On The Detection Of Adversarial Attacks Through Reliable AI

On The Detection Of Adversarial Attacks Through Reliable AI

Ivan Vaccari
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute
Genoa, Italy
ivan.vaccari@ieiit.cnr.it

Alberto Carlevaro
University of Genoa
DITEN department
CNR - IEIIT
Genoa, Italy
alberto.carlevaro@edu.unige.it

Sara Narteni
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute
Genoa, Italy
Politecnico di Torino
DAUIN Department
sara.narteni@ieiit.cnr.it

Enrico Cambiaso
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute
Genoa, Italy
enrico.cambiaso@ieiit.cnr.it

Maurizio Mongelli
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute
Genoa, Italy
maurizio.mongelli@ieiit.cnr.it

Abstract—Adversarial machine learning manipulates datasets to mislead machine learning algorithm decisions. We propose a new approach able to detect adversarial attacks, based on eXplainable and Reliable AI. The results obtained show how canonical algorithms may have difficulty in identifying attacks, while the proposed approach is able to correctly identify different adversarial settings.

Index Terms—machine learning, detection algorithms, adversarial machine learning, reliable

I. Introduction

A. Background

Machine learning (ML) is being increasingly adopted in many fields of our lives today. It is used for image analytics [1], diseases prevention [2], cyber-attacks discovery [3], [4], in Industry 4.0 [5] and many other applications.

Because of this great spread, the risk of possible attacks on ML systems increased in recent years, giving rise to the adversarial machine learning. The main scope of these attacks is the injection of malicious data (perturbed by an attacker starting from legitimate data) with the aim of making the algorithm fail its predictions [6]. The original idea of adversarial attacks was related to misclassification of images [7], then it was extended to other fields such as intrusion detection systems [8].

B. Contribution

In this paper, we focus on a tough adversarial ML setting, both in terms of the number of attacks, their aggressiveness and with respect to a case study that is already difficult by its nature. This demands for a brand new approach, beyond canonical ML. Custom Reliable AI approaches (built on existing explainable and black box approaches) are then elaborated to individuate the adversarial attacks; reliability allows to guarantee zero statistical error and maximize the number of the detected attacks. These approaches are compared to canonical ML as well.

Specifically, two Reliable AI solutions are investigated. The first is a novel scheme, in the black-box Support Vector Data Description (SVDD) [9], [10] framework, here re-designed to surround the adversarial attacks within a controlled region [11], which we call the adversarial region. The methodology also involves explainability through proper rule extraction.

The second approach comprises three methods based on a natively explainable model (the LLM), yet re-designed for reliability [12]. The aim is the same as above for the SVDD, but it is obtained through sensitivity analysis of rules thresholds, until the constraint on false positives (FPs) has come to convergence.

The performance evaluation of the case study will corroborate the reliability of the threat detection, which is otherwise very hard through canonical ML and shows that at least one of the proposed algorithms outlines adversarial regions with a good trade-off between false positives and false negatives.

II. Related work

The topic of adversarial machine learning has been largely investigated in the scientific community in recent years. The consequent impact on AI certification is becoming an urgent problem as well, see, e.g., [13] in the avionic field (other examples may be related to automotive). Fig. 1 summarizes the EASA [13] concept of the ML lifecycle, possible cyber-threats and related defense; the poisoning attacks are those that corrupt the training data and lead to contamination of the generated ML model, thus altering predictions on new data.

Furthermore, [14] proposes a vision on the possible adversarial attacks exploiting the CIA (confidentiality, integrity and availability) requirements, with a focus on a poisoning attack against images. Also [15] categorizes the possible adversarial attacks occurring in cyber warfare contexts, with focus on privacy issues.

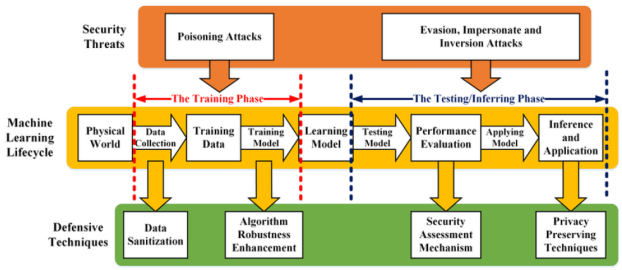


Fig. 1. Illustration of defensive techniques of machine learning in [13]

[16]–[18] instead analyze the bad consequences to which these attacks can lead to, like misclassifications in the medical field, where algorithm failures may not recognize severe diseases.

The adversarial ML framework is also studied to enhance malware detection systems, where ML algorithms are adopted to detect malicious mobile apps [19], [20]. Due to the high sensitivity of smartphones data, a correct protection against malware is fundamental [21], [22]

Speech recognition is another field subject to adversarial attacks.In [23], [24] the robustness of neural networks for speech recognition to possible adversarial attacks is investigated. Authors discover the weaknesses of the recognition models towards these attacks.

A critical context where ML algorithms are widely used is the Internet of Things (IoT). [25], [26] focus on how an adversarial attack could alterate the detection of a cyber-attack against IoT devices, generating unwanted alarms. [27] studies an adversarial ML attack by using a partial-model attack to manipulate the data fusion/aggregation process of IoT: aim of this work is to lead the model to take a wrong decision with respect to the input data of the IoT sensors.

III. Work concept

A. Detection

We considered the following attacks: Carlini-Wagner (CW) [28], the Fast Gradient Sign Method (FGSM) [29] and the Jacobian based Saliency Map (JSMA) [6].

The detection phase involves the solution of canonical supervised learning problem, combining legitimate and adversarial data and training further ML models. Both canonical ML algorithms and Reliable AI methods are adopted to discriminate the attacks. The adopted classifiers are designed to identify as many attacks as possible (minimize false positives). In this way, some legitimate data may be misclassified as malicious (increase of false negatives), but a good balance is seeked under the proposed Reliable AI. As to Fig. 1 again, our approach consists in a defensive technique through robustness enhancement outside the main training model, which is specific for the target application, e.g., visual landing, predictive maintenance, see, e.g., the Annex 2 of the EASA doc [13]. Our detection (through reliable ML) understands if the inputs provided to the machine learning lifecycle (yellow box in Fig. 1) are corrupted.

B. Attacker assumption

Adversarial machine learning algorithms require an underlying algorithm as the victim of their attack. Assuming that an attacker does not know the algorithms of a detection system, in this paper we decided to adopt a neural network as the victim of the various adversarial ML attacks.

In particular, we implemented a neural network made up of 3 layers with 512, 256 and 128 neurons respectively and the output layers. The network is trained with ReLu activation function for the hidden layers, a sigmoid function for the output, an Adam optimizer with learning rate of $1.0e-5$, 300 epochs and a batch size of 16. The accuracy is stably around 95% during all the training phase.

IV. Reliable AI

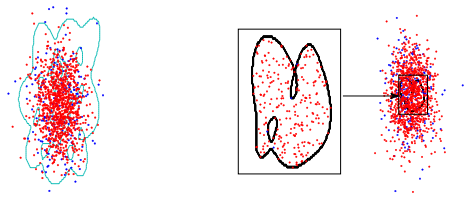
The proposed approaches identify means to surround the adversarial class through confidence envelopes with zero statistical error.

A. Safe SVDD

The SVDD algorithm [9], [10] is a versatile ML tool that is well suited to the field of safety engineering and cybersecurity [11], [30]. The zeroFPRSVDD algorithm performs successive iterations of the SVDD on the safe region, found with a preliminary SVDD, until there are no more negative points inside it. We achieve convergence when we reach a fixed number of iterations or when the condition on FPR is satisfied.

Algorithm 1 zeroFPRSVDD
Data set $\mathcal{X} \times \mathcal{Y}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and test set $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$.
A threshold of ε is set.

1. $\mathcal{S}_1 = \text{SVDD}(\mathcal{X}_{tr}, \mathcal{Y}_{tr})$
2. Test $\mathcal{S}_1$ on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
3. maxiter=1000; i=2;
4. while(i < maxiter)
4.1. $\mathcal{X}_{tr_i} = \mathcal{S}_i(\mathcal{X}_{ts})$;
4.2. $\mathcal{S}_i = \text{SVDD}(\mathcal{X}_{tr_i}, \mathcal{Y}_{tr_i})$
4.3. Test $\mathcal{S}_i$ on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
4.4. if(FPR< ε)
4.4.1. return $\mathcal{S}^* = \mathcal{S}_i$
4.5. end
4.6 $i = i + 1$;
5. end



(a) FPR=0.925 (b) FPR=0.079

Fig. 2. Application of Algorithm 1 on a data set of 2000 target objects sampled from a gaussian with mean $[1, 1]$ and variance 4 and 100 negative examples sampled from a gaussian with mean $[1, 1]$ and variance 5. (a) is the first iteration of the algorithm and (b) is the convergence at the 97th iteration.

B. Logic Learning Machine

Logic Learning Machine (LLM) is a supervised method [31], developed by Rulex [32]. Considering classification tasks, the LLM builds a set of M intelligible rules in if-then format, predicting an output class value based on the logical product of conditions on input variables.

Each rule $\mathbf{r}_k, k = 1, \ldots, M$ can be evaluated by two useful metrics, the covering $C(\mathbf{r}_k)$ and the error $E(\mathbf{r}_k)$, defined as follows:

$$C(\mathbf{r}_k) = \frac{TP(\mathbf{r}_k)}{TP(\mathbf{r}_k) + FN(\mathbf{r}_k)} \quad , \quad E(\mathbf{r}_k) = \frac{FP(\mathbf{r}_k)}{TN(\mathbf{r}_k) + FP(\mathbf{r}_k)} \tag{1}$$

with $TP(\cdot), FP(\cdot), TN(\cdot), FN(\cdot)$ being the confusion matrix values associated to the classification of the data through the rule. Both covering and error are useful to define feature ranking and value ranking. Feature ranking finds out the variables that have a greater impact on the output based on a measure of relevance, which depends on the error and covering measures (Eq. 4 in [33]). Value ranking instead individuates the intervals of values, for each attribute, that impact more on the rules output class.

1) LLM-driven reliable AI: For a XAI-driven detection of adversarial attacks, we propose the novel application of three methods, previously introduced in our work [12], that exploit feature and value ranking to detect the highest number of adversarial attacks with zero FPR: reliability from outside, reliability from inside and LLM with zero error.

The first two methods share the same methodological approach, as summarized in Algorithm 2.

Let $\mathcal{X}$ be the input dataset, with D_1 samples for class $y = 1$ (adversarial class) and D_0 samples for class $y = 0$ (legitimate class).

Algorithm 2 ReliabilityFromLLM
Inputs: dataset $\mathcal{X}$; number of features N_{FR}; candidate perturbations $\Delta = (\boldsymbol{\delta}_1, ..., \boldsymbol{\delta}_{N_{FR}})$, $\boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j})$, $j = 1, ..., N_{FR}$.

1. Apply LLM on $\mathcal{X}$;
2. Select features f_j from feature ranking;
3. Find $[s_j, t_j]$ from value ranking;
4. Define logical OR: $I = \bigcup_{j=1}^{N_{FR}} [s_j, t_j]$;
5. Find hyper-rectangle: $P(\Delta) = \bigcup_{j=1}^{N_{FR}} [s_j \mp \delta_{s_j} \cdot s_j, t_j \pm \delta_{t_j} \cdot t_j]$;
6. Find optimal perturbations Δ^*

Both methods consist in determining the optimal shape of an hyper-rectangle $\mathcal{P}(\cdot)$, by finding the optimal perturbations Δ^* of value ranking thresholds.

This can be differently formalized in the following way.

Reliability from outside method starts from the other class with respect to the target (Sec. 5.1 in [12]), $y = 0$ in our case. Hence, the solution is found by solving:

$$\Delta^* = arg \min_{\Delta: N_0 = D_0} \mathcal{V}(\mathcal{P}(\Delta)) \tag{2}$$

being N_0 the number of elements in X classified as $y = 0$ and included into $\mathcal{P}$, with $\mathcal{V}$ being the volume of $\mathcal{P}$. Since the optimal $\mathcal{P}$ contains all legitimate points, the complementary is considered as adversarial region.

Reliability from inside starts from the target class (see Sec. 5.2 in [12]), i.e. class $y = 1$ in this case. The optimal solution is as follows, with $P(\Delta^*)$ being the adversarial region:

$$\Delta^* = arg \max_{\Delta: N_0 = 0} \mathcal{V}(\mathcal{P}(\Delta)) \tag{3}$$

Since hyper-rectangles shape might be too simple to follow potentially complex boundaries between output classes, another solution consists in joining (in OR operation) the m^0 highest covering rules obtained for the adversarial class by training the LLM with 0% maximum error (LLM 0% from now on, see Section 5.3 in [12]), obtaining predictor $\hat{r}$. Again, feature ranking can be exploited to select N_{FR} features and apply perturbations $\boldsymbol{\delta}$ on their most stringent thresholds present in $\hat{r}$, thus having $\hat{r}(\boldsymbol{\delta})$. The optimal perturbations are chosen according to the following problem:

$$\boldsymbol{\delta}^* = arg \max_{\boldsymbol{\delta}: E(\hat{r}(\boldsymbol{\delta})) = 0} C(\hat{r}(\boldsymbol{\delta})) \tag{4}$$

V. Tests and obtained results

A. DNS tunneling dataset

The used dataset represents a challenging scenario for the detection even outside the adversarial scope. It deals with covert channel detection in cybersecurity [4].

B. Canonical supervised learning with hyperparameter optimization

In order to provide a first possible protection from the adversarial machine learning attacks, we focused on the adoption of classic machine learning (ML) algorithms. As presented in Section V-A, we used the DNS tunneling dataset for the experiments. We implemented different classification algorithms.The algorithms were implemented through the Sklearn [34] library, an open source ML library for Python.

The dataset, composed by balanced legitimate and malicious samples, was split in 70% of training and 30% of test set.

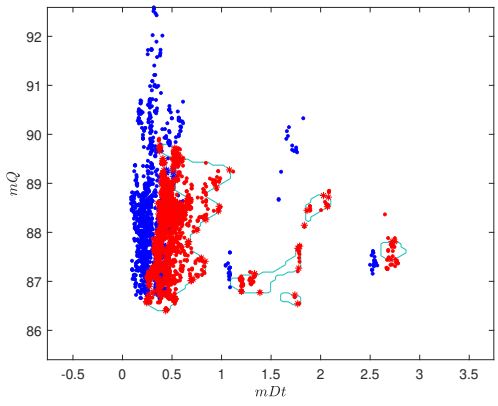


Fig. 3. 2D graph of the "adversarial region" (the red points are the attacked ones) with mDt (average interarrival time between query and answer packet over 1000 sample) and mQ (average size of query packet) as input features of the jsma-DNS dataset. The star points are the SVs of the description, coloured referring their specific label.

We consider hyperparameters optimization through Optuna [35], which allows to find out a set of optimal parameters for the models to improve their performance.

In our tests, 1000 parameter combinations with different values (chosen by Optuna according to its intrinsic logic) were performed for each algorithm to allow efficiency and high variety of combinations.

Performance metrics are: false positive rate (FPR), true positive rate (TPR), false negative rate (FNR) and true negative rate (TNR). They are reported in Table I.

Looking at the results, in the FGSM and the JSMA only the native SVM guarantee almost good performance, except for CW in virtue of its larger complexity.

C. Obtained results with zeroFPRSVDD algorithm

The aim is now to determine the largest region of parameters with no false positives (i.e. prediction of attack, but no attack in reality). To do this, we applied the algorithm proposed in Section IV-A using $C_1 = 1/\nu_1 N_1$, where $N_1 = \#\{y_i = +1\}$ and $\nu_1 = 0.01$ (i.e. we allow the acceptance of up to 1% of negative objects in the target class), $C_2 = 1/\nu_2 N_2$ where $N_2 = \#\{y_i = -1\}$ and $\nu_2 = 0.05$ (i.e. we allow up to 5% negative objects to be included in the classifier shape) and RBF kernel with σ determined with cross-validation. The results are shown in Table II for the DNS tunneling dataset (a normalization with the z-score has been performed to improve the computational speed of the algorithm), where FPR, TPR, TNR and FNR are the usual metrics for the confusion matrix, #iter is the number of algorithm iterations, #time (s) is the time in second for the convergence, R^2 is the squared hypersphere's radius. The last column holds the precision on the target class $\frac{TP}{TP+FP}$. When compared to SVM algorithm (to which the SVDD is closely related [9]) we can observe that the results have been improved. It is also possible to combine SVDD and XAI to obtain intelligible rules from the black box [11], [30]. The derivation of intelligible rules is made by proper rule extraction. Differently from [11], we need a more refined sampling of SVDD classification to derived the new dataset, as performed in [30]. The sapling is performed by setting a threshold ε, such that the extracted observations are sufficiently close to the boundary of the trained and tested SVDD. The first highest-covering rule (i.e. the rule involving the largest number of data points, (1)) for the class attack is

$$\begin{aligned}&\text{if} \quad (30931149 < v_A \leq 166588766) \wedge \\ &(211 < v_Q \leq 2604) \ \wedge \ (3779 < vDt \leq 155832) \wedge \\ &(360 < \ sDt \leq 392) \wedge (52 < s_A \leq 326) \wedge \\ &(368 < kDt \leq 4874 \wedge (29 < k_A \leq 328) \\ &\text{then attack}\end{aligned}$$

The fact that the rules are very intricate and that each rule involves almost all input parameters is because we are approximating the nonlinear form of SVDD with hyper-rectangles. To ensure acceptable prediction confidence with these rules, a large amount of them is required: for the cases in example, CW-DNS, the total number of rules generated is 146, respectively. Moreover, having a high number of rules means having low coverage for each rule: this may suggest that, first, the task is very difficult but, second, that the regions developed by SVDD are widely and sporadically distributed inside the space of the input parameters.

Algorithm 3 ExplainableSVDD
Get $\mathcal{S}^*$ from zeroFPRSVDD algorithm. Fix ε.

---

1. Sample uniformly a new dataset $\mathcal{X}_{new}$ s.t. $x_i \in \mathcal{X}_{new} \iff | \ ||x_i - \mathbf{a}||^2 - R^2 \ | < \varepsilon$
2. Classify $\mathcal{X}_{new}$ in $\mathcal{Y}_{new}$ through optimal zeroFPRSVDD (w.r.t. $\mathcal{S}^*$)
3. Solve a classification problem via LLM w.r.t. $[\mathcal{X}_{new}, \mathcal{Y}_{new}]$
4. The LLM rules defines an explained zeroFPRSVDD region $\mathcal{R}$
5. return $\mathcal{R}$

Below, Table III, are the statistics obtained by applying Algorithm 3 for the previous data: the results are quite good, keeping in mind that the classification problem is very difficult and that the procedure of extracting intelligible rules from the model can be affected by overfitting and approximations.

D. Detection through LLM-driven Reliable AI

We now test the methods described in Sec. IV-B1 on the DNS tunneling dataset. The LLM was trained with the default 5% maximum error on a 70% portion of data as training set (the same used for the other detection algorithms).

	CW				JSMA				FGSM			
	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Decision tree	0.50	1.00	0.50	0.00	0.25	1.00	0.75	0.00	0.50	1.00	0.50	0.00
Gradient boost	0.48	1.00	0.52	0.00	0.50	1.00	0.50	0.00	0.03	0.36	0.97	0.64
KNN	0.97	1.00	0.03	0.00	0.89	1.00	0.11	0.00	0.11	0.28	0.89	0.72
Logistic regression	0.49	0.99	0.51	0.01	0.09	0.98	0.91	0.02	0.03	0.99	0.97	0.01
Random forest	0.49	1.00	0.51	0.00	0.50	1.00	0.50	0.00	0.03	0.32	0.97	0.68
SVM	0.39	0.65	0.61	0.35	0.09	0.98	0.91	0.02	0.15	0.95	0.85	0.05

TABLE I
Canonical machine learning with hyperparameters optimization. FPR, TPR, TNR and FNR for each ML algorithm on the adversarial attacks.

	FPR	TPR	TNR	FNR	# iter	# time (s)	R^2	PPV
CW	0.0422	0.3544	0.9578	0.6456	6	59.432	0.5019	0.8936
JSMA	0.1522	0.8589	0.8478	0.1411	2	30.445	0.6064	0.8495
FGSM	0.0344	0.7789	0.9656	0.2211	3	43.282	0.4129	0.9131

TABLE II
zeroFPRSVDD. Algorithm statistics after applying Algorithm 1 for the DNS tunneling dataset.

	FPR	TPR	TNR	FNR
CW	0.2388	0.3544	0.7611	0.6455
JSMA	0.2866	0.5366	0.7133	0.4633
FGSM	0.2866	0.2833	0.7133	0.7166

TABLE III
ExplainableSVDD. FPR, TPR, TNR and FNR for each attacked DNS tunneling dataset based on the SVDD-LLM algorithm for classification.

Concerning reliability from outside and inside methods, we chose $N_{FR} = 2$ and obtained the adversarial regions shown in Tab. IV, along with the performance metrics (FPR, TPR, TNR and FNR) as obtained when the regions were evaluated on the test set.

	METHOD	ADVERSARIAL REGIONS	FPR	TPR	TNR	FNR
CW	Inside	$m_A > 275.7 \vee sDt > 70.65$	0.03	0.45	0.97	0.55
CW	Outside	$mDt < 0.34 \wedge v_A < 25923$	0	0.01	1	0.99
JSMA	Inside	$m_A > 275.7 \vee k_A > 6.99$	0.03	0.93	0.97	0.07
JSMA	Outside	$m_A > 276.58 \wedge v_A < 39286$	0	0.72	1	0.28
FGSM	Inside	$s_A \leq 1.63 \vee m_A > 270.9$	0.04	0.62	0.96	0.38
FGSM	Outside	$s_A \leq 1.68 \wedge m_A > 275.02$	0	0.25	1	0.75

TABLE IV
Inside and Outside. Adversarial regions obtained for DNS tunneling with outside and inside methods.

By looking at the results, we can observe that a TPR higher than 0.60 is reached for JSMA attack detection, with both inside and outside methods, and for FGSM detection with inside method: this means that more than 60% of attacks is detected in these cases. In particular, it is worth underlying the surprising result on JSMA, that can be recognized very well by using the inside method, as shown in the plot in Figure 4). A way to look for more

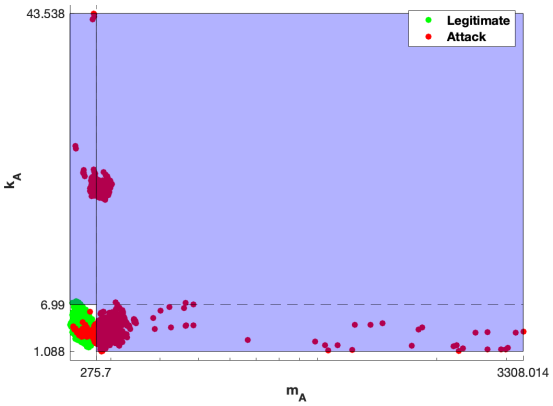


Fig. 4. Adversarial Region obtained for JSMA attack in DNS tunneling dataset by perturbing the intervals thresholds for features m_A and k_A with inside method (TPR=0.93, FPR=0.03, TNR=0.97, FNR=0.07).

refined regions than the sharp rectangles obtained so far is provided by our third method: LLM0% (Eq. 4). First of all, we trained the LLM model by setting the maximum error allowed in each rule to 0%: this resulted in 364 rules for CW attack with covering up to 38%, 40 rules with covering up to 47% for FGSM and 7 rules with covering up to 79% for JSMA.

For each attack case, we decided to select the first 5 highest-covering rules for the adversarial class and merged them in logical OR. Then, we optimized the thresholds of the conditions involving the first two most important features, chosen according to LLM feature ranking, as expressed by Equation 4. The obtained performance metrics are shown in Table V.

Although the optimal solution (zero FPR) is never achieved, it is relevant to underline that LLM0% is the only method, compared to inside and outside, that works better on CW attack than on JSMA or FGSM. This is in reason of the higher complexity involved in LLM 0%

	FPR	TPR	TNR	FNR
CW	0.04	0.44	0.96	0.56
JSMA	0.47	0.50	0.53	0.50
FGSM	0.39	0.42	0.61	0.58

TABLE V
LLM0%. Results of LLM0% on DNS tunneling dataset.

method (see Sec. IV-B1). For CW attack, we report the LLM0% suboptimal predictor below, where the threshold perturbations were applied to features m_A and sDt.

$$\begin{aligned}&\text{if} \quad (m_A > 291.83) \vee \\ &(m_A > 274.55 \ \wedge \ 26257 < v_A \leq 39245) \vee \\ &(m_A > 271.67 \ \wedge \ sDt > 8.14) \vee \\ &(m_A > 269.84 \ \wedge \ 8.98 < vDt \leq 11179 \\ &\wedge \ kDt > 55.19) \ \vee (mDt > 0.95 \\ &\wedge \ m_A > 265.03 \ \wedge \ 223.15 < k_Q \leq 543101255) \\ &\text{then attack}\end{aligned}$$

E. On the choice of the best defense

The best algorithm is chosen as having the minimum FPR (the target of Reliable AI), still maintaining a good balance of FNR. This definition leads to inside method as the best on CW (0.03 FPR, 0.55 FNR) and JSMA (0.03, 0.07) and zeroFPRSVDD for FGSM detection (0.03, 0.22). zeroFPRSVDD and inside thus figure as the most competitive and should be considered jointly if one wants to build the right firewall in front of unknown adversarial threats.

VI. Conclusion and future works

In this paper, we investigated an innovative approach to detect adversarial machine learning attacks by comparing canonical ML algorithms with an innovative approach focused on a Support Vector Data Description (SVDD) and a reliable approach based on explainable AI.

The study may extend the testing through deeper cross-validation in the presence of a large amount of data, including the adoption of explainable data augmentation [2]. The characterization of the placement of the adversarial points, as through rules or other means, deserves further study to understand the behaviour of the attack and profile personalized counterattacks. [33].

References

[1] M. Pak and S. Kim, "A review of deep learning in image recognition," in 2017 4th international conference on computer applications and information processing technology (CAIPT). IEEE, 2017, pp. 1–3.

[2] I. Vaccari, V. Orani, A. Paglialonga, E. Cambiaso, and M. Mongelli, "A generative adversarial network (gan) technique for internet of medical things data," Sensors, vol. 21, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3726

[3] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," Sensors, vol. 20, no. 22, p. 6578, 2020.

[4] M. Aiello, M. Mongelli, and G. Papaleo, "Dns tunneling detection through statistical fingerprints of protocol messages and machine learning," International Journal of Communication Systems, vol. 28, no. 14, pp. 1987–2002, 2015.

[5] I. S. Candanedo, E. H. Nieves, S. R. González, M. T. S. Martín, and A. G. Briones, "Machine learning predictive model for industry 4.0," in International Conference on Knowledge Management in Organizations. Springer, 2018, pp. 501–510.

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016, pp. 372–387.

[7] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," International Journal of Automation and Computing, vol. 17, no. 2, pp. 151–178, 2020.

[8] Y. Pacheco and W. Sun, "Adversarial machine learning: A comparative study on contemporary intrusion detection datasets." in ICISSP, 2021, pp. 160–171.

[9] D. Tax and R. Duin, "Support vector domain description," Pattern Recognition Letters 20, pp. 1191–1199, 1999.

[10] ——, "Support vector domain description," Machine Learning, pp. 45–66, 2004.

[11] A. Carlevaro and M. Mongelli, "Reliable ai trough svdd and rule extraction," International IFIP Cross Domain (CD) Conference for Machine Learning & Knowledge Extraction (MAKE), CD-MAKE 2021., 2021.

[12] S. Narteni, M. Ferretti, V. Orani, I. Vaccari, E. Cambiaso, and M. Mongelli, "From explainable to reliable artificial intelligence," International IFIP Cross Domain (CD) Conference for Machine Learning & Knowledge Extraction (MAKE), CD-MAKE 2021., 2021.

[13] "Easa concept paper: First usable guidance for level 1 machine learning applications, a deliverable of the easa ai roadmap," European Union Aviation Safety Agency, Daedalean, AG, Standard, Apr. 2021.

[14] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," Pattern Recognition, vol. 84, pp. 317–331, 2018.

[15] V. Duddu, "A survey of adversarial machine learning in cyber warfare," Defence Science Journal, vol. 68, no. 4, p. 356, 2018.

[16] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," Science, vol. 363, no. 6433, pp. 1287–1289, 2019.

[17] A. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "Adversarial attacks to machine learning-based smart healthcare systems," arXiv preprint arXiv:2010.03671, 2020.

[18] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," arXiv preprint arXiv:2001.08103, 2020.

[19] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," computers & security, vol. 73, pp. 326–344, 2018.

[20] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," in 2018 26th European signal processing conference (EUSIPCO). IEEE, 2018, pp. 533–537.

[21] L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Explaining vulnerabilities of deep learning to adversarial malware binaries," arXiv preprint arXiv:1901.03583, 2019.

[22] ——, "Functionality-preserving black-box optimization of adversarial windows malware," IEEE Transactions on Information Forensics and Security, 2021.

[23] S. Latif, R. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness," arXiv preprint arXiv:1811.11402, 2018.

[24] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018, pp. 1–7.

[25] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Iot network security from the perspective of adversarial deep learning," in 2019 16th

Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2019, pp. 1–9.
[26] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," in 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019, pp. 1–6.
[27] Z. Luo, S. Zhao, Z. Lu, Y. E. Sagduyu, and J. Xu, "Adversarial machine learning based partial-model attack in iot," in Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 2020, pp. 13–18.
[28] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39–57.
[29] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.
[30] A. Carlevaro and M. Mongelli, "A new svdd approach to reliable and explainable ai," IEEE Intelligent Systems, no. 01, pp. 1–1, oct 5555.
[31] M. Muselli, "Switching neural networks: A new connectionist model for classification," pp. 23–30, 2005.
[32] Rulex analytics platform, https://www.rulex.ai/.
[33] M. Mongelli, "Design of countermeasure to packet falsification in vehicle platooning by explainable artificial intelligence," Computer Communications, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421002504
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[35] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.