IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 19, 2020, accepted April 13, 2020, date of publication April 21, 2020, date of current version May 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989126

From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection

EDER S. GUALBERTO[1], RAFAEL T. DE SOUSA, JR.[1], (Senior Member, IEEE), THIAGO P. DE B. VIEIRA[1], JOÃO PAULO C. L. DA COSTA[1], (Senior Member, IEEE), AND CLÁUDIO G. DUQUE[2]

[1]Department of Electrical Engineering, University of Brasilia, Brasilia 70910-900, Brazil
[2]Faculty of Information Science, University of Brasilia, Brasilia 70910-900, Brazil

Corresponding author: Eder S. Gualberto (edergual@gmail.com)

This work was supported in part by the CNPq - Brazilian National Research Council, under Grant 312180/2019-5 PQ-2, Grant LargEWiN BRICS2017-591, and Grant (INCT in Cybersecurity) 465741/2014-2, in part by the CAPES - Brazilian Higher Education Personnel Improvement Coordination under Grant FORTE 23038.007604/2014-69 and Grant PROBRAL 88887.144009/2017-00, in part by the FAP-DF - Brazilian Federal District Research Support Foundation under Grant UIoT 0193.001366/2016 and Grant SSDDC 0193.001365/2016, in part by the Brazilian Ministry of the Economy under Grant DIPLA 005/2016 and Grant ENAP 083/2016, 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, in part by the General Attorney of the Union under Grant AGU 697.935/2019, and in part by the DPI/DPG/UnB–Decanates of Research and Innovation and Postgraduate Studies of the University of Brasilia.

ABSTRACT Phishing is a type of fraud attempt in which the attacker, usually by e-mail, pretends to be a trusted person or entity in order to obtain sensitive information from a target. Most recent phishing detection researches have focused on obtaining highly distinctive features from the metadata and text of these e-mails. The obtained attributes are then used to feed classification algorithms in order to determine whether they are phishing or legitimate messages. In this paper, it is proposed an approach based on machine learning to detect phishing e-mail attacks. The methods that compose this approach are performed through a feature engineering process based on natural language processing, lemmatization, topics modeling, improved learning techniques for resampling and cross-validation, and hyperparameters configuration. The first proposed method uses all the features obtained from the Document-Term Matrix (DTM) in the classification algorithms. The second one uses Latent Dirichlet Allocation (LDA) as an operation to deal with the problems of the ‘‘curse of dimensionality’’, the sparsity, and the text context portion included in the obtained representation. The proposed approach reached marks with an F1-measure of 99.95% success rate using the XGBoost algorithm. It outperforms state-of-the-art phishing detection researches 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.

INDEX TERMS Feature engineering, feature extraction, natural language processing, phishing detection, topics modeling, XGBoost.

I. INTRODUCTION

According to [1], in 2019, the total number of e-mails transacted every day exceeds half-trillion, and about 80% of this e-mail traffic is spam messages. Although some of these spam messages are just legitimate marketing e-mails, in this amount, there are also malicious e-mails through which sensitive information can be exposed or subtracted. A successful malicious e-mail can lead to critical incidents such as financial frauds and hacked or hijacked systems, accounts, or profiles. These malicious messages are denominated phishing e-mails.

In this type of fraud attempts, the attacker pretends to be a trusted person or entity, and through this false impersonation tries to obtain sensitive information from a target [2], [3]. A typical example is that one in which scammers try to pass through a known institution, claiming the need to update a register or an immediate action from the client-side, and for this personal and financial data are requested. A variety of

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

features such as fake web pages, malicious code installation, or form filling are employed along with the e-mail itself to perform this type of action [4].

There are many pieces of research aimed to develop applications that can correctly detect phishing cases [3]–[6]. Some of them focus on the phishing sites detection [7]–[10], whereas other part focus on the phishing e-mails detection [11]–[13]. This paper is concentrated on phishing e-mail detection. Most of these studies utilize natural language processing (NLP) techniques combined with machine learning techniques to perform such detection activities based on classification tasks. Information such as body and header text of e-mail messages, URLs, and tags are processed and used as input data for the sorting algorithms to be employed [14].

Since phishing aims to appear being a legitimate message, detection techniques based only on filtering rules, such as blacklisting and heuristic, has limited effectiveness [6], in addition to being potentially forged. As alternatives, through data-driven techniques [15], features can be extracted from the e-mail body and the header texts using techniques that explain the similarity and significance relations between the words present in a specific e-mail, as well as in the whole set of messages samples. The most common approach for this type of feature engineering is based on Vector Space Models (VSM) [16], also called Distributional Representations. In this type of representation, each message is represented using numerical values for each of its terms (words, for instance) as symbols in a vector space.

Three main problems are discussed about the representation proposed by this model: the denominated ''Curse of Dimensionality'' [17], the sparsity [18] and the context portion represented together in the VSM [19]. The first one refers to the high number of dimensions in which a text is represented by its words ranking. While the second one refers to the fact that as the data dimensionality increases, the data sparsity also increases. These two problems associated raise the computation to process and store big corpus, besides potentially causing overfitting due to some features being rarely observed. Finally, the third problem points to the few context properties of the text that are incorporated in the VSM representation type.

In order to overcome the three mentioned main problems, statistical measures, feature extraction, and distributed models are three commonly used techniques in the literature.

Statistical measures can be used to select fewer features, a subset of original features, that are supposed to be more representative than the other [20], which could solve the first two problems of the VSM representation, its high dimensionality, and sparsity.

It is also possible to use feature extraction techniques [21]–[23], which, starting from the initial high-dimensional matrix, obtain more discriminative features from the original features extracted from the text in the analysis. This option is an answer to the three listed problems of the VSM representation, including some degree of information from the textual context from which the features are extracted.

Instead of selecting more representative features, or performing mathematical transformations or probabilistic calculations on the VSM representation to extract more distinctive features, representing such texts in a fixed shared low dimensional space, also called distributed models [13], [24]–[26], is also an approach. In this paradigm, a vector and its pre-fixed dimensions represent a word and its contextual information (such as relations with other words, and its semantic and syntactic similarities). In this sense, this option also addresses the three problems.

The contribution of this paper focuses on proposing methods through the use of combined techniques to obtain more representative features attributes from the body texts of the e-mails. These features are obtained by feature extraction of the distributional representation. Then, they are submitted to machine learning classification algorithms using improved learning techniques.

Our strategy is concentrated in a holistic procedure based on lemmatization, bag of words, latent dirichlet allocation, and powerful classification algorithms. After passing the e-mails to an array structure, a preprocessing step over the e-mail texts is executed. Next, it is conducted a lemmatization using the WordNet lexical database as a dictionary to obtain a semantic-based reduction. Then, it is extracted a document-term matrix, from the BoW model, that is used in two different fronts: directly as the classification algorithms features attributes (Method 1), and as the input for Latent Dirichlet Allocation, whose obtained topics are used to express new reduced features in function of their proportions in each message (Method 2). Finally, it feed the same algorithms with these two different sets of features, concluding each method.

The objective of this work is to propose a feature engineering process for phishing detection with approaches that increase the precision and accuracy of these algorithms predictions. Our proposal achieved measures with a 99.95% success rate using the XGBoost algorithm. It is, to the best of our knowledge, the highest result in phishing detection researches for an accredited data set basing only on the body of the e-mails, not taking into account other e-mail features such as its header, IP information or number of links on the e-mail body.

The proposed approach presents itself not only as an answer to the listed problems, but also demonstrates an optimal representation capacity, since it uses a number of new extracted features less than 0.02% of the original amount of features, but still with better performance measures than those with any dimensionality reduction action. It shows that these features provide an enhanced distinction of messages from the selected datasets (as phishing or legitimate e-mails). These well-known collections of data: Phishing Corpus dataset ([27] and SpamAssassin dataset [28]), are the most frequently used for this type of research, which allows a comparison with other related works results and ratifies the obtained marks.

The remainder of this work is segmented into five sections. In Section II, it is described the related works based on natural language processing and machine learning techniques for phishing detection, the baseline study. The data model is presented in Section III, where the chosen datasets and the modeling data are addressed. The architecture of the proposed approach is shown in Section IV, as well as the adopted methodology. In Section V, the proposed methods are evaluated using real data and the baseline study, whereas, in Section VI, conclusions are drawn and future works are summarized.

II. RELATED WORKS

Approaches to detect phishing e-mails based on machine-learning techniques require features that may be useful in distinguishing between a phishing e-mail and a legitimate e-mail. The feature engineering from the e-mail text is an area that has been highly focused and which has received significant attention from the researchers [4], [3], [6], and [29]. The key is to obtain highly distinctive features present in the e-mails, arising from its structure or its content.

An approach for phishing e-mails detection based on e-mail properties and on external sources, such as if these e-mails contain IP-based URLs, the age of linked-to domain names, or the number of links in the e-mail, is presented in [30]. This research submits these features to the Random Forest classification algorithm, wherewith it correctly identify over 96% of the phishing e-mails. Similarly, [31] proposes a hybrid feature selection approach based on the combination of content-based and behavior-based, which reached a 94% accuracy rate with the Bayes Net Algorithm. The dataset of these works is obtained from the Phishing-Corpus [27] and from the Spamassassin PublicCorpus [28], which are the same datasets adopted to evaluate the proposed approach.

More focused on data mining, a data-driven approach is proposed in [22], where the features obtained are a set of structural features extracted directly from the text of e-mails with features derived from keyword extraction, along with those derived from the use of Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) techniques. They used these features in 3 classification algorithms: Support Vector Machines (SVM), Naive Bayes and Logistic Regression, obtaining their best performance measure with SVM (an F1 Score of 99.58% for a set of 1017 features). This cited work used the PhishingCorpus [27] and the Spamassassin PublicCorpus [28].

Still, regarding word count-based phishing detection works, [20] is an approach based on Term Frequency-Inverse Document Frequency (TF-IDF) and domain features, and in [21] is presented an approach that besides TF-IDF, still employed Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF). They respectively achieved F1-measure of 98% (using Logistic Regression classification algorithm) and 94.6% (using k-Nearest Neighbor classification algorithm). Two datasets were used, the biggest one with 5,700 in its training set and 4,300 in its test set [32].

In [33], PhishNet-NLP is a scheme based not only on header information, body, and links in an e-mail but also on natural language processing techniques that identify whether an e-mail is actionable or informative. From all these input data, it determines whether the e-mail is a phishing or a ham(legitimate) mail.

Also, based on natural language processing, a multi-layered methodology proposed in [23] uses three layers to detect phishing e-mails. To do this, it employs Probabilistic Latent Semantic Analysis (PLSA) to build a topic model, the AdaBoost algorithm to build a robust classifier, and Co-training to handle with labeled and unlabeled examples. The output of each of these techniques is the input for the next. This approach reached an F1-measure of 98.3% for 25 and 10 topics used as features into the classification task.

Another methodology, based on LDA and Conditional Randon Field (CRF) [34], builds its features through the hidden topics extracted by LDA and the names of entities/organizations that attackers impersonate during phishing attacks discovered by CRF, and feed into the AdaBoost algorithm with them. Based on knowledge discovery, [14] proposes a feature selection approach based on information gain measure. They obtained a 99.1% accuracy rate.

In [35], is presented a comparative approach between feature selection (Chi-Square and Information Gain measures) and feature extraction (LSA and Principal Component Analysis - PCA) techniques for dimensionality reduction in phishing e-mail detection. This approach had an accuracy rate near to 98%. These last four approaches ([23], [33], [34] and [35]) used the PhishingCorpus [27] and the Spamassassin PublicCorpus [28].

Based on a distributed method, the approach present in [13] uses Word Embedding and Convolutional Neural Networks to build a model to distinguish phishing e-mails from legitimate e-mails. Through this proposed architecture, it reached an accuracy rate of 96.8%.

Also, using word embedding, through Doc2Vec, the approach presented in [24] compares the TF-IDF matrix and Doc2Vec formed for legitimate and phishing e-mails, feed them into various traditional machine learning classifiers for classification. They observed that machine learning classifiers with Doc2Vec representation had performed well in comparison to the TF-IDF representation, reaching an 88.4% accuracy rate.

Through FastText, Barathi Ganesh *et al.* [25], [26] present word embedding approaches that assess the performance of distributed representation in the detection of phishing e-mails as a text classification problem. They observed that word embedding and neural bag-of-ngrams facilitates to extract syntactic and semantic similarity of e-mails, obtaining a

99% F1 Score in both research experiments. These last three researches used the dataset proposed in [32].

In [36], is used artificial neural networks in an approach based on deep packet inspection (DPI) and software-defined networking (SDN) to detect and mitigate phishing e-mails. They achieved an accuracy rate of 98.39%. By the use of convolutional neural networks (RCNN) and Word2Vec techniques, [37] reached an accuracy measure of 99.84% and a false positive rate (FPR) of 0.043%, using the headers and bodies of the e-mails.

Although natural language processing and machine learning have been largely utilized in phishing detection, methods that, together with these techniques, be also based on semantic and similarity enrichment and established training techniques for machine learning algorithms, have not been appropriately addressed into this context.

Approaches that result in more distinctive features for phishing detection and in better prediction rates for this problem successively are suitable for this scenario. Therefore, this work concerns to assess the performance of features obtained from some robust representation perspectives. Our interest goes beyond showing an optimal accuracy (or other isolated metrics) for the models, but rather exposing the obtained results in the various utility measures (that are complementary [38]) in order to demonstrate the overall performance of the proposed approach, not just its capabilities in one of the classes of this classification problem.

III. DATA MODEL

The notations used in this paper are defined as follows: vectors are denoted by lowercase boldface letters (for instance: **a**, **b** and **c**), and matrices are described by uppercase boldface letters (such as **A**, **B** and **C**). The matrix elements are denoted by this shape: $\mathrm{a}_{i \times j}$, i.e., the element of the matrix **A** located at line i column j).

In this section, the chosen datasets and this paper modeling data are presented. The datasets and its details are described in the Subsection III-A, and the modeling data are introduced in Subsection III-B.

A. DATASETS

The datasets used in this work were obtained from two collections of e-mails (both publicly available sources): PhishingCorpus from [27] as the Phishing Dataset and the SpamAssassin PublicCorpus from [28] as the Ham Dataset, that according to [29] are the most widely used datasets in phishing e-mail classification researches.

Some authors that employed them are [14], [22], [23], [30], [31], [33]–[35], [39], and [40]).

From the phishing dataset, all the phishing e-mails in phishing3.mbox are used, obtaining 2,279 phishing e-mails. From the ham dataset, are obtained 4,150 ham e-mails because the remaining of its 6.047 messages are spam. It totals 6,429 e-mails for the experiments proposed in this paper.

B. MODELING DATA

Along this paper, the rows of a matrix denote its instances (the e-mails), and its columns refer to their respective attributes (some data that describes or represents it in some dimension). As described in Subsection III-A, it start with 6,429 e-mails, and during the tasks of the proposed approach, they are parsed and transformed through vector and matrix structures. In Section IV, the proposed approach steps are discussed.

IV. PROPOSED APPROACH FOR PHISHING DETECTION

In this section, it is presented the details of the proposed approach, its associated experimental scenario and its methodology. In Subsection IV-A, there is an overview of the proposal. In Subsections IV-B and IV-C, the parsing and the pre-processing steps are described. In Subsection IV-D, it is presented the feature extraction process through the bag of words model and the Document-Term Matrix, along with a description of Method 1 architecture. In Subsection IV-E, the feature extraction process through the Latent Dirichlet Allocation as a dimensionality reduction procedure is portrayed, as well as the Method 2 architecture is explained, and in Subsection IV-F, we detail the classification plan of action and introduce the employed classification algorithms.

A. PROPOSAL OVERVIEW

The contribution of this work concentrates on generating more expressive features from the existing terms/words in e-mails (documents), and subjecting them to different machine learning algorithms, using enhanced techniques, in order to obtain improved results in classification tasks.

The main architecture of our proposal is presented in Fig. 1. As shown in this figure, the e-mails of the chosen datasets go through a parsing process, in which the body text of all the e-mails is extracted (from which all the necessary features are obtained), keeping their associated labels indicating to each of them whether it is a phishing mail or a ham mail.

The e-mail bodies are submitted to pre-processing in order to: eliminate words/terms that do not add much to the semantics of the documents, strengthen and enrich relationships of synonymy and polysemy, and assign a higher weight to words/terms that better disclose classes of documents among others.

In the pre-processing task, the texts are transformed into lowercase, and the punctuation marks, special characters, possible accents, and stopwords[1] are removed. The terms obtained are then exchanged for their common base form, from the reduction of their respective inflectional forms and derivationally related forms (lemmatization process). This process also allows a potential moderate feature dimensionality reduction by the terms semantics and their synonyms.

Based on the term/word count present in e-mails texts after the pre-processing step, a matrix (Document-Term

[1] 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 that represents all the stopwords, most cases take prepositions and articles as such.

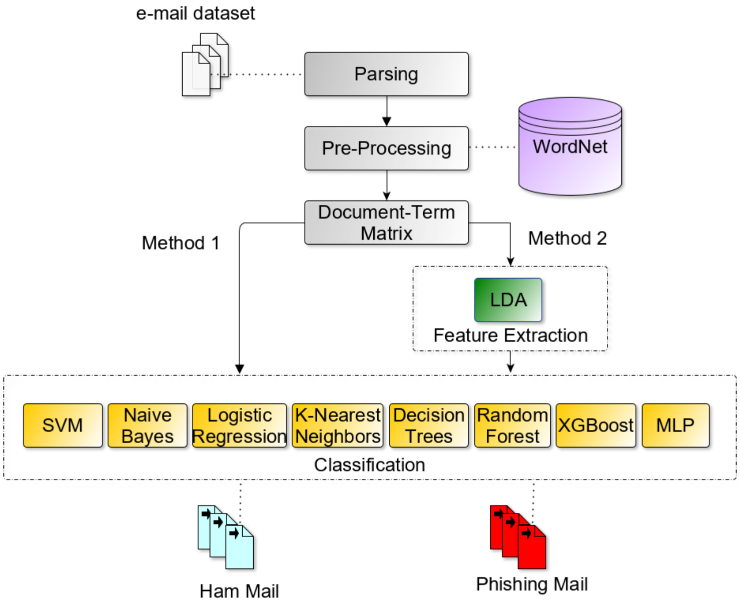

FIGURE 1. The main architecture of the proposed approach.

Matrix - DTM), that relates these terms to the pre-processed e-mails texts, is obtained. The word representations methods that use the DTM are based on distributional semantics. In which the semantic similarities between terms are quantified based on their distributional properties in large text corpora, with discrete symbols representing the terms and its interactions [16].

From the DTM, two methods are followed. Using all the terms obtained arranged in the matrix as features (Method 1), that is, without acting directly on the high dimensionality and sparsity problems. Alternatively, extracting new features from this Matrix by Latent Dirichlet Allocation (Method 2). This second perspective addresses the problem of sparse and high-dimensionality features matrices (extracting more instructive and discriminative features from the pre-processed texts, allowing a low-rank representation of the data), and it includes some degree of information from the e-mail bodies textual context.

The LDA was chosen to Method 2 objective due to its characteristics such as [41]: it is a generative probabilistic model designed for text corpora, that provides a compressed explicit representation of a document (as a finite mixture over an underlying set of topics), and it generalizes easily to new documents.

These two methods to represent the dataset are then submitted to classification algorithms, that after a learning process with enhanced techniques, could predict if an e-mail is phishing or a ham mail, with excellent results.

The choices of which machine learning algorithms to use, as well as the dataset, were based on previous works related to phishing detection for comparison purposes, and also on the algorithms performance power, in order to obtain the best possible results. In this way, it is possible to measure not only the effectiveness of the techniques used in this proposed approach, concerning the results of previous works, but also to introduce more robust algorithms in this research area in order to obtain state-of-the-art marks.

During the classifications tasks, many experiments are performed to find the best configuration of the hyperparameters of each learning algorithm, as well as to implement proposed strategies for dividing and folding the training/validation dataset, before to perform the obtained models on the test set.

Each of these steps is described in detail in the following subsections.

B. PARSING

The e-mails of both datasets go through a parsing process, in which the texts of their respective bodies are extracted. Through this process, these texts are arranged in a vector structure. Thus, at this step, we pass from 6,429 e-mails to a vector with 6,429 rows, represented by:

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_{6,429} \end{bmatrix}. \tag{1}$$

Another vector, in turn, indicates whether these e-mails type, where each of these texts is labeled as phishing and legitimate messages correspondingly. This second vector is represented by:

$$\boldsymbol{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_{6,429} \end{bmatrix}. \tag{2}$$

The concatenation of (1) and (2) is represented by a matrix with 6,429 rows and 2 columns, given by:

$$\boldsymbol{A} = \begin{bmatrix} a_{1\times 1} & a_{1\times 2} \\ a_{2\times 1} & a_{2\times 2} \\ a_{3\times 1} & a_{3\times 2} \\ \vdots & \vdots \\ a_{6,429\times 1} & a_{6,429\times 2} \end{bmatrix}. \tag{3}$$

This number of lines quantity is the same amount of the proposed e-mails (6,429), and the extracted texts represent each of them. The first column refers to these texts (vector e), whereas the second one (vector l) refers to the label of each e-mail (a phishing mail or a ham mail).

C. PRE-PROCESSING

The pre-processing steps are presented in Fig. 2. As can be seen, after being arranged in an array structure, the texts of the e-mails bodies are transformed into lowercase and have the punctuation marks, special characters, and any accents removed. After that, this structure pass for a tokenization process, in which the terms/words of the texts are separated by using white spaces (space, tab, and newline) as the delimiters. Then, the stopwords are removed. Finally, the terms go through a lemmatization process, enhance by the WordNet-based processing. The details of the WordNet-based processing and lemmatization process are explained in Subsection IV-C.1.

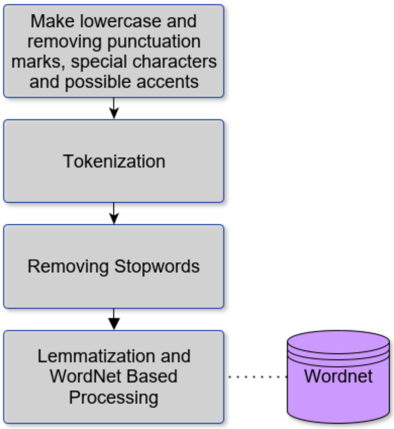
Make lowercase and removing punctuation marks, special characters and possible accents
Tokenization
Removing Stopwords
Lemmatization and WordNet Based Processing
Wordnet

FIGURE 2. **The pre-processing steps.**

During the pre-processing steps, our data keeps its shape with 6,429 rows and 2 columns, but each e-mail representation undergoes the proposed transformations. The output of this step is expressed by:

$$\boldsymbol{B} = \begin{bmatrix} b_{1\times 1} & b_{1\times 2} \\ b_{2\times 1} & b_{2\times 2} \\ b_{3\times 1} & b_{3\times 2} \\ \vdots & \vdots \\ b_{6,429\times 1} & b_{6,429\times 2} \end{bmatrix}. \tag{4}$$

1) LEMMATIZATION AND WORDNET BASED PROCESSING

Lemmatization is a process that, such as stemming, aims to convert a word to its common base form, by reducing its inflectional forms and sometimes its derivationally related forms. Although they share the same objective, these processes differ in the extent that stemming just cut the beginning, or the end of a word (based on a list of prefixes and suffixes that are usually found in inflected words), whereas lemmatization does it based on morphological analysis of the words, thereby providing for each of these words a meaningful base form [42]. For this task, lemmatization needs a dictionary, that is used to find a lemma[2] that best represents the words. In the approach proposed in this work, the lemmatization process is performed with the WordNet database.

WordNet refers to an extensive lexical database of English Language [43]. By its synsets (sets of cognitive synonyms), it tries to express the meaning of a concept. These sets include words from nouns, verbs, adjectives, and adverbs grammatical classes. The interlink among the synsets provides a network of meaningfully related words and concepts that can be used for better results in natural language processing. Some researches use it as an ontology[3].

In this work, WordNet is used to provide a semantic-based reduction. That is, by the use of the synsets and lemmatization process, it is obtained some feature size reducing by replacing groups of terms that have common synonyms with their lemma (that is, reducing its dimensionality). By this semantic-based reduction, the number of potential features decreased from 63,448 to 59,623.

This semantic-based reduction with WordNet also can be made with a part-of-speech (pos) tagging process, which indicates the grammatical class of each token (in WordNet case, is possible tag a term as being a noun, a verb, an adjective or an adverb). With this option, the proposed semantic-based reduction decreases the number of potential features from 63,448 to 56,523. It is the number of features used in all the Methods in this paper before the implementation of dimensionality reduction techniques.

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

Bag of Words (BoW) refers to a model in which a text (an e-mail body for example) is represented as a list of words

[2] Lemma is a word or expression, a particular form, that is chosen to represent a lexeme (a unit of meaning, also named dictionary form).

[3] According to [44]; an ontology can be understood as an engineering artifact based on specific formal vocabulary, whose use allows the description/representation of a knowledge domain.

The set of concepts are representations of real-world objects, which have properties with other objects (relational properties or relationships) and descriptive properties (attributes with values of certain data types) that describe them (representing states, events, or processes of these entities). Instances of a concept represent a particular object and its description (attributes and relationships) within a set of objects of the same type. The set of instances associated with the ontology constitutes a knowledge base relative to that domain.

(or another n-gram[4]) and how many times each of them occurs in the text under the feature extraction process, with no contextual information such as grammatical class and order of occurrence of those words. In our approach, from this model, it is obtained the Document-Term Matrix that represents each text or document in a row, and each term in a column. Its elements are the ranking of each document and term. This ranking usually is represented by its occurrence count or the term frequency-inverse document frequency (TF-IDF)[5]

calculus over it [42]. It is also a type of feature extraction.

In the proposed approach, according to Fig. 1, the ranking based on occurrence count is used by itself directly in the classification step (Method 1) and with a feature extraction by LDA (Method 2).

The approach where the Document-Term Matrix with the ranking based on text occurrence count is used directly in the classification step constitutes the first method proposed in this paper.

After all actions proposed by the pre-processing, the bag of words model and the DTM steps, the representation of our data is now a matrix with 6,429 rows and 56,524 columns, expressed by:

$$\boldsymbol{C} = \begin{bmatrix} c_{1\times 1} & c_{1\times 2} & c_{1\times 3} & \cdots & c_{1\times 56,524} \\ c_{2\times 1} & c_{2\times 2} & c_{2\times 3} & \cdots & c_{2\times 56,524} \\ c_{3\times 1} & c_{3\times 2} & c_{3\times 3} & \cdots & c_{3\times 56,524} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{6,429\times 1} & c_{6,429\times 2} & c_{6,429\times 3} & \cdots & c_{6,429\times 56,524} \end{bmatrix}. \tag{6}$$

Its rows represent each e-mail from our dataset, and the columns represent our 56,523 features (terms extracted from the text) and the labels (phishing or legitimate) of these e-mails.

The whole discussion of this paper until now has been dedicated to exposing the actions/implementations to prepare and promote robust data to be submitted to the methods that will obtain features (the best we can bring out) for the classification activity. The perspective of Method 1 directly submits the DTM (with its ranking based in terms of occurrence count) to the classification step. While, in the Method 2, its dimensionality reduction through LDA can potentially provides benefits such as reducing computational complexity, processing time and variance, as well as preventing overfitting, gaining a better understanding of the process that underlies the data and even allowing better visual analysis of such data [42], [46]. In the next subsections, this second approach will be described in detail, in order to elucidate its respective paradigm and to allow a clear understanding of the obtained results.

E. DIMENSIONALITY REDUCTION THROUGH LATENT DIRICHLET ALLOCATION (LDA)

The feature extraction methods obtain new features from the original features set, generally features of lower dimensionality. Some transformations do it over the original feature space, that is, the new feature space dimensions are combinations of the original high dimensional data. These new features intended to be more representative, concentrating relevant information from the underlying data in a non-redundant shape.

For the purposes of this work, the Latent Dirichlet Allocation is used, whose input data is the DTM based on occurrence count (**C**). This perspective is denominated Method 2.

Latent Dirichlet Allocation is a generative probabilistic model, from the topic model class, in which the documents may be represented as random mixtures of topics, and each topic may be modeled as a distribution over words/terms present on the dataset (vocabulary) [41]. It means that the latent topics probabilities provide an explicit representation of all collection of documents (in our case, all the e-mails of the dataset). From the dataset, the probability distributions are estimated, and from that, the latent topics are inferred. Then, the topics extracted may be used as input features, once each e-mail can be represented as a vector that indicates the probability distribution of this document over the selected topics.

The basic idea is that: to write a text, there are some pre-defined topics to use on the texts set. Their distributions obey the Dirichlet distribution. It is assumed that during the process of drafting the text, its generation process, the author exchange by several of these topics, using the words belonging to each of them. That is, the words from different topics are allocated by the result of the Dirichlet distribution sample result, and, by this process, the document is populated. Important to note that documents may have the same topics but still be different because these documents contain different proportions of these topics.

According to [47], to elaborate a document, first choose a distribution, after this, for each position of the document terms, choose a topic assignment, and finally choose the word from the corresponding topic.

Thus, considering the generative process explained above and making α as the priori Dirichlet probability distribution parameter, related to term-document distribution, β as the priori Dirichlet probability distribution parameter, related to topic-term distribution, z as the topic distribution associated to the terms in the documents, w as all the terms of the vocabulary, ϕ as the topic distribution over all the terms of

[4] N-gram refers to a continuous sequence of n items of a sample, such as characters, syllables, or words.

[5] Statistical measure assigns weights to the importance of a term for a document (or for a text sample, such as an e-mail body), which is inserted in a collection of documents (or in a corpus) [45]. Given by:

$$w_{i,j} = tf_{i,j} \cdot \log\left(\frac{N}{df_i}\right) \tag{5}$$

where: $\mathrm{tf}_{\mathrm{i,j}}$ is the number of occurrences of the term i in document j, N is the total number of documents, and df_{i} is the number of documents containing i.

vocabulary, θ as the topic distribution over the documents, the probability distribution of all the hidden and observed variables is given by the equation 7.

$$p(z, w, \phi, \theta|\alpha, \beta) = \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{v=1}^{V} p(z_{m,v}|\theta_m) p(w_{m,v}|z_{m,v}, \phi_{m,k}) \tag{7}$$

In the equation 7, K is the number of topics, ϕ_k is a vector with the vocabulary terms proportion for the topic k, M is the number of documents, θ_m is a vector with the topic proportion for the document $\mathrm{d_m}$, V is the number of words in the vocabulary, $\mathrm{z_{m,v}}$ is the topic distribution associated to term $\mathrm{w_{m,v}}$ in the document $\mathrm{d_m}$, $\mathrm{w_{m,v}}$ is the term $\mathrm{w_v}$ in the document $\mathrm{d_m}$. Where K varies from 1 to K, m varies from 1 to M and V varies from 1 to V.

Given the words observed for the proposed vocabulary, w, and using the Bayes' theorem, the hidden structure, that is, the assignments of topics for documents, document distributions by topics, and topics distributions by terms, can be obtained by the posterior distribution of the latent variables, given the words observed. This relation is expressed in equation 8.

$$p(z, \phi, \theta|w, \alpha, \beta) = \frac{p(z, w, \phi, \theta|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{8}$$

This equation is intractable to compute, due to its denominator [41], [47]. It is the marginal probability of the observations and can be expressed as equation 9.

$$p(w|\alpha, \beta) = \int_\phi \int_\theta p(w|\alpha, \beta) \tag{9}$$

Equation 9 is computationally intractable because summing the joint distribution over all the terms found in the collection vocabulary is exponentially large. In this sense, the LDA algorithms provide an approximate inference to this posterior distribution, disclosing its related topics ϕ, topic proportions θ, and topics assignments z, that is, the documents latent structure. The number of desired topics is also necessary. In our approach, the topic quantity was set as 100, 95, 35, 10, 5, and 3.

This technique is used in this work as follows: after passing it through the pre-processing steps explained above, generating the initial matrix (Document-Term Matrix) that relates terms to the documents in analysis (the e-mails bodies) through the BoW representation based on word unigrams, choose the number of topics that we want to work on (based in the perplexity and coherence measures, explained in SubsectionIV-E.1) and perform the LDA process. In this way, from the extracted topics, the e-mails can be represented as portions of topics' probability distribution on a latent, low-dimensional Topic space-based.

Here, our data present a new shape, a low-dimensional one. It has the same rows quantity (6,429), and options of 100, 95, 35, 10, 5, and 3 columns (plus one that refers to the label of each e-mail). For instance, our best setting for Method 2 is with ten topics, and its matrix has 6,429 rows and 11 columns. This matrix is represented by:

$$\boldsymbol{D} = \begin{bmatrix} d_{1\times1} & d_{1\times2} & d_{1\times3} & \cdots & d_{1\times11} \\ d_{2\times1} & d_{2\times2} & d_{2\times3} & \cdots & d_{2\times11} \\ d_{3\times1} & d_{3\times2} & d_{3\times3} & \cdots & d_{3\times11} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{6,429\times1} & d_{6,429\times2} & d_{6,429\times3} & \cdots & d_{6,429\times11} \end{bmatrix}. \tag{10}$$

1) PERPLEXITY AND COHERENCE MEASURES

In Method 2, the feature quantity choices are particularly noteworthy. For LDA, setting the number of topics to work on can be based on the perplexity [41] or on the coherence measures [48].

Perplexity refers to a metric that gives the average uncertainty provided by the model to each word in the dataset [49] [50]. In general terms, the idea is that the lower the model's perplexity score, the better its generalization performance. The equation 11 gives the perplexity score.

$$Perplexity = exp\left(-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right) \tag{11}$$

In the equation 11, p $(\mathrm{w_d})$ is the likelihood denoted by the equation 9 to our corpus of e-mails D, and $\mathrm{N_d}$ is the total number of keywords in d-th document of D.

According to [51], perplexity is not aligned with human interpretability. This research showed that these perspectives of the topic models are not correlated. In this context, to obtaining a measure closer to human judgment, the topic coherence measures are discussed. These measures offer a score that helps to assess how much the obtained topics are semantically interpretable, while perplexity is a score that assesses the topics as artifacts of statistical inference. Two measures of coherence are adopted: $\mathrm{C_{UCI}}$ and $\mathrm{C_{UMass}}$. While the first one refers to a measure that compares all words, through all possible combinations of pairs, an extrinsic measure, the second one refers to an intrinsic measure, a measure that compares a word not with all the other words, but with its preceding and succeeding words [48].

$\mathrm{C_{UCI}}$ coherence measure is calculated over all the word pairs of the given top words. It is a measure based on a sliding window, and the Pointwise Mutual Information[6] (PMI) [52].

[6] Pointwise Mutual Information is a utility measure to assess the associativity between two words. The equation 12 gives PMI

$$PMI(w_i, w_j) = \log\left(\frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}\right) \tag{12}$$

where P($\mathrm{w_i}$, $\mathrm{w_j}$ is the probability of the words $\mathrm{w_i}$ and $\mathrm{w_j}$ occur in the same word window, P($\mathrm{w_i}$ and P($\mathrm{w_j}$ are, respectively, the probabilities of $\mathrm{w_i}$ and $\mathrm{w_j}$ occur individually, and ϵ is and added value to avoid logarithm of zero.

The C_{UCI} score is given by the equation 13.

$$C_{UCI} = \frac{2}{N \cdot (N-1)} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j) \tag{13}$$

In equation 13, N refers to the N-top words of a topic, and the remaining terms, as indicated in equation 12.

C_{UMass} coherence measure is based on document co-occurrence counts, which is the document frequencies of the original documents from which the topics are learned. It is an asymmetrical measure based on segmentation and logarithmic conditional probability [53]. The C_{UMass} score is given by the equation 14.

$$C_{UMass} = \frac{2}{N \cdot (N-1)} \cdot \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \left(\frac{P(w_i, w_j) + \epsilon}{P(w_j)} \right) \tag{14}$$

F. CLASSIFICATION

Classification is a supervised learning[7] activity whose objective is to obtain a discriminating function that separates the samples into different classes. For the purposes of this paper, there are two classes: Phishing Mail and Ham Mail (the term used for legitimate e-mails).

As explained previously, some techniques are used to obtain the features from the e-mails of the datasets used in this work. Thus, we have tested: a method that obtains these features without reduction of dimensionality (Method 1), from the BoW model (the most traditional and straightforward of them, expressed by the Document-Term Matrix with term occurrence count ranking) and a method that reduced dimensionality through feature extraction (Method 2): LDA over DTM with term occurrence count ranking, and represent the instances features in terms of the Dirichlet distributions of the topics and the words. Thus, all the classification algorithms used in this work are trained in all these perspectives and their variations.

In order to estimate the hyper-parameters of each classification algorithm, the course of action proposed in [54] is adopted. It suggests dividing the training/validation set (which corresponds to 70% of the entire dataset) into two subsets (folds), each with 50% of the samples. They are used as the training and validation sets, respectively, and then the inverse. This process is repeated five times (ten runs in total) for each combination of the various parameters of the running classification algorithm, using the cross-validation technique. At the end of each run, a new random sampling of 2 folds is performed on the samples, with the restriction of maintaining the proportion of the classes observed in the total training/validation set in each of the two subsets, that is, a proper stratification.

After obtaining adequate hyper-parameters values for the phishing classification problem proposed in this work, the models are tested in the test set (that corresponds to the remaining 30% of the entire dataset).

The training set consists of 4,500 e-mails (70% of 6,429, as explained in subsection III-A) e-mails represented by their body text, 2,916 ham mails, and 1,584 phishing mails. While the test set consists of 1,929 e-mails (30% of 6,429), 1,251 ham e-mails, and 678 phishing e-mails, also called support.

These sets, for Method 1 and Method 2, are represented in Fig. 3, where it is observed that the Method 2 representation approach has a much smaller number of columns than the perspective of Method 1.

For the classification activities, as well as for comparison purposes, eight classification algorithms are used, namely: Support Vector Machines (SVM), Naive Bayes Classifier, Logistic Regression for classification, k-Nearest Neighbor, Decision Trees, Random Forest, Extreme Gradient Boosting (XGBoost) and Multilayer Perceptron (MLP). The main characteristics of the cited classification algorithms are exposed in the subsections below.

1) SUPPORT VECTOR MACHINES (SVM)

Support Vector Machines (SVM) refers to a supervised learning algorithm, in which the objective is to find a hyperplane in the input variable space to best separate the data points in two classes. This choice is based on that hyperplane that has the most significant margin, which is that hyperplane that presents the maximum distance between data points of both classes. By doing this, new data points can be sorted with more accuracy and precision.

Those points that are closer to the hyperplane are named Support Vectors. They influence the position and orientation of the hyperplane, as well as the number of features influence the dimension of the hyperplane [46].

2) NAIVE BAYES CLASSIFIER

This kind of classifier assumes that features are independent of each other on applying Bayes' Theorem (conditional probability). The expression naive comes from the fact that it assumes that all the features independently contribute to the probability of the given class, which is a strong assumption and unrealistic for real data.

Mathematically, what this algorithm does is assume that the off-diagonals values of the covariance matrix to be 0, i.e., they are independent. Then the joint distribution is the product of individual univariate densities (assuming that they have Gaussian distribution) [55].

3) LOGISTIC REGRESSION

This classifier, that is similar to linear regression to classification tasks, is based on find the values for the coefficients $(\mathrm{B}_0, \mathrm{B}_1, \mathrm{B}_2, \cdots, \mathrm{B}_n)$ that weight each feature $(\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \cdots \mathrm{X}_{\mathrm{n}})$, after that, it does its predictions transforming the output through the logistic function [46]. Thus, the probability of an e-mail being considered a phishing e-mail (class 1) or a

[7] In this type of learning, the goal is to learn a mapping from the input data for a given output. The correctness is provided along with the input data (i.e., there is supervision).

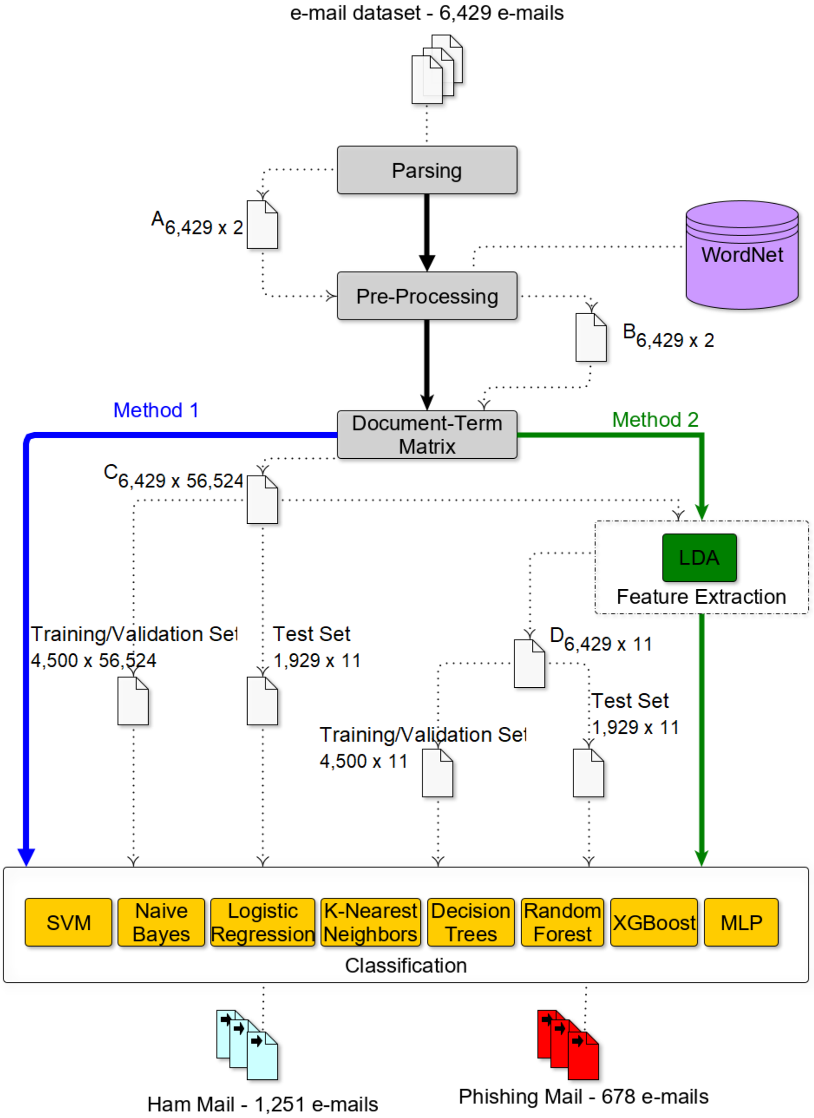

FIGURE 3. The proposed architecture dataflow.

legitimate e-mail (class 0), could be given by:

$$P(Class = 1) = \frac{1}{1 + e^{-g(x)}} \tag{15}$$

where:

$$g(x) = B_0 + B_1X_1 + B_2X_2 + \cdots + B_nX_n \tag{16}$$

These weights are estimated from the e-mails dataset, by the Maximum Likelihood method. If $P(Class = 1) > 0.5$, then this e-mail is phishing, and if $P(Class = 1) < 0.5$, the this e-mail is legitimate.

4) K-NEAREST NEIGHBORS (KNN)

This algorithm is based on the idea that similar data points are arranged nearby in an n-dimensional space. This similarity is measured by the distance between the points (usually the Euclidean Distance, or the Mahalanobis Distance) [55]. Thus, for a new data point, its classification is predicted by a validation of the local posterior probability of each class existing by the average of class membership over its K-nearest neighbors.

This algorithm is susceptible to the ‘‘curse of dimensionality’’, due to it be based on the distance between data points and its dimensions.

5) DECISION TREES

Decision Trees algorithms are built over the binary tree representation, in which each node corresponds to a single input variable, and given a split point on it, there are branches that bind to new nodes, according to the instance data point values for this feature, that will be below or above of this split point value (for a numeric variable). This procedure is done until the path arrives at a leaf node, in which the algorithm predicts the output variable class. At any of these nodes, what is done is splitting the input set into subsets for the next nodes. When analyzing e-mails to check if they are phishing or ham mail, the amount of times the words present in them, the weights of this amount, or the vectors represents theses e-mails are analyzed according to several split points (to find an optimal one, by information gain or Gini index criteria for example), and at a leaf node level, is predicted whether it is a phishing or a legitimate mail [46].

6) RANDOM FOREST

Random forest is an ensemble[8] algorithm based on Bootstrap Aggregation (bagging[9] technique), that creates a set of decision trees on randomly multiple samples of the training set, gets a prediction from each tree and, employing voting of these trees results, gives a better estimation for the final class of the test object. In its approach, instead of gets optimal split points for trees, by the randomness of the selected subset of the training set, it selects suboptimal splits. Due to this, different models will be created, which will be aggregated by combining their results [46].

7) XGBoost

Extreme Gradient Boosting (XGBoost) refers to an ensemble that implements (using the boosting[10] technique) a scalable and accurate version of gradient boosting on decision trees.

This algorithm offers high speed, performance, portability, and flexibility. It can optimize the loss function through three methods: gradient boosting, stochastic gradient boosting, or regularized gradient boosting method. Its default base learners are tree ensembles, in which each of them is a set of classification trees (CART). How explained about boosting technique, these trees are added sequentially, and each of them tries to reduce the misclassification of itself previous learners [56].

8) MULTILAYER PERCEPTRON (MLP)

It refers to an artificial neural network based on perceptrons[11] (artificial neurons[12]), in which each row of these neurons is a layer. The Multilayer Perceptron (MLP) neural networks have three layers of nodes. The first one, the input layer, is responsible for receiving the external stimuli. Then, there is a hidden layer that could be composed of one or more layers. Its function is to extract the environment's behavior, approximating any continuous function. Lastly, in this topology, there is the output layer that provides the answers for the received stimuli.

This structure utilizes the back-propagation technique for training, that, by the gradient descent, fit the network parameters (including with non-linear activation functions) to better express a training set with its labels, iteratively reducing its error. It works in two steps; in the forward pass, the stimuli are passed from layer to layer until the output layer, which provides predictions. These results are compared with the expected output provided by the training set. The prediction errors and its function are then used in the second step called the backward pass, in which the weights and biases of the model are updated by the partial derivatives of the error function, the back-propagation algorithm does these operations. This process is done until the network converges, or for a predetermined number of epochs[13] (in which case the network will not necessarily converge) [46].

V. RESULTS AND APPROACH EVALUATION

This section provides a detailed evaluation of the proposed approach through its prediction results. The utility measures to assess the results of our methods are present in Subsection V-A. The results are described in Subsection V-B, and, in the Subsection V-C, some pertinent observations are discussed.

A. MEASURES

The following measures are used to evaluate the classification algorithms performance in each perspective of the proposed approach. Their equations are based on true positive (tp), false positive (fp), false negative (fn), and true negative (tn) rates.

Accuracy:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{17}$$

Precision:

$$Precision = \frac{tp}{tp + fp} \tag{18}$$

Recall, True Positive Rate (TPR) or Sensitivity:

$$Recall = \frac{tp}{tp + fn} \tag{19}$$

False Positive Rate (FPR):

$$FPR = \frac{fp}{fp + tn} \tag{20}$$

[8] Ensemble is a machine learning technique that combines several base learning algorithms in order to produce a better predictive performance model.

[9] Bagging is a technique that uses the bootstrap algorithm to obtain a random sample from a given dataset with replacement. Then, it trains the base learners and aggregates their outputs to provide a lower variance model.

[10] Boosting is a technique that trains models in succession, with each new base learner being trained to correct the errors made by the previous learners. Through the use of weighted versions of the data, more weight is increasingly being given to misclassified examples. Learners are included sequentially until no further improvements can be made. The final predictions are obtained by weighted majority voting.

[11] Perceptron is a learning algorithm, based on an artificial neuron, for binary classifiers, that can solve only linearly separable problems.

[12] Artificial Neurons are the essential components of an artificial neural network, in which occurs a process that simulates a biological neuron working. There are input connections, that emulates the synapses and their forces, by assigning a weight to each input signal. These input values are sum by a linear combiner, that is also responsible for generating an activation potential (the internal network activation), by comparing this sum with an activation threshold. Thus, a non-linear activation function (a sigmoidal function, in MLP case) provides the neuron output signal.

[13] An epoch indicates how many times all of the training vectors are used once to update the weights. This measure varies according to the type of learning, that is, whether it is in online or in batch mode.

Specificity or True Negative Rate (TPR):

$$Specificity = 1 - FPR \tag{21}$$

F1 Score:

$$F1Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{22}$$

B. RESULTS

The results expressed below are the weighted measures obtained from both phishing detection classes (phishing e-mail or ham e-mail) and their respective samples.

Besides the feature joins (which correspond to words with the same synonym/lemma) made possible by WordNet synsets, it is also used to select only the features that correspond to words present in it. By the addition of this WordNet utilization, the number of features would arise around 18,000, but the obtained results were not better than those already available in the related works. Thus, as already mentioned in subsection IV-C.1, the number of features used in all the Methods in this paper before the implementation of dimensionality reduction techniques is 56,523, which corresponds to the result of lemmatization process with pos tagging (provided by wordnet) over the built vocabulary.

In order to highlight the effects brought to the proposed approaches of using the Wordnet database in the lemmatization process, the results obtained by these methods without the use of this resource also will be presented.

1) METHOD 1 - APPROACH BASED ON DOCUMENT-TERM MATRIX WITHOUT FEATURE REDUCTION TECHNIQUES

The values arranged in Table 1 refer to the perspective based on the Document-Term Matrix that uses all the terms obtained from the bag of words model as features.

TABLE 1. Method: bag of words and document-term matrix - 56,523 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9593	0.9592	0.9593	0.9592
Naive Bayes	0.9635	0.9635	0.9635	0.9635
Logistic Regression	0.9937	0.9937	0.9937	0.9937
KNN	0.8675	0.8986	0.8675	0.8704
Decision Trees	0.9817	0.9818	0.9817	0.9818
Random Forest	0.9901	0.9901	0.9901	0.9901
XGBoost	0.9875	0.9876	0.9875	0.9874
MLP	**0.9943**	**0.9943**	**0.9943**	**0.9943**

As stated earlier, this method does not address the high dimensionality (56,523 dimensions), sparsity (roughly 0.9982, that is, only about 0.18% of the data are non-zero values) and the represented context portion in the vector space model issues of the obtained matrix. However, it was measured to serve as a baseline/benchmark for the other method proposed in this paper. Thus, due to these problems, it demands more processing capacity and time than a method that attend to this questions (due to its higher complexity), although this perspective has reached in its best rate an f1-score of 99.43% (and an FPR of 0.32%, that is a specificity of almost 99.68%). These measures are achieved through the Multilayer Perceptron (MLP) classification algorithm, with the size of mini-batches and the maximum number of iterations as 10 (hyperparameters), and the rest of its parameters in the default setting.

This method, which just uses the e-mail bodies, performed better than similar approaches, with features derived from header, body, and links in e-mails, described in [33] and [14], which obtained respectively 97% and 99.1% as their best phishing detection rate. It also outperforms other classical approaches based on e-mail properties, such as [30], that achieved a measure of 96% identifying phishing e-mails.

Table 2 brings the results obtained to the same method, but without using the wordnet based lemmatization process. It is possible to verify that, although the Logistic Regression results are equal to those expressed in Table 1, all other algorithms perform worse.

TABLE 2. Method: bag of words and document-term matrix without lemmatization and wordnet-based processing - 63,448 features.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.8263	0.8614	0.8263	0.8084
Naive Bayes	0.9598	0.9598	0.9598	0.9598
Logistic Regression	**0.9937**	**0.9937**	**0.9937**	**0.9937**
KNN	0.8565	0.8927	0.8565	0.8598
Decision Trees	0.9750	0.9751	0.9750	0.9750
Random Forest	0.9849	0.9850	0.9849	0.9848
XGBoost	0.9817	0.9821	0.9817	0.9816
MLP	0.9917	0.9917	0.9917	0.9916

2) METHOD 2 - APPROACH BASED ON FEATURE EXTRACTION

For this method, Latent Dirichlet Allocation (LDA), whose prediction assessment values are in the tables from 3 to 9, is used as feature extraction approach. From the topics extracted from the e-mails bodies, representations of the e-mails are obtained in terms of the probability distribution of these topics, specific feature vectors for each of them.

The number of features to generate through this process is chosen based on two utility measures, perplexity, equation 11, and coherence (given by the C_{UCI} and C_{UMass} scores, equations (13) and (14), respectively). Given the values obtained, the best two scores of each measure were chosen as the number of topics for the LDA models, and also, for comparison, the worst score of each one.

Fig. 4 shows the log perplexity for some amounts of topics between 3 and 100. Except for the three topics option, for all others, the more topics, the lower the value for log perplexity. The best two scores for log perplexity are obtained

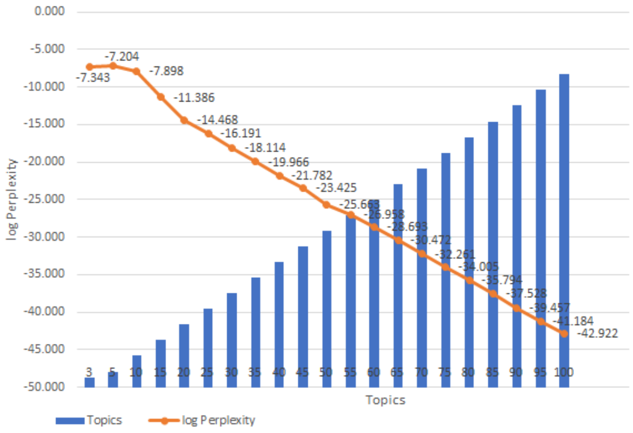

FIGURE 4. The log perplexity of the proposed LDA models.

for 5 and 3 topics, while the worst mark is achieved for 100 topics.

Regarding the coherence scores expressed in Fig. 5, there are many oscillations of these measures while the number of topics in the LDA model is increased, it is observed that the two highest scores obtained in the C_{UCI} are for 10 and 35 topics, and in the C_{UMass} are for 35 and 95 topics, while the worst mark of these measures are found for 5 and 100 topics respectively.

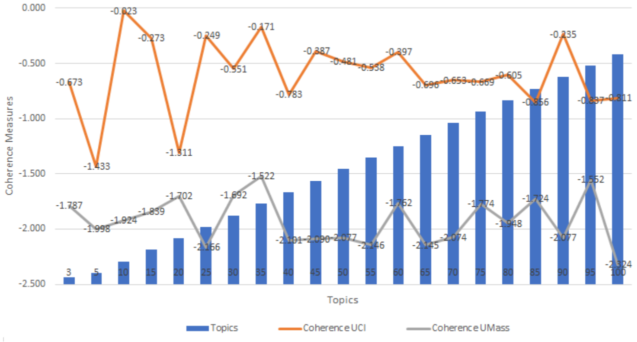

FIGURE 5. The coherence measures of the proposed LDA models.

Therefore, the chosen number of topics for the LDA models to use in Method 2 are 3, 5, 10, 35, 95, and 100.

The approach based on the LDA process, as shown in its metrics in tables from 3 to 9, presents better results than those obtained without any dimensionality reduction. These marks are presented below, by the number of topics, in descending order of their respective best scores.

Table 3 presents the marks achieved by Method 2 through LDA with ten topics. This approach obtains accuracy, precision, recall, and F1 score measures of 99.95%, FPR of 0%, and a neat specificity of 100%, which is, to the best of our knowledge, the highest result in phishing detection researches. 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, the rest of its parameters in the default setting. For this variation of the Method 2, LDA model with ten topics, all the used classification algorithms have marks equal or higher to 99.43%, the best measure presented by the Method 1 (F1 score of 99.43% for MLP).

TABLE 3. Method 2 - with 10 features extracted from 10 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9969	0.9969	0.9969	0.9969
Naive Bayes	0.9943	0.9943	0.9943	0.9943
Logistic Regression	0.9974	0.9974	0.9974	0.9974
KNN	0.9974	0.9974	0.9974	0.9974
Decision Trees	0.9958	0.9958	0.9958	0.9958
Random Forest	0.9974	0.9974	0.9974	0.9974
XGBoost	**0.9995**	**0.9995**	**0.9995**	**0.9995**
MLP	0.9974	0.9974	0.9974	0.9974

The marks reached by the Method 2 through the LDA with thirty-five topics (Table 4) are also the best present in this paper, as well as those present in the Table 4, with the same accuracy, precision, recall and F1 score measures of 99.95% through the XGBoost classification algorithm.

TABLE 4. Method 2 - with 35 features extracted from 35 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9844	0.9847	0.9844	0.9843
Naive Bayes	0.9650	0.9651	0.9650	0.9651
Logistic Regression	0.9896	0.9897	0.9896	0.9895
KNN	0.9974	0.9974	0.9974	0.9974
Decision Trees	0.9984	0.9984	0.9984	0.9984
Random Forest	0.9990	0.9990	0.9990	0.9990
XGBoost	**0.9995**	**0.9995**	**0.9995**	**0.9995**
MLP	0.9844	0.9846	0.9844	0.9843

The results expressed in Table 5 refer to the marks obtained through the Method 2 where the classification algorithms are fed with 95 input features attributes extracted in terms of the LDA model of 95 topics. It reaches a percentage of 99.90% in Precision, Recall (sensitivity) and F1 Score measures, also with XGBoost algorithm.[14] For these two variations of Method 2, models with 95 and 100 topics, the marks are below 99% just for Naive Bayes classification algorithm.

[14] This 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.

TABLE 5. Method 2 - with 95 features extracted from 95 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9958	0.9958	0.9958	0.9958
Naive Bayes	0.9744	0.9746	0.9744	0.9745
Logistic Regression	0.9984	0.9984	0.9984	0.9984
KNN	0.9979	0.9979	0.9979	0.9979
Decision Trees	0.9974	0.9974	0.9974	0.9974
Random Forest	0.9969	0.9969	0.9969	0.9969
XGBoost	**0.9990**	**0.9990**	**0.9990**	**0.9990**
MLP	0.9979	0.9979	0.9979	0.9979

TABLE 6. Method 2 - with 100 features extracted from 100 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9948	0.9948	0.9948	0.9948
Naive Bayes	0.9713	0.9718	0.9713	0.9714
Logistic Regression	0.9958	0.9958	0.9958	0.9958
KNN	**0.9984**	**0.9984**	**0.9984**	**0.9984**
Decision Trees	0.9963	0.9964	0.9963	0.9963
Random Forest	0.9963	0.9964	0.9963	0.9963
XGBoost	**0.9984**	**0.9984**	**0.9984**	**0.9984**
MLP	0.9901	0.9901	0.9901	0.9901

For 100 features, for example, it is achieved an F1 Score of 99.84% through XGBoost[15] and KNN[16] algorithms.

The variation that extracts features from the LDA model with three topics has as its best mark an F1 score of 99.74% with the Decision Trees[17] and Random Forest[18] algorithms.

For the variation obtained from the LDA model with five topics, it has achieved a 99.69% F1 Score (0.24% of FPR, which is a specificity of almost 99.76%) in XGBoost.[19]

Although features extracted from the LDA model with 3 and 5 topics, in their best marks, have reached the worst results among the variations utilized in the Method 2, their best results are better than the best mark of the Method 1. These measures are excellent, since it start the feature extraction step from DTM with 56,523 features, and, with just three or five features, derived from the LDA Model topics, achieve such classification prediction values.

TABLE 7. Method 2 - with 3 features extracted from 3 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9922	0.9922	0.9922	0.9922
Naive Bayes	0.9859	0.9859	0.9859	0.9859
Logistic Regression	0.9880	0.9880	0.9880	0.9880
KNN	0.9969	0.9969	0.9969	0.9969
Decision Trees	**0.9974**	**0.9974**	**0.9974**	**0.9974**
Random Forest	**0.9974**	**0.9974**	**0.9974**	**0.9974**
XGBoost	0.9969	0.9969	0.9969	0.9969
MLP	0.9963	0.9964	0.9963	0.9963

TABLE 8. Method 2 - with 5 features extracted from 5 topics LDA model.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9870	0.9870	0.9870	0.9870
Naive Bayes	0.9833	0.9838	0.9833	0.9834
Logistic Regression	0.9870	0.9870	0.9870	0.9869
KNN	0.9958	0.9958	0.9958	0.9958
Decision Trees	0.9963	0.9963	0.9963	0.9963
Random Forest	0.9958	0.9958	0.9958	0.9958
XGBoost	**0.9969**	**0.9969**	**0.9969**	**0.9969**
MLP	0.9906	0.9906	0.9906	0.9906

TABLE 9. Method 2 - with ten features extracted from 10 topics LDA model without Wordnet-based lemmatization process.

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	0.9885	0.9885	0.9885	0.9885
Naive Bayes	0.9885	0.9885	0.9885	0.9885
Logistic Regression	0.9875	0.9875	0.9875	0.9875
KNN	0.9932	0.9932	0.9932	0.9932
Decision Trees	0.9922	0.9922	0.9922	0.9922
Random Forest	0.9953	0.9953	0.9953	0.9953
XGBoost	**0.9958**	**0.9958**	**0.9958**	**0.9958**
MLP	0.9896	0.9896	0.9896	0.9896

As Method 2 variation with ten topics, in a general way, have reached the best results for this method, in the Table 9, it is showed the results if the input features do not go through the wordnet-based lemmatization process. It is noted that these results are consistently lower than those in the Table 3, but even so, more significant than those presented in the Method 1, and equal to the best mark presented in related works.

[15] This 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.

[16] This measure is achieved through the K-Nearest Neighbors classification algorithm, with the number of neighbors as 100, the weight function as distance, and the rest of its parameters in the default setting.

[17] This measure is achieved through the Decision Trees classification algorithm, with the entropy as function to measure the quality of a split, four as the depth maximum, and the rest of its parameters in the default setting.

[18] This measure is achieved through the Random Forest classification algorithm, with the entropy as function to measure the quality of a split, $\log_2 2$ as the number of features to consider when looking for the best split, three as the minimum number of samples required to split an internal node, and the rest of its parameters in the default setting.

[19] This measure is obtained 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.

C. DISCUSSIONS

Comparing the tabulated results presented in Section V with those of the prominent predictive researches, described in the baseline study (Section II), it is observed that the obtained marks of all Method 2 variations are higher than the mark hit in [22] (99.58% F1 Score), which besides LDA, uses LSA, keyword extraction, and other structural features from e-mails.

They also are higher than the results presented in [37], which uses distributed method Word2Vec and RCNNs, with 99.84%, and in [23] and [34], where topics Models are used, including LDA, with 99.1% as the best accuracy rate.

When compared with approaches based on other feature extraction techniques such as LSA and PCA [57] (accuracy of 98%), the Method 2 of our approach brings an even more significant difference.

The results achieved through the Method 2 show consistency, since, in each variation, for at least half of the eight classification algorithms used in this paper, the measured metrics return scores higher than 99%, and in its respective best mark higher than 99.50%.

Although Method 2 achieved the best marks among all the proposed methods, it is essential to note that each one of the Methods reached optimized results in its respective category, as can be observed in comparison with the related works presented. These marks highlight that the proposed models and their associated techniques brought essential boosts in the performance of phishing detection approaches.

When the variations of the methods with and without the lemmatization and the wordnet based process are compared, for both, Method 1 and 2, it was observed that the results obtained without this step, in a general way, are slightly lower than those that implement it. Besides that, the preprocessing steps, as well as the resampling/cross-validation techniques used in this work, made the classification algorithms obtain better results than those described in the related literature. That is why despite they do not present the best marks for the proposed methods, even those approaches variations without the wordnet-based process are also good results for the proposed problem.

Regarding the effectiveness of the methods in obtaining the most distinctive features, proposed by this paper, Method 2 not just presents the best results in this feature-based text classification, but also mitigates the posed obstacles related to VSM representation, providing a dense and low-dimension matrix that compress the data in the proposed texts with reduced noisy information. These problems are related to high dimensionality, sparsity, and contextual information that may be integrated into the proposed representation. They are problematic not only for phishing detection but for most natural language processing researches.

In addition to the excellent classification results, the LDA models also adds relevant information about the texts and the list of words used in it, as well as other relevant trends and measures, such as the dominant topics in each document, number of words in the text of the corpus, and the keywords of each topic.

It is also observed that the best results in six of the seven variations of method 2 were obtained with the use of the XGBoost algorithm, which demonstrates a pattern of good scores for its use, based on the appropriate configuration of its hyperparameters according to the proposed scheme for this task (section IV-F).

VI. CONCLUSION AND FUTURE WORK

Phishing has been a continuous cybercrime problem for all the e-mails users, especially in the corporate environments where security measures to deal with this type of incident have been increasingly refined and specialized, but also where this fraudulent practice seems to be ever more insightful. Among the proposed phishing detection techniques, those based on natural language processing and machine learning, in a data-driven approach, demonstrated greater effectiveness and higher accuracy than those based on filtering rules. Given this scenario, this paper, by combining enhanced techniques of text processing, feature extraction, topics modeling, training, and improved classification algorithms, propose an approach to obtain more distinctive/characteristic features for phishing detection issue in order to achieve optimized precision, recall, accuracy, and F1 score marks.

Each of the two proposed methods attained valuable results. In its respective best results, an F1 Score of 99.43% was achieved by the Method 1, and 99.95% using XGBoost algorithm by the Method 2 (that is 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). The results demonstrated in these measurements are not only due to the excellent performance of the classification algorithms, but also owing to the proposed combining techniques such as those textual processing procedures (for instance the pos tagging lemmatization), improved learning techniques for resampling and cross-validation, and estimators hyperparameters configuration. Method 2, besides the best performance, also demonstrated avoiding ẗhe curse of the dimensionalityänd the high sparsity, as well as providing relevant contextual information to the document representation. Therefore, this paper presents itself as a significant research contribution to the phishing e-mails detection and feature-based text classification fields.

Future works will focus on approaches that, combined with the techniques used in our pre-processing step, employ word embedding as a technique to generate features of the e-mails, capturing semantic and syntactic regularities from a corpus. Also, we will combining word embedding with techniques such as Latent Dirichlet Allocation (LDA2Vec), that together can provide and structure more information about the text under analysis, turning the results of all the proposed phishing detection process, as well as other natural language processing tasks more interpretable.

Other research questions, expected to be addressed in the context of phishing detection regards word embedding, refer to a better fit of pre-trained models to new NLP tasks (database sharing among organizations), as well as actions regarding the maintenance of prior knowledge contained in the representations employed.

REFERENCES

[1] (Jun. 2019). *Total Global Email & Spam*. [Online]. Available: https://www.talosintelligence.com/reputation_center/email_rep

[2] E. E. H. Lastdrager, "Achieving a consensual definition of phishing based on a systematic review of the literature," *Crime Sci.*, vol. 3, no. 9, pp. 1–10, 2014. [Online]. Available: https://crimesciencejournal.springeropen.com/articles/10.1186/s40163-014-0009-y

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

[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. 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.

[7] G. Sonowal and K. S. Kuppusamy, "PhiDMA—A phishing detection model with multi-filter approach," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 1, pp. 99–112, Jan. 2020, doi: 10.1016/j.jksuci.2017.07.005.

[8] R. B. Basnet, A. H. Sung, and Q. Liu, "Learning to detect phishing Urls," *Int. J. Res. Eng. Technol.*, vol. 3, no. 6, pp. 11–24, 2014, doi: 10.15623/ijret.2014.0306003.

[9] P. Yang, G. Zhao, and P. Zeng, "Phishing Website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.

[10] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "OFS-NN: An effective phishing Websites detection model based on optimal feature selection and neural network," *IEEE Access*, vol. 7, pp. 73271–73284, 2019, doi: 10.1109/ACCESS.2019.2920655.

[11] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Work. Groups 2nd Annu. Ecrime Res. Summit (eCrime)*, 2007, pp. 60–69, doi: 10.1145/1299015.1299021.

[12] N. A. Unnithan, N. B. Harikrishnan, R. Vinayakumar, K. P. Soman, and S. Sundarakrishna, "Detecting phishing E-mail using machine learning techniques," in *Proc. 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, 2018, pp. 51–54.

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

[14] 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.

[15] 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, p. 1, Dec. 2015, doi: 10.1186/s40537-014-0007-7.

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

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

[18] K. Liu, A. Bellet, and F. Sha, "Similarity learning for high-dimensional sparse data," in *Proc. Mach. Learn. Res. (PMLR)*, vol. 38, G. Lebanon and S. V. N. Vishwanathan, Eds. San Diego, CA, USA, 2015, pp. 653–662. [Online]. Available: http://proceedings.mlr.press/v38/liu15.html

[19] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, no. 1, pp. 141–188, 2010, doi: 10.1613/jair.2934.

[20] N. A. Unnithan, N. B. Harikrishnan, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Machine learning based phishing E-mail detection," in *Proc. CEUR Workshop (CEUR-WS)*, vol. 2124, 2018, pp. 64–68.

[21] N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "A machine learning approach towards phishing email detection," in *Proc. CEUR Workshop (CEUR-WS)*, vol. 2124, 2018, pp. 21–28.

[22] G. L'Huillier, A. Hevia, R. Weber, and S. A. Ríos, "Latent semantic analysis and keyword extraction for phishing classification," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, May 2010, pp. 129–131, doi: 10.1109/ISI.2010.5484762.

[23] 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.

[24] N. Unnithan, N. Harikrishnan, R. Vinayakumar, K. Soman, and S. Sundarakrishna, "Detecting phishing e-mail using machine learning techniques," in *Proc. CEUR Workshop (CEUR-WS)*, vol. 2124, 2018, pp. 50–56.

[25] H. B. Ganesh, R. Vinayakumar, M. A. Kumar, and K. Soman, "Distributed representation using target classes: Bag of tricks for security and privacy analytics," in *Proc. CEUR Workshop (CEUR-WS)*, vol. 2124, 2018, pp. 10–15.

[26] R. Vinayakumar, H. Barathi Ganesh, M. Anand Kumar, K. Soman, and P. Poornachandran, "Deepanti-phishnet: Applying deep neural networks for phishing Email detection," in *Proc. CEUR Workshop (CEUR-WS)*, vol. 2124, 2018, pp. 39–49.

[27] J. Nazario. *Phishing Corpus*. [Online]. Available: https://monkey.org/~jose/phishing/

[28] J. Mason. *The Apache Spamassassin Public Corpus*. [Online]. Available: https://spamassassin.apache.org/old/publiccorpus

[29] 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.

[30] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing Emails," in *Proc. Int. World Wide Web Conf. Committee (IW3C2)*, 2007, pp. 649–656.

[31] 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.

[32] A. Elaassal, L. De Moraes, S. Baki, R. Verma, and A. Das. (2018). *IWSPA-AP: 1st Anti-Phishing Shared Task Pilot at 4th ACM IWSPA*. [Online]. Available: http://ceur-ws.org/Vol-2124/

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

[34] 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.

[35] M. Zareapoor and K. R. Seeja, "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, p. 60, Mar. 2015.

[36] 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.

[37] 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.

[38] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432, doi: 10.1371/journal.pone.0118432.

[39] O. A. Adewumi and A. A. Akinyelu, "A hybrid firefly and support vector machine classifier for phishing Email detection," *Kybernetes*, vol. 45, no. 6, pp. 977–994, Jun. 2016, doi: 10.1108/K-07-2014-0129.

[40] I. R. A. Hamid and J. Abawajy, "Phishing Email feature selection approach," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 916–921.

[41] D. M. Blei, A. Y. Ng, M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[42] C. D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.

[43] G. A. Miller, "Wordnet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[44] N. Guarino, "Formal ontologies and information systems," in *Proc. 1st Int. Conf. (FOIS)*, 1998, pp. 3–15.

[45] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, 2003.

[46] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2014.

[47] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012, doi: 10.1145/2133806.2133826.

[48] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring topic coherence over many models and many topics," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (MNLP-CoNLL)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 952–961.

[49] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Using LDA to uncover the underlying structures and relations in smart city data streams," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1755–1766, Jun. 2018, doi: 10.1109/JSYST.2017.2723818.

[50] Y.-K. Tang, X.-L. Mao, H. Huang, X. Shi, and G. Wen, "Conceptualization topic modeling," *Multimedia Tools Appl.*, vol. 77, no. 3, pp. 3455–3471, Feb. 2018, doi: 10.1007/s11042-0175145-4.

[51] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei, "Reading tea leaves: How humans interpret topic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2009, pp. 288–296.

[52] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proc. 8th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2015, pp. 399–408, doi: 10.1145/2684822.2685324.

[53] F. Rosner, A. Hinneburg, M. Röder, M. Nettling, and A. Both, "Evaluating topic coherence measures," *Comput. Res. Repository (CoRR)*, vol. abs/1403.6397, pp. 1–4, 2014.

[54] 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.

[55] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Germany: Springer-Verlag, 2006.

[56] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[57] 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," *ArXiv*, vol. abs/1905.02845, 2019. [Online]. Available: https://arxiv.org/abs/1905.02845

EDER S. 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 Brasilia, 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 at the National Agency of Telecommunications of Brazil (Anatel). His research interests include computer networks and information security, machine learning, and natural language processing.

RAFAEL T. 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 École 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, France, in 1988.

He was a Visiting Researcher with the Group for Security of Information Systems and Networks (SSIR), Ecole Supérieure d'Electricité - Supélec, Rennes, France, from 2006 to 2007. He worked in the private sector, from 1988 to 1996. Since 1996, he has been a Network Engineering Associate Professor at the Electrical Engineering Department, University of Brasília, Brazil, where he is a Coordinator of the Professional Post-Graduate Program on Electrical Engineering (PPEE) and supervises the Decision Technologies Laboratory (LATITUDE). He is 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.

Dr. Sousa has received the 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 P. DE B. 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 Brasilia (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 for information management at the National Agency of Telecommunications of Brazil (Anatel).

JOÃO PAULO C. L. 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 Brasilia (UnB), Brazil, in 2006, and the Ph.D. degree in electrical and information engineering from the Ilmenau University of Tecnology (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. Since March 2019, he has been a Senior Development Engineer at EFS on the area of autonomous driving. Since October 2019, he has also been a Lecturer with the Ingolstadt University of Applied Sciences on the area of autonomous vehicles. He has published more than 170 scientific publications and patents. His research interests are autonomous vehicles, beyond 5G, GNSS, and adaptive and array signal processing. He received six best paper awards in international conferences.

CLÁUDIO G. 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, and the master's degree in psycholinguistics from the Graduate Program in Linguistic Studies, Faculty of Letters, Federal University of Minas Gerais, Belo Horizonte, 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 Geraism, Belo Horizonte, 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), Brasilia, Brazil. His research interests include information architecture, information retrieval, deep learning, blockchain, and natural language processing.