2018 10th International Conference on Cyber Conflict
CyCon X: Maximising Effects
T. Minárik, R. Jakschis, L. Lindström (Eds.)
2018 © NATO CCD COE Publications, Tallinn

Permission to make digital or hard copies of this publication for internal use within NATO and for personal or educational use when for non-profit or non-commercial purposes is granted providing that copies bear this notice and a full citation on the first page. Any other reproduction or transmission requires prior written permission by NATO CCD COE.

On the Effectiveness of Machine and Deep Learning for Cyber Security

Giovanni Apruzzese
Department of Engineering
'Enzo Ferrari'
University of Modena
and Reggio Emilia
Modena, Italy
giovanni.apruzzese@unimore.it

Michele Colajanni
Department of Engineering
'Enzo Ferrari'
University of Modena
and Reggio Emilia
Modena, Italy
michele.colajanni@unimore.it

Luca Ferretti
Department of Engineering
'Enzo Ferrari'
University of Modena
and Reggio Emilia
Modena, Italy
luca.ferretti@unimore.it

Alessandro Guido
Department of Engineering
'Enzo Ferrari'
University of Modena
and Reggio Emilia
Modena, Italy
alessandro.guido@unimore.it

Mirco Marchetti
Department of Engineering
'Enzo Ferrari'
University of Modena
and Reggio Emilia
Modena, Italy
mirco.marchetti@unimore.it

Abstract: Machine learning is adopted in a wide range of domains where it shows its superiority over traditional rule-based algorithms. These methods are being integrated in cyber detection systems with the goal of supporting or even replacing the first level of security analysts. Although the complete automation of detection and analysis is an enticing goal, the efficacy of machine learning in cyber security must be evaluated

with the due diligence. We present an analysis, addressed to security specialists, of machine learning techniques applied to the detection of intrusion, malware, and spam. The goal is twofold: to assess the current maturity of these solutions and to identify their main limitations that prevent an immediate adoption of machine learning cyber detection schemes. Our conclusions are based on an extensive review of the literature as well as on experiments performed on real enterprise systems and network traffic.

Keywords: *machine learning, deep learning, cyber security, adversarial learning*

1. INTRODUCTION

The appeal and pervasiveness of machine learning (ML) is growing. Existing methods are being improved, and their ability to understand and answer real issues is highly appreciated. These achievements have led to the adoption of machine learning in several domains, such as computer vision, medical analysis, gaming and social media marketing [1]. In some scenarios, machine learning techniques represent the best choice over traditional rule-based algorithms and even human operators [2]. This trend is also affecting the cyber security field where some detection systems are being upgraded with ML components [3]. Although devising a completely automated cyber defence system is yet a distant objective, first level operators in Network and Security Operation Centres (NOC and SOC) may benefit from detection and analysis tools based on machine learning. This paper is specifically addressed to security operators and aims to assess the current maturity of these solutions, to identify their main limitations and to highlight some room for improvement.

Our study is based on an extensive review of the literature and on original experiments performed on real, large enterprises and network traffic. Other academic papers compare ML solutions for cyber security by considering one specific application (e.g.: [4], [3], [5]) and are typically oriented to Artificial Intelligence (AI) experts rather than to security operators. In the evaluation, we exclude the commercial products based on machine learning (or on the abused AI term) because vendors do not reveal their algorithms and tend to overlook issues and limitations. First, we present an original taxonomy of machine learning cyber security approaches. Then, we map the identified classes of algorithms to three problems where machine learning is currently applied: intrusion detection, malware analysis, spam and phishing detection. Finally, we analyse the main limitations of existing approaches. Our study highlights pros and cons of different methods, especially in terms of false positive or false negative alarms. Moreover, we point out a general underestimation of the complexity of managing ML architectures in cyber security caused by the lack of publicly available and labelled

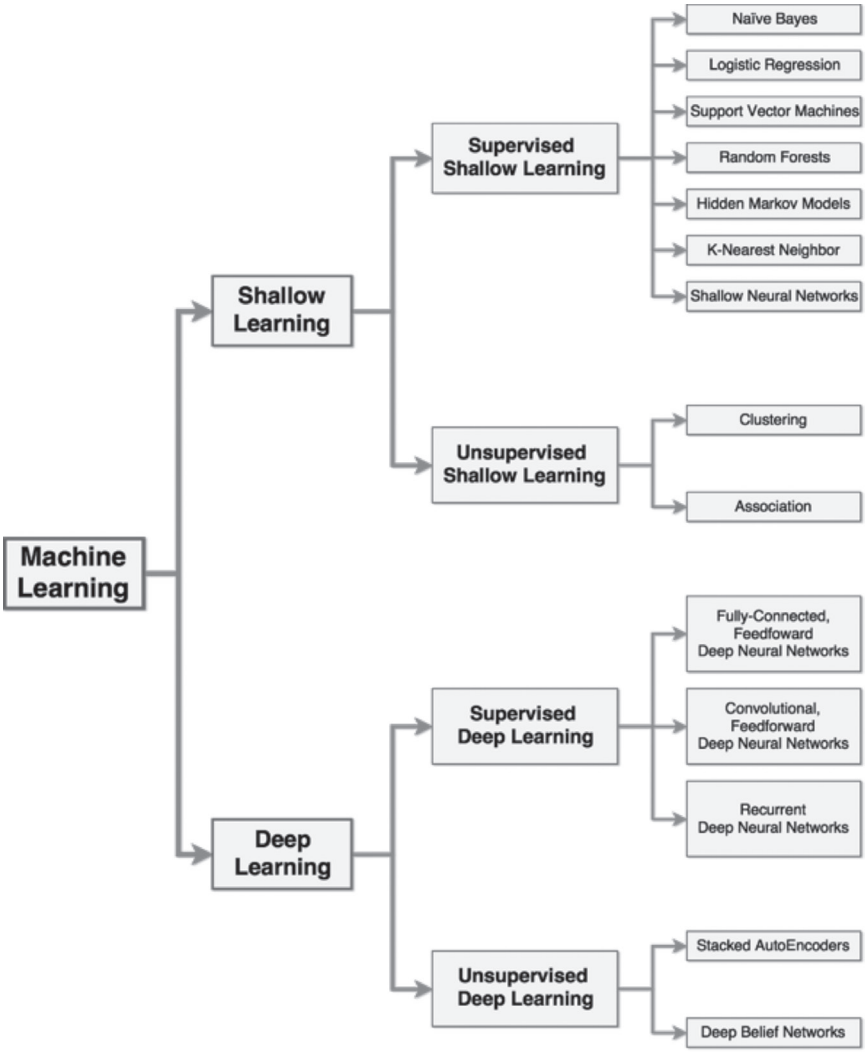
data for training, and by the time required for fine-tuning operations in a domain characterized by continuous change. We also consider recent results emphasizing the effectiveness of *adversarial attacks* [6] [5] in evading ML detectors. The evidenced drawbacks pave the way to future improvements that ML components require before being fully adopted in cyber defence platforms.

This paper is structured as follows. Section 2 proposes our original taxonomy of ML algorithms applied to cyber security. Section 3 outlines the three classes of cyber security problems considered in this paper and Section 4 compares and evaluates ML solutions for cyber security. Section 5 concludes the paper with some final remarks.

2. CLASSIFICATION OF MACHINE LEARNING ALGORITHMS FOR CYBER SECURITY

Machine learning includes a large variety of paradigms in continuous evolution, presenting weak boundaries and cross relationships. Furthermore, different views and applications may lead to different classifications. Hence, we cannot refer to one fully accepted taxonomy from literature, but we prefer to propose an original taxonomy able to capture the differences among the myriad of techniques that are being applied to cyber detection, as shown in Figure 1. This taxonomy is specifically oriented to security operators and avoids the ambitious goal of presenting the ultimate classification that can satisfy all AI experts and application cases. The first discriminant evidenced in Figure 1 is between the traditional ML algorithms, which today can be referred to as **Shallow Learning (SL)**, in opposition to the more recent **Deep Learning (DL)**. Shallow Learning requires a domain expert (that is, a *feature engineer*) who can perform the critical task of identifying the relevant data characteristics before executing the SL algorithm. Deep Learning relies on a multi-layered representation of the input data and can perform feature selection autonomously through a process defined *representation learning*.

FIGURE 1. CLASSIFICATION OF ML ALGORITHMS FOR CYBER SECURITY APPLICATIONS.



SL and DL approaches can be further characterized by distinguishing between *supervised* and *unsupervised* algorithms. The former techniques require a training process with a large and representative set of data that have been previously classified by a human expert or through other means. The latter approaches do not require a pre-labelled training dataset. In this section, we consider and compare the most popular categories of ML algorithms, which appear as the leaves of the classification tree in Figure 1. We remark that each category can include dozens of different techniques[1].

1 For a detailed list of existing ML algorithms, see: https://cran.r-project.org/web/views/MachineLearning.html

A. Shallow Learning

1) Supervised SL algorithms

- **Naïve Bayes (NB).** These algorithms are probabilistic classifiers which make the a-priori assumption that the features of the input dataset are independent from each other. They are scalable and do not require huge training datasets to produce appreciable results.
- **Logistic Regression (LR).** These are categorical classifiers that adopt a discriminative model. Like NB algorithms, LR methods make the a-priori independency assumption of the input features. Their performance is highly dependent on the size of the training data.
- **Support Vector Machines (SVM).** These are non-probabilistic classifiers that map data samples in a feature space with the goal of maximizing the distance between each category of samples. They do not make any assumption on the input features, but they perform poorly in multi-class classifications. Hence, they should be used as binary classifiers. Their limited scalability might lead to long processing times.
- **Random Forest (RF).** A random forest is a set of *decision trees*, and considers the output of each tree before providing a unified final response. Each decision tree is a conditional classifier: the tree is visited from the top and, at each node, a given condition is checked against one or more features of the analysed data. These methods are efficient for large datasets and excel at multiclass problems, but deeper trees might lead to overfitting.
- **Hidden Markov Models (HMM).** These model the system as a set of states producing outputs with different probabilities; the goal is to determine the sequence of states that produced the observed outputs. HMM are effective for understanding the temporal behaviour of the observations, and for calculating the likelihood of a given sequence of events. Although HMM can be trained on labelled or unlabelled datasets, in cyber security they have mostly been used with labelled datasets.
- **K-Nearest Neighbour (KNN).** KNN are used for classification and can be used for multi-class problems. However, both their training and test phase are computationally demanding as to classify each test sample, they compare it against all the training samples.
- **Shallow Neural Network (SNN).** These algorithms are based on neural networks, which consist in a set of processing elements (that is, *neurons*) organized in two or more communicating layers. SNN include all those types of neural networks with a limited number of neurons and layers. Despite the existence of unsupervised SNN, in cyber security they have mostly been used for classification tasks.

2) Unsupervised SL algorithms

- **Clustering.** These group data points that present similar characteristics. Well known approaches include k-means and *hierarchical* clustering. Clustering methods have a limited scalability, but they represent a flexible solution that is typically used as a preliminary phase before adopting a supervised algorithm or for anomaly detection purposes.
- **Association.** They aim to identify unknown patterns between data, making them suitable for prediction purposes. However, they tend to produce an excessive output of not necessarily valid rules, hence they must be combined with accurate inspections by a human expert.

B. Deep Learning

All DL algorithms are based on Deep Neural Networks (DNN), which are large neural networks organized in many layers capable of autonomous representation learning.

1) Supervised DL algorithms

- **Fully-connected Feedforward Deep Neural Networks (FNN).** They are a variant of DNN where every neuron is connected to all the neurons in the previous layer. FNN do not make any assumption on the input data and provide a flexible and general-purpose solution for classification, at the expense of high computational costs.
- **Convolutional Feedforward Deep Neural Networks (CNN).** They are a variant of DNN where each neuron receives its input only from a subset of neurons of the previous layer. This characteristic makes CNN effective at analysing spatial data, but their performance decreases when applied to non-spatial data. CNN have a lower computation cost than FNN.
- **Recurrent Deep Neural Networks (RNN).** A variant of DNN whose neurons can send their output also to previous layers; this design makes them harder to train than FNN. They excel as sequence generators, especially their recent variant, the *long short-term memory*.

2) Unsupervised DL algorithms

- **Deep Belief Networks (DBN).** They are modelled through a composition of *Restricted Boltzmann Machines* (RBM), a class of neural networks with no output layer. DBN can be successfully used for pre-training tasks because they excel in the function of feature extraction. They require a training phase, but with unlabelled datasets.

- **Stacked Autoencoders (SAE)**. They are composed by multiple *Autoencoders*, a class of neural networks where the number of input and output neurons is the same. SAE excel at pre-training tasks similarly to DBN, and achieve better results on small datasets.

3. APPLICATIONS OF MACHINE LEARNING ALGORITHMS TO CYBER SECURITY

We consider the three areas where most cyber ML algorithms are finding application: *intrusion detection, malware analysis*, and *spam detection*. An outline of each field is presented below.

Intrusion detection aims to discover illicit activities within a computer or a network through Intrusion Detection Systems (IDS). *Network* IDS are widely deployed in modern enterprise networks. These systems were traditionally based on patterns of known attacks, but modern deployments include other approaches for anomaly detection, threat detection [7] and classification based on machine learning. Within the broader intrusion detection area, two specific problems are relevant to our analysis: the detection of *botnets* and *of Domain Generation Algorithms* (DGA). A botnet is a network of infected machines controlled by attackers and misused to conduct multiple illicit activities. Botnet detection aims to identify communications between infected machines within the monitored network and the external command-and-control servers. Despite many research proposals and commercial tools that address this threat, several botnets still exist. DGA automatically generate domain names, and are often used by an infected machine to communicate with external server(s) by periodically generating new hostnames. They represent a real threat for organizations because, through DGA which relies on language processing techniques, it is possible to evade defences based on static blacklists of domain names. We consider DGA detection techniques based on ML.

Malware analysis is an extremely relevant problem because modern malware can automatically generate novel variants with the same malicious effects but appearing as completely different executable files. These polymorphic and metamorphic features defeat traditional rule-based malware identification approaches. ML techniques can be used to analyse malware variants and attributing them to the correct malware family.

Spam and phishing detection includes a large set of techniques aimed at reducing the waste of time and potential hazard caused by unwanted emails. Nowadays, unsolicited emails, namely *phishing*, represent the preferred way through which an attacker establishes a first foothold within an enterprise network. Phishing emails

include malware or links to compromised websites. Spam and phishing detection is increasingly difficult because of the advanced evasion strategies used by attackers to bypass traditional filters. ML approaches can improve the spam detection process.

TABLE 1. APPLICATION OF ML TO CYBER SECURITY PROBLEMS.

		Intrusion Detection			Malware Analysis	Spam Detection
		Network	Botnet	DGA		
Deep Learning	Supervised	RNN [8]	RNN [9]		FNN [10] CNN [11] RNN [12]	
	Unsupervised	DBN [13] SAE [14]			DBN [15] SAE [16]	DBN [17] SAE [18]
Shallow Learning	Supervised	RF [3] NB [3] SVM [3] LR [3] HMM [3] KNN [3] SNN [3]	RF [19] NB [19] SVM [19] LR [20] KNN [21] SNN [22]	RF [23] HMM [23]	RF [24] NB [24] SVM [24] LR [24] HMM [25] KNN [24] SNN [26]	RF [27] NB [28] SVM [28] LR [27] KNN [27] SNN [27]
	Unsupervised	Clustering [29] Association [30]	Clustering [5]	Clustering [31]	Clustering [24] Association [32]	Clustering [33] Association [34]

In Table 1 we report the main ML algorithms that have been proposed to address the previously identified cyber security problems. In this table, rows report the family of algorithms presented in Section 2, while columns denote cyber issues. Each cell indicates which ML algorithms are used for each problem; empty cells denote that, to the best of our knowledge, there is no proposal for that class of problems. From this table, it emerges that SL algorithms are applied to all considered problems. Supervised DL algorithms find wide application to malware analysis, less to intrusion detection; spam detection relies only on unsupervised DL algorithms. Despite its relatedness to natural language processing [2], no DL algorithm is applied to DGA detection. As expected, the overall number of algorithms based on DL is considerably smaller than those based on SL. Indeed, DL proposals based on huge neural networks are more recent than SL approaches. This gap opens many research opportunities.

Finally, we highlight a significant difference among supervised and unsupervised approaches: the former algorithms are used for classification purposes and can implement complete detectors; the latter techniques perform ancillary activities [35]. Unsupervised SL algorithms are often used for grouping data with similar characteristics independently of predefined classification criteria, and excel at identifying useful features whenever the data to be analysed present high dimensionality [16].

4. EVALUATION

In this section we present seven issues that must be considered before deciding whether to apply ML algorithms in NOC and SOC. We can anticipate that, at the current state-of-the-art, no algorithm can be considered fully autonomous with no human supervision. We substantiate each issue through experimental results from literature or original experiments performed on large enterprises. We begin by describing the testing environments of our experiments, and the metrics considered for evaluation. The experiments focus on DGA Detection and Network Intrusion Detection, and leverage two ML algorithms: Random Forest and Feedforward Fully Connected Deep Neural Network.

For **DGA Detection**, we compose two labelled training datasets containing both DGA and non-DGA domains. The former dataset contains DGA created through known techniques, while the latter contains DGA created using more recent approaches. Non-DGA domains are randomly chosen among the Cisco Umbrella top-1 million. We report the meaningful metrics of the training datasets in Table 2. Moreover, we build a testing dataset of 10,000 domains extracted evenly from each of the training datasets. We also rely on a real and unlabelled dataset composed of almost 20,000 domains contacted by a large organization. The features extracted for this dataset are: *n-gram* normality score [36]; meaningful characters ratio [36]; number-to-character ratio; vowel-to-consonant ratio; and domain length. These datasets are used to train and test a self-developed Random Forest classifier composed of 100 decision trees leveraging the CART (classification and regression tree) algorithm.

TABLE 2. TRAINING DATASETS FOR DGA DETECTION EXPERIMENTS.

Dataset	DGA technique	DGA count	non-DGA count
1	Well-known	21,355	20,227
2	Well-known and recent	37,673	8,120

For **Network Intrusion Detection**, we use three labelled real training datasets composed of benign and malicious network flows[2] collected in a large organization of nearly 10,000 hosts. The labels are created by flagging as malicious those flows that raised alerts by the enterprise network IDS and reviewed by a domain expert. Meaningful metrics of these training datasets are reported in Table 3. We also generate a testing dataset of 50,000 flows evenly extracted among the training datasets. The considered features for these datasets include: source/destination IP address, source/destination port, number of incoming/outgoing bytes and packets, TCP flags, protocol used, duration of the flow and list of alerts raised. These datasets are used to test and train two self-developed classifiers, one based on Random Forests and one on

2 *Cisco Netflow:* https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

Feedforward Fully-connected Deep Neural Network. Different topologies have been considered for each algorithm. The RF is composed by 100 decision trees leveraging the CART algorithm. For the FNN, the overall number of neurons ranges from 128 to 16,384, distributed between 2 to 16 layers; the hidden layers leverage the *ReLU* activation function, whereas the output layer uses a *sigmoid* activation function.

TABLE 3. TRAINING DATASETS FOR NETWORK INTRUSION DETECTION EXPERIMENTS.

Dataset	Malicious flows	Benign flows
1	1,000	100,000
2	2,500	250,000
3	5,000	500,000

The quality of each classifier is measured through common performance metrics, namely *Precision, Recall, F1-score*, which are computed as follows:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where *TP*, *FP*, and *FN* denote true positives, false positives, and false negatives, respectively. For completeness, we consider a true positive to be a correct detection of a malicious sample. Precision indicates how much a given approach is likely to provide a correct result. Recall is used to measure the detection rate. The F1-score combines Precision and Recall into a single value. We do not rely on Accuracy[3] because, in a real organization, the number of legitimate events is several orders of magnitude greater than illegitimate events. Hence, all the Accuracy values are close to 1 and these results prevent capturing the true effectiveness of a classifier. Finally, to reduce the possibility of biased results, each evaluation metric is computed after performing 10-fold cross validation.

A. Shallow vs Deep Learning

Deep Learning is known to outperform Shallow Learning in some applications, such as computer vision [2]. This is not always the case for cyber security where some well configured SL algorithms may prevail, even given the DL proposals are scarce with respect to SL techniques in this domain. Just to give an example, we experimentally compare the performance of the two self-developed classifiers for Network Intrusion Detection, one based on RF (Shallow Learning) and another based on FNN (Deep Learning). Both are trained with the third dataset described in Table 3 and tested on the network intrusion detection testing dataset. To obtain more refined results, we repeat the training and test phase of these classifiers multiple times using different topologies. In Table 4, we show the classification results achieved by each method; for the FNN we report the results obtained by the best topology consisting

3 $Accuracy = \frac{TP+FN}{TP+TN+FN+FP}$, where *TN* denotes true negatives.

in 1.024 neurons spread across 4 hidden layers. The RF classifier performed better than the FNN, with an F1-score of nearly 0.8, against the 0.6 obtained by the FNN. Our takeaway is that security administrators should not be charmed by the alluring neuronal multi-layer approach offered by Deep Learning, as some of these methods might still be immature for cyber security.

TABLE 4. COMPARISON BETWEEN DL AND SL CLASSIFIERS.

Classifier	F1-score	Precision	Recall
Random Forest (SL)	0.7985	0.8727	0.736
Fully-connected Feedforward Deep Neural Network (DL)	0.6085	0.7708	0.5027

B. General vs specific detectors

Products based on machine learning are often promoted by vendors as catch-all solutions to a broad array of cyberattacks. However, unbiased experimental results show that ML algorithms may provide superior performance when they focus on *specific* threats instead of trying to detect multiple threats at once. We devise multiple intrusion detection systems based on the self-developed RF classifiers for network intrusion detection, each focusing on a specific type of attack, such as buffer overflows, malware infection, DoS. The training dataset for each classifier is based on the third dataset presented in Table 3. We train and test each classifier, and then compare their classification results with the classifier described in the first row of Table 4 that is our baseline. Table 5 shows the Precision, Recall and F1-score of the six classifiers that obtained the best results, alongside the baseline reported in the bottom row. These attack-specific classifiers obtain promising results on real traffic data with F1-scores of over 0.95, while the 'general-purpose' classifier performs significantly poorly. We conclude that entrusting a single ML detector to identify malicious flows is an enticing but as yet unfeasible goal. On the other hand, by having multiple detectors, each focusing on one attack type, it is possible to produce a defensive scheme with superior detection capabilities.

TABLE 5. CLASSIFICATION RESULTS FOR ATTACK-SPECIFIC CLASSIFIERS AND THE GENERAL CLASSIFIER.

Attack Name	F1-score	Precision	Recall
DOS attempt	0.9953	0.9938	0.9969
Overflow attempt	0.9939	0.9933	0.9946
SSH Brute Force login	0.9916	0.9941	0.9892
Suspicious DNS query	0.9753	0.9953	0.9586
Cache Poisoning attempt	0.9676	0.9872	0.9506
Possible Malware infection	0.9587	0.9939	0.9337
General approach (baseline)	0.7985	0.8727	0.7360

C. Vulnerability to adversarial attacks

Competent adversaries use novel strategies to evade detectors based on machine learning algorithms [5]. These activities, namely *adversarial attacks*, may attack the integrity, the availability, or the privacy of the target system [6]. Integrity violations evade a classification or a clustering algorithm by producing attacks classified as licit activities. Availability violations produce a multitude of normal events that are classified as an attack thus causing detectors to raise a huge amount of false alarms. Privacy violations let the attacker acquire information on the target network by exploiting the defensive ML algorithm. Moreover, recent advances in Deep Learning led to the development of *generative adversarial networks* (GAN) [37], which are DNN capable of automatically producing adversarial samples against a target ML system.

TABLE 6. DETECTION RATES OF THE RF CLASSIFIER AGAINST DIFFERENT DGA TECHNIQUES [36].

DGA method	Recall
corebot	1
cryptolocker	1
dircrypt	0.99
kraken_v2	0.96
lockyv2	0.97
pykspa	0.85
qakbot	0.99
ramdo	0.99
ramnit	0.98
simda	0.96
DeepDGA GAN	**0.48**

To demonstrate the effectiveness of a GAN in evading classifiers we analyse the case study of DeepDGA [36]. The authors initially train an RF classifier to detect DGA using known datasets, and then show that this classifier identifies DGA with good detection rates. Then, they develop a GAN to generate domains that evade such classifier. Results are presented in Table 6, where the first ten rows show the detection rate against ten real DGA, while the last row denotes the detection rate against samples generated by the DeepDGA GAN. We observe that the performance of the classifier (always above 0.85, and above 0.96 for nine out of ten DGA) drops below 50% for GAN-generated samples.

TABLE 7. DETECTION RATES OF THE RF CLASSIFIER AGAINST DIFFERENT DGA BEFORE AND AFTER HARDENING [36].

DGA method	Baseline Recall	Hardened Recall
corebot	0.97	**0.97**
dircrypt	0.95	0.93
qakbot	0.94	**0.94**
ramnit	0.94	**0.94**
lockyv2	0.87	0.84
cryptolocker	0.87	**0.88**
simda	0.75	**0.79**
krakenv2	0.72	**0.76**
pykspa	0.67	**0.71**
ramdo	0.54	**0.54**

To counter adversarial attacks, novel proposals introduce the paradigm of *adversarial learning* [6], in which adversarial samples are included in the training dataset to harden the ML detector. As an example, authors in [36] demonstrate the advantages of adversarial learning by enriching the training set of the classifier with adversarial samples produced by the GAN. Table 7 compares the detection rates of the RF classifier before and after this hardening process. Cells with a grey background represent the DGA for which the detection rate improved after adversarial learning (it should be noted that the dataset used for this test is different than that used for the experiments reported in Table 6). Detection rates for 8 out of 10 DGA families improved, thus showing the validity of adversarial learning.

D. Selection of a machine learning algorithm

Unbiased comparison of the effectiveness of two ML algorithms requires that they are both trained on the *same* training dataset and tested on the *same* dataset [3]. Even though many cyber security proposals rely on few and old public datasets, their results are not comparable due to several causes: the two algorithms consider different features; one or both algorithms may implement pre-filtering operations that alter the training dataset; and they may use a different split between test and training dataset. For these reasons, meaningful comparisons between detection performance in literature are extremely difficult. For example, papers such as [4] and [5] discuss ML methods for two cyber security problems, but they do not consider the different training and testing environments of the analysed works. Hence, although some solutions achieve higher accuracy than others, it is possible that results change significantly under different training settings. Furthermore, there is no guarantee that a method performing best on a test dataset confirms its superiority on different datasets.

Security administrators should be aware of this issue, and should thoroughly question the evaluation methodology before accepting the performance results of different machine learning algorithms.

E. False positives and false negatives

The implicit cost of a misclassification in the cyber security domain is a serious problem. False positives in malware classification and intrusion detection annoy security operators and hinder remediation in case of actual infection. In phishing detection, they might cause important, legitimate messages to not be delivered to end users. In contrast, failing to detect malware, a network intrusion or a phishing email can compromise an entire organization. We explore this problem by considering the performance of ML solutions devoted to malware analysis and phishing detection [27], while we perform an original experiment for intrusion detection that is oriented to detect DGA in a real, large enterprise.

For malware analysis, we consider the approach in [24] that proposes an original and effective method for malware classification. This paper contains a detailed analysis and comparison of different ML techniques which were trained and tested on the same datasets, thus satisfying the requirements for valid comparison of different techniques. Hence, we deem this paper to be a good representation of the state-of-the-art of ML for determining the family to which a malware sample belongs. The evaluation is performed on the DREBIN dataset;[4] for large malware families the proposed approach, which outperforms all other baselines, obtains an F1-score of 0.95, whereas for small malware families it achieves an F1-score of 0.89.

For phishing detection, we report the results described in [27] that, to the best of our knowledge, is the only paper on phishing email detection which compares different ML algorithms against the same comprehensive dataset. Therefore, we consider this work as a valid overview of the efficacy of different ML methods. The authors created a custom dataset of ~3,000 phishing emails on which several ML classifiers were tested: the best results were obtained by RF (lowest false positives) and LR (lowest false negatives), obtaining an F1-score of 0.90 and 0.89, respectively.

The scenario for intrusion detection is different, as modern solutions can achieve higher Accuracy scores [3]. Although near-perfect Accuracy may seem an appreciable result, the massive amounts of events generated daily in a large enterprise account for hundreds to thousands of false positives that need to be manually triaged by security operators. We highlight this problem through an original experiment. We consider two DGA detectors based on the self-developed Random Forest classifiers trained on the first and second datasets of Table 2, respectively. We then validate them on the real domain dataset. Results are summarized in Table 8 which presents the number

4 DREBIN dataset: https://www.sec.cs.tu-bs.de/~danarp/drebin/

of domains that are flagged as DGA by both classifiers, alongside its percentage on the total amount of domains included in the dataset. We can observe that the two classifiers obtain comparable detection performances on real traffic data, as they both signal about 400 domains. However, manual inspection revealed that they were not DGA, hence all the domains flagged as DGAs are actually false positives. As anticipated, even a false positive rate of 2% can account to hundreds of false alarms in a real organization.

TABLE 8. PERFORMANCE OF THE DGA DETECTION CLASSIFIERS WHEN USED ON REAL DATA.

Classifier	Training Dataset	Domains classified as DGA
1	Well-known	431 (2.16%)
2	Well-known and recent	397 (1.99%)

Despite these apparently promising results which are well beyond acceptable levels in other fields such as image recognition, these approaches are affected by an excessive number of false positives and false negatives to be considered for cyber defences without human supervision.

F. Re-training issues

A well-known limitation of traditional detection approaches based on static detection rules is the need for frequent and continuous updates (e.g., daily updates of antivirus definitions). A similar issue also influences advanced ML approaches; reliance on outdated training datasets leads to poor detection performance. This is a critical problem for all supervised learning approaches requiring labelled training datasets; the manual creation of similar datasets is an expensive process because they need to be sufficiently large and comprehensive to allow the algorithm to learn the difference between the classes. Furthermore, these operations are error prone and may lead to incorrect classifications. Finally, most organizations are unwilling to share their internal network data. This scenario leads to an overall scarcity of publicly available and labelled data for cyber security, thus rendering periodic retraining extremely difficult or impossible.

To show the detrimental effects of obsolete training sets, we perform an experiment comparing the performance of two instances of the same self-developed RF classifier for DGA detection. The first and second instances are trained with the first and second datasets reported in Table 2. Both classifiers are tested against the same synthetic domain dataset described in Section 4. We report the results in Table 9, which shows the Precision, Recall and F1-score obtained by the two classifiers for DGA detection. As expected, the performance of the second classifier is significantly better because

it obtains an F1-score for DGAs of 0.89 against 0.33. These results demonstrate that classifier performances are extremely sensitive to the freshness of the training set.

TABLE 9. PERFORMANCE OF THE DGA DETECTION CLASSIFIERS WHEN TRAINED ON OUTDATED AND RECENT DATASETS.

Classifier	Training Dataset	F1-score	Precision	Recall
1	Well-known	0.3306	0.1984	0.9913
2	Well-known and recent	0.8999	0.9126	0.8875

G. Deployment process

Security solutions based on ML achieve appreciable detection rates only if the training dataset is appropriate and the parameters of the algorithms are finely tuned. In most scenarios, these operations are still executed empirically and represent a resource intensive task that presents several risks. If these steps are not performed rigorously and/or training is not based on the right datasets, the results are underwhelming. We highlight these issues through a set of ML experiments applied to network intrusion detection. The goal is to show the considerably different results achieved by the same ML algorithm in different environments where either the number of features or the training dataset is changed. To this purpose, we rely on the RF classifier for network intrusion detection. We train it using the third dataset reported in Table 3 by choosing 5, 7, 10 or 12 features, selected through a *feature agglomeration* process; the testing phase is performed on the test dataset. We report the Precision, Recall and F1-score for the five sets of features in Table 10, where we observe that the same classifier yields different results, especially with regards to its Recall, with values ranging from 0.57 to 0.74.

TABLE 10. PERFORMANCE OF THE INTRUSION DETECTION CLASSIFIER WHEN TRAINED WITH DIFFERENT FEATURES.

Features	F1-score	Precision	Recall
12	0.7985	0.8727	0.7361
10	0.7801	0.8684	0.7093
7	0.7476	0.8893	0.6448
5	0.6920	0.8724	0.5734

Then, we keep the number of features fixed at 12 and we repeat the training process two more times by using the first and then the second dataset reported in Table 3, and then test them on the same testing dataset. Table 11 reports the Precision, Recall and F1-score for the three training datasets. These results confirm that the Recall between the best and the worst case may differ by 10% or over.

TABLE 11. PERFORMANCE OF THE INTRUSION DETECTION CLASSIFIER WHEN TRAINED ON DIFFERENT DATASETS.

Training Dataset	F1-score	Precision	Recall
1	0.7306	0.8753	0.6270
2	0.7757	0.8703	0.6996
3	0.7985	0.8727	0.7361

5. CONCLUSIONS

Machine and deep learning approaches are increasingly employed for multiple applications and are being adopted also for cyber security, hence it is important to evaluate when and which category of algorithms can achieve adequate results. We analyse these techniques for three relevant cyber security problems: intrusion detection, malware analysis and spam detection. We initially propose an original taxonomy of the most popular categories of ML algorithms and show which of them are currently applied to which problem. Then we explore several issues that influence the application of ML to cyber security. Our results provide evidence that present machine learning techniques are still affected by several shortcomings that reduce their effectiveness for cyber security. All approaches are vulnerable to adversarial attacks and require continuous re-training and careful parameter tuning that cannot be automatized. Moreover, especially when the same classifier is applied to identify different threats, the detection performance is unacceptably low; a possible mitigation can be achieved by using different ML classifiers for detecting specific threats. Deep learning is still at an early stage and no final conclusion can be drawn. Significant improvements may be expected, especially considering the recent and promising development of adversarial learning. Our takeaway is that machine learning techniques can support the security operator activities and automate some tasks, but pros and cons must be known. The autonomous capabilities of ML algorithms must not be overestimated, because the absence of human supervision can further facilitate skilled attackers to infiltrate, steal data, and even sabotage an enterprise.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015.

[3] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, 2015.

[4] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, 2008.

[5] J. Gardiner and S. Nagaraja, "On the Security of Machine Learning in Malware C8C Detection," *ACM Computing Surveys*, 2016.

[6] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial Machine Learning," in *ACM workshop on Security and artificial intelligence*, 2011.
[7] F. Pierazzi, G. Apruzzese, M. Colajanni, A. Guido, and M. Marchetti, "Scalable architecture for online prioritization of cyber threats," in *International Conference on Cyber Conflict (CyCon)*, 2017.
[8] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," in *IEEE International Conference on Platform Technology and Service (PlatCon)*, 2016.
[9] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of Recurrent Neural Networks for Botnet detection behavior," in *IEEE Biennial Congress of Argentina (ARGENCON)*, 2016.
[10] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.
[11] G. D. Hill and X. J. Bellekens, "Deep Learning Based Cryptographic Primitive Classification," *arXiv preprint*, 2017.
[12] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.
[13] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *IEEE National Aerospace and Electronics Conference (NAECON)*, 2015.
[14] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016.
[15] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and Its Applications*, 2015.
[16] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL4MD: A Deep Learning Framework for Intelligent Malware Detection," in *International Conference on Data Mining* (DMIN), 2016.
[17] G. Tzortzis and A. Likas, "Deep belief networks for spam filtering," in *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2007.
[18] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *International Conference in Swarm Intelligence*, 2015.
[19] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2014.
[20] S. Ranjan, *Machine learning based botnet detection using real-time extracted traffic features*, Google Patents, 2014.
[21] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "Peerrush: mining for unwanted p2p traffic," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2013.
[22] A. Feizollah and e. al, "A study of machine learning classifiers for anomaly-based mobile botnet detection," in *Malaysian Journal of Computer Science*, 2013.
[23] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: detecting the rise of DGA-based malware," in *USENIX Security Symposium*, 2012.
[24] T. Chakraborty, F. Pierazzi, and V. Subrahmanian, "Ec2: Ensemble clustering and classification for predicting android malware families," *IEEE Transactions on Dependable and Secure Computing*, 2017.
[25] C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," *Journal of Computer Virology and Hacking Techniques*, 2015.
[26] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, "On the feasibility of online malware detection with performance counters," in *ACM SIGARCH Computer Architecture News*, 2013.
[27] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *ACM Proceedings of the Anti-Phishing Working Groups*, 2007.
[28] G. Xiang, J. Hong, C. P. Rose and, L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security (TISSEC)*, 2011.
[29] G. Apruzzese, M. Marchetti, M. Colajanni, G. Gambigliani Zoccoli, and A. Guido, "Identifying malicious hosts involved in periodic communications," in *IEEE International Symposium on Network Computing and Applications (NCA)*, 2017.
[30] F. S. Tsai, "Network intrusion detection using association rules," *International Journal of Recent Trends in Engineering*, 2009.

[31] F. Bisio, S. Saeli, L. Pierangelo, D. Bernardi, A. Perotti, and D. Massa, "Real-time behavioral DGA detection through machine learning," in *IEEE International Carnahan Conference on Security Technology (ICCST)*, 2017.
[32] Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang, "An intelligent PE-malware detection system based on association mining," *Journal in computer virology*, 2008.
[33] W.-F. Hsiao and T.-M. Chang, "An incremental cluster-based approach to spam filtering," *Expert Systems with Applications*, 2008.
[34] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, 2014.
[35] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic Analysis of Malware Behavior Using Machine Learning," *Journal of Computer Security*, 2011.
[36] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-Tuned Domain Generation and Detection," in *ACM Workshop on Artificial Intelligence and Security*, 2016.
[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.
[38] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, 2009.
[39] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," *Journal of Big Data*, 2015.
[40] A. Khan, B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of advances in information technology*, 2010.