages. This stage strategy starts separating the two portions (the training and test sets, already folded since the DTM construction stage) of the feature set of each proposed perspective, such as $\boldsymbol{S}$. The training set refers to 70% of $\boldsymbol{S}$, and the test set the remaining 30%.

During the training step, it is employed, the folding plan presented in [75], to perform the training and validation subsets partitioning. This plan folds two random stratified subsets (each one with 50% of the training set), uses one of them for training and another for validation, and after that the inverse, i.e., the first subset for validation and the remaining for training. This folding scheme is repeated five times.

This strategy for folding was not yet used for phishing detection approaches based on ML algorithms. In this context, our previous work presented in [47] and this approach were the first employment of this plan.

Also, in the training step, a wide variety of hyperparameters settings are tested to estimate the befitting configuration to the proposed e-mail data. The specified folding and cross-validation plan is executed for each of these configuration sets in order to evaluate its respective results.

After this training process, the ML models for the phishing classification problem are reached from each of the employed ML algorithms. These models are then tested using unseen data, those in the test set.

As exposed in Subsection III-A, the data set is based on 6,429 e-mails. Thus, the training set contains 4,500 samples, and the test set contains 1,929 samples. The training set consists of 2,916 ham e-mails and 1,584 phishing mails, and the test set consists of 1,251 ham e-mails and 678 phishing e-mails. This split is performed over $\boldsymbol{e}$, before the DTM construction, in this sense, $\boldsymbol{e}$, $\boldsymbol{M}$, $\boldsymbol{F}$ and $\boldsymbol{S}$ are already divided in training and test portions, to properly fit the proposed operations to the training set, and perform these transformations on both sets.

This classification strategy, integrated with all the proposed architecture, is presented in Fig. 3.

Eight ML algorithms are used to perform the proposed classification (phishing detection task): Support Vector Machines (SVM) [62], [76], Naive Bayes Classifier [59], Logistic Regression for classification [62], k-Nearest Neighbor [59], Decision Trees [62], Random Forest [62], Extreme Gradient Boosting (XGBoost) [77] and Multilayer Perceptron (MLP) [62].

IV. RESULTS AND APPROACH EVALUATION

A detailed evaluation of the proposed approach through its prediction results is presented in this section. In Subsection IV-A, the utility measures to assess the results of our methods are presented. The results are described in Subsection IV-B, and some pertinent discussions are outlined in the Subsection IV-C.

A. MEASURES

To evaluate the classification algorithms performance in each perspective of the proposed methods, it is used the measures: accuracy, precision, recall, false positive rate, specificity and

[13] Classification is a supervised learning that aims to map the input data for a given output. The correctness is provided along with the input data, in the labels shape indicating which class each instance belongs to.

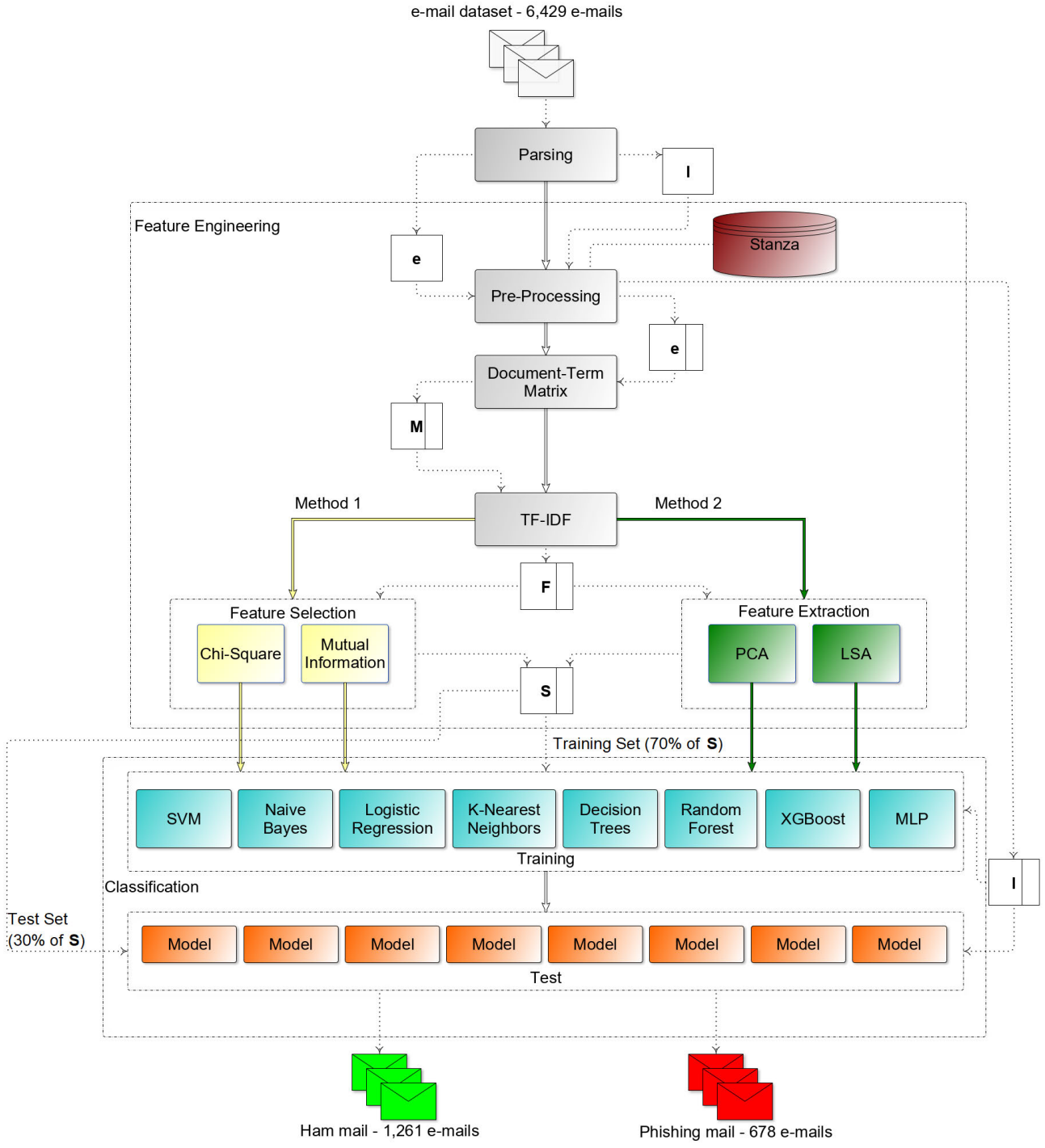


FIGURE 3. The proposal detailed architecture and its dataflow.

F1 score. Their equations are expressed below, in function of true positive (t_p), false positive (f_p), false negative (f_n), and true negative (t_n) rates.

Accuracy (a):

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \tag{11}$$

Precision p:

$$p = \frac{t_p}{t_p + f_p} \tag{12}$$

Recall (r), True Positive Rate (tpr) or Sensitivity:

$$r = \frac{t_p}{t_p + f_n} \tag{13}$$

False Positive Rate (fpr):

$$\text{fpr} = \frac{f_p}{f_p + t_n} \tag{14}$$

Specificity or True Negative Rate (tnr):

$$\text{tnr} = 1 - fpr \tag{15}$$

F1 Score (f_1):

$$f_1 = 2 \cdot \frac{p \cdot r}{p + r} \tag{16}$$

B. RESULTS

As previously announced, 47,107 are the amount of features employed in both methods before performing the feature selection or feature extraction techniques to dimensionality reduction, which is the same number of output terms from the lemmatization process, explained in subsection III-E.

All perspectives are performed in variations based on selecting or extracting 2, 3, 5, 10, 25, 50, and 100 features. Below, the obtained marks are present for each perspective in descending order of their respective best scores. These marks are based on the weighted metrics of the proposed measures for the two classes (phishing or legitimate e-mail labels).

1) METHOD 1 - PERSPECTIVE BASED ON CHI-SQUARE MEASURE

For this Method 1 perspective, Chi-Square measure is used as dimensionality reduction approach. Its prediction assessment values are available in the tables 1 and 2. From the original features in DTM columns weighted through the use of TF-IDF, a desired number of features is selected based on this measure.

The results attained through this perspective with a hundred features are presented in Table 1. In this setting, it was obtained accuracy, precision, recall, F1 score, and Specificity rates of 100%, which is, to the best of our knowledge, the highest result in phishing detection researches using just 100 features, and the best mark using Chi-Square measure in Method 1. This highly prized measure is achieved using Random Forest ML classification algorithm, with the entropy as function to measure the quality of a split, $\log_2 100$ as the number of features to consider when looking for the best split, ten as the minimum number of samples required to split an internal node, and the rest of its parameters in the default setting.

TABLE 1. Results of Perspective Chi-Square of Method 1 - feature set with 100 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVC	0.9974	0.9974	0.9974	0.9974
Naive Bayes	0.9703	0.9708	0.9703	0.9700
Logistic Regression	0.9974	0.9974	0.9974	0.9974
KNN	0.9984	0.9984	0.9984	0.9984
Decision Trees	0.9922	0.9922	0.9922	0.9922
Random Forest	**1.0000**	**1.0000**	**1.0000**	**1.0000**
XGBoost	0.9995	0.9995	0.9995	0.9995
MLP	0.9984	0.9984	0.9984	0.9984

The results expressed in Table 2 refer to those attained in the remaining proposed variations using this perspective.

TABLE 2. Results of Perspective Chi-Square of Method 1 - all feature set variations.

features	Algorithm	Accuracy	F1 Score
2	KNN	0.9708	0.9707
3	KNN	0.9734	0.9733
5	KNN	0.9744	0.9743
10	XGBoost	0.9896	0.9895
25	Random Forest	0.9958	0.9958
50	Random Forest and XGBoost	0.9995	0.9995
100	**Random Forest**	**1.0000**	**1.0000**

2) METHOD 1 - FEATURE SELECTION: PERSPECTIVE BASED ON MUTUAL INFORMATION MEASURE

For this Method 1 perspective, Mutual Information measure is used as dimensionality reduction approach. Its prediction assessment values are available in the tables 3 and 4. From the original features in DTM columns weighted through the use of TF-IDF, a desired number of features is selected based on this measure.

The results attained through this perspective with twenty-five features are presented in Table 3. In this setting, it was obtained accuracy, precision, recall, F1 score, and Specificity rates of 99.90%, which is the best mark using Mutual Information measure in Method 1. This measure is achieved using Random Forest ML classification algorithm, with the entropy as function to measure the quality of a split, 5 as the number of features to consider when looking for the best split, 2 as the minimum number of samples required to split an internal node, and the rest of its parameters in the default setting.

TABLE 3. Results of Perspective Mutual Information of Method 1 - feature set with 25 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVC	0.9969	0.9969	0.9969	0.9969
Naive Bayes	0.9844	0.9844	0.9844	0.9843
Logistic Regression	0.9969	0.9969	0.9969	0.9969
KNN	0.9979	0.9979	0.9979	0.9979
Decision Trees	0.9969	0.9969	0.9969	0.9969
Random Forest	**0.9990**	**0.9990**	**0.9990**	**0.9990**
XGBoost	0.9984	0.9984	0.9984	0.9984
MLP	0.9969	0.9969	0.9969	0.9969

The marks reached by Method 1 through the Mutual Information measure in the remaining proposed variations are presented in Table 4.

3) METHOD 2 - FEATURE EXTRACTION: PERSPECTIVE BASED ON PRINCIPAL COMPONENT ANALYSIS

For this Method 2 perspective, Principal Component Analysis is used as dimensionality reduction approach. Its prediction

TABLE 4. Results of Perspective Mutual Information of Method 1 - all feature set variations.

features	Algorithm	Accuracy	F1 Score
2	KNN	0.9797	0.9796
3	Decision Trees	0.9838	0.9838
5	Decision Trees	0.9932	0.9932
10	K-Nearest Neighbors	0.9974	0.9974
25	**Random Forest**	**0.9990**	**0.9990**
50	Random Forest and XGBoost	0.9984	0.9984
100	Random Forest and XGBoost	0.9984	0.9984

TABLE 5. Results of Perspective PCA of Method 2 - feature set with 10 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVC	0.9969	0.9969	0.9969	0.9969
Naive Bayes	0.9760	0.9764	0.9760	0.9759
Logistic Regression	0.9969	0.9969	0.9969	0.9969
KNN	0.9990	0.9990	0.9990	0.9990
Decision Trees	0.9969	0.9969	0.9969	0.9969
Random Forest	0.9984	0.9984	0.9984	0.9984
XGBoost	**0.9995**	**0.9995**	**0.9995**	**0.9995**
MLP	0.9969	0.9969	0.9969	0.9969

TABLE 6. Results of Perspective PCA of Method 2 - all feature set variations.

features	Algorithm	Accuracy	F1 Score
2	K-Nearest Neighbors	0.9953	0.9953
3	K-Nearest Neighbors	0.9974	0.9974
5	Decision Trees	0.9984	0.9984
10	**XGBoost**	**0.9995**	**0.9995**
25	XGBoost	0.9995	0.9995
50	Logistic Regression	0.9995	0.9995
100	XGBoost	0.9995	0.9995

TABLE 7. Results of Perspective LSA of Method 2 - feature set with 25 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9984	0.9984	0.9984	0.9984
Naive Bayes	0.9734	0.9734	0.9734	0.9733
Logistic Regression	0.9995	0.9995	0.9995	0.9995
KNN	0.9990	0.9990	0.9990	0.9990
Decision Trees	0.9984	0.9984	0.9984	0.9984
Random Forest	0.9995	0.9995	0.9995	0.9995
XGBoost	**1.0000**	**1.0000**	**1.0000**	**1.0000**
MLP	0.9995	0.9995	0.9995	0.9995

assessment values are available in the tables 5 and 6. From the original features in DTM columns weighted through the use of TF-IDF, a desired number of features is extracted based on principal components, projecting the original feature set in a reduced low-dimension space.

The results attained through this perspective with ten features are presented in Table 5. In this setting, it was obtained accuracy, precision, recall, F1 score, and Specificity rates of 99.95%, which is the best mark using PCA in Method 2. This measure is achieved by using the XGBoost classification algorithm, with the subsample as 0.6, the minimum split loss reduction - gamma as 0.5, the maximum depth of a tree as 4, the minimum sum of instance weight needed in a child as 1, and the rest of its parameters in the default setting.

The results expressed in Table 6 refer to those attained in the remaining proposed variations using this perspective.

4) METHOD 2 - FEATURE EXTRACTION: PERSPECTIVE BASED ON LATENT SEMANTIC ANALYSIS

For this Method 2 perspective, Latent Semantic Analysis is used as dimensionality reduction approach. Its prediction assessment values are available in the tables 7 and 8. From the original features in DTM columns weighted through the use of TF-IDF, a desired number of features is extracted based on singular values, projecting the original feature set in a reduced low-dimension space. Below, the obtained marks to 2, 3, 5, 10, 25, 50, and 100 features are presented in descending order of their respective best scores.

The results attained through this perspective with twenty-five features are presented in Table 7. In this setting, it was obtained accuracy, precision, recall, F1 score, and Specificity rates of 100%, which is, to the best of our knowledge, the highest result in phishing detection researches using just 25 features, and the best mark using LSA measure in Method 2. This highly prized measure is achieved through the XGBoost classification algorithm, with the subsample as 0.6, the minimum split loss reduction - gamma as 0.5, the maximum depth of a tree as 4, the minimum sum of instance weight needed in a child as 1, and the rest of its parameters in the default setting.

The marks reached by Method 2 through the Latent Semantic Analysis in the remaining proposed variations are presented in Table 8.

C. DISCUSSIONS

Based on the prediction results for the proposed approach expressed in subsection IV-B, it is plotted a chart with the best performance marks in each variation of the proposed perspectives. This chart is displayed in Fig 4.

Method 2 has the best performance overall: with all the variations and all marks above or equal to 99.53%. LSA has the best fulfillment of the proposed approach, achieving 100% in accuracy, precision, recall, and F1 score with just

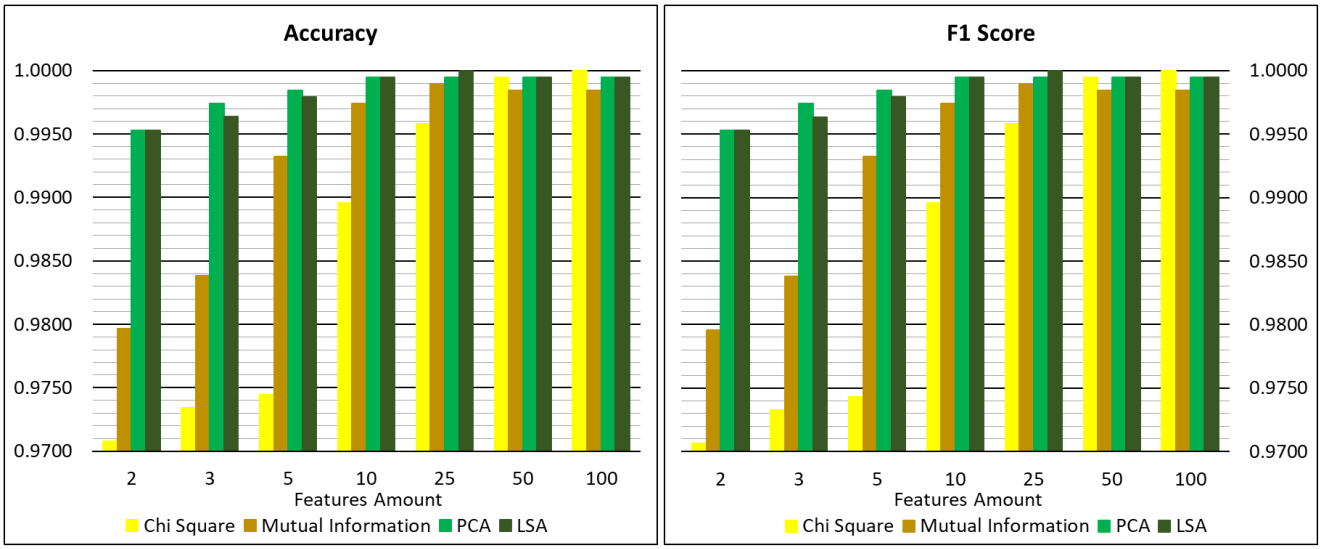


FIGURE 4. Accuracy and F1 Score of the proposed methods in their respective perspectives in each tested feature amount variations.

TABLE 8. Results of Perspective LSA of Method 2 - all feature set variations.

features	Algorithm	Accuracy	F1 Score
2	XGBoost	0.9953	0.9953
3	KNN and Random Forest	0.9963	0.9963
5	XGBoost and Randon Forest	0.9979	0.9979
10	Random Forest	0.9995	0.9995
25	**XGBoost**	**1.0000**	**1.0000**
50	Random Forest and XGBoost	0.9995	0.9995
100	XGBoost	0.9995	0.9995

25 features. Both perspectives of Method 2, PCA and LSA, have four out seven feature set variations reaching above or equal 99.95%. Using 2 features, these perspectives have the same result, an accuracy of 99.53%. After that PCA has better results when employing 3 and 5 features. For ten features, they present the same mark of 99.95% for F1 Score. PCA maintains this results for 25 features, whereas LSA reaches 100% successful in all marks. They yet attain an F1 score 99.95% for the variations with 50 and a 100 features.

The Chi-Square perspective of Method 1 also achieves 100% in accuracy, precision, recall, and F1 score, but performs it with 100 features, which is also an excellent result. Comparing the two perspectives of Method 1, it is observed that Mutual Information perspective has better results for fewer features than the Chi-Square perspective, which, in turn, is slightly more accurate for 50 and 100 features. Chi-Square perspective yet attains the same mark as LSA and PCA using 50 features (F1 score of 99.95%). Mutual information perspective achieves its best mark (F1 score of 99.90%) using 25 features.

All four perspectives have their worst marks using sets of 2, 3 and 5 features respectively.

In Table 9, it is confronted this proposal performance, using the marks displayed in Subsection IV-B, with some state-of-the-art research intended to detect phishing, already described in the baseline study (Section II).

Our best performance, achieved in the LSA perspective of the Method 2 (with 25 features) and in Chi-Square perspective of the Method 1 (with 100 features), are the highest among those compared. These are, to the best of our knowledge, the highest result in phishing detection researches for an accredited data set based only on the body of the e-mails. In [34], an F1 score of 100% is also attained, but it is done using 200 features, a feature set eight times greater than ours.

Using ten features, the LSA and the PCA perspectives of Method 2 reach the same marks (Accuracy and F1 score of 99.95%) as the previous work [47]. When using five features, the PCA perspective outperforms it achieving an F1 score of 99.84% (against a mark of 99.69% of [47]). PCA perspective yet obtains an F1 Score of 99.74% when employing three features, the same mark presented in [47].

The representation based on two features is particularly valuable due it enables visualization of the samples scattering, and how each perspective establishes the frontier between the classes. It is portrayed in Fig. 5, in a pair plot way, i.e., creating a grid of axes, where each feature is shared in the y-axis across a single row and in the x-axis across a single column, displaying pairwise relationships in the dataset.

For Method 1, in Chi-Square perspective, the two selected features are the tokens "be" and "pic", whereas in the Mutual Information, they are the tokens "be" and "http". For Method 2, in both perspectives, PCA and LSA, the two features are obtained projecting portions of the original feature set in a 2-dimensional space.

TABLE 9. Performance Comparison.

Reference	Best Metric Value	Amount of Features	Dataset Source
Proposed M2 - LSA	100%	25	[49] and [48]
Proposed M1 - Chi-Square	100%	100	[49] and [48]
Ramanathan and Wechsler [34]	100%	200	Includes some samples of [49] and [48]
Gualberto et al. [47]	99.95%	10	[49] and [48]
Proposed M2 - PCA	99.95%	10	[49] and [48]
Proposed M1 - Mutual Info	99.90%	25	[49] and [48]
Gualberto et al. [47]	99.90%	95	[49] and [48]
Fang et al. [46]	99.85%	256	Includes some samples of [49] and [48]
Akinyelu and Adewumi [31]	99.70%	15	[49] and [48]
L'Huillier et al. [33]	99.58%	1017	[49] and [48]
Gangavarapu et al. [43]	99.40%	21	[49] and [48]
Daeef et al. [26]	99.40%	48	[49] and [48]
Vinayakumar et al. [78]	99.10%	200	Includes some samples of [49] and [48]
Yasin and Abuhasan [29]	99.10%	16	[49] and [48]
Barathi Ganesh et al. [39]	99%	100	Includes some samples of [49] and [48]
Gangavarapu and Jaidhar [42]	99%	240	[49] and [48]
Halgas et al. [45]	98.63%	5000	[49] and [48]
Chin et al. [44]	98.39%	30	Includes some samples of [49] and [48]
Fette et al. [24]	97.64%	10	[49] and [48]
Islam and Abawajy [27]	97.00%	21	[49] and [48]

From the figures 4 and 5, and the results presented in Subsection IV-B, it is possible to infer that the arrangement of the samples in the perspectives based on feature extraction techniques better separate the proposed classes when dealing with a reduced feature set, and that the similarities between the samples of the same class becomes more evident in the perspectives based on the feature selection measures as the feature set becomes larger.

The marks attained through the two methods show consistency. In the Method 1, for the Chi-Square perspective (in variations from 25 to 100 features) and for the Mutual Information (in variations from 5 to 100 features), the measured metrics return scores equal or higher than 99.32%. In Method 2, for both perspectives (for all the feature set variations), this percentage is not less than 99.53% for any feature set.

The use of Stanza is also a noteworthy item. Obtaining the excellent prediction results described goes through its use in tokenization, POS tagging and lemmatization tasks. For instance, using 10 features, if it were used only for tokenization, the best result would be an F1 score of 98.75% using the KNN ML algorithm. Besides that, two other factors influenced the achievement of these remarkable results. They are the pre-processing steps and the resampling/cross-validation techniques employed in this research. These components of the proposed architecture, jointly with the dimensionality reduction perspectives, fed the adopted ML algorithms. They, optimally set, result in models that obtain better results than those described in the baseline study.

The proposed methods and their perspectives handled successfully with the VSM problems. Without their proposed dimensionality reduction and the lemmatization step for instance, considering all the DTM features in this setting (54,680), the best result would be an F1 score of 99.74%. A good result, but at the cost of much greater computational complexity and processing time. Thus, the proposal provides a dense and low-dimension feature set in many size variations that address these troubles and increase the prediction results of the original feature set. When all the proposed stages of the architecture are implemented, before the methods perspectives are applied over the DTM weighted by the TF-IDF measure, the sparsity of the $\boldsymbol{F}$ matrix is around 99,77%, and after this implementation, the sparsity become about 33% (feature set of 2 attributes) to 50.35% (feature set of 100 attributes) in $\boldsymbol{S}$.

Beyond being the ML algorithms that achieve an F1 score of 100%, Random Forest and XGBoost frequently achieve the best mark in each variation. Random Forest in Method 1, and XGBoost in Method 2. These measures indicate a pattern of great prediction results for their use (optimally set) in phishing detection, using features based on the text of the e-mail bodies.

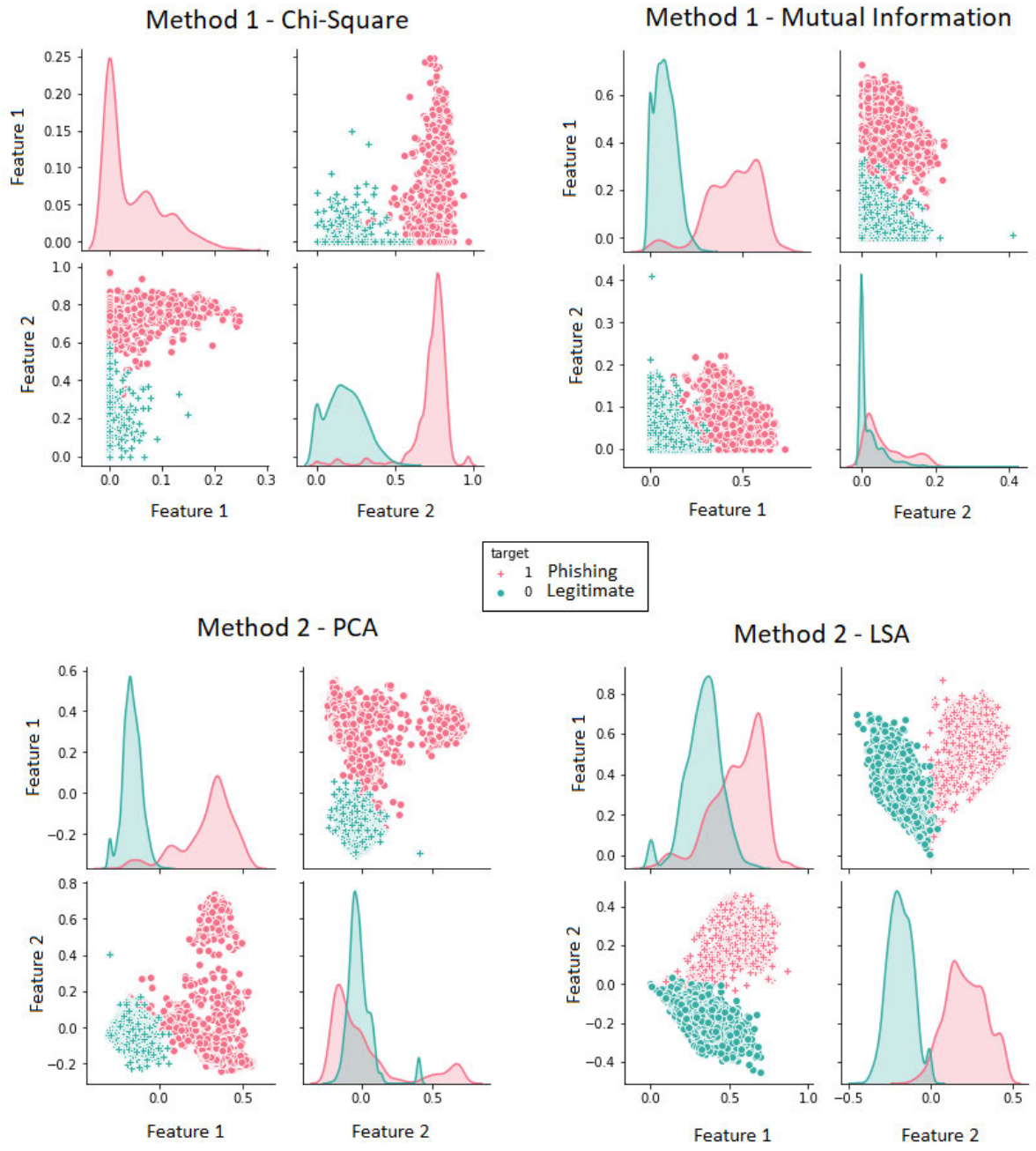


FIGURE 5. Representations Based on Two Features.

V. CONCLUSION AND FUTURE WORK

Considering the phishing detection landscape, this research proposes a multi-stage approach aimed to detect phishing e-mails with an overall approach based on combined techniques related to text processing, feature engineering, feature selection, feature extraction, machine learning training techniques, and improved classification algorithms. The central aspect is to synthesize enhanced features attributes for phishing detection, which feed the training and testing tasks of the ML classification algorithms and yield to improved predictions.

The proposed methods demonstrated optimal performances with reduced features sets based only on the text, when compared with several state-of-the-art research. The LSA perspective of Method 2 attains an F1 score of 100% using the XGBoost algorithm fed with twenty-five features, whereas the Chi-Square perspective of Method 1 reaches the same prominent mark using the Random Forest algorithm

fed with one hundred features. Most of their tested variations had very high-quality results, even using a set of only two features.

These improved results are achieved thanks to the use of selected techniques in each proposed stage, such as the POS tagging and lemmatization tasks implemented with Stanza, the improved learning strategy for re-sampling and cross-validation, and the estimation of hyper-parameters configuration, as well as the overall feature engineering process based on dimensionality reduction. All the perspectives dealt with ''the curse of the dimensionality" and the high sparsity, as well as they improved the representation of the texts contextual information related to phishing. In this sense, the proposed architecture is a significant research contribution to detect this type of cybercrime.

As a prospect of future research objectives, the use of the word embedding technique is listed, since its employment, combined with our pre-processing approach, can generate document representations based on fix-sized dense vectors directly from the extracted tokens.

We also aim to implement approaches to detect phishing based on deep learning, language models, and transformers, considering that their employment can provide advantages as, for instance, a more refined fit to pre-trained models or their use with other languages different from the initial datasets language.

REFERENCES

[1] I. W. Stats. *Internet Usage Statistics—The Internet Big Picture: World Internet Users and 2020 Population Stats*. Accessed: Jun. 26, 2020. [Online]. Available: https://www.internetworldstats.com/stats.htm

[2] CISCO. *Cisco Annual Internet Report (2018-2023) White Paper*. Accessed: Jun. 26, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[3] (APWG). (May 2020). *Phishing Activity Trends Reports: 1st Quarter 2020 Plus Covid-19 Coverage*. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2020.pdf

[4] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Expert Syst. Appl.*, vol. 106, pp. 1–20, Sep. 2018, doi: 10.1016/j.eswa.2018.03.050.

[5] J. Singh, "Detection of phishing e-mail," *Int. J. Comput. Sci. Technol.*, vol. 2, no. 3, pp. 547–549, 2011.

[6] A. Aleroud, L. Zhou, "Phishing environments, techniques, and countermeasures," *Comput. Secur.*, vol. 68, pp. 160–196, Apr. 2017, doi: 10.1016/j.cose.2017.04.006.

[7] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2070–2090, 4th Quart., 2013, doi: 10.1109/SURV.2013.030713.00020.

[8] M. V. Kunju, E. Dainel, H. C. Anthony, and S. Bhelwa, "Evaluation of phishing techniques based on machine learning," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, May 2019, pp. 963–968.

[9] J. Martínez Torres, C. Iglesias Comesaña, and P. J. García-Nieto, "Review: Machine learning techniques applied to cybersecurity," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2823–2836, Oct. 2019, doi: 10.1007/s13042-018-00906-1.

[10] C. N. Gutierrez, T. Kim, R. D. Corte, J. Avery, D. Goldwasser, M. Cinque, and S. Bagchi, "Learning from the ones that got away: Detecting new forms of phishing attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 988–1001, Nov. 2018.

[11] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. 2nd Annu. Crime Researchers Summit*, New York, NY, USA, 2007, pp. 60–69, doi: 10.1145/1299015.1299021.

[12] V. R. Nidhin A. Unnithan, and N. Harikrishnan, "Detecting phishing e-mail using machine learning techniques," in *Proc. 8th ACM Conf. Data Appl. Secur. Privacy*, 2018, pp. 51–54.

[13] R. Hassanpour, E. Dogdu, R. Choupani, O. Goker, N. Nazli, "Phishing e-mail detection by using deep learning algorithms," in *Proc. ACMSE Conf.*, New York, NY, USA, 2018, p. 1, doi: 10.1145/3190645.3190719.

[14] Y. Goldberg, G. Hirst, *Neural Network Methods for Natural Language Processing*. San Rafael, CA, USA: Morgan & Claypool, 2017, doi: 10.2200/S00762ED1V01Y201703HLT037.

[15] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing," in *Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[16] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, Feb. 2010, doi: 10.1613/jair.2934.

[17] D.-h. Shin, "Applications of artificial intelligence in the export control domain," in *Proc. Intell. Syst. Conf.*, Y. Bi, S. Kapoor, R. Bhatia, Eds. Cham, Switzerland: Springer, 2018, pp. 1005–1011, doi: 10.1007/978-3-319-56991-8.

[18] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. 8th Int. Conf. Artif. Neural Netw.* Berlin, Germany: Springer-Verlag, 2005, pp. 758–770, doi: 10.1007/11494669_93.

[19] K. Liu, A. Bellet, F. Sha, "Similarity learning for high-dimensional sparse data," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, G. Lebanon, S. V. N. Vishwanathan, Eds., vol. 38, San Diego, CA, USA, 2015, pp. 653–662.

[20] K. Erk and S. Padó, "A structured vector space model for word meaning in context," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 897–906.

[21] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, Dec. 2015, doi: 10.1186/s40537-014-0007-7.

[22] T. Galibus, T. P. de B. Vieira, E. P. de Freitas, R. de O. Albuquerque, J. A. P. C. L da Costa, R. T. de Sousa Jánior, V. Krasnoproshin, A. Zaleski, H. E. R. M. Vissia, and G. D. Galdo, "Offline mode for corporate mobile client security architecture," *Mobile Netw. Appl.*, vol. 22, no. 4, pp. 743–759, Aug. 2017, doi: 10.1007/s11036-017-0839-4.

[23] D. Tenório, J. Costa, and R. Souza Jánior, "Greatest eigenvalue time vector approach for blind detection of malicious traffic," in *Proc. 8th Int. Conf. Forensic Comput. Sci.*, Aug. 2013, pp. 1–5.

[24] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 649–656.

[25] I. R. A Hamid, J. Abawajy, and T.-H. Kim, "Using feature selection and classification scheme for automating phishing email detection," *Stud. Informat. Control*, vol. 22, no. 1, pp. 61–70, Mar. 2013.

[26] A. Y. Daeef, R. Ahmad, Y. Yacob, Y. Naãimah, and K. N. F. K. Azir, "Multi stage phishing email classification," *J. Theor. Appl. Inf. Technol.*, vol. 83, no. 2, pp. 1–7, 2016.

[27] M. R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *J. Netw. Comput. Appl.*, vol. 36, pp. 324–335, Apr. 2013, doi: 10.1016/j.jnca.2012.05.009.

[28] F. Toolan and J. Sarthy, "Feature selection for spam and phishing detection," in *Proc. eCrime Researchers Summit*, 2010, pp. 1–12.

[29] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *Int. J. Netw. Secur. Appl.*, vol. 8, no. 4, pp. 55–72, Jul. 2016, doi: 10.5121/ijnsa.2016.8405.

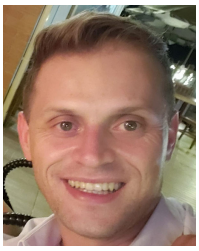
[30] R. Verma, N. Shashidhar, and N. Hossain, "Detecting phishing emails the natural language way," in *Computer Securit*, S. Foresti, M. Yung, F. Martinelli, Eds. Berlin, Germany: Springer, 2012, pp. 824–841, doi: 10.1007/978-3-642-33167-1_47.

[31] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *J. Appl. Math.*, vol. 2014, pp. 1–6, Dec. 2014, doi: 10.1155/2014/425731.

[32] M. Zareapoor, "Feature extraction or feature selection for text classification: A case study on phishing email detection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 7, no. 2, pp. 60–65, Mar. 2015.

[33] G. L'Huillier, A. Hevia, and R. Weber, "Latent semantic analysis and keyword extraction for phishing classification.," in *Proc. ISI*, 2010, pp. 129–131, doi: 10.1109/ISI.2010.5484762.

[34] V. Ramanathan and H. Wechsler, "PhishGILLNET—Phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training," *EURASIP J. Inf. Secur.*, vol. 2012, no. 1, p. 1, Dec. 2012, doi: 10.1186/1687-417X-2012-1.
[35] V. Ramanathan and H. Wechsler, "Phishing detection and impersonated entity discovery using conditional random field and latent Dirichlet allocation," *Comput. Secur.*, vol. 34, pp. 123–139, May 2013, doi: 10.1016/j.cose.2012.12.002.
[36] N. A. Unnithan, N. B. Harikrishnan, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Machine learning based phishing e-mail detection security-ceneamrita," in *Proc. CEUR Workshop Proc.*, 2124, 2018, pp. 64–68.
[37] N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "A machine learning approach towards phishing email detection cen-security Iwspa 2018," in *Proc. CEUR Workshop Process.*, vol. 2124, 2018, pp. 21–28.
[38] A. Vazhayil. (2017). *Ped-ML: Phishing Email Detection Using Classical Machine Learning Techniques Censec Amrita*. [Online]. Available: http://ceur-ws.org/Vol-2124/#paper_11
[39] H. Barathi Ganesh, R. Vinayakumar, M. Anand Kumar, and K. Soman, "Distributed representation using target classes: Bag of tricks for security and privacy analytics Amrita-Nlpiwspa-2018," in *Proc. CEUR Workshop Process.*, vol. 2124, 2018, pp. 10–15.
[40] R. Vinayakumar, H. Barathi Ganesh, M. Anand Kumar, K. Soman, and P. Poornachandran, "Deepanti-phishnet: Applying deep neural networks for phishing email detection cen-aisecurityiwspa-2018," in *Proc. CEUR Workshop Process.*, vol. 2124, 2018, pp. 39–49.
[41] N. Moradpoor, B. Clavie, and B. Buchanan, "Employing machine learning techniques for detection and classification of phishing emails," in *Proc. Comput. Conf.*, Jul. 2017, pp. 149–156.
[42] T. Gangavarapu and C. D. Jaidhar, "A novel bio-inspired hybrid meta-heuristic for unsolicited bulk email detection," in *Computational Science*, V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, J. Teixeira, Eds. Cham, Switzerland: Springer, 2020, pp. 240–254.
[43] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020, doi: 10.1007/s10462-020-09814-9.
[44] T. Chin, K. Xiong, and C. Hu, "Phishlimiter: A phishing detection and mitigation approach using software-defined networking," *IEEE Access*, vol. 6, pp. 42516–42531, 2018, doi: 10.1109/ACCESS.2018.2837889.
[45] L. Halgaš, I. Agrafiotis, J. R. C. Nurse, "Catching the Phish: Detecting phishing attacks using recurrent neural networks (RNNs)," in *Information Security Applications*, I. You, Ed. Cham, Switzerland: Springer, 2020, pp. 219–233.
[46] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *IEEE Access*, vol. 7, pp. 56329–56340, 2019, doi: 10.1109/ACCESS.2019.2913705.
[47] E. S. Gualberto, R. T. De Sousa, T. P. De B. Vieira, J. P. C. L. Da Costa, and C. G. Duque, "From feature engineering and topics models to enhanced prediction rates in phishing detection," *IEEE Access*, vol. 8, pp. 76368–76385, 2020.
[48] J. Nazario. *Phishing Corpus*. Accessed: Nov. 3, 2020. [Online]. Available: https://monkey.org/~jose/phishing/
[49] *The Apache Spamassassin Public Corpus*. Accessed: Nov. 3, 2020. [Online]. Available: https://spamassassin.apache.org/old/publiccorpus
[50] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email classification research trends: Review and open issues," *IEEE Access*, vol. 5, pp. 9044–9064, 2017, doi: 10.1109/ACCESS.2017.2702187.
[51] T. Gangavarapu, C. Jaidhar, B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 4, pp. 1–63, Feb. 2020, doi: 10.1007/s10462-020-09814-9.
[52] S. Maldonado and G. L'Huillier, "SVM-based feature selection and classification for email filtering," in *Pattern Recognition Applications and Methods*, P. Latorre Carmona, J. S. Sánchez, A. L. Fred, Eds. Berlin, Germany: Springer, 2013, pp. 135–148.
[53] C. D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.
[54] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python natural language processing toolkit for many human languages," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2020, pp. 1–5. [Online]. Available: https://nlp.stanford.edu/pubs/qi2020stanza.pdf
[55] I. D. Dinov, *Data Science and Predictive Analytics—Biomedical and Health Applications using R*. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-72347-1.
[56] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instruct. Conf. Mach. Learn.*, 2003, pp. 1–4.
[57] D. Asir, S. Appavu, and E. Jebamalar, "Literature review on feature selection methods for high-dimensional data," *Int. J. Comput. Appl.*, vol. 136, no. 1, pp. 9–17, Feb. 2016.
[58] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.
[59] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.
[60] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, May 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968
[61] B. Ghojogh, M. N. Samad, S. A. Mashhadi, T. Kapoor, W. Ali, F. Karray, and M. Crowley, "Feature selection and feature extraction in pattern analysis: A literature review," *CoRR*, vol. abs/1905.02845, pp. 1–8, Apr. 2019.
[62] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2014.
[63] J. Costa, E. Freitas, A. Serrano, and R. Sousa Jánior, "Improved parallel approach to PCA based MaliciousActivity detection in distributed honeypot data," *Int. J. Forensice Comput. Sci.*, vol. 7, no. 2, pp. 8–20, Dec. 2012.
[64] T. P. B. Vieira, D. F. Tenório, J. P. C. L. da Costa, E. P. de Freitas, G. D. Galdo, and R. T. de Sousa Jánior, "Model order selection and eigen similarity based framework for detection and identification of network attacks," *J. Netw. Comput. Appl.*, vol. 90, pp. 26–41, Jul. 2017, doi: 10.1016/j.jnca.2017.04.012.
[65] J. Shlens, "A tutorial on principal component analysis," Dec. 2005.
[66] I. Jolliffe, *Principal Component Analysis*. Cham, Switzerland: Springer-Verlag, 2002.
[67] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychol. Rev.*, vol. 104, no. 2, pp. 211–240, 1997.
[68] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[69] T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, *Handbook of Latent Semantic Analysis*. New York, NY, USA: Taylor & Francis, 2007. [Online]. Available: https://books.google.com.br/books?id=Jm_NgzzDntYC
[70] C. D. Martin and M. A. Porter, "The extraordinary SVD," *Amer. Math. Monthly*, vol. 119, no. 10, pp. 838–851, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/tamm/tamm119.html#MartinP12
[71] C. D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.
[72] J. Kim and C. Mueller, *Factor Analysis: Statistical Methods and Practical Issues*. Beverly Hills, CA, USA: Sage, 1978.
[73] A. Rea and W. Rea, "How many components should be retained from a multivariate time series PCA?" 2016, *arXiv:1610.03588*. [Online]. Available: http://arxiv.org/abs/1610.03588
[74] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, Apr. 1998.
[75] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.
[76] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," Massachusetts Inst. Technol., Cambridge, MA, USA, AI Memo 1602, 1997.
[77] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[78] V. R. Kb, "Deepanti-phishnet: Applying deep neural networks for phishing email detection cen-aisecurity@iwspa-2018," in *Proc. IWSPA*, 2010, pp. 40–50. [Online]. Available: http://ceur-ws.org/Vol-2124/#paper_9
[79] S. Baki, A. Das, A. Elaassal, P. Goyal, D. Marchette, and L. F. T. D. Moraes, "Anti-phishing shared task pilot at the 4th ACM IWSPA," in *Proc. IWSPA-AP*, Aachen, Germany, 2018, pp. 1–5. [Online]. Available: http://ceur-ws.org/Vol-2124/



EDER SOUZA GUALBERTO received the bachelor's degree in licensing in computer science, the B.Sc. degree in computer science, and the M.Eng. degree in electrical engineering from the University of Brasília, in 2008, 2010, and 2011, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

He is also an Information Security Analyst in the Brazil public sector, and an Advisor for the Information Security Policy of National Agency of Telecommunications of Brazil (Anatel). His research interests include computer network and information security, machine learning, and natural language processing.

RAFAEL TIMOTEO DE SOUSA, JR. (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the Federal University of Paraíba (UFPB), Campina Grande, Brazil, in 1984, the master's degree in computing and information systems from the Ecole Supérieure d'Electricité–Supélec, Rennes, France, in 1985, and the Ph.D. degree in telecommunications and signal processing from the University of Rennes 1, Rennes, in 1988.

He was a Visiting Researcher with the Group for Security of Information Systems and Networks (SSIR), Ecole Supérieure d'Electricité–Supélec, from 2006 to 2007. He worked in the private sector from 1988 to 1996. Since 1996, he has been a Network Engineering Associate Professor with the Department of Electrical Engineering, University of Brasília, Brazil, where he is currently the Coordinator of the Professional Post-Graduate Program on Electrical Engineering (PPEE) and supervises the Decision Technologies Laboratory (LATITUDE). He is also the Chair of the IEEE VTS Centro-Norte Brasil Chapter (IEEE VTS Chapter of the Year 2019). His professional experience includes research projects with Dell Computers, HP, IBM, Cisco, and Siemens. He has coordinated research, development, and technology transfer projects with the Brazilian Ministries of Planning, Economy, and Justice, as well as with the Institutional Security Office of the Presidency of Brazil, the Administrative Council for Economic Defense, the General Attorney of the Union, and the Brazilian Union Public Defender. He received research grants from the Brazilian research and innovation agencies CNPq, CAPES, FINEP, RNP, and FAPDF. He has developed research in cyber, information and network security, distributed data services, and machine learning for intrusion and fraud detection, as well as signal processing, energy harvesting, and security at the physical layer.

THIAGO PEREIRA DE BRITO VIEIRA received the B.Sc. degree in business administration from the Federal University of Paraíba (UFPB), the B.Sc. degree in telematics from the Federal Institute of Paraíba (IFPB), the M.Sc. degree in computer science from the Federal University of Pernambuco (UFPE), and the Ph.D. degree in electrical engineering from the University of Brasília (UNB).

From 2007 to 2019, he was an IT Systems Analyst and Systems Architect with the National Agency of Telecommunications of Brazil (Anatel). He is currently a Data Scientist and an Advisor of information management with the National Agency of Telecommunications of Brazil (Anatel).

JOÃO PAULO CARVALHO LUSTOSA DA COSTA (Senior Member, IEEE) received the diploma degree in electronic engineering from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil, in 2003, the M.Sc. degree in telecommunications from the University of Brasília (UnB), Brazil, in 2006, and the Ph.D. degree in electrical and information engineering from the Ilmenau University of Technology (TU Ilmenau), Germany, in 2010.

From 2010 to 2019, he coordinated the Laboratory of Array Signal Processing (LASP) and several research projects. For instance, from 2014 to 2019, he coordinated the main project related to distance learning courses at the National School of Public Administration and a Special Visiting Researcher (PVE) project related to satellite communication and navigation with the German Aerospace Center (DLR) supported by the Brazilian Government. From March 2019 to July 2020, he was a Senior Development Engineer with the EFS on the area of autonomous driving. Moreover, from October 2019 to July 2020, he was a Lecturer with the Ingolstadt University of Applied Sciences on the area of autonomous vehicles. Since August 2020, he has been a Permanent Professor of applied electrical engineering with the Hamm-Lippstadt University of Applied Sciences, Germany. He has published more than 195 scientific publications and patents. His research interests include autonomous vehicles, beyond 5G, GNSS, and adaptive and array signal processing. He received seven best paper awards in international conferences.

CLÁUDIO GOTTSCHALG DUQUE received the bachelor's degree in licensing in modern languages (Portuguese and German) from the Faculty of Letters, Federal University of Minas Gerais, Belo Horizonte, Brazil, in 1994, the master's degree in psycholinguistics from the Graduate Program in Linguistic Studies, Faculty of Letters, Federal University of Minas Gerais, in 1998, the Sandwich Ph.D. degree in computer science from the Angewandte Sprachwissenschaft und Computerlinguist - Justus-Liebig-Universitat Giessen, Giessen, Germany, in 2004, and the Ph.D. degree in information production and management from the Graduate Program in Information Science, School of Information Science, Federal University of Minas Gerais, in 2005.

He is currently a Coordinator of the research group "Research Expert Group for Intelligent Information in Multimodal Environment using Natural Language Technologies and Ontologies" (R.E.G.I.I.M.E.N.T.O.). He is also a Permanent Member of the Graduate Program in Information Science (PGCINF-UNB), Brasília, Brazil. His research interests include information architecture, information retrieval, deep learning, blockchain, and natural language processing.