International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-018-00906-1

ORIGINAL ARTICLE

Review: machine learning techniques applied to cybersecurity

Javier Martínez Torres[1] · Carla Iglesias Comesaña[2] · Paulino J. García-Nieto[3]

Received: 20 October 2017 / Accepted: 18 December 2018

Abstract
Machine learning techniques are a set of mathematical models to solve high non-linearity problems of different topics: prediction, classification, data association, data conceptualization. In this work, the authors review the applications of machine learning techniques in the field of cybersecurity describing before the different classifications of the models based on (1) their structure, network-based or not, (2) their learning process, supervised or unsupervised and (3) their complexity. All the capabilities of machine learning techniques are to be regarded, but authors focus on prediction and classification, highlighting the possibilities of improving the models in order to minimize the error rates in the applications developed and available in the literature. This work presents the importance of different error criteria as the confusion matrix or mean absolute error in classification problems, and relative error in regression problems. Furthermore, special attention is paid to the application of the models in this review work. There are a wide variety of possibilities, applying these models to intrusion detection, or to detection and classification of attacks, to name a few. However, other important and innovative applications in the field of cybersecurity are presented. This work should serve as a guide for new researchers and those who want to immerse themselves in the field of machine learning techniques within cybersecurity.

Keywords Cybersecurity · Detection systems · Internet threats · Machine learning · Security

1 Introduction

Internet has become an essential resource for people: in 2014, about 40% of the world's population uses the Internet and this figure increases up to 78% in the developed countries [1]. The North Atlantic Treaty Organization (NATO) identifies the internet as "a critical national resource for governments, a vital part of national infrastructures, and a key driver of socio-economic growth and development" [2]. Associated to the spread of Internet usage, malicious code and software have appeared to compromise computer systems, attacking and destroying the information they contain [3]. This type of attacks are designed to gather users' information such as credit card numbers or passwords, but also for distributing information without the user's consent [3].

Malware is defined as software capable of damaging data and systems [4]. It is a threat not only for the individuals but also to organizations, companies and even governments, including both civil and military infrastructures [5], that are at risk of losing valuable information as well as their reputation [6]. Many examples can be found in recent years involving the steal of credit and debit cards from Web payment systems, the steal of part of Google's intellectual property, or the exposure of users personal information, to name a few [6]. Another essential sector is the power sector, a target to cyber-attacks whose security has also been regarded (see [7–9] and references within).

However, if there is a cyber-attack that should be highlighted, it is the attacks suffered by Estonia in 2007. For 3 weeks, Estonia experienced what is considered the first cyberwar provoked by the removal of a Soviet monument erected in Tallin in 1947. The target were the websites of different Estonian organizations such as banks, universities or newspapers. This first cyberwar lead to the announcement of a Policy on Cyber Defence as part of the NATO Bucharest Summit Declaration in 2008 [10]. Since then, NATO nations have participated in multinational projects to enhance their cyber defense capabilities, and the protection

✉ Javier Martínez Torres
javier.martineztorres@unir.net

[1] Universidad Internacional de la Rioja, Logroño, Spain

[2] University of Vigo, Vigo, Spain

[3] University of Oviedo, Oviedo, Spain

Published online: 04 January 2019

of the communications and information systems of the Alliance is regarded as a priority.

In the case of malware, these software have the ability of updating and adjusting in a manner that they avoid their detection, thus needing security systems capable to automatically learn from experience [11]. The cost of this type of attack is difficult to calculate, since many implicit costs are associated, but some authors have estimated it as 0.2–0.4% of global GDP of the country [12]. According to the NATO, G20 economies have lost about 2.5 million jobs due to counterfeiting and piracy, and governments and consumers $125 billion per year [2].

Different definitions are to cybersecurity but one of them it is the definition of Kaspersky lab: *Cyber security is the practice of defending computers and servers, mobile devices, electronic systems, networks and data from malicious attacks. It is also known as information technology security or electronic information security. The term is broad-ranging and applies to everything from computer security to disaster recovery and end-user education.*

The term “cybersecurity” has been formal defined by the ISO/IEC 27032:2012 as “the preservation of confidentiality, integrity and availability of information in the Cyberspace” (the so-called CIA Principle) [13]. It includes other concepts represented in Fig. 1. These concepts are defined by the ISO as follows:

- Information security: “concerned with the protection of confidentiality, integrity, and availability of information in general, to serve the needs of the applicable information user”.
- Network security: “concerned with the design, implementation, and operation of networks for achieving the purposes of information security on networks within organizations, between organizations, and between organizations and users”.
- Internet security: “concerned with protecting internet-related services and related ICT systems and networks as an extension of network security in organizations and at home, to achieve the purpose of security. Internet Security also ensures the availability and reliability of Internet services”.
- Critical information infrastructure protection: “ensures that those systems and networks are protected and resilient against information security risks, network security risks, internet security risks, as well as Cybersecurity risks”.
- Cybercrime: “criminal activity where services or applications in the Cyberspace are used for or are the target of a crime, or where the Cyberspace is the source, tool, target, or place of a crime”.
- Cybersafety: “condition of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event in the Cyberspace which could be considered non-desirable”.
- Finally ICT security has not been defined by the ISO, but is usually referred to the technical origins of computer security and the CIA principle.

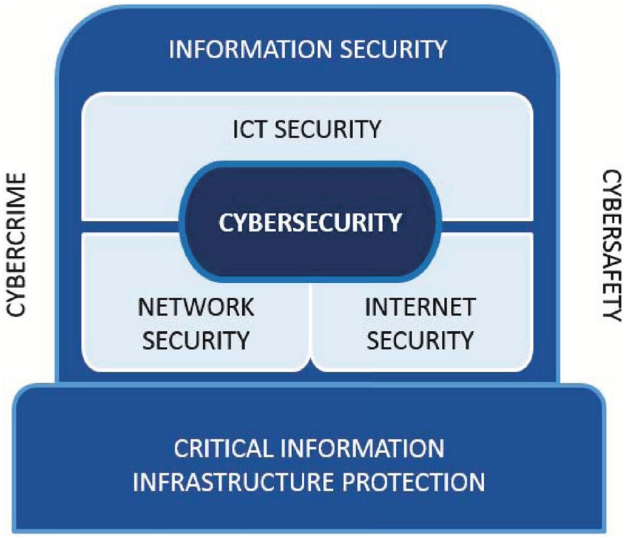

Fig. 1 Cybersecurity and other security domains [2]

Cybersecurity has to protect personal, governmental and business data from misuse or manipulation by other people, focusing on three main tasks: (a) taking measures to protect equipment, software and the information they contain, (b) guaranteeing the state or quality of being protected from the several threats; and (c) implementing and improving these activities [14].

In recent years, many are the organizations and projects that have been created with the aim of facing these threats. One of them is the Open Web Application Security Project (OWASP), an international non-for-profit charitable organization that focuses on the application security [15]. They identify a series of software vulnerabilities and describe the ten most important in their top ten project, whose latest report was published in 2013 and included the following security risks [16]: injection, broken authentication and session management, cross-site scripting (XSS), insecure direct object references, security misconfiguration, sensitive data exposure, missing function level access control, cross-site request forgery (CSRF), using components with known vulnerabilities, and unvalidated redirects and forwards. With similar philosophy, Microsoft’s Security Development Lifecycle offers developers information and tools to build secure software [17].

So far, a descriptive analysis has been made of the different ways, from the legal point of view, that cybersecurity can present. Next, in the following sections, a more detailed analysis of a multitude of applications of machine learning techniques will be carried out within the different areas. That is why this work is considered a good starting point for the knowledge of this field of application of machine learning techniques. However, it is recommended to delve much more following the bibliography in each of the aspects of interest since the depth and quantity of bibliography destined for this purpose is abundant and it has been impossible to collect it all in this work, highlighting in it that which has been considered more relevant and topical.

2 Machine learning

All the capabilities of machine learning techniques are to be regarded in order to improve the models, considering many factors such as computation time, actualization capability and complexity. Depending on the application, the priority may vary. Furthermore, different performance criteria are considered beyond the error rate, since other indicators such as have shown several advantages.

The capacity of automatic classifiers to correctly identify malware has been tested, tackling also the false-positives cases with successful results using classifiers based on perceptron [18–20]. Machine learning techniques have been used for developing threat-detection systems, using Bayesian regularized neural networks, Naive Bayes, Bayesian classifiers, support vector machines (SVM), neural network classifiers or self-organization maps ([21] and references within, [3]) to name a few. Goseva-Popstojanova et al. [22] conclude that machine learning techniques such as SVM and decision trees can successfully distinguish attack Web sessions. Fuzzy logic and neural networks have been successfully combined for malware detection, studying the most important API calls [4].

Phishing attacks are a particular crime that obtain personal information from users by fraudulent web sites, and is the most common method for the identity theft [23]. Machine learning approaches have been used to identify the authorship of the phishing attack [24] and to detect phishing e-mails comparing different machine learning techniques [25].

2.1 Summary of machine learning techniques

This subsection includes the mathematical description of different machine learning techniques used for the detection and management of software attacks. The following aspects are commented:

- Their structure: network-based or not.
- Their learning process: supervised or unsupervised.
- Their complexity.

According to the above mentioned features, the different machine learning models are classified based on their learning strategy:

1. Unsupervised learning. These methods are usually used in the so-called exploratory data analysis, when the natural groups to be found are not known beforehand and a big dataset is to be analysed. They are also commonly used when the classes are known beforehand and we want to validate the training process and the sets of variables chosen. The best known algorithms are the one used for clustering (k-means algorithm [26]) and Kohonen's self-organizing maps [27]. This type of learning might have different objectives such as grouping, generation of hierarchies, dimensionality reduction or interpretation and visualization.
2 Supervised learning. It is a type of automatic learning where the algorithm used is provided with a number of examples with their corresponding answers, that is to say, the model is created using its output. The most used supervised methods are decision trees, SVM and artificial neural networks (ANN) in its most popular version multilayer perceptron (MLP).

2.1.1 Clustering

The aim of cluster analysis is to group items into homogeneous groups based on the similarities between them. As this is an unsupervised method, groups must be determined without prior information about the classes, and will be deduced exclusively from the information of the data. Thus, the objectives of clustering are briefly the following:

- To explore the data.
- To reduce or simplify the data or their variables.
- To formulate statistical hypothesis that could be checked afterwards.
- To predict from the different groups.

There are different methods or approaches classified into hierarchical and partitional. Hierarchical methods are based on establishing a hierarchy among the clusters, that is to say, they provide a series of consecutive partitions where each partition is obtained joining or dividing clusters [28]. On the other hand, partitional methods rely on reaching the optimal partition of an unknown distribution in the input space by forming a defined number of regions (clusters) based on a certain similarity measure, so the items belonging to each particular group can be represented by a single point (the centre of the cluster). The most known algorithm of this kind of methodology is the k-means algorithm.

The algorithm divides a set of n vectors into k groups and its aim is to find the centres of the clusters by minimizing an inequality function based on the distances between each point and its centroid [29]. Hence, the algorithm follows these steps:

1. Set the initial centroids.
2. Determine a logic matrix that indicates the position of each point in a group.
3. Calculate the inequality function and minimize it.
4. Calculate the new centroids.

The proper operation of this algorithm depends on the initial positions of the centroids [30], so it does not guarantee an optimal solution.

2.1.2 Self-organizing maps

Kohonen [27] presented a model of competitive neural network capable of forming feature maps through a matrix organization of artificial neurons. It generates a typological map to place in an optimal way a fixed number of vectors in an input space of greater dimension, thus making data easier to understand [31]. Self-organizing maps (SOMs) are a popular non-linear model of unsupervised neural network for the solution of dimensionality reduction problems. SOM's learning algorithm follows these steps:

1. Random selection of the input patterns and present it to the network.
2. Calculation of the distances of the map of neurons generated initially and determination of the closest neuron to the input vector (minimum distance).
3. Update of the weights of the winner neuron and its neighbours from a topological point of view.
4. Repetition of the previous steps until the selected stop criterion of the method is satisfied.

2.1.3 Classification and regression trees (CARTs)

Decision, classification or identification trees are one of the most outstanding unsupervised learning models. Its main virtue is it is a simple classification model, easy to understand and to represent graphically. Decision trees' simplicity makes them an appealing alternative to the final user of a knowledge extraction system. The aim of the process of construction of decision trees is to obtain a tree which reveals interesting information to make predictions. Decision trees are constructed recursively through an induction process with a top-down strategy, from general concepts to particular examples. This is why the acronym TDIDT ("Top-Down Induction on Decision Trees") is used to refer to the family of algorithms for constructing decision trees. TDIDT family of algorithms includes classical ones such as ID3 [32], C4.5 [33] or CART [34]. A key aspect regarding classification trees is how to divide each node and pruning rules that indicate until which level we want to deepen into the classification rule [35].

2.1.4 Random forest

An extension of CARTs are random forests or random decision forests because they are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. There are in the literature different applications of this technique as for example in web user identification [36], intrusion detection [37] or determine spam volume [38].

2.1.5 Support vector machines

Vapnik [39] is considered as the pioneer in introducing the concept of optimum separating hyperplane of a dataset in a classification problem, concept upon which SVM lie. Separating hyperplanes have two main weaknesses: the need of linear separability of the sample and its lineal character. Different key milestones can be outlined in the development of SVM networks:

- The generation of a feature space from the input space by means of a transformation. Through inverse transformation, the linear frontiers of the separating hyperplanes give rise to non-linear frontiers in the input space. This is known as the Kernel Trick method.
- The appearance of the Soft-Margin algorithm for problems where the perfect separability is of no interest (e.g. problems with noise among the observations).
- The generalization of SVMs to regression problems through Vapnik's ϵ-insensitive loss [40].

In order to use SVMs in an efficient, effective way, it is necessary to bear in mind two key aspects:

- The training algorithm. For this purpose, several authors have developed many algorithms [41–43].
- Hyperparameter-selection methods. There are different approaches that can be used based on the concept of VC-dimension [44].

2.1.6 Neural networks

Neural networks models come from the biological model of neurons back in the forties, based on the work of psychiatrist McCulloch and mathematician Pitts [45]. Applications of this kind of models are very large ([46] and its references). This model consists of two main elements:

- A structure made of a set of basic units, called neurons, organised in layers. The network has three layers: an input layer, a hidden layer and an output layer. Each neural unit has the following components:
  - A set of input connections together with a set of weights that regulate the intensity of each of the input signals.
  - A value (the activation threshold) that is subtracted from the aggregation of input signals transmitted.
  - An activation function that acts upon input signals.
  - The output of the neuron as a function of input signals, called transfer function.

  This structure is usually called network architecture, and we can classify the networks based on the number of layers, the interconnection degree of the structure or the character of the connections.
- A training or learning algorithm calibrates the weights of the network and the other parameters based on the deviations between the outputs given by the network and the real values.

The typical training algorithm for MLP is known as back-propagation algorithm. This back-propagation approach intends to calculate the derivatives of the target function with respect to the parameters in an efficient way. Many authors [31, 47, 48] refer to and develop the algorithm. An important step to be considered regarding the training algorithm is the initialization. The common method is the one proposed by [49], who proposes, basically, choosing the initial values of the weights so the domain of the input space is distributed to the outputs of the network. Furthermore, the consistent character of the algorithm has to be noted, even achieving convergence rates and demonstrating its asymptotic normality.

2.1.7 Other models

New models have been emerging within machine learning techniques such as fuzziness-based learning [50–53]. As well as the methods described previously, different classical and modern models have to be regarded since they are widely used in several aspects of cybersecurity, as described in this chapter. These models or approaches are:

- Bayesian approach. Bayesian classifiers.
- Markov models.

2.1.8 Deep learning

Machine learning and deep learning are so close but it is a difference between them. Deep learning is a new field in machine-learning research. Its motivation lies in the establishment of a neural network that simulates the human brain for analytical learning. It mimics the human brain mechanism to interpret data such as images sounds and texts [54]. As a new technique, there are a lot of studies for these model in cybersecurity as for example: anomalies detection in 5G networks [55], distributed attack detection in fog-to-things computing [56] or classification of malware programs [57]. It is recommended to extend the applications revise the paper of Xin et al. [58].

2.2 Error criteria

Many researches demonstrate the importance of this analysis, since error rate criterium is not always the key factor. It is necessary to consider more factors such as computation time, actualization capability and complexity, taking into account that the priority could be different in each application. When it comes to classifying incoming e-mails, for instance, a new error criteria arises: false positives and false negatives rates. Especially important are false positives, since the consequences and losses of information due to the misclassification of an e-mail. Hence, cybersecurity methods are usually assessed according to the following terms:

- True positive (TP): number of harmful applications correctly classified.
- True negative (TN): number of benign applications correctly classified.
- False positive (FP): number of benign applications misclassified as harmful. It is regarded as the main drawback of these classification methods [59].
- False negative (FN): number of harmful applications misclassified as benign.

  Bearing in mind these definitions, a number of measures are regarded [60, 61]:
- Precision: percentage of correct positives over the total number of positives identified (Eq. 1).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- Recall (or sensitivity): percentage of positive items correctly identified as harmful over the total (Eq. 2).

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- Accuracy: percentage of correct predictions (positive and negative) (Eq. 3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- False alarm rate: benign instances incorrectly classified over the total number of benign samples (Eq. 4).

$$False\ alarm\ rate = \frac{FP}{TN + FP} \tag{4}$$

- Miss rate: harmful instances incorrectly classified over the total number of harmful samples (Eq. 5).

$$Miss\ rate = \frac{FN}{FN + TP} \tag{5}$$

- Error rate: incorrectly classified instances over the total (Eq. 6).

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{6}$$

- False positive ratio: false positives over the total number of positives identified (Eq. 7).

$$False\ positive\ ratio = \frac{FP}{TP + FP} \tag{7}$$

- False negative ratio: false negatives over the total number of negatives identified (Eq. 8).

$$False\ negative\ ratio = \frac{FN}{TN + FN} \tag{8}$$

- F-Measure: from precision and recall, this parameter measures the accuracy of the method according to Eq. (9).

$$F - Measure = 2 * \frac{precision * recall}{precision + recall} \tag{9}$$

- Total cost ratio: the cost of misclassified instances is calculated according to Eq. (10), where λ is the relative cost of both errors.

$$TCR = \frac{FN + TP}{\lambda(FP + FN)} \tag{10}$$

- Weighted error: it is calculated using a specified weight λ (Eq. 11).

$$WErr = \frac{\lambda TN + TP}{\lambda FP} \tag{11}$$

- ROC curve: TP rate is plotted against FP rate.

3 Cybersecurity threats

Many are the threats related to computer security and fast is their development. Nonetheless, phishing and malware can be identified as the most important threats, for example to detect malicious executables in the wild [3] or phishing counter measures and their differences [23]. Phishing is aimed to get personal information fraudulently by impersonating a trustworthy person or entity [62]. Using web pages that look like real web pages, users are cheated to visit them (usually by clicking on links sent to them by email) and enter their personal information [23, 62]. Other phishing techniques can be found in [63].

Regarding malware, it comprises three main categories depending on how they spread through the cyberspace: viruses, Trojan horses and worms [3]. Viruses can be found in infected executables or in virus loaders, and affect to existing programs that infect other programs when executed. Trojan horses perform malicious functions despite appearing as benign programs. And worms are self-contained programs able to propagate over a network thanks to vulnerabilities of the system. Their objective is usually the same: to cause harm to computer systems, to destroy the information, and even to gather personal information or to distribute information without the user's knowledge [3].

A third major concern regarding cybersecurity is spam, defined as "an e-mail message that is unwanted" [61]. Spam e-mails can be not only a time-consuming task for recipients but a source of Java applets that may execute automatically when the message is read [64].

Apart from the above mentioned threats, SANS Institute identifies the following malicious spyware actions as the most frequent, malicious activities [65]: (1) changing network settings, (2) disabling antivirus and antispyware tools, (3) turning off the Microsoft Security Center and/or automatic updates, (4) installing rogue certificates, (5) cascading file droppers, (6) keystroke logging, (7) URL monitoring, form scraping and screen scraping, (8) turning on the microphone and/or camera, (9) pretending to be an antispyware or antivirus tool, (10) editing search results, (11) acting as a spam relay, (12) planting a rootkit or altering the system to prevent removal, (13) installing a bot for attacker remote control, (14) intercepting sensitive documents and exfiltrating them, or encrypting them for ransom, (15) planting a sniffer.

4 Machine learning applied to cybersecurity

Many methods and procedures have been developed for the detection of threats in the cyberspace. In the case of malware, commercial software (antivirus) can be used with good results, although their mechanisms can be disabled by viruses, whose evolution and improvement is usually faster than the development of malware detection software [3]. This rapid evolution of the threats led to the adoption of learning methods for the detection of unknown malware, including a variety of machine learning techniques.

Taking Scopus database as the main source of this literature review, we can describe quantitatively the state-of-art of machine learning techniques applied to cybersecurity: in June 2015, the search "machine learning AND Spam" gave 473 results; "machine learning AND malware", 326 results; finally, "machine learning AND phishing", 94 results. In view of such numbers, these three threats are regarded as the most important ones (or the most studied), so the subsequent sections will focus on them.

4.1 Spam detection

Spam is usually related to incoming e-mails, but other targets can be identified such as blogs [66, 67], search engines [68–70] and even tweets [71]. The detection of spam is based on the use of filters that analyse the content and decide whether or not they are spam or legitimate messages, blogs or websites.

The first filters, which were user-defined, were easily dodged by spammers with content "obfuscation" [72]. The adoption of machine learning techniques improved the detection of spam, and several methods have been developed in recent years. Guzella and Caminhas made an exhaustive review in [73] in 2009, but many more research works have been published since then in this matter.

Any algorithm aimed at detecting spam must address the following characteristics [74]: (1) changing class distributions, (2) message-misclassification costs, (3) complex text patterns, (4) changing target and (5) intelligent adaptive adversaries. Two main strategies can be followed to detect spam: (1) textual analysis and (2) image-based analysis. Textual analysis was the original methodology and is basically a text categorization problem [73]. As spam messages evolved and involved images with embedded texts, it was necessary to perform image-based analysis as well [73].

Filters have a common structure: first, they extract the key words (tokens) of the message, blog, etc. and reduce them to their root forms; then, common and irrelevant words are deleted, and the resulting set of words is the input of the classifier [73]. The different classifiers will be discussed in the following paragraphs.

Bayesian classifiers are widely used and consider the probabilities of a message being spam or not. These classifiers usually rely on a bag-of-words (BoW) model, a simplified representation of words used in messages widely applied to document classification. Two are the main algorithms: Sahami et al. [75] set the basis of spam filtering with Bayesian classifiers back in 1998; later, Graham [76] improved the classifier selecting the most relevant features online before performing the classification.

Wang et al. [77] developed a spam filter with good results in e-mail classification using information gain algorithm [78] to select the words in the messages. Almeida et al. [79] tested the performance of Naive Bayes classifiers and how dimensionality reduction affects them, concluding that the selection of attributes and terms is critical in the training stage, and that Boolean Naive Bayes and Basic Naive Bayes classifiers achieved best performance. Naive Bayes' tractability, easy implementation, speed and competitive performance are the features that make this technique one of the most popular [80]. Cascaded models combining Naive Bayes classifiers with other classifiers in a tree or chain-based structure have been proposed ([80] and references within). This kind of structure uses several consecutive binary classifications, thus simplifying multi-class problems.

Despite Bayesian classifiers being the most popular for spam detection, other machine learning approaches are found in the literature. Support vector machines have been applied to this matter since 1998 [61]. Since then, different research works have proven SVM are appropriate for spam e-mail filtering through text classification approaches [81, 82]. Amayri and Bouguila performed in [81] a thorough study on spam filtering using SVM with different kernels, making clear the computational cost of some kernels and their high training time. In view of these drawbacks, Caruana et al. [83] proposed an SVM algorithm to perform scalable spam filtering and reduce the training time of this technique. Ontology semantics and a distributed computing framework called MapReduce [84] are used to split the training set and improve the accuracy and training stage of the algorithm.

A different approach is presented by Wu [85]. Wu states that spam-detection methods are usually based on the comparison of keywords frequently used by spammers, but the improvement of spam e-mails, sometimes tailored by advanced programs that write them as ham messages, makes their detection increasingly difficult. To tackle this problem, Wu and Tseng and Wu [86] propose the concept of "spamming behaviors" to characterize spam messages and detect them. These "spamming behaviors" have the advantage of being more resistant with respect to the change of time, unlike specific keywords.

In [85], back-propagation neural networks were used for spam filtering with satisfactory results, showing very low misclassification rates and high accuracy.

Image-based spam has also been studied as it is the subsequent weapon of spammers that makes text-analysis tools ineffective [73, 87, 88]. Biggio et al. [89] a set of computer vision and patter recognition techniques applied to the detection of spam messages embedded in images. Other recent studies presenting image-spam detection methods can also be found [90–94].

Spammers have found other venues apart from e-mails for spreading their spam contents. Spam blogs ("splogs") were created with the aim of attracting traffic from blog search engines in order to promote splogs [67]. Several machine learning techniques such as Naive Bayes, neural networks and SVM were tested in [95]. Neural networks were also successfully used in [96]. However, SVM is the most popular technique and was used to detect spam blogs by Abu-Nimeh and Chen [97], Kolari et al. [98], Yoshinaka et al. [99], Sculley and Wachman [100], to name a few.

The spread of social networks in recent years has made them a target for spammers, being Twitter the most attacked one due to its fast growth [101]. Machine learning methods have been applied to spam detection in tweets: McCord and Chuah [101] used random forests classifier [102] (combined tree predictors that depend on the values of an independent random vector) to distinguish spammers from legitimate users; fuzzy K-means algorithm was used to detect spam tweets based on trending topics [103]; Chu et al. [104] designed a system to classify Twitter users that includes a Bayesian classifier to detect text patterns; decision trees, neural networks, SVM and a Bayesian classifier were also used to identify spam bots on Twitter, showing the latter the best performance [105, 106]; Martinez-Romo and Araujo [71] tested decision trees, Naive Bayes, Logistic regression, SVM and random forest in spam filtering tasks in tweets, being SVM the best in performance; similar methods were used [107], where random forest outperformed Bayesian networks, decision trees, k-nearest neighbor and SVM; Zangerle and Specht used SVM to detect hacked Twitter accounts [108].

Spam based on videos is also present in social networks such as YouTube. The detection of video spammers has been studied using SVM as a first approach to classify spam or legitimate users based on some attributes chosen manually [109]. Later, lazy associative classifier (LAC) was tested for the same task with slightly better results [110]. Other authors approach this issue by training SVM on a certain number of collections of nearest neighbors, considering content, individual and social attributes [111].

4.2 Malware detection

The detection of malicious code in its different forms has been approached by means of several machine learning techniques, analyzing code patterns and similarities. The first research works found in this matter in Scopus database were published in 2005 and used an unsupervised machine learning system to examine the accesses to a computer [112] and Markov models to study the spread of malware [113]. Later, clustering algorithms were applied to classify software as malware [114–116], and SVM were tested in computer systems [117] as well as in Symbian-OS mobile handsets [118].

Some methods are based on the concept of n-grams, contiguous sequences of n items from a text used in a variety of research fields such as natural language processing and computer science. Using n-grams to extract features from files or messages, the results are then used for classification tasks based on different machine learning methods [119–140].

Shabtai et al. published in 2009 a state-of-the-art survey on machine learning classifiers based on static features, including decision trees, ANN, Naive Bayes, SVM, boosting methods and other classifiers such as k-nearest neighbor classifier [131]. Later, one of the key research works used Naive Bayesian classifiers to detect malicious JavaScript code [141]. Naive Bayes model was successfully used in subsequent works such as [142–144].

Recently, considerable efforts have been made in malware detection and SVM method has been widely applied with satisfactory results [119, 122, 140, 145–153]. Biffio et al. completed a thorough study on SVM for security applications such as malware detection, and identified the main harmful actions that the classification algorithm could suffer, namely poisoning (misleading the algorithm), evasion (evading the detection) and privacy breaches (obtaining information from the internal parameters) [154].

Apart from Naive Bayes and SVM, other machine learning methods have been applied such as decision trees, successfully applied to malware detection in [155], where outperformed SVM, Naive Bayes and MLP, as well as in [21, 156–159]. Regarding clustering, it was used to detect and classify malware in [160–162]. The performance of Naive Bayes, decision trees and k-nearest neighbor was also jointly tested in [163]. Boosted Bayesian networks had the highest accuracy and lowest false positive rate in Android's application files in [164] when compared to decision trees, Naive Bayes, PART, Bayesian networks, boosted decision trees and random forest.

Malware is not only a threat for computer systems but for smartphones, especially those which run on Android OS. Third-party applications can infect smartphones and affect them very similarly as they affect computers, so many efforts have been made since 2010 to prevent malware in this kind of device. Decision trees were tested in a number of research works [165–173] with good results. Naive Bayes method was also tested in [174, 175], performing better than other classification methods such as decision trees, Bayesian networks or SVM in [137, 176]. Bayesian networks outperformed SVM, decision trees and k-nearest neighbor in [177], while k-nearest neighbor gave better results than the other approaches in [178]. SVM had a good performance detecting android malware in recent works [118, 179–182]. However, Sheen et al. [183] obtained the best results when using a multifeature collaborative decision fusion that comprised SVM, Naive Bayes and decision trees classifiers.

Finally, Allix and Bissyandé studied the performance of SVM and decision trees both in the laboratory and "in the wild" and concluded that android malware detectors had poor overall performance in the wild, unlike in the laboratory tests [184]. They identified some parameters that may explain this difference such as the size and quality of training sets, and concluded that validation scenarios should be carefully chosen and training data should include a cleaned goodware set. These authors studied the influence of chosen malware datasets for training detectors in [185].

4.3 Phishing detection

Phishing attacks were especially noticeable in January 2006, when a record number was reported [186]. Before then, few studies can be found applied to phishing, but in 2007 two main research works were published focused on phishing e-mails detection with machine learning techniques: [25, 186]. In the first one, Fette et al. used a machine-learning based classification approach called PILFER, which is based on random decision trees, with high accuracy and low false negative and positive rates. In the second one, Abu-Nimeh et al. compared some machine learning methods namely logistic regression, decision trees, SVM, random forests and neural networks with inconclusive results: random forests had the lowest error rate, but logistic regression gave the lowest false positive rate and weighted error rate. Furthermore, they supported the conclusion of Zhang and Yao [187]: the analysis of e-mail headers improves the performance of classifiers.

SVM method has been applied in many occasions since then: SVM classified phishing webs in [188, 189], detected phishing URLs in [190] or detected phishing e-mails in [191]. Decision trees gave good results in several studies: Ma et al. [192] used neural networks (MLP), Naive Bayes, random forest, SVM and decision trees to detect phishing e-mails from content, orthographic and derived features, and decision trees outperformed the other methods; Lakshmi and Vijaya [193] obtained better classification results with decision trees than with MLP; they classified phishing e-mails in [194]; decision trees and neural networks performed well in [195].

Other classifiers such as Bayesian have hardly been used for filtering phishing e-mails with good results [196], since their results were not usually noticeable compared to those of SVM and decision trees, basically. However, Bayesian networks outperformed SVM and decision trees in [197], and a Naive Bayes classifier was combined with clustering in a hybrid model with good results in [198].

Recently, Almomani et al. [60] published a survey on the different techniques for phishing e-mail filtering, including not only different classification approaches but also other measures such as network level protection and authentication techniques. They also present a summary of the advantages and disadvantages of the different filters and classifiers, being the computational cost, the needed time and the need of continuous feeding the main drawbacks.

Feature selection and its influence in the performance of classifiers has been studied as well, since it is a key point for phishing detection and filtering and studies have shown an improvement in classification accuracies [199, 200]. Finally, Purkait's review [23] includes a thorough discussion of the countermeasures that had been published until 2012.

5 Conclusions

In this article, a brief review of machine learning methods has been made, emphasizing its applications in the field of cybersecurity.

Using the source https://www.cisecurity.org/cybersecurity-threats/, there are very recent events: *On June 20, 2018, the Cyber Threat Alert Level was evaluated and is remaining at Blue (Guarded) due to multiple vulnerabilities in Google, Apple, and Microsoft products. On June 13, the MS-ISAC released an advisory for a vulnerability in Google Chrome, which could allow for arbitrary code execution. On June 14, the MS-ISAC released an advisory for a vulnerability in Apple Xcode for macOS High Sierra, which could allow for arbitrary code execution. On June 20, the MS-ISAC released an advisory for multiple vulnerabilities in Microsoft Exchange Server, which could allow for information disclosure. Organizations and users are advised to update and apply all appropriate vendor security patches to vulnerable systems and to continue to update their antivirus signatures daily. Another line of defense includes user awareness training regarding the threats posed by attachments and hypertext links contained in emails especially from un-trusted sources.*

Finally, it is important that the reader who is entering into this field of cybersecurity and intends to apply this type of models, delves into the extensive bibliography that is presented in this review.

Acknowledgements C. Iglesias acknowledges the support of the Spanish Ministry of Education, Culture and Sport for FPU Grant number

12/02283. J. Martinez acknowledges the support of the Spanish Ministry of Education for Grant project ID TIN2016-76770-R.

References

1. International Telecommunication Union (2014) The world in 2014: ICT Facts and figures. Technical report
2. Klimburg A (ed) (2012) National cyber security framework manual. NATO CCD COE Publication
3. Kolter JZ, Maloof MA (2006) Learning to detect and classify malicious executables in the wild. J Mach Learn Res 7:2721–2744
4. Almomani A, Altaher A, Ramadass S (2012) Application of adaptive neuro-fuzzy inference system for information security. J Comput Sci 8(6):983–986
5. Bauer JM, van Eeten MJG (2009) Cybersecurity: stakeholder incentives, externalities, and policy options. Telecommun Policy 33(10–11):706–719
6. Vázquez C (2014) Auditing using vulnerability tools to identify today's threats business performance. SANS Institute, Fredericksburg
7. Parise Furfaro A (2017) Using virtual environments for the assessment of cybersecurity issues in IoT scenarios. Simul Model Pract Theory 73:43–54
8. Hashemi Khorshidpour T (2017) Domain invariant feature extraction against evasion attack. Int J Mach Learn Cybern 9:1–12
9. Kumar VA, Pandey KK, Punia DK (2014) Cyber security threats in the power sector: Need for a domain specific regulatory framework in India. Energy Policy 65:126–133
10. North Atlantic Treaty Organization (NATO) (2008) Bucharest summit declaration. Issued by the Heads of State and Government participating in the meeting of the North Atlantic Council in Bucharest on 3 April 2008
11. Barat M, Bogdan D, P, Gavrilut DT (2013) An automatic updating perceptron-based system for malware detection. In: IEEE 2013 15th international symposium on symbolic and numeric algorithms for scientific computing, pp 303–307
12. Bauer JM, Van Eeten M, Chattopadhyay T, Wu Y (2008) Financial implications of network security: malware and spam. Technical report, report for the international telecommunication union (ITU), Geneva (Switzerland)
13. International Organization for Standardization (2012) ISO/IEC 27032:2012. Information technology—Security techniques—Guidelines for cybersecurity
14. Fischer EA (2005) Creating a national framework for cybersecurity: an analysis of issues and options. Technical report. Congressional Research Service
15. The Open Web Application Security Project (OWASP) (2018) https://www.swascan.com/owasp/
16. The Open Web Application Security Project (2013) OWASP Top 10—the ten most critical web application security risks. The OWASP Foundation
17. Microsoft Security Development Lifecycle (2018) https://www.microsoft.com/enus/securityengineering/sdl/
18. Vatamanu C, Gavriluţ D, Benchea R-M (2013) Building a practical and reliable classifier for malware detection. J Comput Virol Hacking Tech 9(4):205–214
19. Gavrilut D, Benchea R, Vatamanu C (September 2012) Optimized zero false positives perceptron training for malware detection. In: IEEE 2012 14th international symposium on symbolic and numeric algorithms for scientific computing, pp 247–253
20. Gavrilut D, Benchea R, Vatamanu C (2012) Practical optimizations for perceptron algorithms in large malware dataset. In: IEEE 2012 14th international symposium on symbolic and numeric algorithms for scientific computing, pp 240–246
21. Singh K, Guntuku SC, Thakur A, Hota C (2014) Big data analytics framework for peer-to-peer botnet detection using random forests. Inf Sci 278:488–497
22. Goseva-Popstojanova K, Anastasovski G, Dimitrijevikj A, Pantev R, Miller B (2014) Characterization and classification of malicious web traffic. Comput Secur 42:92–115
23. Purkait S (2012) Phishing counter measures and their effectiveness: literature review. Inf Manag Comput Secur 20(5):382–420
24. Ceesay EN (2008) Mitigating phishing attacks: a detection, response and evaluation framework. Ph.D. thesis, University of California
25. Nappa D, Wang X, Abu-Nimeh S, Nair S (2007) A comparison of machine learning techniques for phishing detection. In: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit on—eCrime '07, pp 60–69
26. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: pp 281–297
27. Kohonen T (1982) Self-organizing formation of topologically correct feature maps. Biol Cybern 43:59–69
28. Gordon AD (1992) Hierarchical classification. World Scientific Press, Singapore
29. Albayrak S, Amasyali F (2003) Fuzzy c-means clustering on medical diagnostic systems. In: International twelfth Turkish symposium on artificial intelligence and neural networks (TAINN), pp 1–3
30. Bradley PS, Fayad UM (1998) Refining initial points for k-means clustering. In: Proceedings of the 15th conference on machine learning, Wisconsin, pp 91–99
31. Haykin S (1999) Neural netowrks. A comprehensive foundation. Prentice Hall, Upper Saddle River
32. Quinlan JR (1986) Induction on decision trees. Mach Learn 1:81–106
33. Quinlan JR (1993) C4.5: programas for machine learning. Morgan Kaufmann, Burlington
34. Breiman L, Friedman J (1984) Classification and regression trees. Wadsworth, Belmont
35. Cherkassky V, Mulier F (1998) Learning from data: concepts, theory and methods. Wiley, Berlin
36. Vorobeva A (2017) Influence of features discretization on accuracy of random forest classifier for web user identification. In: Conference of open innovation association, FRUCT
37. Miller S, Busby-Earle C (2017) Multi-perspective machine learning a classifier ensemble method for intrusion detection. In: ICMLSC '17 proceedings of the 2017 international conference on machine learning and soft computing, pp 7–12
38. He S, Lee G, Han S, Whinston A (2016) How would information disclosure influence organizations' outbound spam volume? Evidence from a field experiment. J Cybersecur 2(1):99–118
39. Vapnik V (1982) Estimation of dependences based on empirical data. Springer, Berlin
40. Drucker H, Burges C, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. MIT Press, Cambridge
41. Osuna E, Freund R, Girosi F (1997) An improved training algorithm for support vector machines, In: Proceedings of the 1997 IEEE signal processing society workshop, Amelia Island, Florida, USA, pp 1–10
42. Joachims T (1999) Machine large-scale SVM learning practical. MIT Press, Cambridge
43. Kyriakopoulos Ghanem A (2017) Support vector machine for network intrusion and cyber-attack detection. Sensor Signal Processing for Defence Conference (SSPD2017) 38–41
44. Vapnik V (1998) Statistical learning theory. Wiley, Berlin
45. MacCulloch WS, Pitts WS (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133

46. Dua S, Du X (2011) Data mining and machine learning in cybersecurity. Auerbach Publications, Taylor & Francis Group, Boca Raton, FL, USA
47. Battiti R (1992) First and second-order methods for learning: between steepset descent and newton method. Neural Comput 4:141–166
48. Bishop CM (1995) Neural networks and pattern recognition. Oxford University Press, Oxford
49. Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural network by choosing initial values of the adaptative weights. In: International joint conference on neural networks (IJCNN). IEEE, San Diego, pp 21–26
50. Wang X-Z, Wang R, Xu C (2018) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. IEEE Trans Cybern 48:703–715
51. Wang R, Wang X-Z, Kwong S, Xu C (2017) Incorporating diversity and informativeness in multiple-instance active learning. IEEE Trans Fuzzy Syst 25:1460–1475
52. Ashfaq R, Wang X-Z, Huang J, Abbas H, He Y-L (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci 378:484–497
53. Wang X-Z, Xing H-J, Li Y, Hua Q, Dong CR, Pedrycz W (2017) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst 23:1638–1654
54. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
55. Fernandez Maimo L, Perales Gomez AL, Garcia Clemente FJ, Gil Perez M, Martinez Perez. G (2018) A self-adaptive deep learning-based system for anomaly detection in 5G networks. IEEE Access 6(6):7700–7712
56. Abeshu A, Chilamkurti N (2018) Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Commun Mag 56(2):169–175
57. Kebede TM, Djaneye-Boundjou O, Narayanan BN, Ralescu A, Kapp D (2017) Classification of malware programs using autoencoders based deep learning architecture and its application to the microsoft malware classification challenge (big 2015) dataset. Proc IEEE Natl Aerosp Electron Conf NAECON 2017:70–75
58. Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, Wang C (2018) Machine learning and deep learning methods for cybersecurity. IEEE Access 6:35365–35381
59. Islam R, Abawajy J (2013) A multi-tier phishing detection and filtering approach. J Netw Comput Appl 36(1):324–335
60. Almomani A, Gupta BB, Atawneh S, Meulenberg A, Almomani E (2013) A survey of phishing email filtering techniques. IEEE Commun Surv Tutor 15(4):2070–2090
61. Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw Publ IEEE Neural Netw Counc 10(5):1048–54
62. Jagatic TN, Johnson NA, Jakobsson M, Menczer F (2007) Social phishing. Commun ACM 50(10):94–100
63. Mohammad RM, Thabtah F, McCluskey L (2015) Tutorial and critical analysis of phishing websites methods. Comput Sci Rev 17:1–24
64. Cranor LF, Lamacchia BA (1998) Spam!. Commun ACM 41(8):74–83
65. SANS Institute. Top 15 Malicious Spyware Actions (2018) https://www.sans.org/security-resources/
66. Kim SC, Lee SW, Sung KJ, Kim SK (2012) Splog detection usingstructural similarity between posts and URL biasedness in posts. J Internet Technol 13(5):767–772
67. Zhu L, Sun A, Choi B (2011) Detecting spam blogs from blog search results. Inf Process Manag 47(2):246–262
68. Luckner M, Gad M, Sobkowiak P (2014) Stable web spam detection using features based on lexical items. Comput Secur 46:79–93
69. Prieto VM, Álvarez M, Cacheda F (2013) SAAD, a content based web spam analyzer and detector. J Syst Softw 86(11):2906–2918
70. Scarselli F, Tsoi AC, Hagenbuchner M, Noi LD (2013) Solving graph data issues using a layered architecture approach with applications to web spam detection. Neural Netw Off J Int Neural Netw Soc 48:78–90
71. Martinez-Romo J, Araujo L (2013) Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst Appl 40(8):2992–3000
72. Stern H (2008) A survey of modern spam tools. In: 5th conference on email and anti-spam, CEAS 2008. Conference on email and anti-spam, CEAS
73. Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. Expert Syst Appl 36(7):10206–10222
74. Fawcett T (2003) "In vivo" spam filtering: a challenge problem for KDD. SIGKDD Explor 5(2):140–148
75. Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A Bayesian approach to filtering junk E-mail. Tech. rep. WS-98-05
76. Graham P (2003) A plan for spam. http://paulgraham.com/spam.html. Accessed 26 June 2003
77. Wang ZJ, Liu Y, Wang ZJ (2014) E-mail filtration and classification based on variable weights of the Bayesian algorithm. Appl Mech Mater 513–517:2111–2114
78. Dewdney N, VanEss-Dykema C, MacMillan R (2001) The form is the substance. In: Proceedings of the workshop on human language technology and knowledge management, vol 2001, Morristown, NJ, USA. Association for Computational Linguistics, pp 1–8
79. Almeida J, Almeida T, Yamakami A (2011) Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers. J Internet Serv Appl 1(3):183–200
80. Song Y, Kołcz A, Giles CL (2009) Better Naive Bayes classification for high-precision spam detection. Softw Pract Exp 39(11):1003–1024
81. Amayri O, Bouguila N (2010) A study of spam filtering using support vector machines. Artif Intell Rev 34(1):73–108
82. Hsu W-C, Yu T-Y (2010) E-mail spam filtering based on support vector machines with Taguchi method for parameter selection. J Converg Inf Technol 5(8):78–88
83. Caruana G, Li M, Qi M (2011) A MapReduce based parallel SVM for large scale spam filtering. In: IEEE 2011 eighth international conference on fuzzy systems and knowledge discovery (FSKD), vol 4, pp 2659–2662
84. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
85. Wu C-H (2009) Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert Syst Appl 36(3):4321–4330
86. Tseng L-S, Wu C-H (2003) Detection of spam e-mails by analyzing the distributing behaviors of e-mail servers. In: Proceedings of the third international conference on hybrid intelligent systems, pp 1024–1033
87. Gupta A, Singhal C, Aggarwal S (2012) An improved anti spam filter based on content, low level features and noise. Lect Notes Inst Comput Sci Soc Inf Telecommun Engi LNICST 84(PART 1):563–572
88. Li P, Yan H, Cui G, Du Y (2012) Integration of local and global features for image spam filtering. J Comput Inf Syst 8(2):779–789

89. Biggio B, Fumera G, Pillai I, Roli F (2011) A survey and experimental evaluation of image spam filtering techniques. Pattern Recognit Lett 32(10):1436–1446
90. Hazza ZM, Aziz NA (2015) A new efficient text detection method for image spam filtering. Int Rev Comput Softw 10(1):1–8
91. Liu T-J, Wu C-N, Lee C-L, Chen C-W (2014) A self-adaptable image spam filtering system. J Chin Inst Eng Trans Chin Inst Eng Ser A (Chung-kuo Kung Ch'eng Hsuch K'an) 37(4):517–528
92. Manek AS, Shamini DK, Bhat VH, Shenoy PD, Mohan MC, Venugopal KR, Patnaik LM (2014) Rep-etd: a repetitive preprocessing technique for embedded text detection from images in spam emails. In: pp 568–573
93. Wakade S, Liszka KJ, Chan C-C (2013) Application of learning algorithms to image spam evolution. Smart Innov Syst Technol 13:471–495
94. Attar A, Rad RM, Atani RE (2013) A survey of image spamming and filtering techniques. Artif Intell Rev 40(1):71–105
95. Romero C, Garcia-Valdez M, Alanis A (2010) A comparative study of blog comments spam filtering with machine learning techniques. Stud Comput Intell 312:57–72
96. Yang W, Dong G, Wang W, Hu Y, Shen G, Yu M (2015) A novel approach for bots detection in sina microblog. J Comput Theor Nanosci 12(7):1420–1425
97. Abu-Nimeh S, Chen T (2010) Proliferation and detection of blog spam. IEEE Secur Priv Mag 8(5):42–47
98. Kolari P, Java A, Finin T, Oates T, Joshi A (2006) Detecting spam blogs: a machine learning approach. Proc Natl Conf Artif Intell 2:1351–1356
99. Yoshinaka T, Ishii S, Fukuhara T, Masuda H, Nakagawa H (2010) A user-oriented splog filtering based on a machine learning. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6045 LNCS((M4D)):88–99
100. Sculley D, Wachman GM (2007) Relaxed online SVMS for spam filtering. In: pp 415–422
101. McCord M, Chuah M (2011) Spam detection on twitter using traditional classifiers. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6906 LNCS:175–186
102. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
103. Soman SJ, Murugappan S (2014) Detecting malicious tweets in trending topics using clustering and classification
104. Chu Z, Gianvecchio S, Wang H, Jajodia S (2010) Who is tweeting on twitter: human, bot, or cyborg? In: pp 21–30
105. Wang AH (2010) Detecting spam bots in online social networking sites: a machine learning approach. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6166 LNCS:335–342
106. Wang AH (2010) Don't follow me—spam detection in twitter. In: pp 142–151
107. Santos I, Miñambres-Marcos I, Laorden C, Galán-García P, Santamaría-Ibirika A, García Bringas P (2014) Twitter content-based spam filtering. Adv Intell Syst Comput 239:449–458
108. Zangerle E, Specht G (2014) "sorry, i was hacked" a classification of compromised twitter accounts. In: pp 587–593
109. Benevenuto F, Rodrigues T, Almeida V, Almeida J, Zhang C, Ross K (2008) Identifying video spammers in online social networks. In: pp 45–52
110. Benevenuto F, Rodrigues T, Veloso A, Almeida J, Goncalves M, Almeida V (2012) Practical detection of spammers and content promoters in online video sharing systems. IEEE Trans Syst Man Cybern Part B Cybern 42(3):688–701
111. Indira K, Christal Joy E (2014) Prevention of spammers and promoters in video social networks using SVM-knn. Int J Eng Technol 6(5):2024–2030
112. Stolfo SJ, Hershkop S, Bui LH, Ferster R, Wang K (2005) Anomaly detection in computer security and an application to file system accesses. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3488 LNAI:14–28
113. Chen Z, Ji C (2005) Spatial-temporal modeling of malware propagation in networks. IEEE Trans Neural Netw 16(5):1291–1303
114. Lin J (2008) On malicious software classification. In: pp 368–371
115. Li P, Liu L, Gao D, Reiter MK (2010) On challenges in evaluating malware clustering. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6307 LNCS:238–255
116. Nakazato J, Song J, Eto M, Inoue D, Nakao K (2011) A novel malware clustering method using frequency of function call traces in parallel threads. IEICE Trans Inf Syst E94–D(11):2150–2158
117. Shafiq MZ, Khayam SA, Farooq M (2008) Improving accuracy of immune-inspired malware detectors by using intelligent features. In: pp 119–126
118. Bose A, Hu X, Shin KG, Park T (2008) Behavioral detection of malware on mobile handsets. In: pp 225–238
119. Anderson B, Quist D, Neil J, Storlie C, Lane T (2011) Graph-based malware detection using dynamic analysis. J Comput Virol 7(4):247–258
120. Chandramohan M, Tan HBK, Briand LC, Shar LK, Padmanabhuni BM (2013) A scalable approach for malware detection through bounded feature space behavior modeling. In: pp 312–322
121. Dhaya R, Poongodi M (2015) Detecting software vulnerabilities in android using static analysis. In: pp 915–918
122. Durand J, Atkison T (2012) Applying random projection to the classification of malicious applications using data mining algorithms. In: pp 286–291
123. Ismail I, Marsono MN, Nor SM (2014) Malware detection using augmented naive bayes with domain knowledge and under presence of class noise. Int J Inf Comput Secur 6(2):179–197
124. Lu W, Rammidi G, Ghorbani AA (2011) Clustering botnet communication traffic based on n-gram feature selection. Comput Commun 34(3):502–514
125. Markel Z, Bilzor M (2015) Building a machine learning classifier for malware detection. In: Second workshop on anti-malware testing research (WATeR). IEEE, Canterbury, UK. https://doi.org/10.1109/WATeR.2014.7015757
126. Merkel R, Hoppe T, Kraetzer C, Dittmann J (2010) Statistical detection of malicious pe-executables for fast offline analysis. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6109 LNCS:93–105
127. Moskovitch R, Elovici Y (2008) Unknown malicious code detection—practical issues. In: pp 145–152
128. Ponomarev S, Durand J, Wallace N, Atkison T (2013) Evaluation of random projection for malware classification. In: pp 68–73
129. Reddy DKS, Pujari AK (2006) N-gram analysis for computer virus detection. J Comput Virol 2(3):231–239
130. Santos I, Penya YK, Devesa J, Bringas PG (2009) N-grams-based file signatures for malware detection. In: Volume AIDSS, pp 317–320
131. Shabtai A, Moskovitch R, Elovici Y, Glezer C (2009) Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey. Inf Secur Tech Rep 14(1):16–29 Malware
132. Shahzad F, Farooq M (2012) Elf-miner: using structural knowledge and data mining methods to detect new (linux) malicious executables. Knowl Inf Syst 30(3):589–612

133. Shijo PV, Salim A (2015) Integrated static and dynamic analysis for malware detection. Procedia Comput Sci 46:804–811
134. Siddiqui M, Wang MC, Lee J (2008) A survey of data mining techniques for malware detection using file features. In: pp 509–510
135. Uppal D, Sinha R, Mehra V, Jain V (2014) Malware detection and classification based on extraction of API sequences. In: pp 2337–2342
136. Wressnegger C, Schwenk G, Arp D, Rieck K (2013) A close look on n-grams in intrusion detection: anomaly detection vs. classification. In: pp 67–76
137. Yu W, Zhang H, Ge L, Hardy R (2013) On behavior-based detection of malware on android platform. In: pp 814–819
138. Yuxin D, Wei D, Yibin Z, Chenglong X (2014) Malicious code detection using opcode running tree representation. In: pp 616–621
139. Yuxin D, Xuebing Y, Di Z, Li D, Zhanchao A (2011) Feature representation and selection in malicious code detection methods based on static system calls. Comput Secur 30(6–7):514–524
140. Zolotukhin M, Hämäläinen T (2013) Support vector machine integrated with game-theoretic approach and genetic algorithm for the detection and classification of malware. In: pp 211–216
141. Cova M, Kruegel C, Vigna G (2010) Detection and analysis of drive-by-download attacks and malicious javascript code. In: pp 281–290
142. Zhu K, Yin B (2012) Malware behavior classification approach based on naive bayes. J Converg Inf Technol 7(5):203–210
143. Zhu K, Yin B, Mao Y, Hu Y (2014) Malware classification approach based on valid window and naive bayes. Comput Res Dev (Jisuanji Yanjiu yu Fazhan) 51(2):373–381
144. Bat-Erdene M, Kim T, Li H, Lee H (2013) Dynamic classification of packing algorithms for inspecting executables using entropy analysis. In: pp 19–26
145. Ban T, Isawa R, Guo S, Inoue D, Nakao K (2013) Application of string kernel based support vector machine for malware packer identification. In: The 2013 international joint conference on neural networks (IJCNN). IEEE, Dallas, TX, USA. https://doi.org/10.1109/IJCNN.2013.6707043
146. Divya S, Padmavathi G (2014) A novel method for detection of internet worm malcodes using principal component analysis and multiclass support vector machine. Int J Secur Appl 8(5):391–402
147. Komiya R, Paik I, Hisada M (2011) Classification of malicious web code by machine learning. In: pp 406–411
148. Nissim N, Moskovitch R, Rokach L, Elovici Y (2012) Detecting unknown computer worm activity via support vector machines and active learning. Pattern Anal Appl 15(4):459–475
149. Nissim N, Moskovitch R, Rokach L, Elovici Y (2014) Novel active learning methods for enhanced pc malware detection in windows os. Expert Syst Appl 41(13):5843–5857
150. Okane P, Sezer S, McLaughlin K, Im EG (2014) Malware detection: program run length against detection rate. IET Softw 8(1):42–51
151. Sanjaa B, Chuluun E (2013) Malware detection using linear SVM. In: vol 2, pp 136–138
152. Wang P, Wang Y-S (2015) Malware behavioural detection and vaccine development by using a support vector model classifier. J Comput Syst Sci 81(6):1012–1026
153. Zhao M, Ge F, Zhang T, Yuan Z (2011) Antimaldroid: an efficient SVM-based malware detection framework for android. Commun Comput Inf Sci 243 CCIS(PART 1):158–166
154. Biggio B, Corona I, Nelson B, Rubinstein BIP, Maiorca D, Fumera G, Giacinto G, Roli F (2014) Security evaluation of support vector machines in adversarial environments
155. Firdausi I, Lim C, Erwin A, Nugroho AS (2010) Analysis of machine learning techniques used in behavior-based malware detection. In: pp 201–203
156. Canzanese R, Kam M, Mancoridis S (2013) Toward an automatic, online behavioral malware classification system. In: pp 111–120
157. Dube T, Raines R, Peterson G, Bauer K, Grimaila M, Rogers S (2012) Malware target recognition via static heuristics. Comput Secur 31(1):137–147
158. Haddadi F, Runkel D, Nur Zincir-Heywood A, Heywood MI (2014) On botnet behaviour analysis using gp and c4.5. In: pp 1253–1260
159. Ye W, Cho K (2014) Hybrid p2p traffic classification with heuristic rules and machine learning. Soft Comput 18(9):1815–1827
160. Borgolte K, Kruegel C, Vigna G (2013) Delta: automatic identification of unknown web-based infection campaigns. In: pp 109–120
161. Mohaisen A, Alrawi O (2015) AMAL: high-fidelity, behavior-based automated malware analysis and classification. In: Rhee KH, Yi J (eds) Information security applications, WISA 2014. Lecture notes in computer science, vol 8909. Springer, pp 107–121
162. Rieck K, Trinius P, Willems C, Holz T (2011) Automatic analysis of malware behavior using machine learning. J Comput Secur 19(4):639–668
163. Menahem E, Shabtai A, Rokach L, Elovici Y (2009) Improving malware detection by applying multi-inducer ensemble. Comput Stat Data Anal 53(4):1483–1494
164. Shabtai A, Fledel Y, Elovici Y (2010) Automated static code analysis for classifying android applications using machine learning. In: pp 329–333
165. Huang C-Y, Tsai Y-T, Hsu C-H (2013) Performance evaluation on permission-based detection for android malware. Smart Innov Syst Technol 21:111–120
166. Glodek W, Harang R (2013) Rapid permissions-based detection and analysis of mobile malware using random decision forests. In: pp 980–985
167. Alam MS, Vuong ST (2013) Random forest classification for detecting android malware. In: pp 663–669
168. Ng DV, Hwang J-IG (2015) Android malware detection using the dendritic cell algorithm. In: IEEE international conference on machine learning and cybernetics, Lanzhou, China, pp 257–262
169. Pehlivan U, Baltaci N, Acarturk C, Baykal N (2014) The analysis of feature selection methods and classification algorithms in permission based android malware detection. In: IEEE symposium on computational intelligence in cyber security (CICS), Orlando, FL, USA. https://doi.org/10.1109/CICYBS.2014.7013371
170. Barbareschi M, De Benedictis A, Mazzeo A, Vespoli A (2014) Mobile traffic analysis exploiting a cloud infrastructure and hardware accelerators. In: pp 414–41
171. Yu W, Zhang H, Xu G (2013) A study of malware detection on smart mobile devices. In: vol 8757
172. Yerima SY, Sezer S, Muttik I (2014) Android malware detection using parallel machine learning classifiers. In: pp 37–42
173. Feldman S, Stadther D, Wang B (2015) Manilyzer: automated android malware detection through manifest analysis. In: pp 767–77
174. Gates CS, Li N, Peng H, Sarma B, Qi Y, Potharaju R, Nita-Rotaru C, Molloy I (2014) Generating summary risk scores for mobile applications. IEEE Trans Dependable Secure Comput 11(3):238–251
175. Yu L, Pan Z, Liu J, Shen Y (2013) Android malware detection technology based on improved bayesian classification. In: pp 1338–1341

176. Shabtai A, Kanonov U, Elovici Y, Glezer C, Weiss Y (2012) "Andromaly": a behavioral malware detection framework for android devices. J Intell Inf Syst 38(1):161–190
177. Sanz B, Santos I, Laorden C, Ugarte-Pedrero X, Bringas PG (2012) On the automatic categorisation of android applications. In: pp 149–153
178. Feizollah A, Anuar NB, Salleh R, Amalina F, Ma'arof RR, Shamshirband S (2013) A study of machine learning classifiers for anomaly-based mobile botnet detection. Malays J Comput Sci 26(4):251–265
179. Ham H-S, Kim H-H, Kim M-S, Choi M-J (2014) Linear SVM-based android malware detection. Lect Notes Electr Eng 301:575–585
180. Narayanan A, Chen L, Chan CK (2014) AdDetect: automated detection of android ad libraries using semantic analysis. In: IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP). IEEE, Singapore. https://doi.org/10.1109/ISSNIP.2014.6827639
181. Sahs J, Khan L (2012) A machine learning approach to android malware detection. In: pp 141–147
182. Spreitzenbarth M, Schreck T, Echtler F, Arp D, Hoffmann J (2015) Mobile-sandbox: combining static and dynamic analysis with machine-learning techniques. Int J Inf Secur 14(2):141–153
183. Sheen S, Anitha R, Natarajan V (2015) Android based malware detection using a multifeature collaborative decision fusion approach. Neurocomputing 151(P2):905–912
184. Allix K, Bissyandé TF, Jérome Q, Klein J, State R, Le Traon Y (2014) Empirical assessment of machine learning-based malware detectors for Android. Empir Softw Eng 21:183–211
185. Allix K, Bissyandé TF, Klein J, Traon YL (2015) Are your training datasets yet relevant? an investigation into the importance of timeline in machine learning-based malware detection. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8978:51–67
186. Fette I, Sadeh N, Tomasic A (2007) Learning to detect phishing emails. In: Proceedings of the 16th international conference on World Wide Web (WWW '07), New York (US), ACM, pp 649–656
187. Zhang L, Yao T (2003) Filtering junk mail with a maximum entropy model. In: pp 446–453
188. Gu X, Wang H, Ni T (2013) An efficient approach to detecting phishing web. J Comput Inf Syst 9(14):5553–5560
189. He M, Horng S, Fan P, Khan M Khurram, Run R, Lai J, Chen R, Sutanto A (2011) An efficient phishing webpage detector. Expert Syst Appl 38(10):12018–12027
190. Cao J, Dong D, Mao B, Wang T (2013) Phishing detection method based on url features. J Southeast Univ (English Edition) 29(2):134–138
191. Chandrasekaran M, Narayanan K, Upadhyaya S (2006) Phishing E-mail detection based on structural properties. In: Proceedings of 9th annual NYS cyber security conference, Albany, NY, USA, pp 2–8
192. Ma L, Ofoghi B, Watters P, Brown S (2009) Detecting phishing emails using hybrid features. In: pp 493–497
193. Santhana Lakshmi V, Vijaya MS (2012) Efficient prediction of phishing websites using supervised learning algorithms. Procedia Eng 30:798–805
194. Akinyelu AA, Adewumi AO (2014) Classification of phishing email using random forest machine learning technique. J Appl Math 2014:1–6
195. Webber CG, De Fátima M, Do Prado Lima W, Hepp FS (2012) Testing phishing detection criteria and methods. Adv Intell Soft Comput 133AISC:853–858
196. Del Castillo MD, Iglesias Á, Serrano JI (2007) An integrated approach to filtering phishing e-mails. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4739 LNCS:321–328
197. Xiang G, Hong J, Rose CP, Cranor L (2011) Cantina+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans Inf Syst Secur 14(2):1–28
198. Patil R, Dasharath DB, Dhonde KS, Chinchwade RG, Mehetre SB (2014) A hybrid model to detect phishing-sites using clustering and bayesian approach. Int J Comput Sci Netw Secur 15:92–95
199. Basnet RB, Sung AH, Liu Q (2012) Feature selection for improved phishing detection. Lect Notes Comput Sci (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7345 LNAI:252–261
200. Qabajeh I, Thabtah F (2014) An experimental study for assessing email classification attributes using feature selection methods. In: pp 125–132