Annals of Telecommunications (2023) 78:401–412
https://doi.org/10.1007/s12243-023-00953-y

Towards adversarial realism and robust learning for IoT intrusion detection and classification

João Vitorino[1] · Isabel Praça[1] · Eva Maia[1]

Received: 19 May 2022 / Accepted: 17 February 2023 / Published online: 11 March 2023

Abstract
The internet of things (IoT) faces tremendous security challenges. Machine learning models can be used to tackle the growing number of cyber-attack variations targeting IoT systems, but the increasing threat posed by adversarial attacks restates the need for reliable defense strategies. This work describes the types of constraints required for a realistic adversarial cyber-attack example and proposes a methodology for a trustworthy adversarial robustness analysis with a realistic adversarial evasion attack vector. The proposed methodology was used to evaluate three supervised algorithms, random forest (RF), extreme gradient boosting (XGB), and light gradient boosting machine (LGBM), and one unsupervised algorithm, isolation forest (IFOR). Constrained adversarial examples were generated with the adaptative perturbation pattern method (A2PM), and evasion attacks were performed against models created with regular and adversarial training. Even though RF was the least affected in binary classification, XGB consistently achieved the highest accuracy in multi-class classification. The obtained results evidence the inherent susceptibility of tree-based algorithms and ensembles to adversarial evasion attacks and demonstrate the benefits of adversarial training and a security-by-design approach for a more robust IoT network intrusion detection and cyber-attack classification.

Keywords Adversarial attacks · Adversarial robustness · Machine learning · Tabular data · Internet of things · Intrusion detection

1 Introduction

The internet of things (IoT) is accelerating the digital transformation. It represents decentralized and heterogeneous systems of interconnected devices, which combine wireless sensor networks, real-time computing, and actuation technologies [1]. Due to the integration of physical and business processes, as well as control and information systems, IoT is bridging the gap between operational technology and information technology [2]. However, the convergence of previously isolated systems and technologies faces tremendous security challenges because of the software vulnerabilities and weak security measures of IoT devices [3]. A self-propagating malware can compromise numerous devices and establish a botnet to launch a wide range of cyber-attacks [4], which is particularly concerning for IoT systems that control critical infrastructure like healthcare facilities [5], energy markets [6], and water supply networks [7].

Machine learning (ML) can be very valuable to tackle the growing number and increasing sophistication of cyber-attacks targeting IoT systems, but it is susceptible to adversarial examples: cyber-attack variations specifically crafted to exploit ML [8]. For instance, tree-based algorithms and ensembles are remarkably well-established for network intrusion detection [9, 10]. However, even though the malicious purpose of a cyber-attack causes it to have distinct characteristics that could be recognized in a thorough analysis by experienced security practitioners, an attacker can create perturbations in IoT network traffic to deceive these algorithms and be misclassified as benign. The increasing threat posed by adversarial attacks restates the need for

✉ João Vitorino
jpmvo@isep.ipp.pt

Isabel Praça
icp@isep.ipp.pt

Eva Maia
egm@isep.ipp.pt

[1] Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), School of Engineering, Polytechnic of Porto (ISEP/IPP), 4249-015 Porto, Portugal

better defense strategies for intelligent IoT network intrusion detection systems [11, 12].

To ensure that ML is used in a secure way, organizations should proactively search for vulnerabilities in their intelligent systems. By simulating realistic attack vectors, ML engineers and security practitioners can anticipate possible threats and use that knowledge to improve their countermeasures [13]. But throughout the current scientific literature, various studies apply adversarial evasion attacks to intrusion detection and provide the examples as direct input to an ML model without questioning if they are viable for a real deployment scenario [14], which may result in misleading robustness evaluations where a model seems to be robust because it was tested against examples that it will not encounter in real IoT network traffic [15].

This work addresses the challenge of improving the robustness of tree-based algorithms and ensembles for IoT network intrusion detection. The main contributions are (i) a description of the types of constraints required for an adversarial cyber-attack example to be realistic, (ii) a methodology for a trustworthy robustness analysis with a realistic adversarial evasion attack vector, and (iii) an analysis of several tree-based algorithms and ensembles in binary and multi-class classification scenarios, following the proposed methodology. The initial evaluation carried out in [16] was extended to include adversarial attacks performed with the adaptative perturbation pattern method (A2PM). Three supervised algorithms, random forest (RF), extreme gradient boosting (XGB), and light gradient boosting machine (LGBM), and one unsupervised algorithm, isolation forest (IFOR), were evaluated using the IoT-23 and Bot-IoT datasets. In addition to regular training, the effectiveness of performing adversarial training with realistically perturbed samples was also analyzed.

The present paper is organized into multiple sections. Section 2 provides a survey of previous work on ML robustness for IoT network intrusion detection. Section 3 describes the constraints required to achieve adversarial realism and defines a methodology for a trustworthy robustness analysis. Section 4 describes the experimental evaluation performed following the proposed methodology, including the scenarios, datasets, adversarial method, models, and evaluation metrics. Section 4.4 presents a comparative analysis of the results obtained by each ML model in each scenario. Finally, Sect. 5 addresses the main conclusions and future research topics.

2 Related work

In recent years, the susceptibility of tree-based algorithms to adversarial examples has been drawing attention for network intrusion detection [17, 18]. To better protect these ML models from adversarial attacks, several defense strategies have been developed. Some attempt to improve the intrinsic robustness of entire tree ensembles at once [19, 20], whereas other address each individual decision tree at a time [21, 22]. Nonetheless, the most effective and widespread defense is adversarial training because it anticipates the data variations that an ML model may encounter [23]. Augmenting a training set with examples created by an adversarial evasion attack method enables a model to learn additional characteristics that the samples of each class can exhibit, so it becomes harder for an attacker to deceive it [24].

However, performing adversarial training with unrealistic examples will make a model learn distorted characteristics that will not be exhibited by real samples during its inference phase [25]. This raises a major security concern because training with unrealistic data may not only deteriorate a model's robustness against adversarial data, because it will not learn the subtle nuances that an attacker can exploit, but it may also be significantly detrimental to a model's generalization to regular data, leading to accidental data poisoning and to the introduction of hidden backdoors that make a model even more vulnerable to attacks [26].

Since the focus of adversarial ML has been the computer vision domain, the common attack vector is to freely exploit the internal gradients of artificial neural networks to generate random data perturbations in the pixels of an image [27], which can lead to unrealistic adversarial examples in tabular data. Consequently, most state-of-the-art evasion attack methods do not support other settings nor models that do not have loss gradients [28], which severely limits their applicability to the IoT network intrusion detection domain. To adversarially train a model and improve its robustness with realistic cyber-attack examples, a defender will need to resort to methods that support the specificities of a communication network.

Even though most methods were intended to attack images, a few could be adapted to tabular data. Both the Jacobian-based Saliency Map Attack (JSMA) [29] and the OnePixel attack [30] were developed to minimize the number of modified pixels, which could be used to solely perturb a few features in a network traffic flow. Nonetheless, the perturbations are still randomly generated, so the resulting values for those few features are commonly incompatible with the remaining features of a flow [31]. On the other hand, A2PM [32] was specifically developed for communication networks, assigning an independent sequence of adaptative patterns to analyze the characteristics of each class and create realistic data perturbations that preserve the purpose of a cyber-attack. Due to its suitability for IoT network traffic, it was selected for this work.

To determine the most adequate ML models for IoT network intrusion detection, it is important to understand the results and conclusions of previous performance evaluations.

A comprehensive survey [33] analyzed studies published until 2018, highlighting the advantages and limitations of each model. Tree-based algorithms and ensembles obtained good results in the reviewed performance evaluations, although their robustness was not addressed. In more recent studies, the best performances were achieved by RF in a testbed replicating an industrial plant [34], XGB with the CIDDS-001, UNSW-NB15 and NSL-KDD datasets [35], LGBM with an industrial dataset [36], and IFOR in an IoT testbed for zero-day attacks [37]. Due to their promising results, RF, XGB, LGBM, and IFOR were selected for this work.

To the best of our knowledge, no previous work has analyzed the adversarial robustness of these four algorithms against realistic adversarial examples of cyber-attacks targeting IoT systems nor the effectiveness of an adversarial training approach with realistically perturbed samples.

3 Adversarial realism

This section describes the types of constraints required for an adversarial cyber-attack example to achieve realism and defines a methodology for a trustworthy adversarial robustness analysis with a realistic evasion attack vector.

3.1 Data constraints

In the IoT network intrusion detection domain, cyber-attacks can be identified by analyzing the characteristics of network traffic flows, which are represented in a tabular data format. The features of a flow may be required to follow specific data distributions, according to the specificities of a communication network and the utilized protocols. Furthermore, due to their distinct malicious purposes, different cyber-attacks may exhibit entirely different feature correlations. Since a data sample must represent a real traffic flow, either benign activity or a cyber-attack class, it must fulfill all the constraints of this complex tabular data domain.

To generate adversarial cyber-attack examples that could evade detection in a real IoT system, the constraints must be carefully analyzed. For instance, a key characteristic of an IoT network traffic flow is the inter-arrival time (IAT), which represents the elapsed time between the arrival of two subsequent packets. Its minimum (MinIAT) and maximum (MaxIAT) values are valuable features for the detection of several cyber-attack classes, such as denial-of-service (DoS). A low MinIAT can indicate a short DoS that quickly overloads a server with requests, whereas a high MaxIAT can indicate a lengthy DoS that overwhelms a server by keeping long connections open [38].

When perturbing these features, validity is essential because a successful adversarial attack is not necessarily a successful cyber-attack. If MinIAT was increased to a value higher than MaxIAT, a flow could become an adversarial example that a model would misclassify as benign. However, that would be an invalid network flow that a model would never encounter in a real deployment scenario because it could not be transmitted through a communication network. Therefore, to preserve validity within the network traffic structure, a domain constraint must be enforced: MinIAT must not be higher than MaxIAT. These types of constraints, including value ranges and multiple category membership, have started being investigated in [31] to improve the feasibility of adversarial attacks for intrusion detection.

Nonetheless, validity is not enough for an adversarial attack to be a successful cyber-attack. It is imperative to also address class coherence. Even if the previous domain constraint was fulfilled when increasing MinIAT, the resulting flow could still not be coherent with the intended purpose of a cyber-attack class. Valid adversarial examples with increased MinIATs could be misclassified as benign, but not be quick enough to overload a server in a real scenario. Consequently, those supposed adversarial examples would not actually belong to the short DoS class. Instead, they would represent just regular traffic that would not be useful for a cyber-attack, so an ML model would be correct to label them as benign. Therefore, to preserve coherence, it is necessary to also enforce a class-specific constraint: MinIAT must not be higher than the highest known value of that feature for the short DoS class. These types of constraints are based on the idea initially introduced in [32], where data perturbations were created according to feature correlations.

Even though validity and coherence have previously been investigated, sometimes with different designations, it is pertinent to address them together in a single unifying concept: adversarial realism. Hence, for an adversarial example to be realistic, it must be valid within its domain structure and coherent with the characteristics and purposes of its class, by simultaneously fulfilling all domain and class-specific constraints. Regarding cyber-attacks targeting IoT systems, realistic adversarial examples must be valid traffic capable of being transmitted through a communication network, as well as coherent cyber-attacks capable of fulfilling their intended malicious purpose.

3.2 Analysis methodology

To perform a trustworthy adversarial robustness analysis of multiple ML models, it is imperative to carry out realistic evasion attack vectors that use valid and coherent examples. The proposed methodology is meant to enable a security-by-design approach during the development of an intelligent system and to be regularly replicated with new data

recordings to ensure that the models continue to be adversarially robust.

Considering that network intrusion detection systems are developed in a secure environment and deployed with security measures to encapsulate the utilized models, an attacker will not likely have access neither to a model's training set nor to its internal parameters. Therefore, in addition to fulfilling all domain and class-specific constraints, an adversarial attack method will have to rely solely on a model's class predictions in a black-box or gray-box setting, depending on the available system information about the model and feature set [39]. This attack vector can be simulated by solely giving an evasion attack method access to a holdout set with IoT network traffic that a model has not yet seen. The analysis can be performed in four steps:

1. Prepare the data, creating training and holdout sets.
2. Train and validate an ML model, using the training set.
3. Perform an evasion attack to create a model-specific adversarial holdout set, using the regular holdout set and the model's class predictions.
4. Evaluate the model's performance on the regular and adversarial holdout sets, analyzing its generalization to regular data and its robustness to adversarial data.

In addition to a regularly trained model, an adversarial training approach can be included to analyze the trade-off of performance on regular data to improve the performance on adversarial data. The complete analysis can be performed in five steps:

1. Prepare the data, creating training and holdout sets.
2. Create a simple data perturbation in a copy of each sample of the regular training set, creating an augmented adversarial training set with more data variations.
3. Train and validate two ML models, the first using the regular training set and the second using the adversarial training set.
4. Perform two evasion attacks to create two model-specific adversarial holdout sets, using the regular holdout set and each model's class predictions.
5. Evaluate each model's performance on the regular and adversarial holdout sets, comparing their generalization to regular data and their robustness to adversarial data.

From the comparison performed in the last step, the model with the most adversarially robust generalization can be selected for deployment. Posteriorly, if new data is recorded, this methodology can be replicated to anticipate possible threats and use that knowledge to improve the defense strategy (see Fig. 1).

4 Experimental evaluation

This section describes the experimental evaluation performed following the proposed methodology, including the considered scenarios and datasets, and the utilized adversarial method, ML models, and performance evaluation metrics. The analysis was carried out on a machine with 16 gigabytes of random-access memory, an 8-core central processing unit, and a 6-gigabyte graphics processing unit. The implementation relied on the Python 3 programming language and the following libraries: *numpy* and *pandas* for data preparation and manipulation, *scikit-learn* for the implementation of RF and IFOR, *xgboost* for XGB, and *lightgbm* for LGBM. The previously developed *a2pm* library was used to perform a constrained adversarial example generation.

4.1 Scenarios and datasets

Two distinct scenarios were considered for IoT network intrusion detection: binary and multi-class classification. In the former, the aim of a model was to detect that a network traffic flow was malicious, whereas in the latter, a model had to correctly identify each cyber-attack class and distinguish between them.

Both scenarios included the IoT-23 [40] and Bot-IoT [41] datasets. These are public datasets that contain multiple labeled captures of benign and malicious network flows within IoT systems. The recorded data is extremely valuable because it manifests real IoT network traffic patterns and includes various classes of common cyber-attacks. The former was created in the Stratosphere Research Laboratory and contains twenty-three labeled captures of malware attacks targeting real IoT devices between 2018 and 2019. Despite the latter also incorporating simulated devices and services, it resulted from a realistic testbed of botnet activity developed at the University of New South Wales. Table 1 provides an overview of the main characteristics of the datasets. The class labels were either benign or the name of a cyber-attack class, such as Distributed DoS (DDoS) and Command and Control (C&C).

A preprocessing stage was applied to both datasets, considering their distinct characteristics. First, the features that did not provide valuable information about a flow's benign or malicious purpose, such as origin and destination addresses, were discarded. Then, one-hot encoding was employed to convert the categorical features to numeric values. Due to their high cardinality, low-frequency categories were aggregated into a single category to avoid encoding qualitative values that had a small relevance. Finally, the data was randomly split

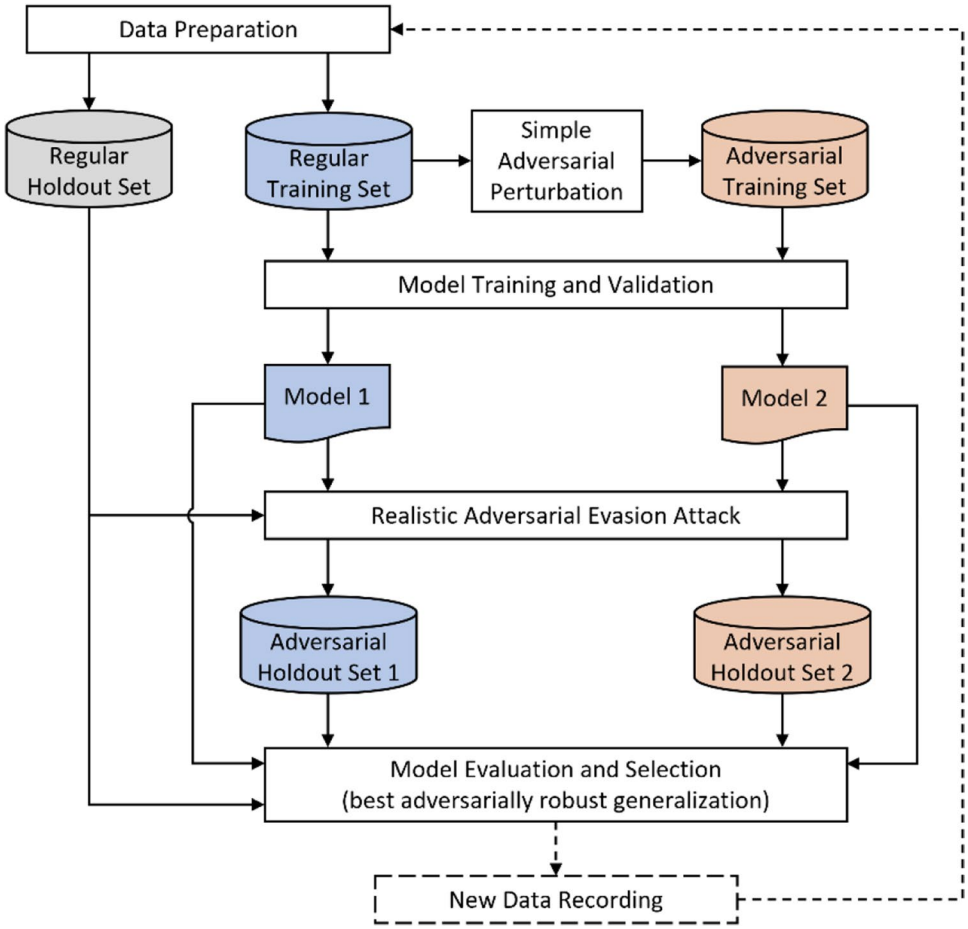

Fig. 1 Adversarial robustness analysis methodology

Table 1 Main characteristics of utilized datasets

Dataset	Selected captures	Total samples	Class samples	Class label
IoT-23	1–1 34–1	1,031,893	539,587	POAHPS
			471,198	Benign
			14,394	DDoS
			6714	C&C
Bot-IoT	Full5pc-4	668,522	576,884	DDoS
			91,082	Recon
			477	Benign
			79	Theft

into training and holdout sets with 70% and 30% of the samples, respectively. To preserve the imbalanced class proportions, the split was performed with stratification. The resulting IoT-23 sets were comprised of four classes and 42 features, 8 numerical and 34 categorical. Similarly, the Bot-IoT sets contained four classes and 35 features, 15 numerical and 20 categorical.

4.2 Adversarial method

The realistic data perturbations required for a trustworthy analysis were created with A2PM [32]. It relies on sequences of adaptative patterns that learn the characteristics of each class. The patterns record the value intervals of individual features and value combinations of multiple features of tabular data. The learnt characteristics are then used to generate constrained adversarial examples that are coherent with the characteristics of their class and simultaneously remain valid within a domain.

Considering that the benign class represents regular IoT network traffic that is not part of an attack, A2PM was applied solely to samples of cyber-attack classes. The method was configured to use independent patterns for specific feature subsets, accounting for the constraints of numerical features and the correlation between encoded categorical features like the destination port, the communication protocol, and the connection flags. Then, two different functionalities were used to perform a simple perturbation and a full evasion attack. These exhibit distinct behaviors and were adapted to different data to prevent any bias in the evaluation of the adversarially trained model.

Simple adversarial perturbation The method was adapted solely to the characteristics of the regular training set, and then a single perturbation was created in a copy of each malicious sample of that set. This resulted in an adversarial training set with twice as many malicious samples as the regular set, so a model could learn not only from a recorded cyber-attack, but also from a simple variation of it.

A security practitioner could perform these simple perturbations manually by analyzing the entire dataset and adding modified samples according to the characteristics of each cyber-attack class. Nonetheless, the automated process of A2PM was preferred. When compared to the training time of an ML model, the few additional seconds required to create the simple sample variations were negligible.

Realistic adversarial evasion attack The method was adapted solely to the characteristics of the regular holdout set, and then a full evasion attack was performed, creating as many data perturbations as necessary in a copy of each malicious sample of that set until every flow was misclassified or a maximum of 30 misclassification attempts were performed. This resulted in an adversarial holdout set with the same size as the regular set, but where each malicious sample was replaced with an adversarial example.

In the multi-class scenario, the performed adversarial evasion attacks could be untargeted, causing any misclassification of malicious samples to different classes, as well as targeted, seeking to misclassify malicious samples as the benign class. In turn, in the binary scenario, both types of evasion attacks were equivalent because all cyber-attacks were aggregated into a single class.

4.3 Models and fine-tuning

The RF, XGB, LGBM, and IFOR algorithms were used to create distinct models for each dataset and scenario, which were fine-tuned through a grid search of well-established hyperparameter combinations for cyber-attack classification. To determine the optimal configuration for each model, a fivefold cross-validation was performed. Therefore, in each iteration, a model was trained with 4/5 of a training set and validated with the remaining 1/5. The macro-averaged F1-score was selected as the validation metric to be maximized in both regular and adversarial training, which will be detailed in the next subsection. After being fine-tuned, each model was retrained with a complete training set and evaluated using the corresponding holdout set.

Random forest RF [42] is a supervised ensemble of decision trees, which are decision support tools that use a tree-like structure. Each individual tree performs a prediction according to a specific feature subset, and the most voted class is chosen. It is based on the wisdom of the crowd—the concept that the collective decisions of multiple classifiers will be better than the decisions of just one.

Table 2 Summary of RF configuration

Parameter	Value
Criterion	Gini impurity
No. of estimators	100
Max. depth of a tree	16
Max. features	$\sqrt{\text{No. of features}}$
Min. samples in a leaf	2 to 4

Table 3 Summary of XGB configuration

Parameter	Value
Method	Histogram
Loss function (objective)	Cross-entropy
No. of estimators	80 to 120
Learning rate	0.01 to 0.2
Max. depth of a tree	8
Min. loss reduction (gamma)	0.01
Feature subsample	0.7 to 0.8

The default Gini impurity criterion was used to measure the quality of the possible node splits, and the maximum number of features selected to build a tree was the square root of the total number of features of each dataset. The optimized value for the maximum depth of a tree was 16, and the minimum number of samples required to create a leaf node was 2 and 4 for the binary and multi-class scenarios, respectively. Table 2 summarizes the configuration.

Extreme gradient boosting XGB [43] performs gradient boosting using a supervised ensemble of decision trees. A level-wise growth strategy is employed to split nodes level by level, seeking to minimize a loss function during the training of the ensemble.

The acknowledged cross-entropy loss was used for both binary and multi-class scenarios, and the histogram method was selected because it computes fast histogram-based approximations to choose the best splits. The key parameter of this model is the learning rate, which controls how quickly the model adapts its weights to the training data. It was optimized to relatively small values for each training set and scenario, ranging from 0.01 to 0.2. Table 3 summarizes the configuration.

Light gradient boosting machine LGBM [44] also utilizes a supervised ensemble of decision trees to perform gradient boosting. Unlike XGB, a leaf-wise strategy is employed, following a best-first approach. Hence, the leaf with the maximum loss reduction is directly split in any level.

Table 4 Summary of LGBM configuration

Parameter	Value
Method	GOSS
Loss function (objective)	Cross-entropy
No. of estimators	80 to 120
Learning rate	0.01 to 0.2
Max. depth of a tree	16
Max. leaves in a tree	32
Min. loss reduction (gamma)	0.01
Min. samples in a leaf	16
Feature subsample	0.7 to 0.8

Table 5 Summary of IFOR configuration

Parameter	Value
No. of estimators	100
Contamination	0.4 to 0.5
Max. features	0.9
Max. samples	256

The key advantage of this model is its ability to use gradient-based one-side sampling (GOSS) to build the decision trees, which is computationally lighter than previous methods and therefore provides a faster training process. The cross-entropy loss was also used, and the minimum samples required to create a leaf were optimized to 16. To avoid fast convergences to suboptimal solutions, the learning rate was also kept at small values for the distinct datasets and scenarios. Table 4 summarizes the configuration.

Isolation forest IFOR [45] isolates anomalies through an unsupervised ensemble of decision trees. The samples are repeatedly split by random values of random features until outliers are segregated from normal observations. Unlike the previous algorithms, IFOR can only perform anomaly detection with unlabeled data. Nonetheless, it can be compared to the remaining models in the binary scenario, so cross-validation was also utilized to optimize its configuration.

This model relies on the contamination ratio of a training set, which must not exceed 50%. Hence, the number of samples intended to be anomalies must be lower than the number of remaining samples; otherwise, outliers cannot be detected. To reduce the contamination of the training data, each cyber-attack class was randomly subsampled with stratification. The optimized ratios of the total proportion of malicious samples were 0.4 and 0.5 for IoT-23 and Bot-IoT, respectively. Therefore, the training data contained 40% and 50% of anomalies. Table 5 summarizes the configuration.

4.4 Evaluation metrics

To analyze a model's robustness, its performance on the regular holdout set was compared to its performance on its respective adversarial holdout set. The considered evaluation metrics and their interpretation are briefly described below [46, 47].

Accuracy is a standard metric for classification tasks that measures the proportion of correctly classified samples. It uses the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) reported by the confusion matrix. However, its bias towards the majority classes must not be disregarded when the minority classes are particularly relevant, which is the case of cyber-attack classification. Since A2PM generated adversarial examples solely for malicious samples, even if all examples evaded detection, an accuracy as high as the proportion of benign flows could still be achieved. Therefore, to correctly exhibit the misclassifications caused by the performed attacks, the accuracy score was calculated using the samples of all classes except the benign class. It can be expressed as:

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

Despite the reliability of accuracy, there are other suitable metrics. For instance, precision measures the proportion of predicted attacks that were actual attacks, which indicates the relevance of a model's predictions. On the other hand, recall, which corresponds to TPR, measures the proportion of actual attacks that were correctly predicted, reflecting a model's ability to identify malicious flows. Another valuable metric is the false positive rate because it measures the proportion of benign flows that were incorrectly predicted to be attacked, leading to false alarms.

These metrics are indirectly consolidated in the F1-score, which calculates the harmonic mean of precision and recall. A high F1-score indicates that malicious flows are being correctly identified and there are low false alarms. It can be macro-averaged to give all classes the same relevance, which is well-suited for imbalanced training data. Due to the consolidation of multiple metrics, the macro-averaged F1-score was the preferred metric for the model fine-tuning. It is mathematically defined as:

$$\text{Macro-averaged F1-Score} = \frac{1}{\text{C}} * \sum_{\text{i}=1}^{\text{C}} \frac{2 * \text{P}_\text{i} * \text{R}_\text{i}}{\text{P}_\text{i} + \text{R}_\text{i}} \tag{2}$$

where P_i and R_i are the precision and recall of class i, and C is the number of classes.

5 Results and discussion

This section presents the results obtained by the four tree-based algorithms in the binary and multi-class scenarios, as well as a comparative analysis of their robustness against adversarial network flow examples, with regular and adversarial training approaches.

5.1 Binary classification

In the binary scenario, the models created with regular training exhibited reasonable performance declines on the IoT-23 dataset. Even though all four models achieved over 99% accuracy on the original holdout set, numerous misclassifications were caused by the adversarial attacks. The lowest score on an adversarial set, 68.35%, was obtained by XGB. In contrast, the models created with adversarial training kept significantly higher scores. By training with one realistically generated example per malicious flow, all models successfully learnt to detect most cyber-attack variations. IFOR stood out for preserving the 99.98% accuracy it obtained on the original holdout set throughout the entire attack, which highlighted its excellent generalization (see Fig. 2).

Regarding the Bot-IoT dataset, the declines were significantly higher. The inability of these tree-based algorithms to distinguish between the different classes evidenced their high susceptibility to adversarial examples. The score of LGBM dropped to 26.04%, followed by IFOR, with 34.31%. Regarding the latter, it could not reach 85% in the original holdout set, possibly due to the occurrence of overfitting. Despite some examples still deceiving them, the models created with adversarial training were able to learn the subtle nuances between each cyber-attack class, which mitigated

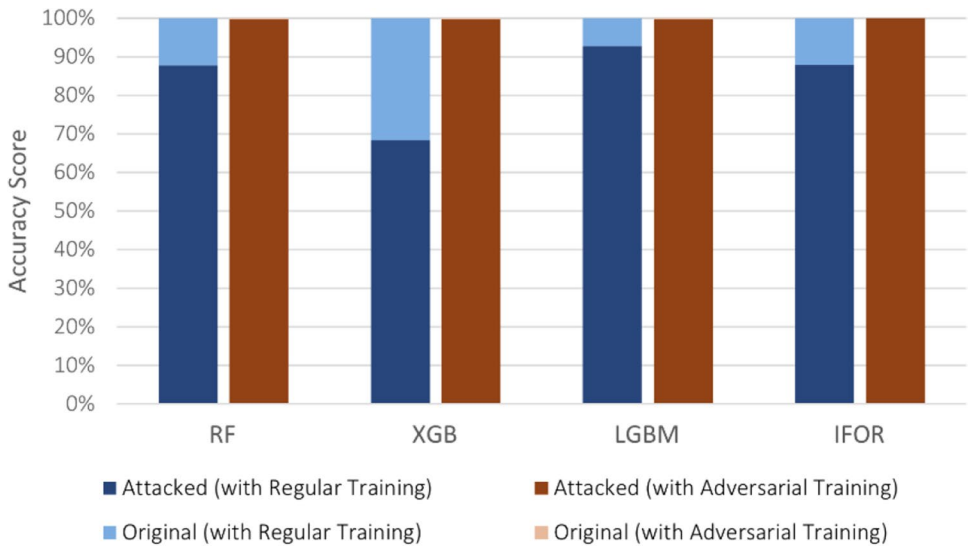

Fig. 2 Accuracy on IoT-23 binary classification

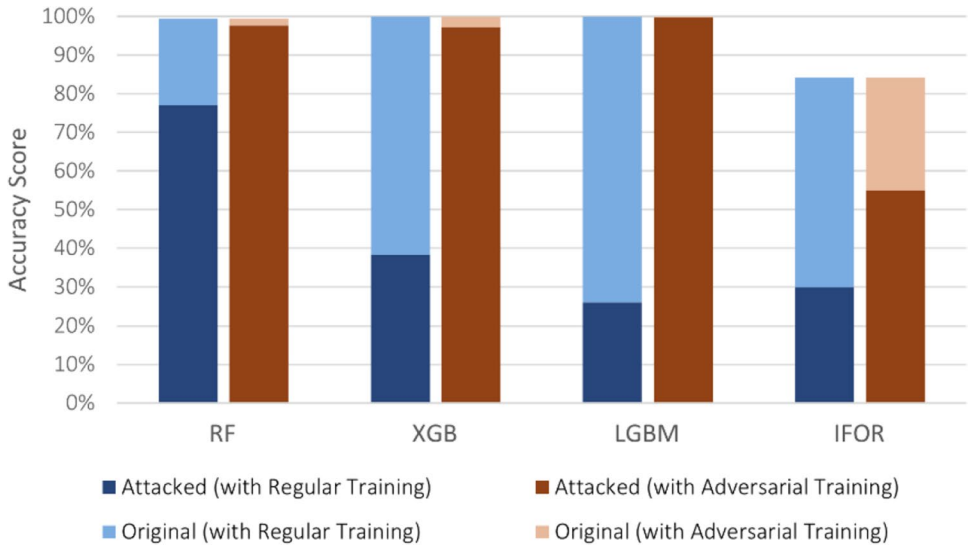

Fig. 3 Accuracy on Bot-IoT binary classification

the impact of the generated examples. Apart from IFOR, the remaining models consistently achieved scores over 97%, which indicated a good robustness (see Fig. 3).

5.2 Multi-class classification

In the multi-class scenario, the targeted and untargeted attacks had different impacts on a model's performance. The former caused malicious flows to be solely predicted as the benign class, whereas the latter caused malicious flows to be predicted as different classes, including other cyber-attack classes. Both attacks decreased the accuracy of the three supervised models on IoT-23, with LGBM being significantly more affected. Nonetheless, it can be observed that its targeted accuracy, 57.78%, was significantly higher than the untargeted, 32.11%, with more misclassifications occurring between different cyber-attack classes. Therefore, despite LGBM being susceptible, the benign class was more difficult to reach in multi-class intrusion detection. Even though performing adversarial training further increased the high scores of XGB, it was surpassed by RF on the targeted attack, which achieved 99.97% (see Figs. 4 and 5).

As in the previous scenario, higher declines were exhibited for the Bot-IoT dataset. The untargeted attacks performed by A2PM dropped the accuracy of RF and XGB to

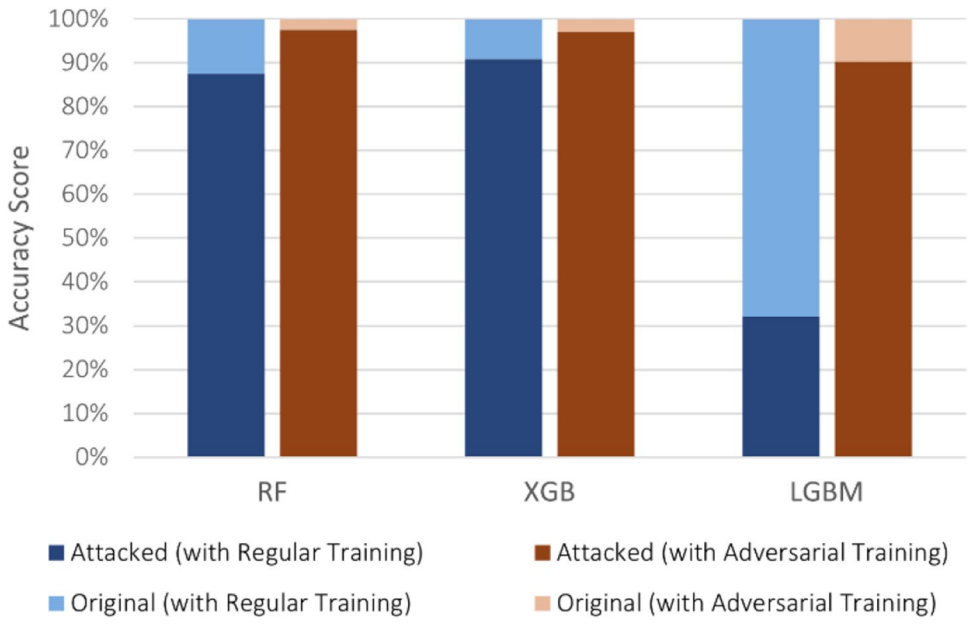

Fig. 4 Untargeted accuracy on IoT-23 multi-class classification

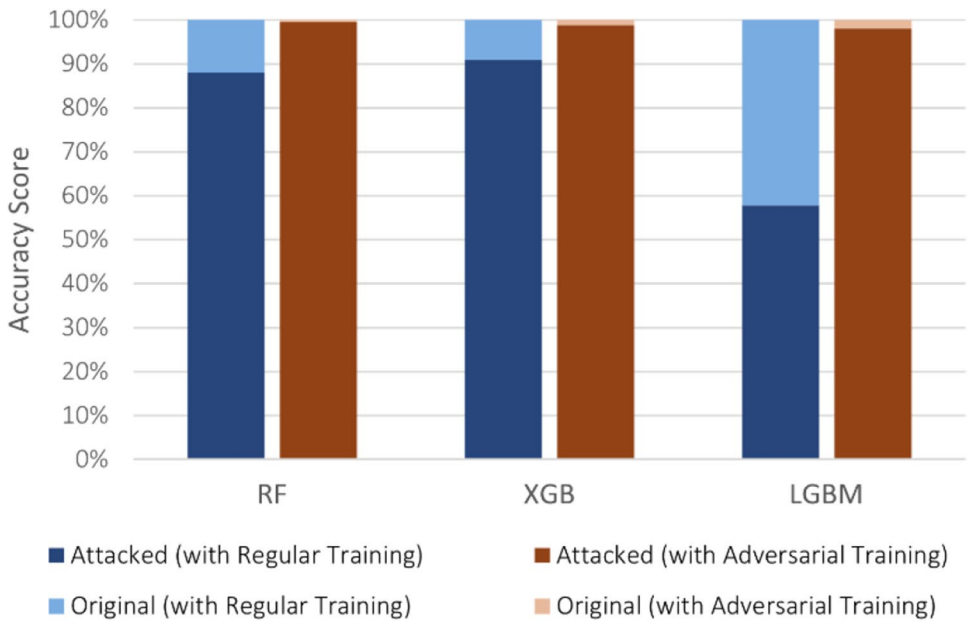

Fig. 5 Targeted accuracy on IoT-23 multi-class classification

nearly 65%, although the targeted attacks only decreased it to 87.50% and 97.14%. Adversarial training contributed to the creation of more robust models, leading to fewer incorrect class predictions. Regarding RF, it could even preserve the 99.98% score it obtained on the holdout set throughout the entire attack. Even though some malicious flows still evaded detection, the robustness of both XGB and LGBM was also successfully improved. Overall, the adversarial robustness of the analyzed tree-based algorithms was significantly improved by augmenting their training data with a simple variation of each cyber-attack (see Figs. 6 and 7).

6 Conclusions

This work addressed the use of ML for IoT network intrusion detection from an adversarial robustness perspective. The types of constraints required for an adversarial cyber-attack example to be valid and coherent were described, and a methodology was proposed for a trustworthy adversarial robustness analysis. The methodology was followed to analyze the robustness of four algorithms, RF, XGB, LGBM, and IFOR, using the IoT-23 and Bot-IoT datasets. Targeted and untargeted adversarial evasion attacks were performed with A2PM,

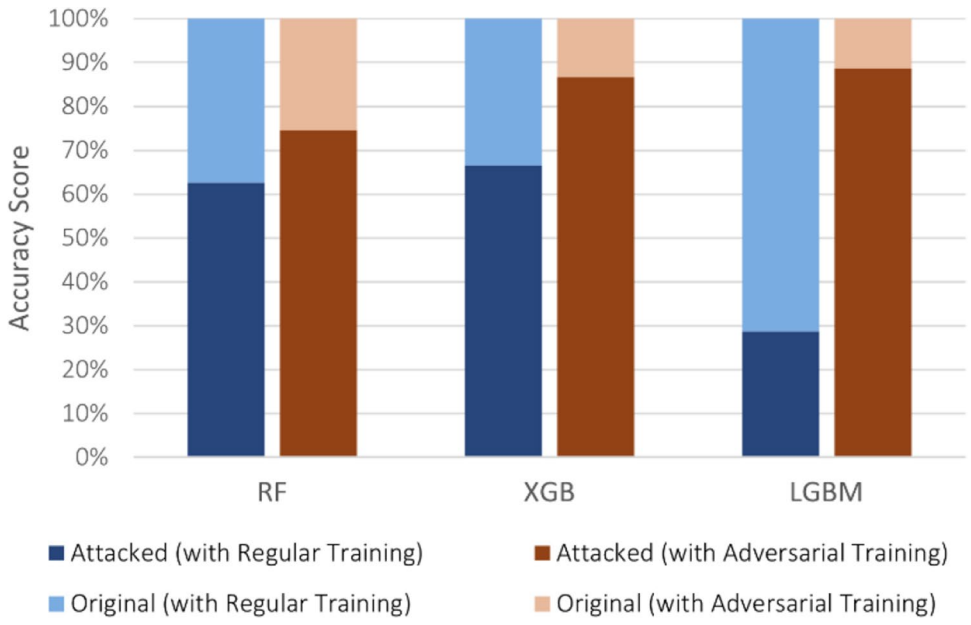

Fig. 6 Untargeted accuracy on Bot-IoT multi-class classification

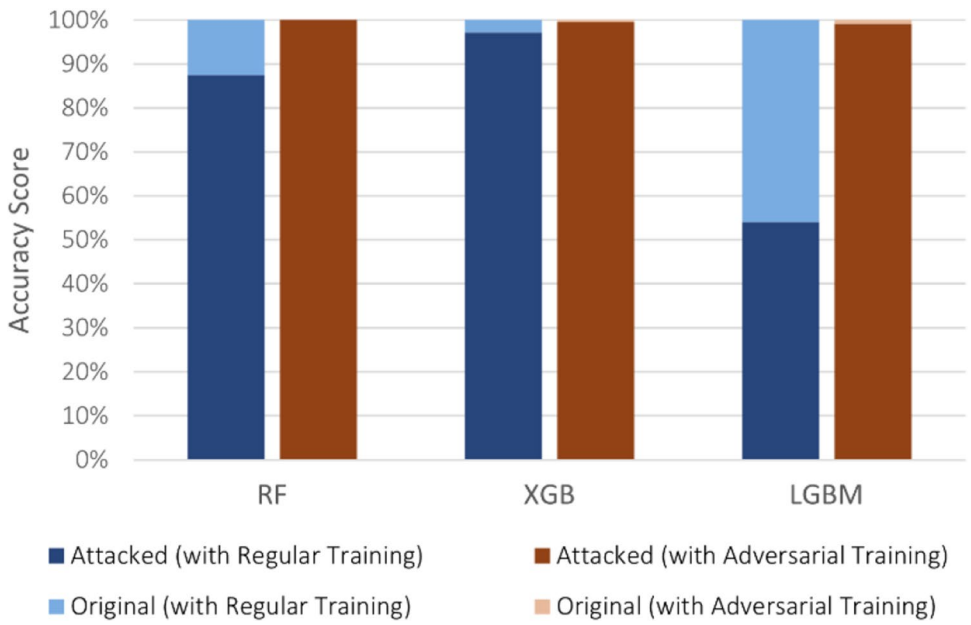

Fig. 7 Targeted accuracy on Bot-IoT multi-class classification

and both regular and adversarial training approaches were evaluated in binary and multi-class classification scenarios.

The models created with regular training exhibited significant performance declines, which were more prominent on the Bot-IoT dataset. Even though RF was the least affected in the binary scenario, XGB consistently achieved the highest accuracy on multi-class classification. Furthermore, when adversarial training was performed, all four models successfully learnt to detect most cyber-attack variations and kept significantly higher scores when attacked. The adversarially trained IFOR and RF stood out for preserving the highest accuracy throughout entire attacks, on binary IoT-23 and multi-class Bot-IoT, respectively. Regarding LGBM, the obtained results suggest that it is highly susceptible to adversarial examples, especially on imbalanced multi-class classification. Nonetheless, this vulnerability can be successfully tackled by augmenting its training data with one realistic adversarial example per malicious flow.

The performed analysis evidenced the inherent susceptibility of tree-based algorithms to adversarial examples and demonstrated that they can benefit from defense strategies like adversarial training to create more robust models. In the future, it is pertinent to further contribute to robustness research by replicating this methodical analysis with novel datasets, ML models, and evasion attack methods. As the threat of adversarial attacks increases, defense strategies must be improved and a security-by-design approach must be followed to ensure that ML models can provide a reliable and robust IoT network intrusion detection and cyber-attack classification.

Author contribution Conceptualization, J.V. and I.P.; methodology, J.V.; software, J.V.; validation, E.M. and I.P.; investigation, J.V. and E.M.; writing, J.V. and E.M.; supervision, I.P.; project administration, I.P.; funding acquisition, I.P. All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). The present work was partially supported by the Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project "Cybers SeC IP" (NORTE-01-0145-FEDER-000044). This work has also received funding from UIDB/00760/2020.

Data availability Publicly available datasets were analyzed in this work. The data can be found at IoT-23 (https://www.stratosphereips.org/datasets-iot23), Bot-IoT (https://research.unsw.edu.au/projects/bot-iot-dataset). A publicly available method was utilized in this work. The method can be found at A2PM (https://github.com/vitorinojoao/a2pm).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Butun I, Osterberg P, Song H (2020) Security of the internet of things: vulnerabilities, attacks, and countermeasures. IEEE Commun Surv Tutorials 22(1):616–644. https://doi.org/10.1109/COMST.2019.2953364
2. Sisinni E, Saifullah A, Han S, Jennehag U, Gidlund M (2018) Industrial internet of things: challenges, opportunities, and directions. IEEE Trans Ind Informatics 14(11):4724–4734. https://doi.org/10.1109/TII.2018.2852491
3. Neshenko N, Bou-Harb E, Crichigno J, Kaddoum G, Ghani N (2019) Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale iot exploitations. IEEE Commun Surv Tutorials 21(3):2702–2733. https://doi.org/10.1109/COMST.2019.2910750
4. Srivastava A, Gupta S, Quamara M, Chaudhary P, Aski VJ (2020) Future IoT-enabled threats and vulnerabilities: state of the art, challenges, and future prospects. Int J Commun Syst 33:12. https://doi.org/10.1002/dac.4443
5. Anand S, and Routray SK (2017) "Issues and challenges in healthcare narrowband IoT," in 2017 Int. Conf. on Inventive Communication and Computational Technologies (ICICCT) pp.486–489https://doi.org/10.1109/ICICCT.2017.7975247
6. Andrade R, Vitorino J, Wannous S, Maia E, Praça I (2022) LEMMAS: a secured and trusted local energy market simulation system, in 2022 18thInt. Conf. on the European Energy Market (EEM) pp. 1–5. https://doi.org/10.1109/EEM54602.2022.9921159
7. Tuptuk N, Hazell P,ll Watson J, and Hailes S, (2021) "A systematic review of the state of cyber-security in water systems," Water 13:1 https://doi.org/10.3390/w13010081
8. European Union Agency for Cybersecurity, A. Malatras, and G. Dede (2020) "AI cybersecurity challenges: threat landscape for artificial intelligence," https://doi.org/10.2824/238222
9. Salman O, Elhajj IH, Kayssi A, Chehab A (2020) A review on machine learning–based approaches for Internet traffic classification. Ann Telecommun 75(11):673–710. https://doi.org/10.1007/s12243-020-00770-7
10. Belavagi MC, Muniyal B (2016) Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Comput Sci 89:117–123. https://doi.org/10.1016/j.procs.2016.06.016
11. European Union Agency for Cybersecurity, A. Malatras, I. Agrafiotis, and M. Adamczyk, (2022) "Securing machine learning algorithms," https://doi.org/10.2824/874249
12. Papadopoulos P, Thornewill von Essen O, Pitropakis N, Chrysoulas C, Mylonas A, Buchanan WJ (2021) Launching adversarial attacks against network intrusion detection systems for IoT. J Cybersecurity Priv 1(2):252–273. https://doi.org/10.3390/jcp1020014
13. Biggio B, Fumera G, Roli F (2014) Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 26(4):984–996. https://doi.org/10.1109/TKDE.2013.57

14. Martins N, Cruz JM, Cruz T, Henriques Abreu P (2020) Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. IEEE Access. 8:35403–35419. https://doi.org/10.1109/ACCESS.2020.2974752
15. G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, (2021) "Modeling realistic adversarial attacks against network intrusion detection systems," Digit. Threat. Res. Prac. 1 1 https://doi.org/10.1145/3469659
16. Vitorino J, Andrade R, Praça I, Sousa O, Maia E (2022) A comparative analysis of machine learning techniques for IoT intrusion detection, in Foundations and Practice of Security 191–207. https://doi.org/10.1007/978-3-031-08147-7_13
17. Anthi E, Williams L, Rhode M, Burnap P, Wedgbury A (2021) Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J Inf Secur Appl 58, no. February, p. 102717. https://doi.org/10.1016/j.jisa.2020.102717
18. Apruzzese G, Andreolini M, Colajanni M, Marchetti M (2020) Hardening random forest cyber detectors against adversarial attacks. IEEE Trans Emerg Top Comput Intell 4(4):427–439. https://doi.org/10.1109/TETCI.2019.2961157
19. Kantchelian A, Tygar JD, Joseph AD (2016) Evasion and hardening of tree ensemble classifiers, 33rd Int. Conf Mach Learn 5:3562–3573
20. Chen Y, Wan S, Jiang W, Cidon A, and Jana S. (2021) "Cost-aware robust tree ensembles for security applications," Proc. 30th USENIX Secur. Symp 2291–2308
21. Chen H, Zhang H, Boning D, and Hsieh CJ (2019) "Robust decision trees against adversarial examples," https://doi.org/10.48550/ARXIV.1902.10660
22. Vos D, Verwer S (2021) Efficient training of robust decision trees against adversarial examples, in 38th Int Conf Mach Learn 139:10586–10595
23. Shafahi A et al (2019) "Adversarial training for free!" in Advances in Neural Information Processing Systems, vol. 32, available: https://proceedings.neurips.cc/paper/ 2019/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf
24. Andriushchenko M, Flammarion N (2020) Understanding and improving fast adversarial training. Adv. Neural Inf. Proces. Syst. 33:16048–16059
25. Stutz D, Hein M, Schiele B (2019) "Disentangling adversarial robustness and generalization", in. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019:6969–6980. https://doi.org/10.1109/CVPR.2019.00714
26. Li Y, Jiang Y, Li Z, and Xia S.-T, (2022) "Backdoor learning: a survey," IEEE Trans. Neural Networks Learn. Syst. pp. 1–18 https://doi.org/10.1109/TNNLS.2022.3182979
27. Yuan X, He P, Zhu Q, Li X (2019) Adversarial examples: attacks and defenses for deep learning. IEEE Trans neural networks Learn Syst 30(9):2805–2824. https://doi.org/10.1109/TNNLS.2018.2886017
28. Pitropakis N, Panaousis E, Giannetsos T, Anastasiadis E, Loukas G (2019) A taxonomy and survey of attacks against machine learning. Comput. Sci. Rev 34:100199. https://doi.org/10.1016/j.cosrev.2019.100199
29. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) "The limitations of deep learning in adversarial settings", in. IEEE European Symposium on Security and Privacy 2016:372–387. https://doi.org/10.1109/EuroSP.2016.36
30. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 23(5):828–841. https://doi.org/10.1109/TEVC.2019.2890858
31. Merzouk MA, Cuppens F, Boulahia-Cuppens N, Yaich R (2022) Investigating the practicality of adversarial evasion attacks on network intrusion detection. Ann Telecommun. https://doi.org/10.1007/s12243-022-00910-1
32. Vitorino J, Oliveira N, and Praça I (2022) Adaptative perturbation patterns: realistic adversarial learning for robust intrusion detection. Future Internet 14(4). https://doi.org/10.3390/fi14040108
33. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surv Tutorials 21(3):2671–2701. https://doi.org/10.1109/COMST.2019.2896380
34. Zolanvari M, Teixeira MA, Gupta L, Khan KM, Jain R (2019) Machine learning-based network vulnerability analysis of industrial internet of things. IEEE Internet Things J 6(4):6822–6834. https://doi.org/10.1109/JIOT.2019.2912022
35. Verma A, Ranga V (2020) Machine learning based intrusion detection systems for IoT applications. Wirel Pers Commun 111(4):2287–2310. https://doi.org/10.1007/s11277-019-06986-8
36. Yao H, Gao P, Zhang P, Wang J, Jiang C, Lu L (2019) Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection. IEEE Netw 33(5):75–81. https://doi.org/10.1109/MNET.001.1800479
37. Eskandari M, Janjua ZH, Vecchio M, Antonelli F (2020) Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Internet Things J 7(8):6882–6897. https://doi.org/10.1109/JIOT.2020.2970501
38. Shorey T, Subbaiah D, Goyal A, Sakxena A, and Mishra AK (2018) "Performance comparison and analysis of slowloris, goldeneye and Xerxes DDoS attack tools," 2018 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2018, pp. 318–322 https://doi.org/10.1109/ICACCI.2018.8554590
39. Rosenberg I, ShabtaA, Elovic Y, and Rokach L (202) Adversarial machine learning attacks and defense methods in the cyber security domain, ACM Comput Surv 54(5). https://doi.org/10.1145/3453158
40. Garcia S, Parmisano A, Erquiaga MJ (Jan.2020) IoT-23: a labeled dataset with malicious and benign IoT network traffic. Zenodo. https://doi.org/10.5281/zenodo.4743746
41. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Futur Gener Comput Syst 100:779–796. https://doi.org/10.1016/j.future.2019.05.041
42. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324
43. Chen T,and Guestrin C (2016) XGBoost: a scalable tree boosting system. Proc ACM SIGKDD Int Conf Knowl Discov Data Min, vol. 13–17-Augu, pp. 785–794. https://doi.org/10.1145/2939672.2939785
44. Ke G et al (2017) "LightGBM: a highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems, 2017, pp. 3147–3155
45. Liu FT, Ting KM, and Zhou ZH (2008) Isolation forest, Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 413–422. https://doi.org/10.1109/ICDM.2008.17
46. Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. Int J Data Min Knowl Manag Process 5(2):1. https://doi.org/10.5121/ijdkp.2015.5201
47. Khraisat A, Gondal I, Vamplew P, and Kamruzzaman J (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2(1). https://doi.org/10.1186/s42400-019-0038-7