IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 23, 2020, accepted January 1, 2021, date of publication January 13, 2021, date of current version January 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051374

Knowledge-Based Approach to Detect Potentially Risky Websites

JUAN CARLOS PRIETO, ALBERTO FERNÁNDEZ-ISABEL, ISAAC MARTÍN DE DIEGO, FELIPE ORTEGA, AND JAVIER M. MOGUERZA

Data Science Laboratory, Rey Juan Carlos University, 28933 Móstoles, Spain

Corresponding author: Juan Carlos Prieto (juancarlos.prieto@urjc.es)

Research supported by grants from the Spanish Ministry of Economy and Competitiveness, under the Retos-Colaboración program: PPI (Ref: RTC-2015-3580-7), UNIKO (Ref: RTC-2015-3521-7), and SABERMED (Ref: RTC-2017-6253-1), and Retos-Investigación program: MODAS-IN (Ref: RTI-2018-094269-B-I00).

ABSTRACT Nowadays, fraudulent and malicious websites are emerging as a harmful and very common problem on the Internet. It causes huge money losses and irreparable damage for both companies and particulars. To face this situation, governments have approved multiple law projects. This way, the legality on the Internet is being enforced and sanctions to those offenders who develop illegal or malicious activities are being imposed. However, governments still need a way to simplify the classification of websites into risky or non-risky, since most of this work is manual. This paper presents the *DOmains Classifier based on RIsky Websites* (*DOCRIW*) framework to detect domains that contain possible fraud or malicious content. It is based on two main components. The first component is a previously built knowledge base containing information from risky websites. The second one complements the system with a binary classifier able to label a website (as risky or not) considering just its domain. The system makes use of web information sources and includes host-based variables. It also applies similarity measures, supervised learning algorithms and optimization methods to enhance its performance. The presented work is experimental, rendering promising outcomes.

INDEX TERMS Risky website detection, malware alerts, knowledge-based systems, similarity metrics, combination of information.

I. INTRODUCTION

Public bodies that prosecute fraudulent and malicious websites dedicate a significant amount of time and resources to detect scam and malware on the Internet [2]. Most of this work is usually manual, which translates into hard and inefficient efforts. For this reason, it has become essential to develop systems able to automate the classification of websites into potentially risky or non-risky according to the features of these sites. In this context, a risky website is one with malicious, unsafe or fraudulent content with dangerous intentions against their visitors [1].

The study by Spanish Information Security Observatory (OSI) captures the magnitude of the risky websites problems in Spain [3]. Among the main results and conclusions of the study, it should be noted that a 53.1% of Spanish Internet users claimed to have been victims of an attempt (not necessarily consummated) of fraud in the last three months. The analysis of potentially fraudulent situations occurred to users while surfing the Web highlights the reception of invitations to visit some suspicious website (34.4%). In the analyzed period, 95.2% of Spanish Internet users share that they have not suffered economic damage in the last three months as a result of a fraud via Internet, while 4.8% have suffered losses. Besides, the empirical analysis of the equipment shows that 39.8% of the computers host some type of Trojan, 6.8% host banking Trojans (malicious code snippets intended to intercept electronic banking credentials of specific entities) and a 5.8% suffer a rogue-ware infection (or fake antivirus). Furthermore, 81.8% of Internet users who have suffered an incident of this type have not changed their habits surfing websites, compared to 5% who have abandoned this activity and 13.2% who have reduced the use of Internet. The Spanish Observatory of Computer Crimes (OEDI) have reported 110, 613 cyber-crimes in Spain in 2018, 74% of them have been fraud [4].

In this paper, two main contributions have been made. The first one consists of a novel Knowledge-Based System (KBS)

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

to automate the detection of potentially risky websites. It is called *DOmains Classifier based on RIsky Websites* (*DOCRIW*). The second one offering mathematical conclusions of similarity metrics, Machine Learning (ML) models and optimization methods that enhance the accuracy of the framework.

DOCRIW includes a knowledge database built using information collected from websites that present illegal and malicious content. These websites have been labeled by experts in the domain. *DOCRIW* also includes a module to predict the risk of websites for those not found in this knowledge database. This module is based on a binary classifier trained using a supervised learning process. It uses a set of domains already labeled as *risky* or *non-risky* and a set of host-based variables related to these domains.

The *DOCRIW* framework has two main work flows. The first one labels the domains, while the second addresses the information gathering from web information sources. Both work flows provide guidelines to users in order to illustrate the global functionalities of *DOCRIW*.

Three different experiments have been exposed in order to show the viability of the proposal. The first experiment is performed in order to find the optimal parameters of the binary classifier. The second experiment uses domains previously labeled as *risky* or *non-risky* in order to evaluate the *accuracy* of the proposed classifier. The third experiment includes new domains and it is presented to show the whole system (i.e., the modules that use the information collected from web sources and the binary classifier).

The rest of paper is organized as follows. Section II establishes the context and describes similar approaches related to scam and malware issues. Section III introduces the framework architecture and its foundations. Section IV explains the main work flows of the system. Section V presents a set of experiments to show the viability of the proposal. Finally, Section VI concludes and provides future lines of work.

II. BACKGROUND

DOCRIW is a KBS built to detect potentially *risky* websites. These websites usually present malicious content and fraud, which are two of the most detrimental problems found regarding Internet browsing. Furthermore, different ML techniques have been included to improve the accuracy by classifying these websites. All these issues are addressed in detail in the following sections. Thus, KBSs are presented in Section II-A. Section II-B introduces the malware and fraud detection solutions. Finally, Section II-C delves into the ML methods applied to malware and fraud detection.

A. KNOWLEDGE-BASED SYSTEMS

KBSs have become a relevant approach nowadays [5]. KBSs are frameworks able to process data and information in order to generate knowledge using Artificial Intelligence (AI) to solve general tasks. These systems usually comprehend a storage component (e.g., a database) to ease the knowledge retrieval in response to specific queries, along with learning and justification, or to transfer knowledge from one domain of knowledge to another. They are formed by different modules to address the needs of the users or to optimize the system [6]. Such systems are capable of cooperating with human users and are being used for problem solving, training and assisting users and experts of the domain for which the systems are developed. Examples of these modules are the visualization interface and a possible set of Machine Learning (ML) techniques [7]. These systems are designed to collect information to make decisions consequently. The most important KBS approaches include ML techniques (usually supervised learning) which are able to identify and interpret relevant features from the data. Approaches have varied from simple rule-based systems to more complex models that use fuzzy logic and artificial neural networks. Natural Language Processing is one of the most widespread scopes in this domain [8].

In the case of *DOCRIW*, it presents functionalities to gather, organize and use external knowledge gathered from web information sources in order to provide support to the classifier in order to avoid scam and malware propagation.

B. RISKY WEBSITES DETECTION

A risky website can be defined as a website that has malicious intentions against their visitors [9]. These websites are prone to distribute different types of malware, fraud and phishing techniques, and other forms of cybercrime acts [1].

Malware and fraud have been exploited as a very common issue in the current society provoking great financial damages to particulars and companies. Heuristics have been the traditional way to fight against these harmful practices. Nevertheless, this kind of analysis is no longer considered effective because fraud instances can be similar in appearance and content, but usually are not identical. Fraud is an adaptive crime, so it needs special methods of intelligent data analysis to detect and prevent it. Hence, automated methods have been developed to detect these threats. The most typical solution for the detection of malware is based on behavior. The analysis includes re-playing the malware in an emulated environment to generate behavior reports [10]. Important methods to detect and prevent fraud are network-based. Phishing detection modules detect fraud attacks by determining that a domain is similar to a known phishing domain, or that an address of the network-based resource from which the content is received has suspicious network properties [11].

However, these solutions have significant drawbacks. Notice that, it is necessary to re-play the malware in a virtual environment or to display the content of an URL. Hence, achieving good results implies high costs both in time and resources. The approach presented in this paper faces this issue by simplifying the entire process. The only input that the proposed system will demand to determine whether a website is potentially *risky* is the domain name and its related features.

In this line, there are similar studies that use the domain to detect malicious websites. However, these studies usually use

textual features [12] or they use an IP address approach [13]. Other alternatives use the Domain Name System (DNS) [14] and the Whois [15] features, which are more similar to the presented proposal. The main different lies in the classification task, as it is carried out using both lexical and host-based variables. Additionally, other differential contribution is the use of similarity measures and assembling methods for the optimal classification. The *DOCRIW* framework uses ML techniques to improve the performance of the binary classifiers according to different evaluation metrics.

C. ML FOR RISKY WEBSITES DETECTION

For the detection of risky websites, statistical data analysis techniques have been traditionally used. Instances of these statistical data analysis techniques are: calculation of statistical parameters such as probability distribution and quantiles [16], time series analysis [17], clustering to find patterns among data sets, data matching used to compare two data sets or regression analysis to detect relationships between variables of interest [18].

However, more advanced AI techniques have recently appeared: expert systems to detect fraud in the form of rules [19], pattern recognition to approximate classes or patterns of suspicious behavior [20], ML to automatically detect risky features [21], neural networks that can learn suspicious patterns from data [22], optimization of weighted extreme learning machines for imbalanced classification in credit card fraud detection [23], transaction fraud detection based on total order relation and behavior diversity [24], online fault detection models and strategies based on clouds [25], and deep representation learning with full center loss for credit card fraud detection [26].

There are several ML techniques used in the state of art in the context of risky websites detection. In [27] a comprehensive survey and a structural understanding of malicious URL detection techniques using ML is presented. Among the most common techniques in this field are the Support Vector Machines (SVM) [28]–[32], Logistic Regression (LR) [31], [33]–[35], Naïve Bayes (NB) [34]–[37], and Decision Tree [31], [38], [39]. In [40] a set of ML models have been evaluated for classifying malicious websites given their URL as input. In addition, a ML method based on SVM to classify malicious websites by using only domain names has been proposed [41].

In the last few years, other relevant works regarding classification algorithms have proposed new paths to avoid problems introduced by traditional prediction methods. In the field of artificial neural networks, a new Dendritic Neuron Model (DNM) has been developed for a better understanding of a biological neuronal system and for providing a more useful method for solving practical problems by considering the non-linearity of synapses [42]. Reliable predictions for Quality of Service (QoS) has also been an important research topic in the domain of service computing. Two interesting lines to make a highly accurate prediction for missing QoS data are to build an ensemble of Non-Negative Latent Factor (NLF) models [43], and to present a Biased Non-Negative Latent Factorization of Tensors (BNLFTs) model for temporal pattern-aware QoS prediction [44]. Regarding the processing of high-dimensional and sparse matrices and imbalanced data, Non-negative Matrix Factorization (NMF) models have proven to be highly effective owing to their fine representativeness of the non-negative data [45] and the embedded feature selection method using the Weighted Gini Index (WGI) has improved the accuracy [46].

In the present paper, ML models are used to detect risky patterns on websites. The *DOCRIW* framework has been tested using a battery of these models. Several studies are performed to compare ML algorithms, and weighted combinations of them. The best ML method for the classification task is selected.

III. FRAMEWORK ARCHITECTURE

The *DOCRIW* framework is an innovative platform focused on detecting potentially risky websites. Thus, it is able to classify websites into *risky* or *non-risky*. For this purpose, it extracts knowledge from external web information sources and makes predictions when no information is available. In order to make these predictions, *DOCRIW* builds similarity measures to train ML algorithms, and uses optimization methods to select the best model and the proper parameters. Notice that the proposed approach refers to direct access of a website from users (i.e., when the users are trying to be scummed).

Regarding the general architecture of the system, it presents four main modules (see Fig. 1): the *Domains Extraction and Validation* module, the *Host-based Variables Extraction* module, the *Classification* module and the *Information Updating* module. Besides, the system also holds a *Graphical Interface* and two databases: the *Knowledge Base* and the *Machine Learning Model*. The *Graphical Interface* is in charge of the interaction with users. The *Knowledge Base* is an ElasticSearch database [48] that organizes the

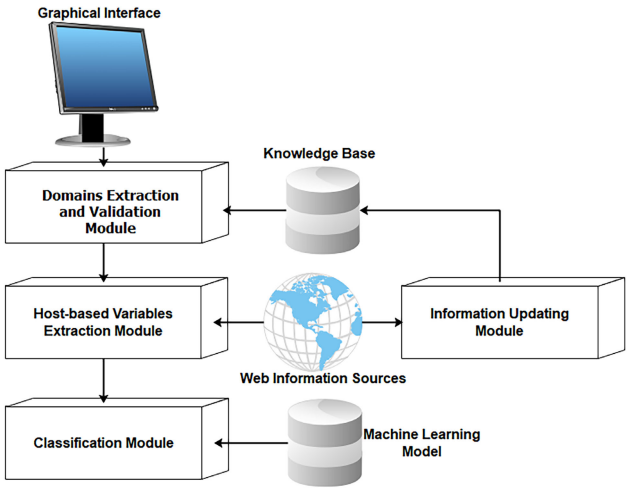


FIGURE 1. Excerpt of the architecture of the *DOCRIW* framework.

knowledge collected from the *Web Information Sources*. The *Machine Learning Model* includes a classifier previously trained. Next, the rest of the modules are described.

A. DOMAINS EXTRACTION AND VALIDATION MODULE

This module processes URLs by extracting their corresponding domains and analyzing them. In order to achieve these tasks, it uses the *Knowledge Base* module to obtain the previously labeled *risky* domains.

The module presents two components: the *URL Analyzer* and the *Domains Evaluator* (see Fig. 2). The first receives information from the *Graphical Interface* and acts in response to the requests made by users. The information provided by the *Graphical Interface* can be entire URLs or domains previously preprocessed. The *URL Analyzer* evaluates the proposed domain in both situations. Thus, it checks if the domain is correct (i.e., status code equals to 200) and it detects possible redirections to landing pages. In this case, all the landing pages are included to be analyzed, extracting the associated domains. The *Domains Evaluator* component matches the obtained domains and the domains stored in the database. When matches are found, the reported domain is labeled as *risky*. When none of the domains are matched, the module sends the original domain to the *Host-based Variables Extraction* module.

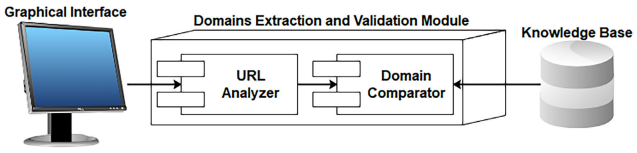


FIGURE 2. Excerpt of the architecture of the ***Domains Extraction and Validation*** module.

B. HOST-BASED VARIABLES EXTRACTION MODULE

The *Host-based Variables Extraction* module collects new host-based variables through the Whois API REST [49], which is part of the *Web Information Sources*. It provides information about city, country, creation date, expiration date and e-mail. Thus, this module characterizes the analyzed domain.

Regarding the architecture of the module (see Fig. 3), it consists of two components: the *Host-based Variables Collector* and the *Data Cleaning*. The first one manages the information provided by the Whois API, building a dataset as output. The second one addresses the cleanup task unifying the results. For instance, the country abbreviations are adapted according to the ISO code [50], and possible mismatches between values are normalized (e.g., a city name with accent marks and the same city name without them).

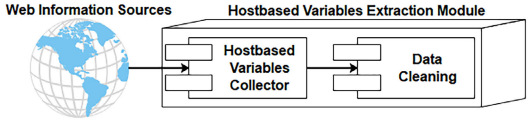


FIGURE 3. Excerpt of the architecture of the ***Host-based Variables Extraction*** module.

C. CLASSIFICATION MODULE

This module classifies domains into *risky* or *non-risky* labels when they are not found in the *Knowledge Base*. It uses the variables generated by the *Host-based Variables Extraction* module to feed the *Machine Learning Model* in order to obtain a predicted value for domains. The *Machine Learning Model* has been selected based on empirical results. The complete study to select the elements related to this model will be explained later. The model includes a definition of the similarity between domains, a LR algorithm, a threshold for the probability provided by the algorithm, and a reference set of domains.

Regarding the architecture of the module (see Fig. 4), it consists of two components: the *Similarity Creator* and the *Classifier*. The first one calculates similarities between the new domain and any of the domains in the reference set, for each variable (i.e., domain name, city, country, creation date, expiration date and e-mail). The similarity based on domain name is calculated using the *Levenshtein* distance [51]. The other five similarities (corresponding to the host-based variables) evaluate whether two domains have the same value for the corresponding variable or not. For instance, for the country variable, the similarity is 0 when the two domains are hosted in two different countries, and it is 1 when the two domains are hosted in the same country. Next, a global similarity between the new domain and any of the domains in the reference set, is calculated as a weighted average of the previous similarities. These weights are provided by the *Machine Learning Model*.

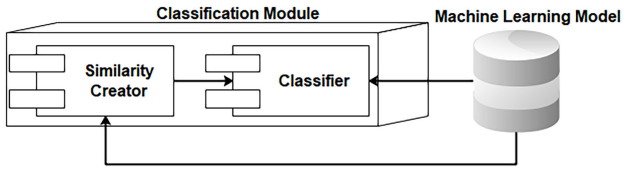


FIGURE 4. Excerpt of the architecture of the ***Classification*** module.

The second component of the *Classification* module is the *Classifier*. The LR algorithm provided by the *Machine Learning Model* is fed with the vector of global similarities previously calculated to obtained a prediction (between 0 and 1) of the riskiness. Given a predefined cut-off probability threshold that maximize the overall performance, the domain is labeled as *risky* or *non-risky*.

D. INFORMATION UPDATING MODULE

This module collects data from the *Web Information Sources* to update the information stored in the *Knowledge Base*. Thus, it obtains new reported malicious domains from *AA419* [52] and *MalwareURL.com* [53], two public websites that

identify risky domains and makes this data available as a public service.

This task is periodically executed by updating former register with the new gathered information. This module stores the information in the *Risky Domains* index of the *Knowledge Base* module.

IV. LABELING DOMAINS PROCESS

The *DOCRIW* framework addresses the process of labeling a domain as *risky* or *non-risky*. This process describes the interactions among the modules of *DOCRIW* to label a domain as *risky* or *non-risky* according to its malicious or fraudulent nature. It implies seven sequential steps and a decision.

The work flow starts by processing the URL provided by users in the *URL processing* step (see Fig. 5). This input could be a domain name or a specific URL. In the second case the URL is processed in order to extract the proper domain name. Then, this domain is compared to the domains stored in the *Knowledge Base* module in the *Domain found in Risky Domains DB* step. If the domain is found, the next step is *Risky domain labeling*. There, the response of the system is provided. This response is the domain labeled as *risky* with probability equals to 1. All these tasks are achieved in the *Domains Extraction and Validation* module. In contrast, if the domain name is not found, the next step is *Host-based variables collection*. There, the system collects information about the domain according to the five host-based features selected (i.e., city, country, creation date, expiration date and e-mail). This information is normalized in the *Data normalization* step. These operations are achieved by the *Host-based Variables Extraction* module. Once this part is completed, the corresponding global similarities are calculated, using the information of the current domain and the domains used to train the ML model. Finally, the process finishes making a prediction in the *Predictive model creation* step and the final labeling is achieved in the *Risky or non-risky domain labeling* step. In the *DOCRIW* architecture, the *Classification* module is responsible for these tasks.

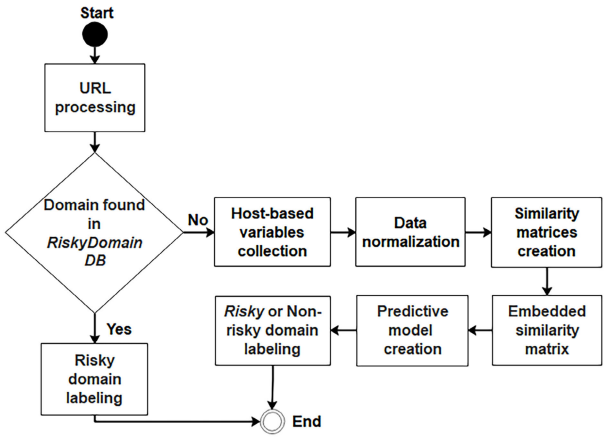


FIGURE 5. Excerpt of the *Domain Labeling* work flow.

Besides, the framework updates the information of the system using the corresponding web information sources in order to acquire new reported risky domains. This is executed by the *Information updating* module once a day.

V. EXPERIMENTS

This section addresses a set of experiments that explain the design of the framework, and evaluate the overall performance of the system.

The first experiment, presented in Section V-A, displays a test battery carried out to justify the selection of the elements included into the *Machine Learning Model* (see Fig. 4). These elements are: the similarity between domains, the ML algorithm, the threshold for the probability provided by the algorithm, and a reference set of domains. In this case, 1, 500 domains previously labeled are used to train and test the model (750 *non-risky* and 750 *risky*).

The purpose of the second experiment is to validate the performance of the *Machine Learning Model*. Section V-B describes an experiment which addresses two different issues. In the first one, the performance for the classification of *risky* domains is evaluated. To this aim, 200 domains extracted from the *Risky Domains* index of the *Knowledge Base* module have been used. In the second issue, it is evaluated the performance to classify *non-risky* domains. In this case, 200 prestigious domains have been tested. This second experiment does not use the *Domains Extraction and Validation Module*, so only the classifier is evaluated.

The third experiment, described in Section V-C, simulates the complete labeling functionality of the system. It uses 100 *risky* domains, 100 *non-risky* domains and 20 inactive domains in order to provide the corresponding predicted labels (*risky* and *non-risky*) and their probabilities.

A. TRAIN AND TEST OF THE MACHINE LEARNING MODEL

This experiment is used to select and test the proper elements of the *Machine Learning Model*. It has been carried out with 1, 500 domains already labeled as *risky* or *non-risky* by experts of the domain.

The dataset has been divided into train (70 %) and test (30 %). The train set has been used to train a set of ML algorithms that predict the labels of the entry domains. The domains of the test set have been used as input of the model to evaluate the performance. Thus, the similarity measures, the weights to combine these measures, the ML algorithm and the cut-off probability threshold that maximize the overall performance have been selected. The train and test datasets have been randomly partitioned 10 times, and a run of the experiment has been done over each partition. Therefore, the mean and standard deviation of the performance measures are presented.

The *Levenshtein* distance is used as similarity measure for the *domain name*. The other similarities were calculated by evaluating whether two domains have the same value for the corresponding variable (similarity equals to 1) or not (similarity equals to 0). Thus, six different similarity

TABLE 1. ML algorithm description.

ML Algorithm	Definition	Reference
AB	AdaBoost is an ensemble method that trains and deploys trees in series. AdaBoost implements boosting, wherein a set of weak classifiers is connected in series such that each weak classifier tries to improve the classification of samples that were misclassified by the previous weak classifier. In doing so, boosting combines weak classifiers in series to create a strong classifier.	[54]
ERT	Extremely Random Tree is the same as Random Forest with the exception that the decision thresholds used to divide the nodes are also chosen randomly instead of selecting the most discriminative ones.	[47]
GB	Gradient boosting produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.	[55]
KNN	K-Nearest Neighbors is a non-parametric method that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).	[57]
LDA	Linear Discriminant Analysis is a learning method that allows finding a linear combination of features that characterize or separate two classes. It is used as a linear classifier or for dimension reduction tasks before classification.	[47]
LR	Logistic Regression is a statistical method that uses a logistic function to model a binary dependent variable, Y, from one or more response variables, X.	[33]
NB	Naïve Bayes is a simple learning algorithm that utilizes Bayes rule together with a strong assumption that the attributes are conditionally independent, given the class. While this independence assumption is often violated in practice, Naïve Bayes nonetheless often delivers competitive classification accuracy.	[47]
RF	Random Forest is an ensemble of random Decision Tree classifiers, that makes predictions by combining the predictions of the individual trees.	[47]
SVM	Support Vector Machines are particular linear classifiers which are based on the margin maximization principle. They perform structural risk minimization, which improves the complexity of the classifier with the aim of achieving excellent generalization performance. The SVM accomplishes the classification task by constructing, in a higher dimensional space, the hyperplane that optimally separates the data into two categories.	[56]

measures were obtained. The global similarity was calculated as a weighted sum of the six individual similarities. The best weights selected to calculate the global similarity were 0.5, 0.15, 0.25, 0.05, 0.05, and 0, corresponding to domain name, city, country, creation date, expiration date, and e-mail, respectively. Thus, in this case the e-mail similarity was not included in the global similarity calculation. These values were selected during the training phase and evaluated in the testing phase.

Several ML algorithms designed to offer a good response as binary classifiers have been evaluated. These ones are the most typical in the literature of the domain [1]. Thus, an algorithm based on LR [33], two bagging algorithms using decision trees (Random Forest (RF) [47] and Extremely Randomized Trees (ERT) [47], two boosting algorithms (Adaboost (AB), [54] and Gradient Boosting (GB)) [55] and an algorithm based on support vectors (SVM) [56]. Additionally, other algorithms have been included to gauge the performance of the previous ones considered: KNN [57], NB [47] and Linear Discriminant Analysis (LDA) [47]. A brief description of these methods is presented in Table 1. Notice that the computational and storage complexities of these algorithms are different. However, the aim of this experiment is to select a unique ML model. Therefore, this issue does not affect the relative performance of the *DOCRIW* framework.

The Grid Search method [58] has been applied to select the optimal parameters values for the ML models. This method tests each algorithm with different values of its parameters and compares the obtained results. In addition, other techniques have been used to find the optimal values, such as the Out-Of-Bag (OOB) Error Rate plot for Random Forest [47]. The parameters that optimize the ML algorithms are the following. LR makes use of Ridge Regression (L2) [59] as regularization function and a penalty parameter equal to 10. RF includes 400 estimators (trees), log2 (logarithm in base 2) as the function that calculates the number of variables per tree, and Gini coefficient [60] as selection criterion. ERT uses the same parameters as RF except for the selection criterion, that has been set up to Entropy instead of Gini coefficient. AB includes 300 estimators and a learning rate equal to 0.1. GB uses the same parameters as AB, adding a depth length equal to 3. SVM uses a linear kernel and a penalty parameter equals to 1. Finally, KNN has been set up to 5 neighbors, a weighed importance for closer neighbors and the Euclidean distance. NB and LDA have no parameters.

The performance metrics considered to test the *Machine Learning Model* are [61]: *accuracy*, *sensitivity*, and *specificity*. The *accuracy* is the proportion of domains (*risky* and *non-risky*) that are correctly identified by the ML method:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}, \tag{1}$$

where:

- TN = *non-risky* websites rightly classified as *non-risky*.
- TP = *risky* websites rightly classified as *risky*.
- FN = *risky* websites wrongly classified as *non-risky*.
- FP = *non-risky* websites wrongly classified as *risky*.

The *sensitivity* is the proportion of *risky* domains that are correctly identified as such:

$$Sensitivity = \frac{TP}{TP + FN}. \tag{2}$$

Finally, the *specificity* is the proportion of *non-risky* domains that are correctly identified as such:

$$Specificity = \frac{TN}{TN + FP}. \quad (3)$$

Table 2 shows the performance metrics for all the ML algorithms evaluated. The LR algorithm was the best one for every performance measure. The cut-off probability threshold that maximize the performance was 0.6. That is, the domains with a predicted probability of being *risky* lower than 0.6 are classified as *non-risky*. By contrast, the domains with a predicted probability of being *risky* greater than or equal to 0.6 are classified as *risky*.

TABLE 2. Results (Mean and Standard Deviation) for the ML algorithms in Experiment 1.

ML Model	Accuracy	Sensitivity	Specificity
AB	0.80 (0.02)	0.77 (0.07)	0.83 (0.05)
ERT	0.81 (0.01)	0.75 (0.03)	0.87 (0.04)
GB	0.84 (0.01)	0.79 (0.01)	0.89 (0.03)
KNN	0.82 (0.01)	0.79 (0.04)	0.85 (0.03)
LDA	0.72 (0.03)	0.72 (0.05)	0.72 (0.02)
LR	0.89 (0.01)	0.85 (0.01)	0.92 (0.02)
NB	0.68 (0.02)	0.67 (0.06)	0.69 (0.04)
RF	0.82 (0.01)	0.80 (0.01)	0.85 (0.02)
SVM	0.86 (0.01)	0.85 (0.01)	0.87 (0.01)

Thus, the *Machine Learning Model* has been built and tested. Its elements are: the LR algorithm using a predefined weighted combination of individual similarity measures, and the cut-off probability threshold, and the set of domains used to learn the model.

B. VALIDATION OF THE MACHINE LEARNING MODEL

The purpose of the second experiment is to validate the performance of the *Machine Learning Model* built in the first experiment. For this intent, two different validations have been carried out. The first one evaluates the classification of 200 domains predefined as *risky* domains. The second one assesses the performance of classifying 200 domains predefined as *non-risky* domains.

1) RISKY DOMAINS

This first validation has been carried out with 200 domains collected from the *Risky Domains* index of the *Knowledge Base* module. Therefore, all domains are *risky*. The main purpose of this validation is to check if the system labels all entries as *risky* domains, as it should. For this, two modules of the *DOCRIW* framework have been used: the *Host-based Variables Extraction* module and the *Classification* module. In this way, the *Machine Learning Model* is validated. Table 3 shows the prediction label and the probability assigned by the system for an excerpt of these domains.

The system has achieved an *accuracy* of 0.86 for the classification of these 200 *risky* domains. Thus, it can be concluded that the classifier has a good performance. Besides, the *accuracy* when classifying *risky* domains should be similar to the *sensitivity* reached in the first experiment, which is 0.85.

TABLE 3. An excerpt of the classification of *risky* domains from *Risky Domains* index.

Domain name	Predict. Label	Prob. (risky)
1080p-torrents.kickass-torrent.biz	risky	0.99
5movies.to	risky	0.97
ddlvalley.me	risky	0.97
filetram.com	risky	0.84
filmstreaminghd.biz	risky	0.90
foumovies.com	risky	0.90
heroturko.net	risky	0.94
limetorrents.co	risky	0.99
madefittoday.com	non-risky	0.49
putlockers.ws	non-risky	0.46
sipeliculas.com	risky	0.99
sockshare.io	risky	0.98
torrentdownloads.unblocked.live	risky	0.99
uwatchfree.tv	risky	0.98
zooqle.com	risky	0.73

Notice that if the *Domains Extraction and Validation* module would have been included in this experiment, it will reach an *accuracy* value of 1. This is due to all these domains would have been found in the *Risky Domains* index and they would be automatically labeled as *risky*.

Furthermore, some problems were detected with inactive domains in order to generate the host-based variables as the web information sources do not provide information about them. This issue is addressed using only active domains to perform this experiment. In the entire *DOCRIW* framework, it is controlled by the *Domains Extraction and Validation* module. It checks if domains are active or not. In the second case, it labels them as inactive (i.e., no *risky* or *non-risky* label is provided for inactive domains).

2) NON-RISKY DOMAINS

This second validation has been carried out with 200 prestigious domains, all of them previously labeled by experts as *non-risky*. This time, the aim is to check if the classifier labels all these entries as legal domains. Table 4 shows the prediction label and the probability assigned by the system for an excerpt of these domains.

The system has achieved an *accuracy* value of 0.88 for the classification of these 200 *non-risky* domains. Albeit the *accuracy* is lower than the *specificity* of the first experiment (0.9), the results are good enough. In total, 24 domains have been classified as risky. 13 out of these 24 domains have achieved a probability among 0.6 and 0.61, so the classifier is not sure either that these domains are really risky.

Notice that when a domain name contains substrings used in *risky* domain names, they could be classified as *risky*. Especially when host-based variables do not provide additional information. Instances of this issue are *linkedin.com* and *uber.com*, which appear in Table 4. '*link*' and '*ube*' appear in several *risky* domain names used to train the LR

TABLE 4. Classification of non-risky domains.

Domain name	Predict. Label	Prob. (risky)
adidas.es	non-risky	0.02
amazon.com	non-risky	0.04
atleticodemadrid.com	non-risky	0.23
audi.es	non-risky	0.02
bancosantander.es	non-risky	0.01
carrefour.es	non-risky	0.17
disney.es	non-risky	0.22
elmundo.es	non-risky	0.04
facebook.com	non-risky	0.55
google.es	non-risky	0.07
hboespana.com	non-risky	0.40
linkedin.com	risky	0.63
telecinco.es	non-risky	0.01
uber.com	risky	0.62
urjc.es	non-risky	0.01

algorithm. Moreover, both are hosted in California, USA (city and country variables). These host-based values do not aid either to characterize the domains as *non-risky* (i.e., only 35% of the *non-risky* domains used to train the LR algorithm are hosted in USA). However, it is not completely clear either that these domains are *risky*. Thus, their probabilities of being *risky* have been 0.63 and 0.62, respectively. Notice that the cut-off probability is 0.6.

Regarding the inactive domains issue, it is not common to find inactive *non-risky* domains. *Non-risky* domains usually have a long life. Instead, *risky* domains shutdowns are more frequent, due to the illegal activities performed by them. As mentioned above, the *DOCRIW* framework controls this situation.

C. SIMULATION OF THE SYSTEM IN PRODUCTION

This experiment simulates the complete domains classification functionality. This functionality encompasses the *Domain Labeling Process*. Thus, the objective is to execute the complete process to simulate the operation of the *DOCRIW* framework in the production stage. Hence, the modules involved in the experiment are: the *Domains Extraction and Validation* module, the *Host-based Variables Extraction* and the *Classification* modules.

Delving into the experiment, a total of 220 domains that have not been previously considered by the framework (i. e. 100 *risky* domains, 100 *non-risky* domains and 20 inactive domains) have been evaluated. A total *accuracy* value of 0.86 has been achieved.

Regarding the *risky* domains, 81 out of 100 domains have been correctly classified. Five of them have been directly labeled by the *Domains Extraction and Validation* module (e.g., *ugtorrent.com*), as they were stored in the *Risky Domains* index. So the rest of the modules are not considered, and these domains were rightly classified with a probability value of 1. In relation to the *non-risky* domains, 89 out of 100 have been properly classified. This proves that the system is designed to minimize the error when classifying *non-risky* domains. Finally, all the inactive domains have been well classified. They have also labeled by the *Domains Extraction and Validation* module.

TABLE 5. An excerpt of the classification using the modules implied in the *Domain Labeling Process*. The *risky* label with asterisk means that it comes from the *Risky Domains* index.

Domain name	Actual Label	Predict. Label	Prob. (risky)
3hdmovies.com	inactive	inactive	1.00
acmefilm.ee	risky	non-risky	0.43
ariamovie7.site	risky	risky	0.92
bigcinema.tv	inactive	inactive	1.00
canon.es	non-risky	non-risky	0.05
cisco.com	non-risky	non-risky	0.34
edreams.es	non-risky	non-risky	0.08
fnac.es	non-risky	non-risky	0.01
freemovieswatchonline.co	inactive	inactive	1.00
harley-davidson.com	non-risky	non-risky	0.31
hdfilme.tv	risky	risky	0.78
hornyblog.eu	inactive	inactive	1.00
iberia.es	non-risky	non-risky	0.08
marca.com	non-risky	risky	0.61
mobilemoviescorner.com	risky	non-risky	0.58
nvidia.es	non-risky	non-risky	0.04
seedpeer.me	risky	risky	0.73
serviwin.com	inactive	inactive	1.00
sony.es	non-risky	non-risky	0.24
templestowepub.com	risky	risky	0.76
ugtorrent.com	risky*	risky	1.00
usabit.com	risky	risky	0.87
vodafone.es	non-risky	non-risky	0.06
watchonline.red	risky	risky	0.83
xdownload.pl	risky	risky	0.97

Table 5 shows the results for the classification of an excerpt of these domains. It presents the domain name, the actual label, the predicted label and the probability of being classified as *risky* domain.

In conclusion, the *DOCRIW* framework has shown that it provides acceptable results in order to classify domains according to their risk. The *Domains Extraction and Validation* module acts as a filter discarding inactive domains and those that have already been stored in the *Risky Domains* index. This allows to minimize noisy features that have to be evaluated by the other two modules that are part of the *Domain Labeling Process*. Thus, it can be said that *DOCRIW* is a functional prototype to detect and classify possible potentially risky domains.

VI. CONCLUSION

This paper presents a novel framework called *DOCRIW* that classifies domains as *risky* and *non-risky*. For this purpose, web information sources are used to collect specific knowledge from potentially risky domains. This functionality is completed with a ML classifier based on a LR algorithm. The classifier has been trained through 1, 500 labeled domains. Promising results have been accomplished through several experiments carried out to prove the viability of the proposal.

Even though the *DOCRIW* framework represents a complete system, future enhancements can be applied to increase the overall performance.

The database that contains the domains labeled as *risky* (*RiskyDomainsDB*) includes only some domains. Thus, it is mandatory to further improve the predictive module. This module, first calculates the similarity matrices to classify the domains as *risky* or *non-risky*. Six features are used to define six different similarities to measure the similarity between domains. The first one uses the normalized *Levenshtein* distance for domain names. The other five similarity measures are defined from the correspondence between the cities, the countries, the creation dates, the expiration dates, and the emails. The global similarity is produced through a weighted combination of all these individual similarities, where the weights represent the influence of each feature. The first limitation is given by the Levenshtein similarity measure due to it disregards semantics implications. Levenshtein has reached better results than TF-IDF + Similarity Cosine; however, several well-known similarity measures [62] could be tested and compared (e.g., edit distance, Smith-Waterman similarity, Jaro-Winklers similarity, or Monge-Elkan similarity). NLP techniques could also be explored to add the semantic component to the similarity matrices, even though it could not enrich the system. Besides, only five host-based variables have been used to upgrade the similarity due to the rest of them do not provide any information to the models. It is likely that only six variables are not enough, it could be interesting to consider new features that could provide relevant information for the domains.

Regarding the ML algorithms, only 1, 500 domains previously labeled by the experts have been used. This sample size limitation could be mitigated by retraining the LR classifier with those domains with high probability of being *risky* or *non-risky* and reinforcement learning techniques could also be included. Notice that, in the real world, the number of non-risky domains is much bigger than the number of risky domains. In this paper, a set of non-risky domains of the same size that the set of risky domains has been considered. In the future, different approaches to deal with unbalanced problems, such as penalizing misclassification in different ways, will be considered. Finally, further research with other classifiers and configurations, and also ensemble methods would be interesting.

REFERENCES

[1] A. Ali Ahmed, "Malicious Website detection: A review," *J. Forensic Sci. Criminal Invest.*, vol. 7, no. 3, pp. 1–4, Feb. 2018.

[2] S. Abraham and I. Chengalur-Smith, "An overview of social engineering malware: Trends, tactics, and implications," *Technol. Soc.*, vol. 32, no. 3, pp. 183–196, Aug. 2010.

[3] (2011). *Study by Observatory of Information Security*. [Online]. Available: https://www.prevent.es/Documentacion/estudio_fraude_4t10.pdf

[4] OEDI. (2018). *Spanish Observatory of Cyber Crimes*. [Online]. Available: http://oedi.es/estadisticas/

[5] R. Akerkar and P. Sajja, *Knowledge-Based Systems*. Burlington, MA, USA,: Jones & Bartlett Publishers, 2010.

[6] A. Fernández-Isabel, J. C. Prieto, F. Ortega, I. Martín de Diego, J. M. Moguerza, J. Mena, S. Galindo, and L. Napalkova, "A unified knowledge compiler to provide support the scientific community," *Knowl.-Based Syst.*, vol. 161, pp. 157–171, Dec. 2018.

[7] I. H. Witten, E. Frank, and M. A. Hall, *Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2005, p. 578.

[8] J. Eisenstein, *Introduction to Natural Language Processing*. Cambridge, MA, USA: MIT Press, 2019.

[9] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Detection of fraudulent and malicious Websites by analysing user reviews for online shopping Websites," *Int. J. Knowl. Web Intell.*, vol. 5, no. 3, pp. 171–189, 2016.

[10] I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in *Proc. 2nd Int. Conf. Adv. Comput., Control, Telecommun. Technol.*, Dec. 2010, pp. 201–203.

[11] J. T. Goodman, P. S. Rehfuss, R. L. Rounthwaite, M. Mishra, G. J. Hulten, K. G. Richards, A. H. Averbuch, A. P. Penta, and R. C. Deyo, "Phishing detection, prevention, and notification," U.S. Patent 7 634 810, Dec. 15, 2009.

[12] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1245–1254.

[13] D. Chiba, K. Tobe, T. Mori, and S. Goto, "Detecting malicious Websites by learning IP address features," in *Proc. IEEE/IPSJ 12th Int. Symp. Appl. Internet*, Jul. 2012, pp. 29–39.

[14] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Sep. 2018.

[15] M. Kuyama, Y. Kakizaki, and R. Sasaki, "Method for detecting a malicious domain by using WHOIS and DNS features," in *Proc. 3rd Int. Conf. Digit. Secur. Forensics (DigitalSec)*, vol. 74, 2016, pp. 74–80.

[16] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Hoboken, NJ, USA: Wiley, 2011.

[17] R. Devaki, V. Kathiresan, and S. Gunasekaran, "Credit card fraud detection using time series analysis," *Int. J. Comput. Appl.*, vol. 3, pp. 8–10, Feb. 2014.

[18] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, Feb. 2011.

[19] C. S. Hilas, "Designing an expert system for fraud detection in private telecommunications networks," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11559–11569, Nov. 2009.

[20] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," in *Statistical Science*. New York, NY, USA: JSTOR, 2002, pp. 235–249.

[21] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," *Inf. Secur. Tech. Rep.*, vol. 14, no. 1, pp. 16–29, Feb. 2009.

[22] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A neural network based database mining system for credit card fraud detection," in *Proc. IEEE/IAFE Comput. Intell. Financial Eng. (CIFEr)*, Mar. 1997, pp. 220–226.

[23] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, and Q. Kang, "Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection," *Neurocomputing*, vol. 407, pp. 50–62, Sep. 2020.

[24] L. Zheng, G. Liu, C. Yan, and C. Jiang, "Transaction fraud detection based on total order relation and behavior diversity," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 796–806, Sep. 2018.

[25] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

[26] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 569–579, Apr. 2020.

[27] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," 2017, *arXiv:1701.07179*. [Online]. Available: http://arxiv.org/abs/1701.07179

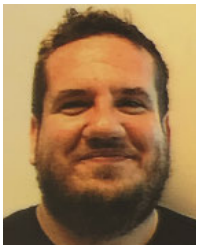
[28] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[29] P. Kolari, R. Finin, A. Joshi, and others, "SVMs for the blogosphere: Blog identification and splog detection," in *Proc. AAAI Spring Symp. Comput. Approaches Analysing Weblogs*, 2006, pp. 1–8.

[30] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, "Detection of malicious Web pages using system calls sequences," in *Proc. Int. Conf. Availability, Rel., Secur.*, 2014, pp. 226–238.

[31] J. Ma, "Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1245–1254.
[32] A. A. Ahmed, A. Jantan, and T.-C. Wan, "Filtration model for the detection of malicious traffic in large-scale networks," *Comput. Commun.*, vol. 82, pp. 59–70, May 2016.
[33] S. Menard, *Applied Logistic Regression Analysis*, vol. 106. Newbury Park, CA, USA: Sage, 2002.
[34] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A fast filter for the large-scale detection of malicious Web pages," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 197–206.
[35] A. A. Ahmed and C. Xue Li, "Analyzing data remnant remains on user devices to determine probative artifacts in cloud environment," *J. Forensic Sci.*, vol. 63, no. 1, pp. 112–121, Jan. 2018.
[36] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen, "Malicious Web content detection by machine learning," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 55–60, Jan. 2010.
[37] A. A. Ahmed, "Investigation approach for network attack intention recognition," *Int. J. Digit. Crime Forensics*, vol. 9, no. 1, pp. 17–38, Jan. 2017.
[38] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of malicious Web pages with static heuristics," in *Proc. Australas. Telecommun. Netw. Appl. Conf.*, Dec. 2008, pp. 91–96.
[39] A. A. Ahmed and N. A. Abdullah, "Real time detection of phishing Websites," in *Proc. IEEE 7th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2016, pp. 1–6.
[40] S. Singhal, U. Chawla, and R. Shorey, "Machine learning & concept drift based approach for malicious Website detection," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 582–585.
[41] N. Davuth and S. Kim, "Classification of malicious domain names using support vector machine and bi-gram method," *Int. J. Secur. Appl.*, vol. 7, no. 1, pp. 51–58, 2013.
[42] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.
[43] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.
[44] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1798–1809, May 2020.
[45] X. Luo, M. Zhou, S. Li, and M. Shang, "An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2011–2022, May 2018.
[46] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.
[47] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. New York, NY, USA: Springer, 2001.
[48] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. Newton, MA, USA: O'Reilly Media, 2015.
[49] Whois.com. (2019). *Whois Domain Information*. Accessed: Jan. 24, 2020. [Online]. Available: https://www.whois.com/whois/
[50] International Organization for Standardization. (2019). *Country Codes–ISO 3166*. Accessed: Jan. 24, 2020. [Online]. Available: https://www.iso.org/iso-3166-country-codes.html
[51] D. Sarkar, *Text Analytics With Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*. New York, NY, USA: Apress, 2016.
[52] db.aa419.org. (2019). *Fake Sites Database*. Accessed: Jan. 24, 2020. [Online]. Available: https://db.aa419.org/fakebankslist.php
[53] MalwareURL.com. (2019). *Malware URLs Database*. Accessed: Jan. 24, 2020. [Online]. Available: https://www.malwareurl.com/
[54] Y. Freund, R. E. Schapire, and N. Abe, "A short introduction to boosting," *J. Japn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, 1999.
[55] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers Neurorobot.*, vol. 7, p. 21, Dec. 2013.
[56] J. M. Moguerza and A. Muñoz, "Support vector machines with applications," *Stat. Sci.*, vol. 21, no. 3, pp. 322–336, 2006.
[57] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. New York, NY, USA: Springer, 2013, p. 18.
[58] P. Lameski, E. Zdravevski, R. Mingov, and A. Kulakov, "SVM parameter tuning with grid search and its impact on reduction of model over-fitting," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Cham, Switzerland: Springer, 2015, pp. 464–474.
[59] M. Gruber, *Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators*. Evanston, IL, USA: Routledge, 2017.
[60] X.-X. Zhang, Y.-M. Wang, S.-Q. Chen, J.-F. Chu, and L. Chen, "Gini coefficient-based evidential reasoning approach with unknown evidence weights," *Comput. Ind. Eng.*, vol. 124, pp. 157–166, Oct. 2018.
[61] W. Zhu, N. Zeng, and N. Wang, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations," in *Proc. NESUG: Health Care Life Sci.*, Baltimore, MD, USA, vol. 19, 2010, pp. 1–9.
[62] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string metrics for matching names and records," in *Proc. KDD Workshop Data Cleaning Object Consolidation*, vol. 3, 2003, pp. 73–78.

JUAN CARLOS PRIETO was born in Madrid, Spain, in 1988. He received the dual master's degree in data science and in telecommunications engineering. He is currently pursuing the Ph.D. degree in statistics and decision science with Rey Juan Carlos University (URJC). He worked as a Researcher with the Data Science Laboratory, URJC, and participated in several projects. Since 2018, he has been managing a software start-up developing AI SaaS products for the automotive sector. His research interests include machine learning, natural language processing, and similarity metrics in various application domains, including sentiment analysis, knowledge-based systems, and reputation and recommender systems.



ALBERTO FERNÁNDEZ-ISABEL was born in Toledo, Spain, in 1984. He received the Ph.D. degree in computer science from the Complutense University of Madrid (UCM), in 2015, and the master's degree in artificial intelligence. He received a scholarship from the Spanish National Research Council (CSIC), as a Technical Assistant. He worked for several years in European and national projects as a Predoctoral Researcher and a Postdoctoral Researcher. Since 2019, he has been an Assistant Professor with the Higher Technical School of Computer Engineering (ETSII), Rey Juan Carlos University (URJC). He has authored more than 30 scientific articles and one book. His research interests include intelligent agents, machine learning, data visualization, and natural language processing in various application domains, including distributed programming, sentiment analysis, agent-based collaboration and negotiation, smart cities, and simulations.

ISAAC MARTÍN DE DIEGO was born in Campaspero, Valladolid, Spain, in 1973. He received the Ph.D. degree in mathematical engineering from the Carlos III de Madrid University, in 2005. In 2018, he was an Associate Professor with the Higher Technical School of Computer Engineering, Rey Juan Carlos University, where he has been a Professor since 2006. He is currently the Co-Founder of the Data Science Laboratory and the Head of the Master in data science with Rey Juan Carlos University. He is also the Head of the ERICSSON Chair on data science applied to 5G. He has authored more than 100 articles. His research interests include methods, processes, and tools for data science in various application domains, such as artificial vision, opinion mining, security, and biostatistics, with special interest on machine learning algorithms and combination of information methods. In 2005, he was a recipient of the Extraordinary Doctorate Award from the Carlos III de Madrid University.

FELIPE ORTEGA was born in Cartagena, Murcia, Spain, in 1980. He received the Ph.D. degree in computer science and mathematical modelling from Rey Juan Carlos University, in 2009. He was an Invited Speaker with Georgia Tech, Atlanta, USA, Xerox PARC, California, USA, and the Cervantes Institute, Madrid, Spain. Since 2016, he has been an Assistant Professor with the Higher Technical School of Telecommunications Engineering (ETSIT), Rey Juan Carlos University. He is currently the Co-Founder and the Coordinator of data engineering with the Data Science Laboratory, Rey Juan Carlos University. He has authored three books and more than 35 articles. His research interests include methodologies and tools for the practice of data science, and data engineering and machine learning with applications in various domains, including text mining, sentiment analysis, cybersecurity, oil and gas, livestock management, software engineering, and open online communities.

JAVIER M. MOGUERZA was born in Granada, Spain, in 1972. He received the Ph.D. degree in mathematical engineering from the University Carlos III of Madrid (UC3M). He worked with the Carlos III University of Madrid and the Pontificia Comillas University of Madrid (ICAI-ICADE). From December 2010 to May 2016, he was an Academician with the Global Young Academy (GYA). From September 2016 to September 2019, he was responsible for the Ericsson Institutional Chair on Data Science applied to 5G with Rey Juan Carlos University. In 2019, he was the Founder Academician of the Young Academy of Spain, created by the Spanish Government. He is currently a Full Professor with Rey Juan Carlos University. He also belongs to the Alumni of the Global Young Academy. His research interests include operations research, such as six sigma quality and optimization of resources, the design of machine learning methods, and data science.