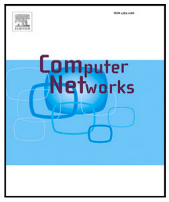
Computer Networks 214 (2022) 109073

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Adversarial machine learning for network intrusion detection: A comparative study

Houda Jmila [a], Mohamed Ibn Khedher [b,*]

[a] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France
[b] IRT-SystemX, 2 Boulevard Thomas Gobert , 91120 Palaiseau, France

ARTICLE INFO

Keywords:
Networ Intrusion detection
AI adversarial robustness
Adversarial attack
Defense technique
NSL-KDD
UNSW-NB15

ABSTRACT

Intrusion detection is a key topic in cybersecurity. It aims to protect computer systems and networks from intruders and malicious attacks. Traditional intrusion detection systems (IDS) follow a signature-based approach, but in the last two decades, various machine learning (ML) techniques have been strongly proposed and proven to be effective. However, ML faces several challenges, one of the most interesting being the emergence of adversarial attacks to fool the classifiers. Addressing this vulnerability is critical to prevent cybercriminals from exploiting ML flaws to bypass IDS and damage data and systems.

Some research papers have studied the vulnerability of ML based IDS to adversarial attacks, however most of them focused on deep learning based classifiers. Unlike them, this paper pays more attention to shallow classifiers that are still widely used in ML-based IDS due to their maturity and simplicity of implementation. In more detail, we evaluate the robustness of 7 shallow ML-based NIDS including Adaboost, Bagging, Gradient boosting (GB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Classifier (SVC) and also a Deep Learning Network, against several adversarial attacks widely used in the state of the art (SOA). In addition, we apply a Gaussian data augmentation defense technique and measure its contribution to improving classifier robustness [1]. We conduct extensive experiments in different scenarios using the NSL-KDD benchmark dataset [2] and the UNSW-NB 15 dataset [3]. The results show that attacks do not have the same impact on all classifiers and that the robustness of a classifier depends on the attack and that a trade-off between performance and robustness must be considered depending on the network intrusion detection scenario.

1. Introduction

Protecting computer systems and networks from cyberattacks has been a growing concern in recent years. Although most systems are built with improved security features, a large number of vulnerabilities still exist. These include unwanted access to systems and information, destruction or alteration of data, etc. Intrusion detection systems play a critical role in the network defense process and allow network operators to accurately identify security attacks. There are mainly two categories of IDS: Network-based IDS and Host-based IDS, described below:

- *Network-based IDSs (NIDS)* monitor and analyze network traffic at different layers to detect intruders.
- *Host-based IDS (HIDS)*, monitor the computer infrastructure to detect internal changes by exploiting host indicators such as sensor log files, disk resources, user account information processes, etc.

This paper focuses on the Network-based IDSs. The continuous increase in the number and types of contemporary network threats [4] motivates this interest.

Both NIDS and HIDS approaches can be classified into the following categories:

- *Misuse-based approaches* (also called *signature-based*) exploit indicators (or signatures) previously extracted from *known* attacks. Signatures are manually generated for each new attack. Therefore, maintaining an up-to-date list of signatures is costly due to the increasing number and diversity of attacks.
- *Anomaly-based approaches* model normal network behavior, as opposed to malicious behavior. Although these approaches are capable of detecting new attacks, they suffer from a high false alarm rate because *new normal behavior* can be detected as malicious.

* Corresponding author.
E-mail address: ibnkhedhermohamed@hotmail.com (M.I. Khedher).

https://doi.org/10.1016/j.comnet.2022.109073
Received 6 October 2021; Received in revised form 26 April 2022; Accepted 23 May 2022
Available online 3 June 2022

Nomenclature

AAC	Anomaly prediction ACcuracy
AdvML	Adversarial Machine Learning
ANN	Artificial Neural Network
BIM	Basic Iterative Method
C&W	Carlini and Wagner
CNN	Convolutional Neural Networks
DDoS	Distributed Denial-of-Service
DNN	Deep Neural Network
DT	Decision Tree
FFNN	Feedforward Neural Network
FGSM	Fast Gradient Sign Method
FNR	False Negative Rate
GAN	Generative Adversarial Networks
GAN	Generative adversarial network
GB	Gradient Boosting
GMM	Gaussian Mixture Model
HIDS	Home based IDS
HSJ	Hop Skip Jump
IDS	Intrusion Detection System
IoT	Internet Of Things
JSMA	Jacobian based Saliency Map Attack
KNN	K-Nearest Neighbors
ML	Machine Learning
NIDS	Network-based IDS
PGD	Projected Gradient Descent
RF	Reinforcement Learning
RNN	Recurrent Neural Network
SOA	State Of the Art
SVC	Support Vector classifier
SVM	Support Vector Machine
TAC	Total prediction ACcuracy
UAP	Universal Adversarial Perturbations
ZOO	Zeroth-order optimization

Anomaly detection is often considered by the community to be more promising than signature-based detection, as it is able to detect *unknown attacks*. Therefore, this paper focuses on anomaly-based NIDS.

In recent years, ML approaches have been widely used for anomaly detection. Existing approaches can be classified into *shallow (or classic) models* [5] and *deep learning models* [6,7]. Deep learning involves several levels of representation and several layers of non-linear processing units. On the contrary, all non-deep learning approaches can be qualified as shallow learning, this includes the majority of conventional machine learning models proposed prior to 2006 and neural networks with only one hidden layer of nodes [8]. The most popular shallow approaches include Random Forest (RF), Decision Tree(DT), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Hidden Markov Models (HMM) and Ensemble Learning. Both shallow and deep learning models have been used with promising results.

While most research focuses on designing new ML-based IDSs, this paper highlights the vulnerabilities of ML systems to adversarial attacks. Adversarial attacks allow a small and carefully designed change in the input of the ML classifier to completely alter the output of the system. *Adversarial Machine Learning* (AdvML) is the research area that studies these vulnerabilities. It has been widely explored in recent years, particularly in the field of computer vision [9]. The study of AdvML in cybersecurity also deserves a great deal of attention given the sensitivity of this field and the need to preserve the confidentiality, integrity, and availability of data and systems. It is essential to evaluate the *robustness* of ML-based intrusion detection systems before deploying them in the network. This prevents cyber criminals from exploiting ML vulnerabilities to bypass IDS and damage data and systems. The *robustness* of an ML classifier is defined as its ability to maintain its accuracy against *adverse samples*. An *adverse sample* is an input instance with a small disturbance that is erroneously predicted. Depending on the results of the robustness assessment, appropriate defense techniques can be applied to improve the robustness of NIDS.

Due to the widespread adoption of deep learning approaches for NIDS, most research work evaluate the robustness of deep learning based NIDS [10]. However, shallow ML models are still widely used in NIDS due to their simplicity and implementation maturity [11]. It is therefore interesting to study their robustness in an adversarial environment. This paper focuses on the evaluation of shallow ML based NIDS against several adversarial attacks widely used in the state of the art.

In this paper, we evaluate the robustness of 7 shallow classifiers including Adaboost, Bagging, Gradient boosting, Logistic regression, Decision Tree, random forest, Support Vector Classifier and also a Deep Learning Network, against a wide range of attacks (an attack is defined as a method of generating adversarial examples). In particular, we consider white-box and gray/black-box attacks. In white-box attacks the attacker has full access to all information about the ML-based NIDS, whereas in gray/black-box attacks, the attacker has little or no knowledge of ML-based NIDS. Gray/black-box attacks are interesting because they represent the most realistic scenario for adversary's attacks. Examining white-box attacks is useful for IDS manufacturers who has full access to their system and wish to evaluate its performance against adversarial attacks.

This document provides the following main contributions:

- A clear and structured survey of most commonly used adversarial attacks and defense techniques, in addition to an exhaustive review of current work on Adversarial ML NIDS.
- An in-depth study of the impact of adversarial attacks on ML based NIDS. Several types of attacks (9 white-box and gray/black-box attacks) are explored with a particular attention to shallow classifiers. Indeed, unlike the overwhelming majority of works that study the behavior of NIDS in an adversarial environment and focus on deep learning approaches, this paper focuses on shallow algorithms, which are still widely used in ML-based NIDS thanks to their simplicity of implementation and maturity. The evaluation of their performance in an adversarial environment is therefore also worth exploring.
- An evaluation of the contribution of a Gaussian data augmentation defense technique to improving the robustness of the classifiers.
- Valuable results and conclusions that can help security researchers improve the robustness of their NIDS. These results are deduced based on extensive experiments conducted under different scenarios.
- The steps in the study conducted represent a framework detailing the steps to be taken to assess the sensitivity of NIDS to adversary attacks and improve their robustness.

The paper is structured as follows. Section 2 describes the challenges in the field of network intrusion detection. Section 3, provides a state of the art of the most commonly used adversarial attacks and defense techniques, as well as an exhaustive study of AdvML approaches in the field of NIDS. Section 4 describes our evaluation study, including the evaluation parameters and protocol. Section 5 details the experimental results. Section 6 provides a discussion and Section 7 concludes the paper.

2. The challenging task of network intrusion detection

Network intrusion detection is a complex task for many reasons. Challenges can be related to the nature of the network traffic data, or to the inherent NIDS decision model, as described below.

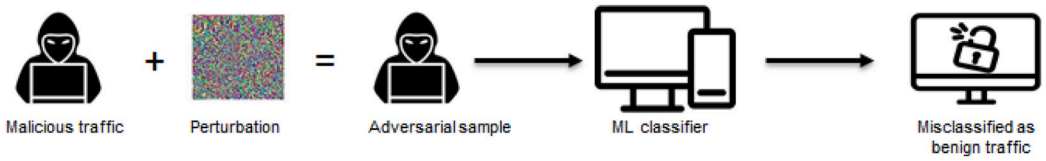


Fig. 1. Adversarial attack generation in NIDS. An adverse sample is generated by adding a small perturbation to the original sample. Thus, malicious perturbed traffic can be misclassified as benign and thus bypass the intrusion detection system. This can have serious consequences for the system.

2.1. Challenges related to the nature of the network traffic data

- *Challenges related to data exploitation*: in network anomaly detection, the captured packets partially represent the entire network traffic because the observation points are often widely distributed. Thus, the captured data is often sparse, huge and contains redundant or uninformative data, which makes it difficult to exploit.
- *Unbalanced data-sets*: In most IDS data-sets, the amount of normal data dominates the data. This is due to the low frequency of attacks compared to normal behavior. This makes the available data-sets unbalanced. Therefore, classical machine learning algorithms need to be adapted to the context of unbalanced data-sets.
- *Variety of attacks*: the attack landscape is frequently changing as attackers are constantly developing new methods. Attack detection and analysis tools must be updated and evolve continuously.

2.2. Challenges related to the decision model

- *Scalability*: this is a common challenge for statistical learning algorithms. It is defined by the ability of the algorithm to function normally even with high dimensional data.
- *Real-time IDS:* The goal of an IDS is to detect attacks and stop them before they damage the system. Therefore, the design of a real-time IDS is very important. A real-time IDS must also be efficient and flexible to run on most commercial computers.
- *High false positive rate:* This is the reporting of a high number of false alarms that correspond to legitimate activity that has been misclassified by the IDS. Recognizing true alarms from the huge volume of alarms is a complicated and time consuming task. Therefore, reducing false alarms is a serious problem to ensure the effectiveness and use of IDS.
- *Vulnerability to Adversarial Attacks:* This aspect has been described above and is the focus of this paper. The objective is to examine and reduce the vulnerability of various NIDS classifiers to adversarial attacks [12,13].

3. Adversarial attacks and defense techniques in NIDS: Background and review

3.1. Preliminaries

Generating an adversarial attack involves adding a small perturbation to the input sample so that the output label is misclassified [14,15]. This is illustrated in Fig. 1 in the context of NIDS. Formally, let x be the original input data sample, f be the classifier, and $y = f(x)$ be the label associated with x. A data sample x' is considered an adverse sample of x when x' is close to x under a specific distance metric while $f(x') \neq y$. Adversarial attacks in network security can be classified along two dimensions: *the attacker's knowledge* and *the attacker's goal*:

1. *The attacker's knowledge*: describes the extent of the adversary's knowledge about the NIDS system. We can characterize three levels of attack danger [16]:
   - *White-box attacks*: the attacker is in the most favorable position where he has full access to all information about the ML-based NIDS. This includes training data and the learning model architecture, decision and parameters (gradient, loss function, etc.). Fortunately, this is generally not feasible in the majority of real adversarial attacks.
   - *Black-box attacks:* This is the opposite case where the attacker completely ignores the ML-based NIDS system and its inputs/outputs. It can be argued that a truly black-box attack is impossible and rarely succeeds.
   - *Gray-box attacks:* this scenario assumes a more realistic approach, where the attacker has some level of knowledge of the ML-based NIDS, and may have limited access to the training data. The adversary does not have the exact information but has enough information to be able to attack the ML system and cause it to fail.

   Note that in the literature, by abuse of language, the term "black-box attacks" is also used for "gray-box attacks" (for example, the ZOO attack is called back-box attack in [17]). In this article, we use the terms "gray/black-box" to refer to the gray-box attacks described below, to nuance between this definition and the term "black-box" widely used in the literature.
2. *The attacker's goal*: depends on whether he simply wants to deceive the system, or to induce a precise prediction for certain inputs. Two forms of attack can be listed:
   - *Targeted attacks:* direct the ML algorithm to a specific class, i.e., the adversary tricks the classifier into predicting all adversary examples as a specific target class.
   - *Non-targeted attack:* aims to misclassify the input sample away from its original class, regardless of the new output class. They are easier to implement because more alternatives are available to reorient the output. Note that in binary classification problems, targeted and untargeted attacks are equivalent.

3.2. Survey of adversarial attack generation approaches

As many adversarial attack generation approaches can be applied to both targeted and untargeted scenarios, we will rather rely on the attacker's knowledge to classify them.

3.2.1. White-box attacks

Fast Gradient Sign Method (FGSM) [18]. creates adversarial examples by adding noise to the original sample along the gradient directions.

Two iterative extension of FGSM, namely, *Basic Iterative Method (BIM) [19]* and *Projected Gradient Descent (PGD) [20]* have been also used in the recent literature.

The Jacobian-based Saliency Map Attack (JSMA) [21]. generates adversarial examples using forward derivatives (i.e., model Jacobian). JSMA iteratively perturbs features/components of the input one at a time instead of perturbing the whole input to fool the classifier.

Universal Adversarial Perturbations (UAP) [22]. are a special type of untargeted attacks that consist on creating a constant perturbation that successfully misclassifies a specified fraction of the input samples.

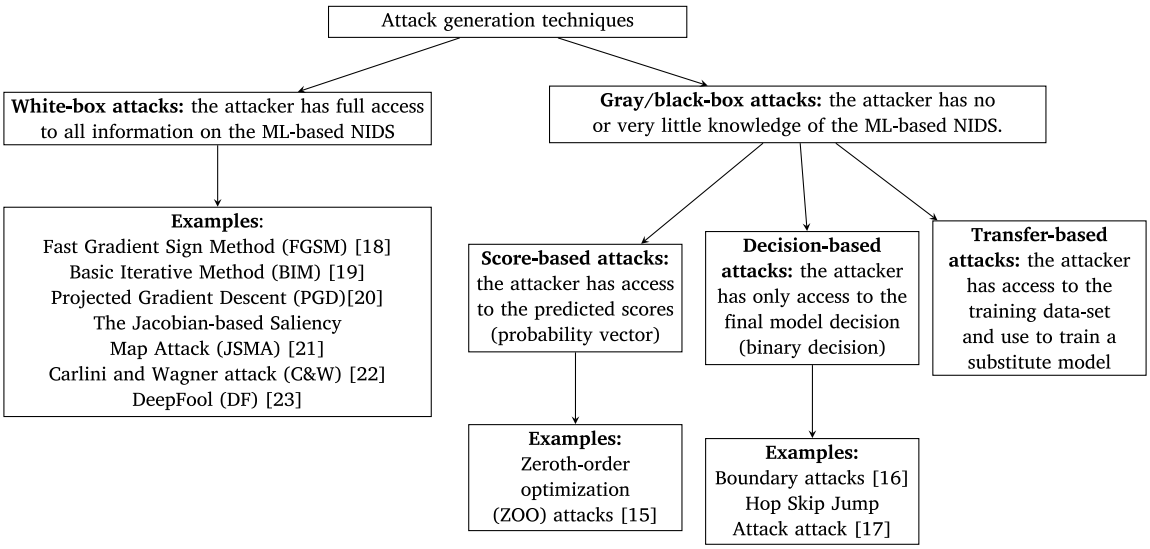


Fig. 2. Adversarial attack techniques.

DeepFool (DF) [23]. is an untargeted attack based on computing the minimum distance between the original input and the decision boundary.

Carlini and Wagner attack (C&W) [24]. The authors formulate the search for an adversarial sample as an optimization problem with the following objective:

$$\min_{\epsilon} D(x, x+\epsilon) + c.f(x+\epsilon) \quad \text{subject to} \quad x+\epsilon \in D$$

where ϵ denotes the adversarial perturbation, $D(.,.)$ denotes the ℓ_0, ℓ_0 or ℓ_∞ distance metric, and $f(x+\epsilon)$ define the cost function such that $f(x+\epsilon) \geqslant 0$ if and only if the model correctly classifies $x+\epsilon$ (i.e., gives it the same label as x).

3.2.2. Gray/black-box attacks

Score-bases attacks. : the attacker has access to the predicted scores, i.e., the probability of each predicted label to belong to the classification classes of the model. Examples include the *Zeroth-order optimization (ZOO) attacks [17].*

Decision-based attacks. : the attacker only has access to the final decision of the model (binary decision) without any confidence score. Examples include *Boundary attacks [25]* and *Hop Skip Jump Attack [26].*

Transfer-based attacks. : the attacker has access to the hole or part of the training data-set and use it to train another fully observable model, called "a substitute model" intending to emulate the attacked model called "target model". Adversarial perturbations that can be synthesized from the "substitute model" are used to attack the "target model".

We refer the reader to [27–30] for more additional information on adversarial attacks. Fig. 2 summarizes the different Adversarial attack generation techniques described above.

3.3. Defense

A defense technique aims at improving the robustness of the model against adversarial attacks. In [31], the three following categories of defense techniques are highlighted:

- *Modify the input data:* These techniques do not deal directly with training models, but rely on modifying the training data during training or modifying the input data during testing. For example *Gaussian data augmentation [32]* technique involves augmenting the original data-set with copies of the original samples to which Gaussian noise has been added. The underlying idea is that forcing the model to make the same prediction for a true instance and its slightly perturbed version should increase its generalization capabilities. This method is widely used because of its simplicity, ease of implementation and effectiveness against both gray/back-box and white-box attacks.
- *Modify the classifier:* This involves modifying the original classification model by changing the loss functions, adding additional layers/sub-networks, etc. For example, the *Gradient Masking* method modifies a machine learning model to mask its gradient from an attacker.
- *Add an external model:* these methods keep the original model intact and add one or more external models to it during testing. For example, the authors of [33] used *Generative Adversarial Networks* (GAN) to train the network along a generator network that attempts to generate a perturbation to that network.

Fig. 3 summarize the different defense techniques described above.

3.4. Defense and attacks in IDS

Table 1 presents and compares recent research on ML based NIDS in adversarial environment. For each research work, we highlight (i) the evaluated ML classifiers (ii) the evaluation data-set (iii) the adversarial attack algorithms and (iv) the defense techniques, if any. In particular, we classify the evaluated ML classifiers into two categories: *shallow* and *deep* learning. We also divide the adversarial attack generation techniques into *State of the art techniques*, i.e. techniques inspired by the field of computer vision, and *new techniques designed by the authors*.

The first row of the table, presents a statistic revealing the trends in the literature. It can be seen that the majority of the literature (95%) evaluate the robustness of deep learning techniques, while a minority (37%) evaluates shallow learning. The majority of the latter focus on a single type of adversarial attack, proposed by the authors, and do not address the various adversarial attacks widely used in the literature.

The evaluation of shallow ML based NIDS under adversarial environment requires further study. To fill this gap, this paper evaluates shallow ML based NIDS against the most used attack generation approaches in the literature. Only [41] have already addressed this issue, but the authors did not explore defense techniques.

In more detail, we evaluate diverse and widely used ML algorithms [76] in the NIDS domain when exposed to white-box and

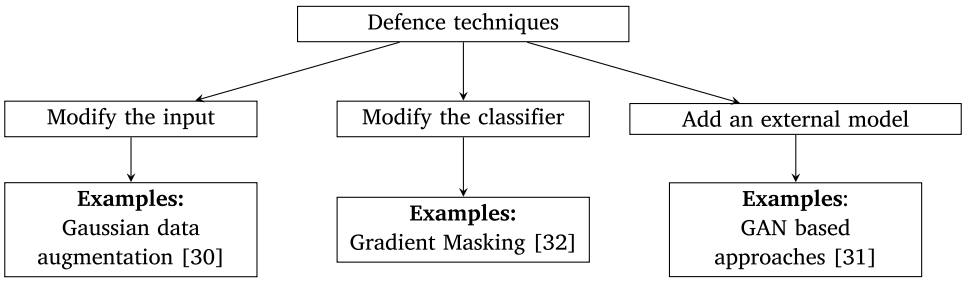


Fig. 3. defense techniques (Ref. [34]).

gray/back-box adversarial attacks generated by well know SOA algorithms. Furthermore, we explore the effect of the Gaussian data augmentation defense technique on different classification settings.

4. Evaluating IDS robustness against adversarial attacks

The white-box attacks investigated in this paper are: FGSM attack, BIM, PGD attack, JSMA, DeepFool attack, Carlini and Wagner attack. The gray/black-box examined attacks are: Zoo Attack, Boundary attack and Hop Skip Jump Attack. Moreover, we generate additional adversary attacks in a naive way using Gaussian noise, with different intensities (σ values are 0.01, 0.1 and 0.2).

In order to fairly compare the robustness of diverse classifiers, we have to apply the same attacks, with the same configuration and the same hyper-parameters to all classifiers. However, white-box attacks are highly dependent on the type of classifier they attack, e.g., FGSM, BIM, PGD, and JSMA use the gradient of the classifier to generate the attacks, and thus can only be applied to gradient-based classifiers.

To overcome this problem, we propose to use an external DNN-based surrogate classifier, which we call "*Generator*" to which we apply all white-box attacks in order to generate the adversary samples, *under the same conditions*. The samples generated by each type of white-box attack are then introduced in the classifiers in order to measure their robustness against such an attack. This idea is based on the *transferable property of adversarial attacks* [77], which shows that the effect of the attack can be transferred to other ML models, including the "Generator" in our case. We use a DNN composed of 7 completely connected layers with dimensions ranging from 1024 to 32. From one layer to another, the dimension is divided by 2. The architecture of the generator is different from the evaluated neural network. It is more complex to make the generation of the adversarial samples more complex.

To improve the robustness of IDSs, defensive techniques can be applied. To obtain the most robust NIDS system, the manufacturer must follow an iterative procedure:

1. Build a basic NIDS
2. Evaluate its robustness against a set of adversarial attacks
3. Apply a defense technique to increase its robustness
4. Repeat (2) and (3) until the level of robustness of the model is acceptable.

In this paper, we focus on steps (2) and (3), i.e. we evaluate the robustness of the classifiers against adversarial attacks, then apply **the Gaussian data augmentation** defense technique and measure its contribution to improving its robustness.

In the following, we describe the experimental setup, then present the results.

4.1. Data-sets description

4.1.1. NSL-KDD data-set [2]

The NSL-KDD data-set is derived from the KDDCup 99 data-set and addresses the problems of the latter, namely irrelevant records and data imbalance between normal and abnormal records. A record is defined by 41 features, including 9 basic features of individual TCP connections, 13 content features within a connection, 9 temporal features calculated within a two second time window, and 10 other features. The data-set contains 24 attack types, grouped into 4 categories of attacks, namely denial of service (DoS), remote to local (R2L), user to root (U2R) and probing. The data-set is divided into training and test subsets containing 100.778 and 25.195 samples respectively. The original training NSL-KDD Train$^+$ and testing NSL-KDD Test$^+$ sets of the NSL-KDD data-set are used in this study.

4.1.2. UNSW-NB15 [3]

The data-set was created by the cyber security research group at the Australian Centre of Cyber Security (ACCS) in 2015. It contains approximately 100 GB of raw data (normal and malicious traffic). 9 different types of attacks were used to generate the malicious traffic: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Each sample is described by 49 features that were generated by several feature extraction tools. The data-set is divided into training and test subsets containing 175.341 and 82.332 samples respectively. The original training UNSW_NB15_training-set and 10% (randomly selected samples) of the testing UNSW_NB15_testing-set are used in this study.

4.1.3. Data pre-processing

Furthermore, the training set is divided into training and validation sets according to **8:2**. In order to provide more suitable data for the classifier, One Hot Encoding and Data Normalization steps are performed. One Hot Encoding involves encoding the categorical attributes, such as protocol, service, and state, into one-hot numeric array. This data is then normalized between 0 and 1 to produce more homogeneous values.

4.2. The evaluated classifiers

We evaluate the performance of various well known ML algorithms: Adaboost, Bagging, Gradient boosting, Logistic regression, Decision Tree, Random Forest, Support Vector Classifier (SVC) and also a Deep Learning Network. Binary classification is considered. All our models were implemented using the TensorFlow [78], Keras [79] and scikit-learn packages [80]. To generate adversarial samples, we use the open-source IBM Robustness Toolbox (ART) framework [81]. All the hyper-parameters of the classifiers and algorithms used to generate the adverse samples *have been set to their default values* to facilitate the comparison of the different evaluation scenarios.

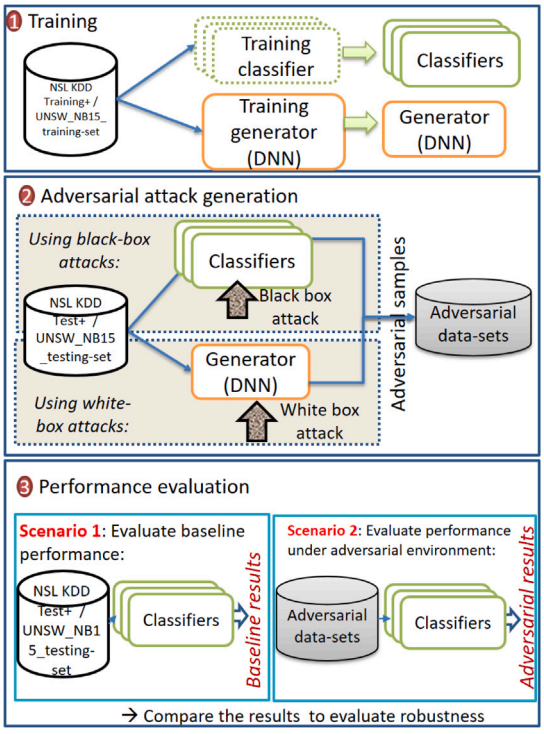
4.3. Robustness indicators

We consider the following two metrics:

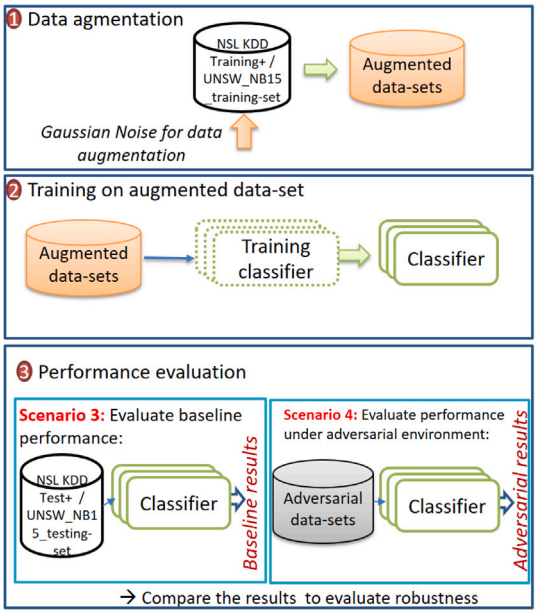
- **Accuracy**: measures the ability of the IDS to correctly classify the malicious and legitimate traffic. It represents the percentage of the number of correctly classified records out of the total number of records. $ACC = \frac{TP+TN}{TP+TN+FP+FN}$

Table 1
Summary of research works related to Document Alignment.

Work	Year	Algorithm		Data-set	Adversarial attack		defense technique
		Classic ML	Deep Learning		SOA	Designed	
(%)		37%	95%		53%	47%	53%
[35]	2018	✗	DNN	NS-LKDD	ZOO, Gan based attack	✗	✗
[36]	2018	✗	DNN	NSL-KDD	FGSM, JSMA DeepFool, C&W	✗	✗
[37]	2019	AdaBoost DT, GB KNN, LR RF, SVM	MLP	CTU [38] CICIDS [39] BOTNET [40]	✗	Alter some features values	Remove altered features
[41]	2019	LR, RF SVM	DNN	NSL-KDD	FGSM, PGD L-BFGS,SPSA	✗	✗
[42]	2019	✗	KitNET [43]	Kitsune [43]	FGSM JSMA C&W Elastic Net Method	✗	✗
[44]	2019	✗	FFNN, SNNs	BoT-IoT [45]	FGSM BIM, PGD	✗	Feature Normalization
[46] [47]	2019 2020	KitNET [43] DAGMM [48] BiGAN [49]	✗	CICIDS [39]	✗	Alter the packets to mimic benign traffic	Reconstruction from Partial Observation
[50]	2020	KNN, LR RF, SVM	✗	DARPA SYN flood [51] CICIDS [39]	✗	Perturb some important features	✗
[52]	2020	DT, LR NB, RF	MPL	ADFA-LD [53] DREBIN [54]	✗	GAN based attack Brute-force attack	✗
[16]	2020	DT, IF LR, SVM	Kitnet [43] MLP	Kitsune [43]	✗	Random Mutation and duplication of selected features	Adversarial training Feature selection Adversarial feature reduction
[55]	2020	✗	CNN, AE	MNIST [56]	FGSM, JSMA	✗	✗
[57]	2020	✗	CNN, RNN ANN	UNSW-NB15	FGSM, BIM, C&W PGD, DeepFool	✗	Min–max
[58]	2020	✗	DNN	UNSW-NB15	FGSM, BGA, BCA	✗	Min–max
[47]	2020	✗	KitNET [43]	MIRAI [59]	✗	Use saliency map to identify critical features and perturb them	✗
[60]	2020	✗	RNN	NSL-KDD	JSMA	✗	✗
[61]	2020	✗	MLP, CNN CNN, LSTM	CICIDS [39]	JSMA	✗	Model Voting Ensembling Ensemble Adversarial Training Adversarial Query Detection
[62] [63]	2020 2021	RF	Wide and Deep	CTU [38] BOTNET [40]	✗	Deep RL attacks	Deep RL based adversarial training
[64]	2021	J48 DT, RF NB, SVM	✗	EclipseIoT [65]	✗	Altering some selected features	✓
[66]	2021	LSTM, CNN, GRU	✗	CSE-CIC-IDS2018 [67]	FGSM	✗	✓
[68]	2021	✗	LSTM, CNN, GRU	CSE-CIC-IDS2018 [67]	FGSM	✗	✓
[69]	2021	✗	DNN	CSE-CIC-IDS2018 [67]	FGSM, BIM JSMA, DeepFool	✗	✓
[70]	2022	✗	MPL, LSTM, CNN	CSE-CIC-IDS2018 [67]	Nes, Boundary HopSkipJump, pointwise, Opt-attack	✗	✓
[71]	2022	RF	MLP	CIC-IDS2017 [67], IoT-23 [72]	✗	Adaptative perturbation pattern method	✗
[73]	2022	✗	LSTM, RNN	MedBIoT [74] and IoTID [75] dataset) [75].	✗	Altering features	✗
This paper	–	Adaboost Bagging DT, GB LR, RF SVC	DNN	NSL-KDD UNSW-NB15	GN,ZOO,BIM FGSM, JSMA, C&W, PGD BA, HSJ	✗	Gaussian data augmentation



(a) Initial robustness evaluation



(b) Applying defence techniques

Fig. 4. Evaluation protocol.

- **False Negative Rate**: is a more specific indicator that highlights the percentage of malicious traffic that has successfully passed the IDS. $FNR = \frac{FN}{FN+TP}$

Where **TP** are the True Positives and represent the number of anomalous records that are correctly identified as anomalies. The **TN** are the True Negatives and calculate the number of normal records that are correctly identified as normal. The **FP** are the False Positives that are the number of normal records that are mis-classified as anomalous. The **TN** are the True Negatives and represent the number of anomalous records that are identified as normal.

A good classifier is a classifier *having high accuracy and a low False Negative Rate.*

4.4. Evaluation protocol

4.4.1. *Initial robustness evaluation*

This scenario (see Fig. 4(a)) evaluates the performance of the classifiers against adversarial attacks.

To generate adversarial samples using white-box attacks, the samples of test data-sets (NSL-KDD Test$^+$ and UNSW_NB15_testing-set) are perturbed using gray/black-box attacks applied *directly to the classifier* and white-box attacks applied *to the "Generator" model*. The generated adversarial samples form new test data-sets, which we call, **"Adversarial data-sets"** (an adversarial data-set generated from NSL-KDD Test$^+$ and another generated from UNSW_NB15_testing-set). The performance of the classifiers are evaluated in the following scenarios:

- *Scenario 1: measuring performance in baseline scenario*
  - **Train:** NSL-KDD Train$^+$ / UNSW_NB15_training-set
  - **Test:** NSL-KDD Test$^+$ / UNSW_NB15_testing-set
- *Scenario 2: measuring performance in adversarial environment*
  - **Train**: NSL-KDD Train$^+$ / UNSW_NB15_training-set
  - **Test:** Adversarial data-sets

The first sub-tables of Tables 4 and 5 in the appendix shows the results of *accuracy*, whereas the second sub-tables shows the results for the *False Negative Rate*.

For each table, the first row shows the results of the first scenario whereas other rows illustrate the results of the second scenario. The difference in performance between the baseline and adversarial scenarios is highlighted **in red** for increase and **yellow** for decrease, for each classifier. For example, if a classifier's accuracy is 50% in the baseline scenario, if its performance decreases to 47% in the adversarial scenario, 3% is highlighted in yellow, if its performance increases to 55% in the adversarial scenario, 5% is highlighted in red.

4.4.2. *Applying defense techniques*

In this scenario (see Fig. 4(b)), we measure the contribution of Gaussian data augmentation to the robustness improvement of NIDS classifiers. Therefore, two new data-set called "augmented data-sets" are generated by applying Gaussian data augmentation on KDD Train$^+$ and UNSW_NB15_training-set. The performance of the classifiers is then evaluated in the following two scenarios:

- *Scenario 3: measuring performance in non adversarial environment with training in augmented data* **Train:** Augmented data-sets, **Test:** NSL-KDD Test$^+$ / UNSW_NB15_testing-set
- *Scenario 4: measuring the impact of defense techniques in adversarial environment* **Train:** Augmented data-sets, **Test:** Adversarial data-sets

The obtained results are described in the third and fourth sub-tables of Tables 4 and 5, for accuracy and FNR respectively. For each sub-table, the first row shows results of scenario 3, and the other rows illustrate the results of scenario 4.

Table 2
Summary of results for evaluation scenario 2.

Behavior against a Gaussian noise attack
• DT and GB are very vulnerable compared to other classifiers • For GB the attacks have caused an increase in False alarms. • Adaboost is the most robust although it has the worst performance in the baseline scenario. • The classifiers classify more malicious traffic as legitimate on UNSW-NB15 than on NSL-KDD (high false negative rates), especially for Bagging and Random Forest.
Behavior against Gray/black-box attacks
• The performance of the classifiers drops drastically when faced to HopSkipJump and Boundary attacks • Classifiers are rather robust to ZOO attack. • DNN with the best baseline performance, is the most vulnerable to the ZOO attack. • Adaboost is robust to three gray/back-box attacks on the NSL-KDD but extremely sensitive to Boundary and HopSkipJum attacks on the UNSW-NB15. • From the point of view of accuracy, the classifiers have the same behavior against attacks in both databases. However, the behavior of FNR is different from one database to the other: it increases for most of the classifiers on the NSL-KDD database, but remains very low for some classifiers (DNN, LR, RF SVC) on the UNSW-NB15 database.
Behavior against white-box attacks
• Classifiers are rather robust to FGSM, PGD, BIM and C&W attack also has a very similar effect. • JSMA is the most powerful attack. • Adaboost and RF are the only classifiers that are robust to all white box attacks. • The classifiers were more vulnerable to these attacks on the UNSW-NB15 database than on the NSL-KDD database, for both the accuracy and False Negative Rate.

5. Experimental results

5.1. Initial robustness evaluation

5.1.1. Scenario 1: measuring the performance in baseline scenario

Results in the first rows of the first sub-tables show that almost all the classifiers perform well, the accuracy varies between 74% and 77% on NSL-KDD except for Adaboost that has the worst performance (accuracy 55.13% on NSL KDD and 60.6% on UNSW-NB15). DNN is the most efficient with an accuracy of 77.68% on NSL-KDD and 78.56% on UNSW-NB15. On the NSL-KDD data-set, these results are confirmed by the False Negative Rates (first raw of the second sub-table). Indeed, Adaboost wrongly classifies 71.35% of the malicious traffic as benign, while the other classifiers have a FNR varying between 29.68% and 39.32%. However, all classifiers have extremely low false negative rates (below 6%) on the UNSW-NB15 database.

5.1.2. Scenario 2: measuring the performance in adversarial environment

Two analysis are of interest in this scenario: (i) measuring the impact of *each adversarial attack* on different classifiers and (ii) measuring the performance of *each classifier* against different types of attacks. The first analysis identifies the most/least vulnerable classifier to a given adversarial attack while the second finds the most/least powerful adversarial attack for each classifier.

5.1.2.1. ***Accuracy***.

Gaussian noise attack:

- There are two different behaviors; some classifiers (DT, GB and bagging on UNSW-NB15) are very vulnerable to the attack of Gaussian noise, even at low intensity ($\sigma = 0.01$), while the other classifiers are more robust. In particular, for $\sigma = 0.01$, the performance of SVC (resp. Logistic regression) remains stable on NSL-KDD, (resp. UNSW-NB15).
- For a high-intensity attack ($\sigma = 0.2$), the behavior is similar, where, on the NSL-KDD dataset, DT loses almost half of its performance and GB drops to a third of its efficiency. However, the accuracy decrease for the other algorithms ranges from 4.7% for LR to 11.5% for Bagging. The results are similar on the UNSW-NB15 database.
- Surprisingly, Adaboost, which has the worst baseline performance, is the least vulnerable to Gaussian noise attacks on the NSL-KDD dataset, beating DT and GB when they all face this type of attack. More surprisingly, a small Gaussian noise perturbation even *increases* Adaboost's performance by 5.6%.

Gray/black-box attacks:

- Compared to the Gaussian noise attack, all classifiers are rather robust to the ZOO attack, with the performance loss ranging from 0.3% for Adaboost to 5% for DNN on NSL-KDD. Thus, the DNN with the best baseline performance is the most vulnerable to the ZOO attack.
- HopSkipJump and Boundary attacks have almost the same impact on all classifiers (whose performance drops drastically). This can be explained by the fact that both attacks are of the same family, i.e. HopSkipJump is an extension of the Boundary attack.
- With the exception of Adaboost, whose accuracy decreases by only 21% on NSL-KDD for both attacks, the other classifiers are much more vulnerable. In fact, on NSL-KDD, DNN and RF lose 71% and 68% of their performance respectively when dealing with the Boundary attack. As for the HopSkipJUmp attack, both Bagging and DT lose almost 68% of their performance.
- The behavior of the majority of classifiers against these attacks is similar in both databases.

White-box attacks:

- FGSM, PGD and BIM which are attacks of the same family have almost the same impact on the classifiers.
- JSMA is the most powerful attack; on NSL-KDD, the decrease in accuracy reaches 36% for GB. C&W is slightly more powerful than the trio of FGSM, PGD, BIM, but these are all weak attacks that result in a drop in accuracy of no more than 4.7%.
- Adaboost and Random Forest are the only classifiers that are robust to all white-box attacks. The decrease in accuracy is limited to 1.4% and 3.7% for Adaboost and RF respectively on NSL-KDD.
- The classifiers are more vulnerable to these attacks on the UNSW-NB15 database than on the NSL-KDD database.

5.1.2.2. ***False negative rate***.

The results are shown in the second sub-tables of Tables 4 and 5. The objective of this evaluation is to measure the ability of classifiers to block malicious traffic. Recall that the FNR measures the percentage of malicious traffic classified as legitimate, so the lower the FNR, the better the performance of the classifier.

Gaussian noise attack:

- The results show that FNR can decrease after data perturbation, i.e., malicious data initially classified as illegitimate are misclassified as legitimate after Gaussian noise perturbation. This reflects an improvement in classifier performance, as is the case for Adaboost, GB, LR, and SVC on NSL-KDD data-set. As for the Bagging, DT, DNN, and RF classifiers, the FNR increases by 23%, 13.2%, 22%, and 11% for a $\sigma = 0.02$, reflecting the general decrease in accuracy described in the first sub-table of Table 4.
- The GB result on NSL-KDD data-set is particularly interesting because the Gaussian noise has a contradictory impact on the accuracy and the FNR. Indeed, the overall performance of the classifier degraded (accuracy decreased by up to 21.68% for $sigma = 0.2$), while the FNR also decreased (by 25.6% for $sigma = 0.2$) which means that more malicious traffic was blocked. Thus, the degradation in overall classifier performance may be due to mis-classifying benign traffic as illegitimate, which causes *False Alarms*.
- We note that in the UNSW-NB15 database, where all classifiers had low false-negative rates in the absence of adversarial attacks, the false-negative rates increase significantly, showing their sensitivity to these attacks, especially for the Bagging and RF classifiers.

Gray/black-box attacks:

- Adaboost is robust to all three gray/back-box attacks on NSL-KDD data-set. Its FNR increase does not exceed 3.1% (in the case of the HopSkipJump attack). However, on UNSW-NB15, the FNR increases exponentially against Boundary and HopSkipJum and reaches 98.23%.
- HopSkipJump and Boundary attacks have the same impact on the other classifiers. Indeed, the FNR reaches 100%, which means that the adverse noise manages to mis-classify all malicious samples into benign ones.
- HopSkipJump and Boundary are more powerful than ZooAttack in most cases.
- Unlike NSL-KDD, some algorithms (DNN, LR, RF, SVC) are very robust to these attacks on the UNSW-NB15 from the point of view of FNR, which does not increase.

White-box attacks:

- on NSL-KDD data-set, JSMA has the greatest influence on the deviation of FNR, whether it is positive or negative. It is mainly noticed that the FNR of DT increases by 41% for DT and by 41% and 10% for Adaboost and LG respectively.
- While the results are not significantly different from the baseline results for FGSM, PDG and BIM on NSL-KDD data-set, we note that DNN is the most impacted classifier (an increase in FNR up to 8.7%), while the FNR of Adaboost and Gradient Boosting decrease slightly, reflecting better classification of malicious traffic.
- The impact of these attacks on the FNR of the classifiers is much more remarkable on the UNSW-NB15 data-set, which increases exponentially for most of the classifiers except for Adabooost which keeps a low FNR.

5.2. Applying defense techniques

5.2.1. Scenario 3: measuring the performance in non adversarial environment with training in augmented data

Comparing the performance of the classifiers trained on the augmented database in non adversarial environment (first rows of Tables 4 and 5) with their performance when trained on initial training data-sets (first row of the sub-tables 3), we notice that the performance of most of them (Bagging, DT, DNN, RF) has remained stable on NSL-KDD data-set. The performance of Adaboost increased from 55.13% to 66.33%, but the performance of LR and SVC decreased from 75% to 67.27% and from 74.29% to 65.36% respectively. The FNR results (first row of Table 4) reflect the same conclusions. The performance degradation is more noticeable in the UNSW-NB15 base.

5.2.2. Scenario 4: measuring the impact of defense techniques

5.2.2.1. ***Accuracy***. The results are shown in the third sub-tables and will be compared to the results of the first sub-tables, where the accuracy was measured without applying defense techniques.

- *Gaussian noise attack:* The defense technique has improved the robustness of most classifiers. The improvement is even more remarkable for a Gaussian perturbation of high intensity ($\sigma = 0.2$). The improvement is almost perfect for SVC and LR that have become more robust but at the cost of a degradation in performance, even in non adversarial conditions. More interesting, the DNN has become more robust while keeping almost the same original performance on NSL-KDD data-set, it is the classifier that has the best accuracy under normal and adversarial conditions on NSL-KDD data-set.
- *Gray/black-box attacks:* The defense technique has different effects depending on the classifier. For example on the NSL-KDD data-set: (i) it contributes to the improvement of the robustness for some classifiers as for LR and SVC whose accuracy decreased, compared to the baseline performance, by only 55.3% and 53.8% (resp.) when confronted to a Boundary attack, versus 67% and 66.8% (resp.) of decrease before having applied the defense technique. (ii) it degrades the robustness of some algorithms as it is the case for Adaboost whose accuracy decreases by 53% for a Boundary attack instead of 22% without defense. (iii) the defense technique has very little effect on the other classifiers who have kept almost the same level of robustness as before. The behavior of the classifiers is similar in both databases.
- *White-box attacks:* The defense technique is effective for almost all the classifiers against the FGSM, PGD and BIM. As for JSMA, the robustness is also improved for all classifiers, and Gradient Boosting has the best performance. The behavior of the classifiers is similar in both databases.

5.2.2.2. ***False negative rate***. The results are shown in the fourth sub-table, and will be compared with the second sub-tables, where the FNR is measured without defense technique.

- *Gaussian Noise attack:* The increase of the FNR is less important with the defense technique for most of the algorithms thus their robustness has improved. However, the ability of these classifiers to block malicious traffic has decreased overall, for example for SVC, the FNR is 50% while it did not exceed 37.52% without defense on NSL-KDD. Moreover, the defense technique did not improve the performance Gradient Boosting and RF on NSL-KDD since their FNR increases from 10.2% to 48.51% and from 45.07% to 55.51% respectively, thus more malicious traffic was blocked without defense. DNN has the lowest FNR (the decrease in FNR does not exceed 2.8%) and a very good robustness. The defense technique has improved its robustness.Curiously, the defense method decreased the robustness of some algorithms to Gaussian noise attacks on UNSW-NB15 (FNR increases more) as is the case for DT, Gradient Boost and Adaboost.
- *Gray/black-box attacks:* on NSL-KDD data-set, the defense has degraded the robustness of Adaboost against Boundary and HopSkipJump (FNR increases by 49% while the increase was limited to 3.1% before defense). It slightly improved the robustness of Bagging, (DNN,RF). The improvement is more significant for LR and SVC. FNR increased by only 52% and 49% for LR and SVC respectively, compared to an increase by 62% for both without defense. On the UNSW-NB15 data-set, the defense method proved to be effective and significantly improved the robustness of the classifiers in terms of FNR.

Table 3
Summary of results for evaluation scenario 4.

Behavior against a Gaussian noise attack
• The defense technique has improved the robustness of most classifiers. • The robustness of SVC and LR has improved but at the cost of performance degradation. • The robustness of the DNN has been improved on NSL-KDD dataset while maintaining its good performance, thus DNN has the best accuracy under normal and adversarial conditions. • The defense method decreased the robustness of some algorithms on the UNSW-NB15 (FNR increases more) as is the case for DT, Gradient Boost and Adaboost.
Behavior against Gray/black-box attacks
• The robustness of Bagging, DT, LR, SVC has improved on NSL-KDD dataset • The robustness of Adaboost has decreased on NSL-KDD dataset • The defense technique has almost no effect on other classifiers. In terms of FNR, the defense method was more effective on the UNSW-NB15 basis, than NSL KDD (FNR rates are lower).
Behavior against white-box attacks
• The robustness of classifiers against FGSM, PGD, BIM attacks is almost the same since they are already quite robust. • Robustness against JSMA has improved for most classifiers but at the cost of performance degradation for most of them. In terms of FNR, the defense method was not effective on UNSW-NB15 and even degraded the performance of some classifiers (e.g. Gradient Boost, DT).

- *White-box attacks:* After defense, the robustness of the classifiers for FGSM, PGD, BIM and C&W is stable on NSL-KDD, as the classifiers are already quite robust. But the improvement of the robustness is more visible for JSMA, although the global performance has degraded for most of the classifiers (higher FNR, compared to Table 2). The defense method was not effective on UNSW-NB15 and even degraded the performance of some classifiers (e.g. Gradient Boost, DT).

5.3. Analysis and discussion

In this section we summarize and discuss our findings.

5.3.1. Global conclusions

- An attack does not impact all classifiers in the same way. Similarly, the robustness of a classifier depends on the attacks. In the same sense, a defense technique is not effective against all attacks and does not have the same effect on different classifiers (it can improve or decrease the classifier's robustness or be ineffective). Similarly, the behavior of a classifier when faced with an attack or a defense method depends on the database.
- The robustness and overall performance of classifiers can be contradictory. As seen in the results on the NSL-KDD data-set, Adaboost is a very robust classifier, but does not have high accuracy Ċonversely, DNN is very efficient but is the most vulnerable to ZOO attack. Therefore, depending on the situation and need of the IDS, robustness or performance can be privileged. Namely, if the IDS operates in a certain environment, it is natural to favor performance, however, if the environment is uncertain, robustness becomes important. Similarly, defense techniques can improve the robustness of classifiers but at the cost of degrading their performance. Thus, a trade-off between these two objectives must be considered. A good defense technique, improves the robustness of the classifier without degrading its performance. In addition, the effectiveness of a defense method against a database attack can vary from database to database.

5.3.2. Specific remarks

- Attacks of the same family (Boundary and HopSkipJump) have the same impact on each classifier.
- Sometimes, an attack can have an opposite effect: the Gaussian noise attack improved Adaboost's performance on NSL-KDD. In addition to mis-classifying more malicious traffic as legitimate, which is the main objective of an adversarial IDS attack, an adversary attack can also increase the False Alarms rate, as was the case with Gradient Boosting against Gaussian noise.
- Boundary and HopSkipJump attacks succeed in mis-classifying 100% of the malicious traffic on NSL-KDD.
- The Gaussian data augmentation defense was especially effective on the Gaussian noise attack probably because they are of the same family.
- A defense method can even degrade the robustness of a classifier (e.g. DT against white box attacks on the UNSW-NB15), so it must be chosen appropriately.

It is also interesting to note that in our experiment, gray/back-box attacks are more effective than white-box attacks, which is counter-intuitive since the latter have access to more information about the classifier. This can be explained by the fact that gray/back-box attacks were applied directly on the classifiers while white-box attacks were applied on a Generator network. On the other hand, the performance of the attacks strongly depends on the setting of the hyper parameters. A better setting of the hyper-parameters would probably enhance the performance of white-box attacks.

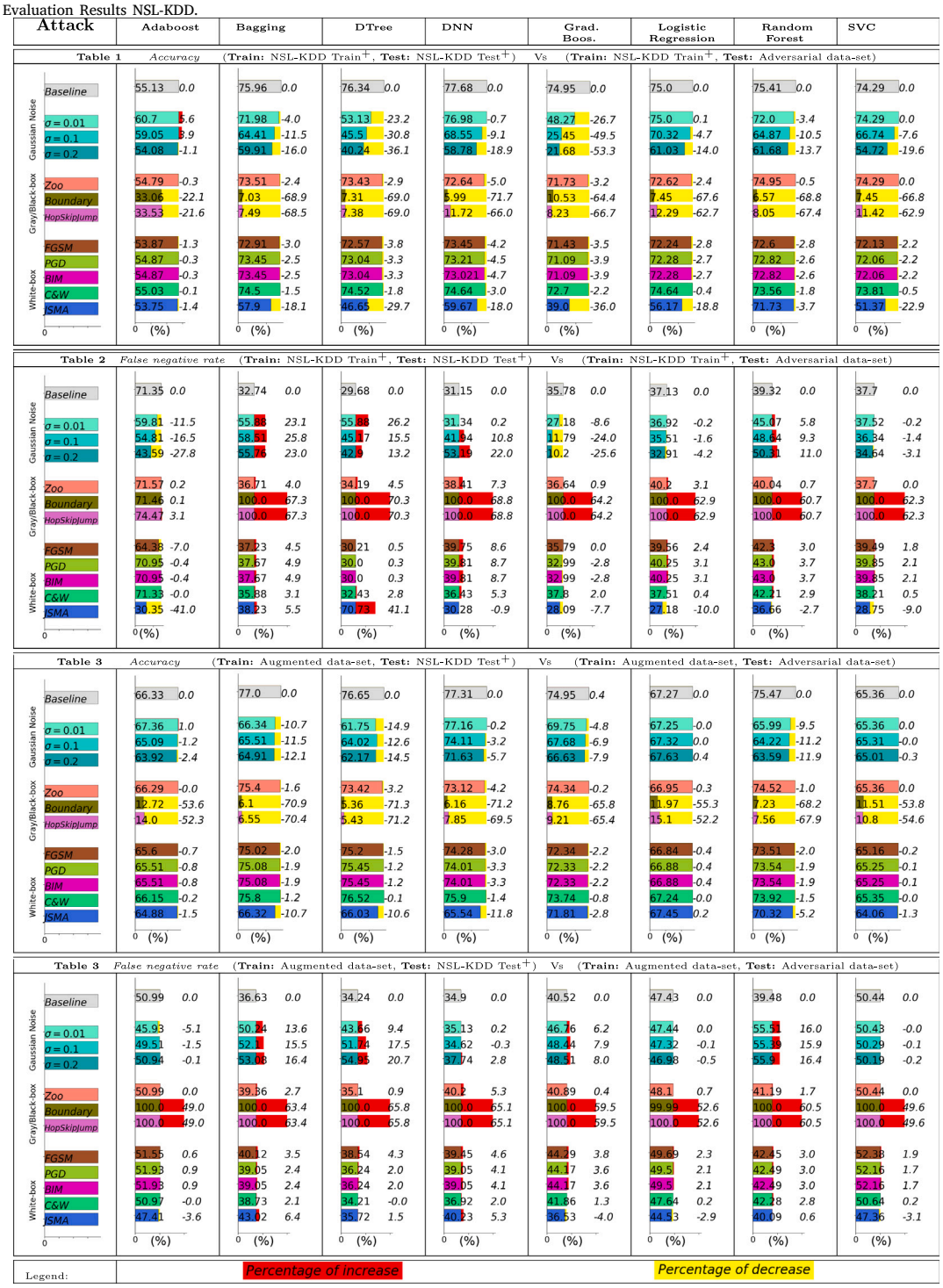
5.4. Comparison with the state of the art

As mentioned in 3.4, the paper [41] is the closest to our work since, like us, the authors evaluate the robustness of *shallow classifiers* against *state of the art adversarial attacks* using the NSL-KDD database.

Since the hyper-parametric information is unavailable in [41], in addition to the difference in the list of classifiers and attacks considered, we opt for qualitative comparison our evaluation framework with the approach presented in [41].

The authors of [41] evaluate the robustness of 3 shallow classifiers (LR, RF, SVM) against three adversarial white-box attacks (FGSM, PDG, L-BFGS) and a black-box attack SPSA, and do not propose a defense method (Cf. Table 1). This paper, however, evaluates a richer and more varied collection of ML classifiers (Adaboost, Bagging, DT, GB, LR, RF, SVC), against more varied and numerous adversarial attacks (Gaussian noise, White-box attacks: FGSM, PGD, BIM, C&W, JSMA, Gray/black-box attacks: Zoo, Boundary, HopSkipJump). Moreover, we complete our study by evaluating the impact of a defense technique (Gaussian data augmentation) on the improvement of the robustness of the classifiers. This allowed us to draw interesting conclusions about the trade-off between robustness and accuracy of classifiers, as discussed above. Note, however, that the results obtained in [41] confirm our findings regarding the vulnerability of ML classifiers to different attacks with varying degrees of sensitivity.

Table 4
Evaluation Results NSL-KDD.

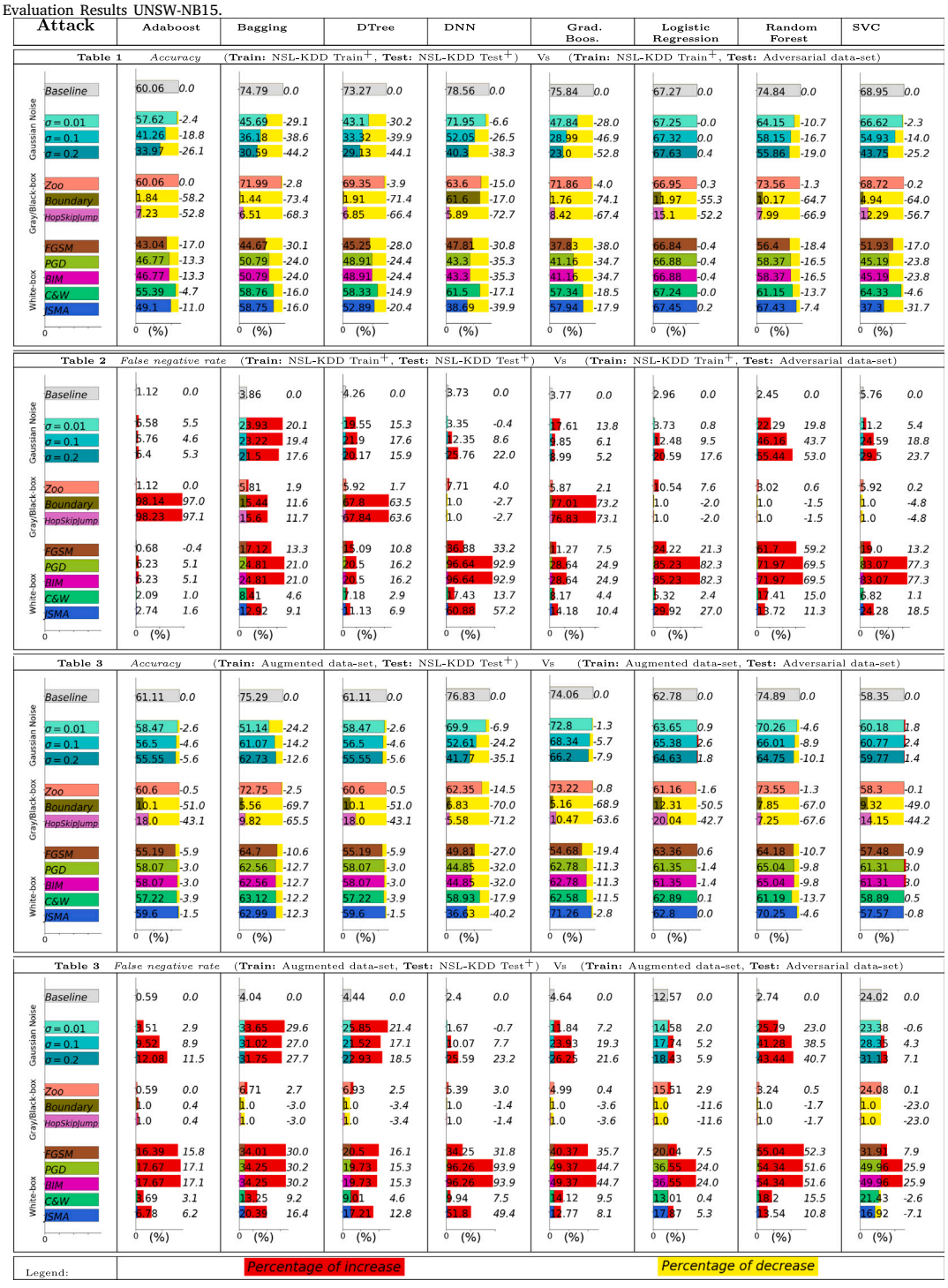


6. Discussion

In this work, changes on the input samples are made on the feature vector, however, an attacker does not have access to the input feature vector of the ML algorithm to be able to modify it, but instead must generate actual traffic that respects the features described by the feature vector generated by the adversarial attack (FGSM, PGD, etc.). This traffic must also retain its original functionality (malicious or benign). This transition from feature vector to actual instance is referred to in the literature as the "feature space" to "problem space" transition [82], it is specific to certain data types, where unlike images, this transition is not trivially reversible, and can be complex.

One way to facilitate this transition is to judiciously perform some modifications on the original traffic sample, in order to obtain a sample with characteristics close to those of the perturbed sample (generated adversary sample), without altering the primary function of the traffic. Examples of possible manipulations include (i) filling and fragmenting or duplicating protocol data units (PDUs, e.g. packet, segment, datagram, etc.) to modify their volumetric characteristics (e.g. flow size, number of packets, etc.), (ii) delaying the transmission of PDUs, to act on their temporal characteristics (e.g. inter-arrival time of packets), (iii)

Table 5
Evaluation Results UNSW-NB15.



modifying the values of some fields, etc. In order not to alter the main function of the traffic, the modifications must be made only on the fields that do not have an impact on this function. To do this, the PDU manipulation tools need to be explored and improved. Fortunately, there are already promising tools such as Scapy [83] which is a packet manipulation program. Custom synthetic traffic generators [84] can also be explored.

7. Conclusion and future work

This paper focuses on the research area of adversarial machine learning. We study the robustness of various widely used ML classifiers against adversarial examples in the context of network IDS. We consider both gray/black-box and white-box attacks. A DNN-based external classifier has been used to generate white-box based adversarial examples. In addition, we studied the impact of a defense technique based on Gaussian data augmentation to improve the robustness of different NIDS. For the evaluation, we consider both the accuracy and the false negative rate. The latter measures the percentage of malicious traffic that successfully bypasses the NIDS. The NSL-KDD and UNSW-NB15 data-sets were used for the evaluation. The results show that attacks do not have the same impact on all classifiers and that the robustness of a classifier depends on the attack. Similarly, a defense technique is not effective for all classifiers, nor against all attacks. Furthermore,

a defense technique may improve the robustness of a classifier but degrade its overall performance, so a trade-off between performance and robustness must be considered depending on the NIDS application scenario.

In future work, we intend to generate more realistic adversarial attacks that project more easily into the problem space. To do so, we will follow some recommendations found in the literature, [85–87], namely (i) restrict the space of features to be perturbed, i.e., avoid perturbing non-differentiable features so that the transformation is reversible, and the features directly related to the functionality of the flow so as not to impact it, (ii) perform small amplitude perturbations and check that the values of the modified features remain valid (domain constraints), and (iii) analyze the consistency of the values taken by the correlated features.

CRediT authorship contribution statement

Houda Jmila: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Mohamed Ibn Khedher:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and suggestions that helped us improve the quality of this work considerably. All authors approved the version of the manuscript to be published.

Appendix

See Tables 1–5.

References

[1] M.I. Khedher, M. Mziou-Sallami, M. Hadji, Improving decision-making-process for robot navigation under uncertainty, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1105–1113.

[2] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, 2009, pp. 1–6.

[3] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6.

[4] N. Moustafa, J. Hu, J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey, J. Netw. Comput. Appl. 128 (2019) 33–55.

[5] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2) (2015) 1153–1176.

[6] S. Gamage, J. Samarabandu, Deep learning methods in network intrusion detection: A survey and an objective comparison, J. Netw. Comput. Appl. 169 (2020) 102767.

[7] H. Jmila, M.I. Khedher, G. Blanc, M.A. El-Yacoubi, Siamese network based feature learning for improved intrusion detection, in: Neural Information Processing - 26th International Conference, ICONIP 2019, 11953, 2019, pp. 377–389.

[8] Y. Xu, Y. Zhou, P. Sekula, L. Ding, Machine learning in construction: From shallow to deep learning, Dev. Built Environ. 6 (2021) 100045.

[9] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, A. Jain, Adversarial attacks and defenses in images, graphs and text: A review, Int. J. Autom. Comput. 17 (2020) 151–178.

[10] I. Rosenberg, A. Shabtai, Y. Elovici, L. Rokach, Adversarial learning in the cyber security domain, 2020, arXiv preprint arXiv:2007.02407.

[11] N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches, Peer-to-Peer Netw. Appl. 12 (2) (2019) 493–501.

[12] M.I. Khedher, H. Ibn-Khedher, M. Hadji, Dynamic and scalable deep neural network verification algorithm, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1122–1130.

[13] M. Mziou-Sallami, M.I. Khedher, A. Trabelsi, S. Kerboua-Benlarbi, D. Bettebghor, Safety and robustness of deep neural networks object recognition under generic attacks, in: Neural Information Processing - 26th International Conference, ICONIP, 1142, 2019, pp. 274–286.

[14] H. Ibn-Khedher, M.I. Khedher, M. Hadji, Mathematical programming approach for adversarial attack modelling, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 343–350.

[15] M.I. Khedher, M. Rezzoug, Analyzing adversarial attacks against deep learning for robot navigation, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1114–1121.

[16] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, X. Yin, Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors, 2020, arXiv: Cryptography and Security.

[17] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 15–26.

[18] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015, URL http://arxiv.org/abs/1412.6572.

[19] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, 2017, URL https://arxiv.org/abs/1611.01236.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018, URL https://openreview.net/forum?id=rJzIBfZAb.

[21] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016, pp. 372–387, http://dx.doi.org/10.1109/EuroSP.2016.36.

[22] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 86–94, http://dx.doi.org/10.1109/CVPR.2017.17.

[23] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574–2582, http://dx.doi.org/10.1109/CVPR.2016.282.

[24] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 Ieee Symposium on Security and Privacy (Sp), IEEE, 2017, pp. 39–57.

[25] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, in: International Conference on Learning Representations, 2018, URL https://arxiv.org/abs/1712.04248.

[26] J. Chen, M.I. Jordan, Boundary attack++: Query-efficient decision-based adversarial attack, 2019, CoRR arXiv:1904.02144, URL http://arxiv.org/abs/1904.02144.

[27] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Adversarial attacks and defenses: A survey, 2018, CoRR arXiv:1810.00069, URL http://arxiv.org/abs/1810.00069.

[28] K. Ren, T. Zheng, Z. Qin, X. Liu, Adversarial attacks and defenses in deep learning, Engineering 6 (3) (2020) 346–360.

[29] M. Ozdag, Adversarial attacks and defenses against deep neural networks: a survey, Procedia Comput. Sci. 140 (2018) 152–161.

[30] X. Wang, J. Li, X. Kuang, Y.-a. Tan, J. Li, The security of machine learning in an adversarial setting: A survey, J. Parallel Distrib. Comput. 130 (2019) 12–23.

[31] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, Ieee Access 6 (2018) 14410–14430.

[32] V. Zantedeschi, M.-I. Nicolae, A. Rawat, Efficient defenses against adversarial attacks, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, in: AISec '17, ACM, New York, NY, USA, 2017, pp. 39–49.

[33] H. Lee, S. Han, J. Lee, Generative adversarial trainer: Defense to adversarial perturbations with gan, 2017, arXiv preprint arXiv:1705.03387.

[34] L. Nguyen, S. Wang, A. Sinha, A learning and masking approach to secure learning, in: International Conference on Decision and Game Theory for Security, Springer, 2018, pp. 453–464.

[35] K. Yang, J. Liu, C. Zhang, Y. Fang, Adversarial examples against the deep learning based network intrusion detection systems, in: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), 2018, pp. 559–564, http://dx.doi.org/10.1109/MILCOM.2018.8599759.

[36] Z. Wang, Deep learning-based intrusion detection with adversaries, IEEE Access 6 (2018) 38367–38384.

[37] G. Apruzzese, M. Colajanni, M. Marchetti, Evaluating the effectiveness of adversarial attacks against botnet detectors, in: 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), IEEE, 2019, pp. 1–8.

[38] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100–123.

[39] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSp, 2018, pp. 108–116.

[40] E.B. Beigi, H.H. Jazi, N. Stakhanova, A.A. Ghorbani, Towards effective feature selection in machine learning-based botnet detection approaches, in: 2014 IEEE Conference on Communications and Network Security, IEEE, 2014, pp. 247–255.

[41] Y. Peng, J. Su, X. Shi, B. Zhao, Evaluating deep learning based network intrusion detection system in adversarial environment, in: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 61–66.

[42] J. Clements, Y. Yang, A. Sharma, H. Hu, Y. Lao, Rallying adversarial techniques against deep learning for network security, 2019, arXiv preprint arXiv:1903.11688.
[43] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: an ensemble of autoencoders for online network intrusion detection, 2018, arXiv preprint arXiv:1802.09089.
[44] O. Ibitoye, M.O. Shafiq, A. Matrawy, Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks, in: 2019 IEEE Global Communications Conference, GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019, IEEE, 2019, pp. 1–6.
[45] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.
[46] M.J. Hashemi, G. Cusack, E. Keller, Towards evaluation of nidss in adversarial setting, in: Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, 2019, pp. 14–21.
[47] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, M. Qiu, Adversarial attacks against network intrusion detection in IoT systems, IEEE Internet Things J. (2020) 1.
[48] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.
[49] H. Zenati, C.S. Foo, B. Lecouat, G. Manek, V.R. Chandrasekhar, Efficient gan-based anomaly detection, 2018, arXiv preprint arXiv:1802.06222.
[50] J. Aiken, S. Scott-Hayward, Investigating adversarial attacks against network intrusion detection systems in SDNs, in: 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2019, pp. 1–7, http://dx.doi.org/10.1109/NFV-SDN47374.2019.9040101.
[51] L. Labrotary, DARPA Intrusion Detection Evaluation Data Set, Vol. 12, Massachusetts Institute of Technology, Cambridge, MA, 1999, p. 2009, Retrieved January.
[52] S. Zhang, X. Xie, Y. Xu, A brute-force black-box method to attack machine learning-based systems in cybersecurity, IEEE Access 8 (2020) 128250–128263.
[53] G. Creech, J. Hu, Generation of a new IDS test dataset: Time to retire the KDD collection, in: 2013 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2013, pp. 4487–4492.
[54] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, C. Siemens, Drebin: Effective and explainable detection of android malware in your pocket, in: Ndss, Vol. 14, 2014, pp. 23–26.
[55] J. Jeong, S. Kwon, M. Hong, J. Kwak, T. Shon, Adversarial attack-based security vulnerability verification using deep learning library for multimedia video surveillance, Multim. Tools Appl. 79 (23–24) (2020) 16077–16091.
[56] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[57] R.A. Khamis, A. Matrawy, Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs, 2020, arXiv:2007.04472.
[58] R.A. Khamis, M.O. Shafiq, A. Matrawy, Investigating resistance of deep learning-based IDS against adversaries using min-max optimization, in: 2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020, IEEE, 2020, pp. 1–7.
[59] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, et al., Understanding the mirai botnet, in: 26th {USENIX} Security Symposium ({USENIX} Security 17), 2017, pp. 1093–1110.
[60] A.U.H. Qureshi, H. Larijani, M. Yousefi, A. Adeel, N. Mtetwa, An adversarial approach for intrusion detection systems using Jacobian saliency map attacks (JSMA) algorithm, Computers 9 (3) (2020).
[61] C. Zhang, X. Costa-Pérez, P. Patras, Tiki-Taka: Attacking and defending deep learning-based intrusion detection systems, in: Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop, 2020, pp. 27–39.
[62] G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, M. Colajanni, Deep reinforcement adversarial learning against botnet evasion attacks, IEEE Trans. Netw. Serv. Manag. 17 (4) (2020) 1975–1987.
[63] A. Venturi, G. Apruzzese, M. Andreolini, M. Colajanni, M. Marchetti, Drelab - deep reinforcement learning adversarial botnet: A benchmark dataset for adversarial attacks against botnet intrusion detection systems, Data Brief 34 (2021) 106631.
[64] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, X. Yin, Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors, IEEE J. Sel. Areas Commun. 39 (8) (2021) 2632–2647.
[65] E. Anthi, S. Ahmad, O. Rana, G. Theodorakopoulos, P. Burnap, EclipseIoT: A secure and adaptive hub for the Internet of Things, Comput. Secur. 78 (2018) 477–490.
[66] E. Anthi, L. Williams, A. Javed, P. Burnap, Hardening machine learning denial of service (DoS) defenses against adversarial attacks in IoT smart home networks, Comput. Secur. 108 (2021) 102352.
[67] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.
[68] X. Fu, N. Zhou, L. Jiao, H. Li, J. Zhang, The robust deep learning–based schemes for intrusion detection in Internet of Things environments, Ann. Telecommun. 76 (5) (2021) 273–285.
[69] J. Wang, J. Pan, I. AlQerm, Y. Liu, Def-IDS: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection, in: 2021 International Conference on Computer Communications and Networks (ICCCN), IEEE, 2021, pp. 1–9.
[70] C. Zhang, X. Costa-Pérez, P. Patras, Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms, IEEE/ACM Trans. Netw. (2022).
[71] J. Vitorino, N. Oliveira, I. Praça, Adaptative perturbation patterns: Realistic adversarial learning for robust intrusion detection, Future Internet 14 (4) (2022) 108.
[72] S. Garcia, A. Parmisano, M.J. Erquiaga, IoT-23: A Labeled dataset with malicious and benign IoT network traffic, 2020, More details here https://www.stratosphereips.org/datasets-iot23.
[73] H. Jiang, J. Lin, H. Kang, Fgmd: A robust detector against adversarial attacks in the IoT network, Future Gener. Comput. Syst. 132 (2022) 194–210.
[74] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, S. Nõmm, MedbIoT: Generation of an IoT botnet dataset in a medium-sized IoT network, in: ICISSP, 2020, pp. 207–218.
[75] H. Kang, D.H. Ahn, G.M. Lee, J.D. Yoo, K.H. Park, H.K. Kim, IoT network intrusion dataset, 2019.
[76] P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutor. 21 (1) (2018) 686–728.
[77] Y. Dong, T. Pang, H. Su, J. Zhu, Evading defenses to transferable adversarial examples by translation-invariant attacks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4312–4321.
[78] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.
[79] A. Gulli, S. Pal, Deep Learning with Keras, Packt Publishing Ltd, 2017.
[80] A.C. Müller, S. Guido, Introduction to Machine Learning with Python: A Guide for Data Scientists, "O'Reilly Media, Inc.", 2016.
[81] M.-I. Nicolae, M. Sinn, T.N. Minh, A. Rawat, M. Wistuba, V. Zantedeschi, I.M. Molloy, B. Edwards, Adversarial robustness toolbox v0.2.2, 2018, CoRR arXiv:1807.01069, URL http://arxiv.org/abs/1807.01069.
[82] F. Pierazzi, F. Pendlebury, J. Cortellazzi, L. Cavallaro, Intriguing properties of adversarial ml attacks in the problem space, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 1332–1349.
[83] R. Rohith, M. Moharir, G. Shobha, et al., Scapy-a powerful interactive packet manipulation program, in: 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), IEEE, 2018, pp. 1–5.
[84] O.A. Adeleke, N. Bastin, D. Gurkan, Network traffic generation: A survey and methodology, ACM Comput. Surv. 55 (2) (2022) 1–23.
[85] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, T. Hou, Manda: On adversarial example detection for network intrusion detection system, IEEE Trans. Dependable Secure Comput. (2022).
[86] R. Sheatsley, N. Papernot, M.J. Weisman, G. Verma, P. McDaniel, Adversarial examples for network intrusion detection systems, J. Comput. Secur. (Preprint) (2022) 1–26.
[87] M.A. Merzouk, F. Cuppens, N. Boulahia-Cuppens, R. Yaich, Investigating the practicality of adversarial evasion attacks on network intrusion detection, Ann. Telecommun. (2022) 1–13.

Houda Jmila received her engineering degree in Computer Science from Telecom Sudparis, Institut Polytechnique de Paris, France in 2011 and the Ph.D. in Telecommunications and Computer science in 2015 from the same university. She is currently a post-doctoral researcher at Telecom SudParis. Her research interests include network security and automated management of resources in virtual networks, 5G networks and the IoT ecosystem as well as the machine learning applications to these domains.

Mohamed Ibn Khedher obtained his engineering degree in 2007, and his master degree in 2009, in Computer Science from the National School of Computer Sciences, Tunisia. He obtained his Ph.D. in computer science from Telecom SudParis with the collaboration of the University of Evry, France. During his Ph.D., he developed software for person re-identification from video sequence. Since 2015, he worked as a software Engineer in a research center specialized in Advanced Driver Assistance Systems. Currently, he is a senior research engineer at IRT systemX, France. He worked in the French Grand defi "Trusted AI" research program as a senior expert in AI robustness. His main interests include Neural Network Verification, AI adversarial robustness, Machine Learning, Video coding standards, Video Surveillance, Biometrics and Handwriting Analysis.