IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 July 2022, accepted 1 August 2022, date of publication 8 August 2022, date of current version 15 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197299

RESEARCH ARTICLE

eXplainable and Reliable Against Adversarial Machine Learning in Data Analytics

IVAN VACCARI[1], ALBERTO CARLEVARO[1,2], SARA NARTENI[1,3], ENRICO CAMBIASO[1], AND MAURIZIO MONGELLI[1], (Member, IEEE)

[1]IEIIT Institute, Consiglio Nazionale delle Ricerche (CNR), 16149 Genoa, Italy
[2]Department of Electrical, Electronics and Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, 16145 Genoa, Italy
[3]Department of Control and Computer Engineering (DAUIN), Politecnico di Torino, 10129 Turin, Italy

Corresponding author: Sara Narteni (sara.narteni@ieiit.cnr.it)

ABSTRACT Machine learning (ML) algorithms are nowadays widely adopted in different contexts to perform autonomous decisions and predictions. Due to the high volume of data shared in the recent years, ML algorithms are more accurate and reliable since training and testing phases are more precise. An important concept to analyze when defining ML algorithms concerns adversarial machine learning attacks. These attacks aim to create manipulated datasets to mislead ML algorithm decisions. In this work, we propose new approaches able to detect and mitigate malicious adversarial machine learning attacks against a ML system. In particular, we investigate the Carlini-Wagner (CW), the fast gradient sign method (FGSM) and the Jacobian based saliency map (JSMA) attacks. The aim of this work is to exploit detection algorithms as countermeasures to these attacks. Initially, we performed some tests by using canonical ML algorithms with a hyperparameters optimization to improve metrics. Then, we adopt original reliable AI algorithms, either based on eXplainable AI (Logic Learning Machine) or Support Vector Data Description (SVDD). The obtained results show how the classical algorithms may fail to identify an adversarial attack, while the reliable AI methodologies are more prone to correctly detect a possible adversarial machine learning attack. The evaluation of the proposed methodology was carried out in terms of good balance between FPR and FNR on real world application datasets: Domain Name System (DNS) tunneling, Vehicle Platooning and Remaining Useful Life (RUL). In addition, a statistical analysis was performed to improve the robustness of the trained models, including evaluating their performance in terms of runtime and memory consumption.

INDEX TERMS Machine learning, detection algorithms, adversarial machine learning, reliable.

I. INTRODUCTION

A. BACKGROUND

Machine learning (ML) has become an increasingly used technology in every aspect of our lives. It is adopted for image classification [1], to prevent health diseases [2], in cybersecurity to detect cyber-attacks [3], [4], in the new industrial era (called industry 4.0) [5] or in other fields. It has a significant impact on daily activities and the use of these algorithms aims to improve daily life by offering services and applications capable of making optimal autonomous decisions. Obviously, the huge adoption of these algorithms is due to the large amount of data produced by the birth of emerging technologies such as the Internet of Things, smartwatches and smartphones. The extensive use of these algorithms and approaches has obviously also brought a benefit to the algorithms themselves as they have been more studied and applied, obtaining an improvement in performance, reliability, precision and calculation times.

Given the great use of ML algorithms, possible attacks on these systems have arisen in recent years. In particular, they are called adversarial machine learning. The main scope of these attacks is to inject malicious data (perturbed by an attacker starting from legitimate data) with the aim of making the algorithm misclassify or lower its accuracy [6]. The initial concept of the adversarial machine learning attack was focused on a misclassification of images [7] then it is moved to other fields such as in intrusion detection systems [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

The detection phase of these attacks on ML algorithms is, to date, an important challenge in the research world as it is complex to identify malicious datasets as adversarial machine learning attacks use minimal global perturbations that make identification complex and challenging.

B. CONTRIBUTION

In this work, we propose a new approach to identify an adversarial machine learning attack against a ML algorithm. As many attacks [9] and defensive approaches [10] are consolidated in the image analysis context, the topic is urgent for the more general framework of data analysis. In image settings, defensive techniques are strictly built around the sensitivity analysis of the functional cost of deep learning classifiers [10], [11]. They may thus result inapplicable to other kinds of classifiers in more general data analytics contexts.

We study a tough adversarial setting, both in terms of the number of attacks, their aggressiveness and with respect to two case studies that are already difficult by their nature. This requires a brand new approach, beyond canonical ML. Two kinds of Reliable AI (white and black boxes) are elaborated, in which the statistical error of misclassification is modulated to zero and, at the same time, the number of points characterized by this property is maximized.

C. WHY eXplainable AI

Before Reliable AI, emphasis is put on the explainability of the detection. This helps understand how the attack works and identify the most sensitive areas in the feature space for the attack. eXplainable AI (XAI) [12] may allow to enter the logic of the adversarial moves, as it deals with intelligible rules, whose interpretable, yet powerful, perturbations, as well as feature and value rankings, may shed new light on the knowledge discovery process of the adversarial configuration.

D. WHY RELIABLE AI

The intrinsic statistical error introduced by any ML algorithm may lead to criticism by safety and security engineers [13]. The topic remains a significant challenge in ML as learning algorithms proliferate in difficult real-world pattern recognition applications. In this context, new methodologies associate reliable measures of confidence to pattern recognition settings including classification, regression, and clustering [14]. The proposed approach follows this direction, by identifying methods to circumvent data-driven adversarial envelopes with statistical zero error.

E. BLACK-BOX AND EXPLAINABLE RELIABLE AI

The following Reliable AI solutions are considered. A novel scheme, in the Support Vector Data Description (SVDD) [15], [16] framework, is designed to enclose the adversarial attack within a controlled and explainable region [17], which we will call the *adversarial-region*. The novelty of the scheme relies on finding the best pair of centre $\mathbf{a}$ and radius R of the SVDD, such that there are as few adversarial points in the adversarial-region as possible. In other words, we want to minimize the presence of False Positives (FPs) within the classification task. Then such an optimized SVDD is interrogated in order to find the explainability of it.

We also study other three methodologies that are explainable by nature, yet re-designed for reliability [18]. The aim is the same as above for the SVDD, but it is achieved by proper sensitivity analysis of rules thresholds, until the constraint on FPs has come to convergence.

F. ACHIEVEMENT

The extensive performance evaluation corroborates the reliability of the threat detection, which is otherwise impossible through canonical ML and shows that at least one of the proposed algorithms finds a decision boundary with a good trade-off between false positives and false negatives.

G. STRUCTURE OF THE PAPER

The remaining of the paper is structured as follows: Section II reports the related work about adversarial machine learning concept. Section III introduces the concept of the work with the assumptions for attacker and detection parts, while Section IV describes the adversarial machine learning attacks considered. The new algorithms based on Reliable AI are described in Section V. Executed tests and obtained results are reported in Section VI. Finally, Section VII concludes the paper and reports further works on the topic.

II. RELATED WORK

A. ADVERSARIAL MACHINE LEARNING

The concept of adversarial machine learning has been widely studied in the scientific literature in the last years. The inherent impact on the way to AI certification is becoming an urgent matter as well. For example, in the avionic sector, the EASA vision of ML life-cycle [13], summarized in Fig. 1, poses the attention on inherent cyber-threats and countermeasures. In particular, poisoning attacks are defined as able to corrupt the training data so as to contaminate the ML model generated in the training phase, thus altering predictions on new data.

Technically speaking, [19] proposes an overview of the possible adversarial attacks to exploit the CIA (confidentiality, integrity and availability) requirements with a focus on a poisoning attack against images. Also [20] discusses all the possible adversarial attacks in a specific cyber warfare with a focus on possible privacy aspects. Then [21] offers a broad overview of the most widely used and efficient methodologies for dealing with adversary attacks in AI fields.

[22], [23], [24] instead analyze the issues that these attacks can lead to such as incorrect classifications or predictions in the medical field where an algorithm error may not identify a serious disease.

The adversarial machine learning concept is also considered in the malware detection approach where ML algorithms are adopted to detect a malicious mobile apps [25], [26].

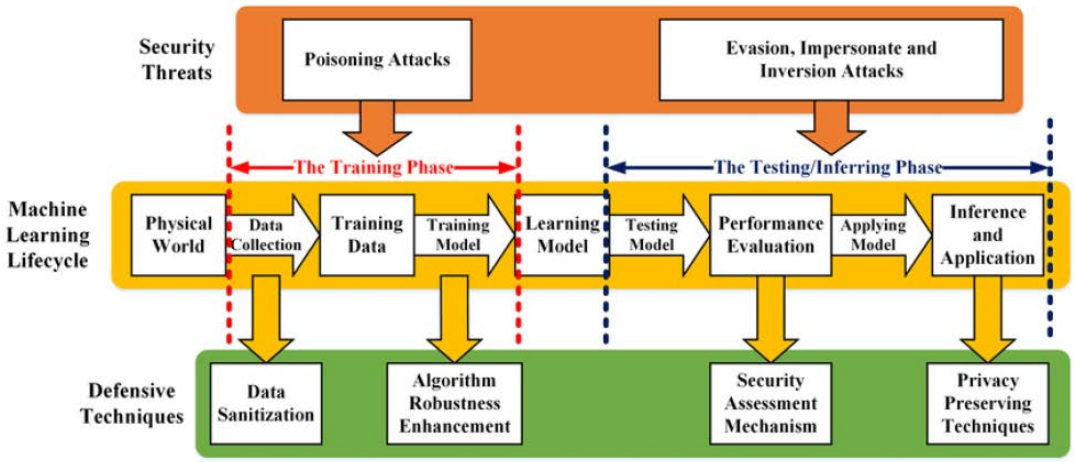


FIGURE 1. Illustration of defensive techniques of machine learning in [13].

This is a critical topic since smartphones contain sensitive information and a malware could retrieve these data, a correct classification aim to protect users from this threats [27], [28].

Another field aimed by adversarial attack is related to speech recognition. [29], [30] discusses the robustness of neural networks, adopted to speech recognition, to possible adversarial attacks. Authors demonstrate weaknesses of the speech algorithm on these attacks.

A critical context where ML algorithms are widely adopted is related to the Internet of Things (IoT). [31], [32] demonstrate how an adversarial attack could cause an alarm in case of fake detection of a cyber-attack against IoT devices. [33] discusses an adversarial machine learning attack by using a partial-model attack in order to manipulate the data fusion/aggregation process of IoT. Scope of this work is to lead the algorithm to make a wrong decision with respect to the input data of the IoT sensors.

Also, in [34], a detection approach of adversarial machine learning attacks is reported and presented. In this work, authors adopted canonical ML approaches to detect two adversarial attacks on a single dataset. Comparing with our work, we evaluated three adversarial attacks with canonical algorithms, innovative SVDD and XAI-based reliable approaches on three different datasets.

These are some examples of possible adversarial attacks against ML algorithms adopted in different contexts. Due to the criticality of this topic, we decided to work on a new approach to detect possible adversarial machine learning attacks by defining new approaches based on reliable AI through native eXplainable AI and SVDD approaches. As obtained results, the proposed algorithms are able to detect adversarial machine learning inference by obtaining a good balance between FPR and FNR. The results obtained demonstrate that the approach through reliable AI is more efficient than classic algorithms, also trained with a hyperparameter optimization. The proposed approach will be deeply discussed in the next sections and the results obtained will be detailed reported in order to demonstrate the efficiency and accuracy of the proposed algorithms and approach. We also We also evaluated our approach on three different datasets focused on different contexts: an intrusion a collision avoidance (platooning) and a predictive maintenance (RUL) scenarios. In the next sections, we will detail the adopted approaches and the obtained results.

B. RELIABLE AI

As far as Reliable AI is concerned, the following considerations are remarkable. Safety regions research is a new trend in ML [14], [17], [35]. The main focus is to control misclassifications, typically associated with dangerous conditions, by searching for regions of a specific target class (e.g., collision in smart mobility scenarios) [36], [37]; also, these studies recently turned into identifying and managing assurance under uncertainty in AI systems [14], [38]. Research topics such as conformal prediction [14], selective prediction [39] and three-way decision making [40] are widely related to the notion of safety region. What all these arguments have in common is to find the largest set of input parameters, such that the prediction can be considered reliable. On the other hand, the hard task consists in finding the right balance between the misclassification and the size of the inherent safety envelope. Our reliable AI approaches pursue this goal by seeking a good balance between FPR and FNR. Moreover, we include an explainable algorithm, either alone or as rule extractor from a black-box (the SVDD) to try to enter the logic of the adversarial attacks.

III. WORK CONCEPT

A. PRINCIPLE BEHIND ADVERSARIAL

ML and artificial intelligence (AI) algorithms have been applied to many and different contexts in recent years, from the healthcare world to intrusion detection systems in the

field of IoT security. Often, ML and AI models are trained using data retrieved from the environment to classify the different classes and make decisions based on the context in which they are applied. However, the trained models which support such systems may also be subject to attacks and thus introduce a new attack vector. Attacks that target the ML models are known as adversarial machine learning. The main aim of these attacks is to exploit the weaknesses of the trained model by manipulating and crafting data by starting from the real one. These perturbations increase the confusion in the decision model since ML algorithms are trained with different data. The perturbations performed by the adversarial machine learning attacks aim to be minimal to fool the model without an obvious change in the data used. Furthermore, another possible target of these attacks is to have the data misclassified in order, for example, to execute a cyber-attack on a system and classify it as legitimate. Although the concept of adversarial ML has been introduced in the field of images, in recent years several research works have dealt with introducing this concept in other contexts such as IoT [31], malware [27] or web applications [41].

B. DETECTION

We considered the following attacks: Carlini-Wagner, the Fast Gradient Sign method and the Jacobian based saliency map. In order to detect an adversarial attack against a victim ML algorithm, we decided to train a further ML binary classifier, by combining legitimate and adversarial data. The detection classifier is designed to identify as many attacks as possible, thus minimizing the False Positive Rate (FPR). In this way, more legitimate data may be misclassified as malicious (increase of false negatives), but a good compromise is seeked anyway under the adopted Reliable AI. After creating the combined dataset, we initially evaluate canonical ML algorithms, including decision tree, random forest, k-nearest neighbors (knn), gradient boost, support vector machine (svm) and logistic regression, with hyperparameters optimization to improve the detection performance.

We chose the mentioned algorithms as they are among the most used ones for binary classification problems in recent machine learning literature [42].

Then, Reliable AI is applied and compared with canonical ML. The workflow is shown in figure 2.

As to Fig. 1 again, our approach introduces a defensive technique through robustness enhancement outside the main training model, which is designed for the target application, e.g., visual landing, predictive maintenance, see, e.g., the Annex 2 of the EASA doc [13]. That means the detection is still made by ML, but through another model that works in parallel with the main one and understands if the inputs provided to the main model are corrupted by adversarial distortions.

C. TARGET APPLICATIONS

The proposed activities are tested and evaluated on three different datasets: the first one is focused on network security

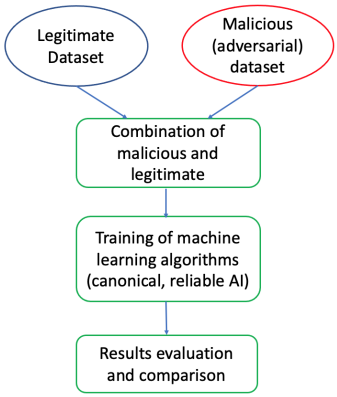


FIGURE 2. Concept approach to detect an adversarial machine learning attack. Datasets are represented by circles (blue for legitimate and red for malicious), while actions are represented by squares (green color).

(in particular, a DNS tunneling communication), the second one focused on vehicle platooning and the third one is a benchmark in predictive maintenance, consisting in Remaining Useful Life (RUL) estimation. The dataset relating to DNS tunneling is simpler as the legitimate and malicious data are divided into more distinct zones than with platooning and RUL, i.e., there are less overlaps of the two classes (legitimate and adversarial). In the platooning dataset, on the other hand, a strong superposition of points between the two classes makes the detection a hard task. Finally, in the RUL estimation original problem, the healthy and fault classes are quite well separated and we will investigate how the different proposed attacks impact on this base performance. In this way, we evaluated the proposed approach on increasing scenarios complexity. More information about the datasets is reported in Section VI-A.

Some more words are necessary for the assumptions made for the attacker and detection systems. This is the subject of the following two subsections.

D. ATTACKER ASSUMPTION

During the adversarial attacks generation, a ML algorithm is required as victim of these attacks. Assuming that an attacker does not know the algorithms of a detection system, in this work we have decided to use a neural network. In particular, we decided to implement a neural network composed of 3 layers with the following numbers of nodes: 512, 256, 128 and a last layer as output. The model is trained with ReLu activation function for the hidden layers, a sigmoid function for the output layer, an Adam optimizer with learning rate set to $1.0e-5$, 300 epochs and batch size set to 16. During the training of the neural network, the accuracy was stably around 95 %. The workflow of the attack creation is reported in Figure 3. The original (legitimate) dataset is split into training and test portions; the victim neural network is then trained on training data and exploited by the adversarial attacks algorithms, that manipulate the test set in such a way to make that network misclassify data. This ends up in

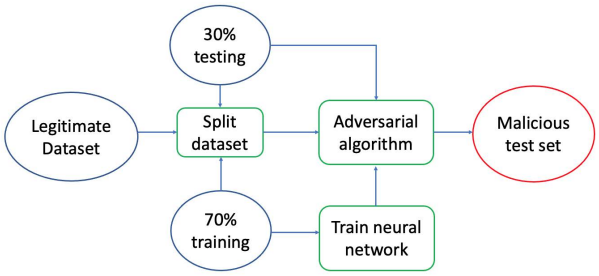


FIGURE 3. Concept approach to execute an adversarial machine learning attack. Datasets are represented by circles (blue for legitimate and red for malicious), while actions are represented by squares (green color).

a malicious test set, which is then combined to legitimate data, as we better detail in the next Section.

E. DETECTION ASSUMPTION

As for ML algorithms used for a specific use case in analysis, there are several considerations to take into account. Often, ML systems are trained using the data present in the system, to carry out classifications or forecasts, where the aim for these systems is to obtain high accuracy and precision metrics without considering adversarial attacks on the ML system. With this approach, an adversarial machine learning attack would be very successful as the algorithms would not be able to identify it. In this paper, we decided to consider possible adversarial machine learning attacks during the algorithm training phase. The classification/prediction system was trained with the different types of adversarial attacks to identify a possible attack on the system. In this way, some legitimate data will be classified as malicious but the aim is to identify a possible attack by sacrificing possible legitimate values.

IV. ADVERSARIAL ATTACKS CONSIDERED

In this section, we report a description of the adversarial machine learning algorithms considered in our work. In particular, we selected the Carlini-Wagner (CW), Fast Gradient Sign Method (FGSM) and Jacobian based Saliency Map attack (JSMA). The tool adopted to generate the adversarial machine learning attacks is the Adversarial Robustness Toolbox (ART) [43]. In the following formulas, we considered x as the legitimate input dataset while $\hat{x}$ as the adversarial dataset produced during the adversarial attacks, usually considered as $\hat{x} = x + \delta$ where $\delta \in [-1, 1]$ (data is supposed normalized in $[0, 1]$ [9]) is the perturbation of the attack.

A. FAST GRADIENT SIGN METHOD

The Fast Gradient Sign Method (FGSM) attack was introduced by Goodfellow in the 2015 [44]. The malicious test set is generated by using the following equation:

$$\hat{x} = x - \varepsilon * sign(\nabla loss_{F,t}(x)) \tag{1}$$

In the above equation (1), x is the original input, t represents the target class and ε is the perturbation parameter, sufficiently small to be undetectable.

In the FGSM attack, a loss function is implemented to elaborate the input data to minimize the loss function. The attack proposed is able to misclassify the output of ML algorithms. This attack is tested and evaluated on different models to demonstrate its efficacy [44], [45], [46]. The main purpose of the FGSM attack is to be faster in the generation of adversarial test set at the expense of an optimal search for the best dataset in terms of perturbations [9]. This is considered the most efficient adversarial attack in terms of computing time and resources.

B. JACOBIAN BASED SALIENCY MAP

Another adversarial attack considered in this work is the Jacobian Based Saliency Map (JSMA); it was introduced in 2016 by Papernot [6]. The concept is based on a simple assumption: understand how inputs affect outputs by modifying samples through the most influential features and tune them to achieve the most subtle, yet detrimental, effect on classification. The saliency map defines the gradients of the output over the input in canonical deep learning structures and it may drive comprehension, via visualization, of the image processing at each layer of the neural chain. The ranking of the values of the saliency map over the feature samples gives feature ranking. The JSMA process iteratively exploits such a feature ranking: the input is perturbed until a misclassification in the target class is achieved. If the desired misclassification is not reached, the JSMA inserts a new feature in the perturbation and tries to misclassify again.

C. CARLINI-WAGNER

The Carlini-Wagner (CW) attack is available in three different versions aimed to obtain low distortion for this metrics [9]: the first aims to minimize the L_2 norm, the second the L_0 norm and finally the third the L_∞ norm. In this work, we adopted the version focused on the L_2 norm.

$$\begin{aligned} &\arg\min_{\delta} \; D(x, \hat{x}) \\ &\text{such that } C(\hat{x}) \neq C^*(x) \\ &\text{where } \hat{x} \in [0, 1]^n \end{aligned} \tag{2}$$

where $\hat{x}$ is considered as $x + \delta$ and C is the classifier considered; $C^*(x)$ is the optimal classification of x and $D(\cdot)$ is a proper distance metric. The CW attack is also robust against defensive distillation [10] or other detection mechanisms [9]. Until now, this attack is considered as the most powerful among the existing attacks. Obviously, since the generation of malicious adversarial data is very accurate, computational times can be very long.

V. RELIABLE AI

The proposed approaches identify means to surround the adversarial class through confidence envelopes with zero statistical error. We show how this guarantee can considerably

limit the size of the adversarial envelope (i.e., by providing detection, yet drastically reducing the population of certified legitimate traffic) and focus on how to strike a good balance between the guarantee and the size of the adversarial space.

The first approach, presented throughout Section V-A, directly originates by white-box predictions made by an explainable AI algorithm, the Logic Learning Machine, which is optimized to infer adversarial regions containing only attack points.

The second is a black box algorithm whose goal is to enclose as many target objects within a hypersphere while keeping out as many negative points as possible. This algorithm is known in the literature as SVDD, [15], [16], and in this paper an improved version is given that attempts to maximize the hypersphere size while keeping the False Positive Rate (FPR) at zero [17], [47]. Then, next, another algorithm is proposed to extract intelligible rules from the model run by SVDD (see Section VI-D).

A. LOGIC LEARNING MACHINE

The Logic Learning Machine (LLM) is a global rule-based supervised method [48], available in Rulex Analytics platform.[1] Given an input vector, the LLM builds intelligible rules in the form of **if** <*premise*> **then** <*consequence*>. The <*premise*> is the logical product ($\wedge$) of conditions on the input features, whereas <*consequence*> indicates the output class.

The model construction relies on three steps, involving the conversion of the input space into a Boolean lattice, the individuation of clusters inside of it through Shadow Clustering [49] and, finally, the translation of clusters into rules.

Given a classification task, the LLM provides M rules $\mathbf{r}_k, k = 1, \ldots, M$, each including d_k conditions $c_{l_k}, l_k = 1_k, \ldots, d_k$.

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

$$C(\mathbf{r}_k) = \frac{TP(\mathbf{r}_k)}{TP(\mathbf{r}_k) + FN(\mathbf{r}_k)} \tag{3}$$

$$E(\mathbf{r}_k) = \frac{FP(\mathbf{r}_k)}{TN(\mathbf{r}_k) + FP(\mathbf{r}_k)}, \tag{4}$$

being $TP(\cdot)$, $FP(\cdot)$, $TN(\cdot)$, $FN(\cdot)$ the confusion matrix values associated to the classification of the data with rule $\mathbf{r}_k$. Both covering and error are the basis for defining feature ranking and value ranking.

1) FEATURE AND VALUE RANKING

Feature and value rankings are useful tools to extract knowledge from the LLM results.

The first one is used to discover which are the most meaningful features for the considered classification task, according to a measure of relevance. Value ranking is useful to individuate the most relevant intervals of values for each of them. The relevance for a single condition is $R(c_{l_k}) = (E(\mathbf{r}'_k) - E(\mathbf{r}_k))C(\mathbf{r}_k)$, where $\mathbf{r}'_k$ is the rule obtained by removing c_{l_k} from $\mathbf{r}_k$. The overall measure of relevance $R_{\hat{y}}(v_j)$, for output class $\hat{y}$ and input feature X_j with values v_j, is then derived by the following equation:

$$R_{\hat{y}}(v_j) = 1 - \prod_k \left(1 - R\left(c_{l_k}\right)\right), \tag{5}$$

where the product is computed on all the rules $\mathbf{r}_k$ that include a condition c_{l_k} on $X_j = v_j$. Feature ranking then consists in sorting the relevance scores obtained for all the features. The same argument can be extended to intervals of values, thus giving rise to *value ranking*.

In this paper, we re-design three methods, introduced in our previous work [18], that exploit feature and value ranking to design adversarial regions with zero False Positive Rate (i.e., the rate of legitimate misclassified as attacks) based on eXplainable AI: `reliability from outside`, `reliability from inside` and `LLM with zero error`.

Before entering the details, we provide some basic notation. Let X be a $D \times N$ matrix of the input samples, with N being the total number of features. Also, let D_1 be the number of instances belonging to the adversarial class, $y = 1$, and D_0 the number of samples in the legitimate class, $y = 0$, so that $D_1 + D_0 = D$.

Our XAI-based detection methodologies are presented in the following sections.

2) RELIABILITY FROM OUTSIDE

As suggested by its name (" from outside"), this method aims at finding the adversarial regions based on the opposite class to the target (which is the adversarial class, y=1, in our case). Hence, our focus is here on the LLM for the legitimate class, denoted with $y = 0$.

The method is based on feature and value ranking properties of the LLM. In particular, we build initial hyper-rectangles by logical disjunction of N^{FR} intervals, selected by inspecting the value ranking of the N^{FR} most relevant features. Then, perturbations to the intervals thresholds are applied until optimal hyper-rectangles are obtained, as described in **Algorithm 1**.

For instance, if we fix $N^{FR} = 2$, $\mathcal{P}(\Delta^*)$ is a rectangle containing all the legitimate points. To get the adversarial region $\mathcal{S}_0$, the complementary surface is then considered:

$$\mathcal{S}_0 = ((-\infty, \tilde{s_1}^*) \vee (\tilde{t_1}^*, \infty)) \wedge ((-\infty, \tilde{s_2}^*) \vee (\tilde{t_2}^*, \infty)) \tag{6}$$

with $\tilde{s_1}^*, \tilde{t_1}^*, \tilde{s_2}^*, \tilde{t_2}^*$ being the optimized thresholds.

3) RELIABILITY FROM INSIDE

This method performs the same search for adversarial regions with a conceptually similar approach to the previous method, except for that it starts with N^{FR} most important features for the adversarial class, which is our target in this case.

[1] https://www.rulex.ai/

Algorithm 1 `Reliability From Outside`

Inputs: dataset $\mathcal{X}$; number of features N^{FR};
Candidate perturbations
$\mathcal{H} = \{\Delta_i | \Delta_i = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N^{FR}}),\ \boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j}),$
$j = 1, \ldots, N^{FR}, i = 1, \ldots, N_c\}$

1. Apply **LLM** on $\mathcal{X}$;
2. **Select** N^{FR} features from **feature ranking** of class $y = 0$;
3. Get $[s_j, t_j]$ from **value ranking** for class $y = 0$;
4. Define intervals **logical union**: $I = \bigcup_{j=1}^{N^{FR}} [s_j, t_j]$;
5. $\forall \Delta_i \in \mathcal{H}$, do:
   Define an **hyper-rectangle**:
   $P(\Delta_i) = \bigcup_{j=1}^{N^{FR}} [s_j - \delta_{s_j} \cdot s_j, t_j + \delta_{t_j} \cdot t_j]$
   $\doteq \bigcup_{j=1}^{N^{FR}} [\tilde{s_j}, \tilde{t_j}]$;
6. Find **optimal perturbations**:
   $\Delta^* = arg\ \min_{\Delta_i : N_0 = D_0} \mathcal{V}(\mathcal{P}(\Delta_i))$
7. **Return adversarial region** as complementary to $\mathcal{P}(\Delta^*)$

Algorithm 2 `Reliability From Inside`

Inputs: dataset $\mathcal{X}$; number of features N^{FR};
Candidate perturbations
$\mathcal{H} = \{\Delta_i | \Delta_i = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N^{FR}}),\ \boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j}),$
$j = 1, \ldots, N^{FR}, i = 1, \ldots, N_c\}$

1. Apply **LLM** on $\mathcal{X}$;
2. **Select** N^{FR} features from **feature ranking** of class $y = 1$;
3. Get $[s_j, t_j]$ from **value ranking** for class $y = 1$;
4. Define intervals **logical union**: $I = \bigcup_{j=1}^{N^{FR}} [s_j, t_j]$;
5. $\forall \Delta_i \in \mathcal{H}$, do:
   Define **hyper-rectangle**:
   $P(\Delta_i) = \bigcup_{j=1}^{N^{FR}} [s_j + \delta_{s_j} \cdot s_j, t_j - \delta_{t_j} \cdot t_j]$
   $\doteq \bigcup_{j=1}^{N^{FR}} [\tilde{s_j}, \tilde{t_j}]$;
6. Find **optimal perturbations**:
   $\Delta^* = arg \max_{\Delta_i : N_0 = 0} \mathcal{V}(\mathcal{P}(\Delta_i))$
7. **Return adversarial region** $\mathcal{P}(\Delta^*)$

Perturbations of the initial intervals are now made in such a way to leave out all the legitimate points from the hyper-rectangle, as detailed in **Algorithm 2**.

For $N^{FR} = 2$, the adversarial region is a 2D surface $\mathcal{S}_1$ containing, at the optimal solution, no points from legitimate class:

$$\mathcal{S}_1 = (\tilde{s_1}^*, \tilde{t_1}^*) \vee (\tilde{s_2}^*, \tilde{t_2}^*), \tag{7}$$

with $\tilde{s_1}^*, \tilde{t_1}^*, \tilde{s_2}^*, \tilde{t_2}^*$ being the optimized thresholds.

4) LLM WITH ZERO ERROR

Since hyper-rectangles may be too simple to follow potentially complex shapes of the output classes, a more refined solution consists in looking for more complex adversarial regions, still preserving the zero statistical error constraint. This is performed by training the LLM with 0% error, so that the rules do not cover points of the incorrect class (`LLM 0%`, in the following).

Algorithm 3 `LLM0%`

Inputs: dataset $\mathcal{X}$; number of features N^{FR};
Candidate perturbations
$\mathcal{H} = \{\Delta_i | \Delta_i = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N^{FR}}),\ \boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j}),$
$j = 1, \ldots, N^{FR}, i = 1, \ldots, N_c\}$

1. Apply **LLM** on $\mathcal{X}$ s.t. $E(\mathbf{r}_k) = 0\ \forall k \in [1, m]$;
2. **Select** $m^0 < m$ **highest-covering** rules r_z for class $y = 1$
3. **Select** N^{FR} features from **feature ranking** for class $y = 1$
4. **Define** $\hat{r} \doteq \bigcup_{z=1}^{m^0} r_z$
5. $\forall \Delta_i \in \mathcal{H}$ do:
   for $f_j \in [s_j, t_j],\ j = 1, \ldots, N^{FR}$:
   if $f_j \in \hat{r}$:
   $s_j \leftarrow s_j + \delta_{s_j} \cdot s_j$
   $t_j \leftarrow t_j - \delta_{t_j} \cdot t_j$
   end
   Build $\hat{r}(\Delta_i)$
   end
6. Solve $\Delta^* = \arg\max_{\Delta_i : E(\hat{r}(\Delta_i)) = 0} C(\hat{r}(\Delta_i))$
7. Get the **adversarial region** $\hat{r}(\Delta^*)$

The search for adversarial regions is performed as explained in **Algorithm 3**.

Differently from the two previous methods, in this algorithm we are interested in joining a number of entire *rules*, instead of single intervals, thus giving rise to a new predictor $\hat{r}$ with more complex geometry. Our goal is always to have zero false positive rate (FPR=0): this is pursued here by perturbing a subset of conditions contained in the rules making up $\hat{r}$, chosen for a number N^{FR} of top-performing features selected via feature ranking (see Step 5 in **Algorithm 3**).

B. SAFE SVDD

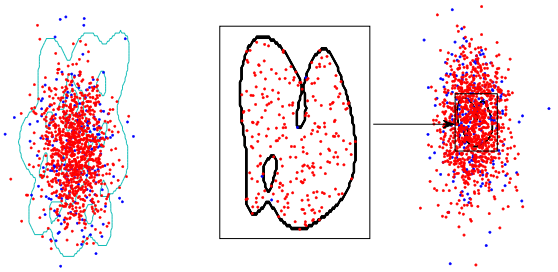
Two native SVDD algorithms for reliable classification are presented in this section. The first controls the false positive rate by iterating successive SVDDs within the target region until the desired classification error threshold is reached. The second extracts intelligible rules from the SVDD classification boundary.

1) `ZeroFPRSVDD`

As in [17], [47], the `zeroFPRSVDD` algorithm performs successive iterations of the SVDD on the target initial region, found with a preliminary SVDD, until there are no more negative points inside it. The convergence is achieved when a fixed number of iterations is reached or when the condition on FPR is satisfied.

We performed this algorithm in Matlab[2] and we tested it using data from [50]. In Fig.(4) it is reported an example with

[2]https://www.mathworks.com/matlabcentral/fileexchange/106375-zerofpr_svdd



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

FIGURE 4. Application of Algorithm 4 on a dataset 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.

a 2 dimensional Gaussian dataset. Behind the pseudo-code of the algorithm is reported, regarding which we denoted with Ξ the region of the input parameters within the SVDD boundary and with σ the kernel parameter (if any).

Algorithm 4 `ZeroFPRSVDD`

dataset $\mathcal{X}\times\mathcal{Y}$ 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. SVDD-**cross-validation** on $\mathcal{X}_{tr}\times\mathcal{Y}_{tr}$
2. $[\mathbf{a},R^2]$=SVDD$(\mathcal{X}_{tr}, \mathcal{Y}_{tr}, C_{-1}, C_{+1}, \sigma)$
3. **Test** SVDD on $\mathcal{X}_{ts}\times\mathcal{Y}_{ts}$
4. maxiter=1000;
5. i=1;
6. **while** (i<maxiter)
6.1. $\left[\mathcal{X}_{tr_i}, \mathcal{Y}_{tr_i}\right] = \Xi(\mathcal{X}_{ts}, \mathcal{Y}_{ts})$;
6.2. SVDD-**cross-validation** on $\mathcal{X}_{tr_i}\times\mathcal{Y}_{tr_i}$
6.3. $[\mathbf{a}_i, R_i^2]$=SVDD$(\mathcal{X}_{tr_i}, \mathcal{Y}_{tr_i}, C_{-1}, C_{+1}, \sigma)$
6.4. **Test** SVDD on $\mathcal{X}_{ts}\times\mathcal{Y}_{ts}$
6.5. **if**(FPR$< \varepsilon$)
6.5.1. **return** $[\mathbf{a}^*, R^{*2}] = [\mathbf{a}_i, R_i^2]$;
6.5. **end**
7. $i = i + 1$;
end

2) eXplainable SVDD

The combination of `ZeroFPRSVDD` and XAI yields a a new explainable classifier which extracts intelligible rules from the black box structure of the SVDD [17], [47]. The derivation of intelligible rules is made as follows. After that a `zeroFPRSVDD` has been optimized, a new dataset of observations *sampled around the edge of the* `zeroFPRSVDD` is provided and the classification via `zeroFPRSVDD` is registered. The new dataset is then elaborated via a XAI algorithm, the LLM (see Section V-A) and the rules extracted. In this work, differently from [17], the sampling of the new dataset to be processed by the LLM is more refined: the sampling is performed by setting a threshold ε such that the extracted observations are sufficiently close to the boundary of the trained and tested `zeroFPRSVDD`. The threshold is set a priori and it depends on the dataset: given a set $X = \{x_i\}_i$ of synthetic data, sampled uniformly from the test set, in order to extract points close to the radius we take into account the quantity $t := \mid ||x_i - \mathbf{a}||^2 - R^2|| \mid$ and we choose $\varepsilon \in (\min(t), \max(t))$. It is quite clear that values too close to $\min(t)$ do not allow enough samples to be extracted while values too close to $\max(t)$ extract too many points away from the edge of the `zeroFPRSVDD`. A good balance for the choose of ε then can be the average $(\min(t) + \max(t))/2$ or values close to it.

Algorithm 5 `eXplainableSVDD`

Get $\mathbf{a}^*$, R^* from `ZeroFPRSVDD` algorithm. Fix ε.

1. **Sample** uniformly a new dataset
$\mathcal{X}_{new}$ s.t. $x_i \in \mathcal{X}_{new} \iff$
$\mid ||x_i - \mathbf{a}||^2 - R^2 \mid < \varepsilon$
2. **Classify** $\mathcal{X}_{new}$ in $\mathcal{Y}_{new}$ through optimal `ZeroFPRSVDD` (w.r.t. $[\mathbf{a}^*, R^{*2}]$)
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** $\mathcal{R}$

VI. TESTS AND OBTAINED RESULTS

In this section, we initially present the datasets considered in the proposed work. Then, we discuss a first approach to detect adversarial machine learning attacks by using classical algorithms with hyperparameters optimization to improve performance metrics. Then, we move to reliable approaches for the detection of adversarial attacks based on XAI optimizations (Section VI-C)and, finally, we test the reliable SVDD approach, also combined with rules extraction. As metrics to measure the detection of adversarial attacks, we adopt the confusion matrices in order to evaluate the correct classification of legitimate and malicious data.

A. DATASETS

The used datasets represent two challenging scenarios for detection even without the adversarial component.[3] The first one deals with covert channel detection in cybersecurity [4]; more specifically, the aim is detecting the presence of Domain Name Server intruders by an aggregation-based monitoring that avoids packet inspection, in the presence of silent intruders and quick statistical fingerprints generation. By modulating the quantity of anomalous packets in the server, we are able to modulate the difficulty of the inherent supervised learning solution via canonical classification schemes (Bayes decision theory, neural networks). More specifically, let q and a be the packet sizes of a query and the corresponding answer, respectively (what answer is related to a specific query can

[3]The adopted datasets, both the original legitimate and the attacked ones, are available as open-source in the following repository: https://www.kaggle.com/datasets/cnrieiit/adversarial-machine-learning-dataset.

be understood from the packet identifier) and Dt the time-interval intercurring between them. The information vector is composed of the statistics (mean, variance, skewness and kurtosis) of q, a and Dt for a total number of 12 input features:

$$\mathbf{I} = [m_A, m_Q, m_{Dt}, v_A, v_Q, v_{Dt}, s_A, s_Q, s_{Dt}, k_A, k_Q, k_{Dt}]$$

The corresponding vectors are: $\mathbf{m}$, $\boldsymbol{\sigma}$, $\mathbf{s}$, $\mathbf{k}$. High-order statistics give a quantitative indication of the asymmetry (skewness) and heaviness of tails (kurtosis) of a probability distribution, they help improve detection inference.

The second dataset addresses collision prediction in vehicle platooning [35], which is widely considered one of the most challenging problems in smart mobility scenarios. It consists of a group of vehicles interconnected via wireless that travel autonomously, based on the widespread Cooperative Adaptive Cruise Control (CACC) technology [51]; the aim is to find a compromise between performance (e.g., maximize speed and minimize reciprocal distance, thus minimizing air drag resistance and fuel consumption, too) and safety (no collision, even in the presence of anomalous events, such as sudden brakes [35]).

The behavior of the platooning system is synthesised by the following vector of features:

$$\mathbf{I} = [N, F_0, PER, d_0, v_0]$$

where N is the total number of vehicles of the platoon, F_0 is the braking force applied by the leader, PER is the probability of packet loss, and d_0 and v_0 are the mutual distance and speed between each pair of vehicles in the initial condition. The ML solution is based on a supervised classification task that maps current speed, distance, acceleration, weight of vehicles, as well as quality of service of the communication channel, into a potential collision into the near future. As shown later, the adversarial component makes a detrimental impact on the chances to find such a mapping.

Another realistic application scenario is represented by the Turbofan Engine Degradation Simulation Data Set, made available by NASA [52]. It is an important benchmark in predictive maintenance, since it deals with damage propagation modeling for aircraft engines. The repository contains four different sets of data, called FD001, FD002, FD003 and FD004 [53], corresponding to simulations under different combinations of operational conditions and fault modes. Our analysis here is based on FD001 only. Different functional parameters of aircraft gas turbine engines are collected by sensors over time and describe the trajectory of the system (more information on all the available measurements can be found in the original publication [54]). Features are then extracted, by computing the mean m, variance v, skewness s and kurtosis k for each parameter raw time-series, over a moving time window (observation horizon) of fixed size, obtaining samples making up what we call RUL dataset. The goal is to recognize those trajectories that may result in fault states, based on the extracted features. This implies the definition of the Remaining Useful Life (RUL) variable, which represents how much time is left before a fault occurs. In practice, one would want to understand which conditions are inherent to imminent faults of the engine. The problem can be solved via a ML classification task, where the RUL constitutes the output class. In the original dataset, the RUL class assumes three values: healthy (RUL$>$150), critical (50$\leq$RUL$\leq$150), and faulty (RUL$<$50). For our application, we further elaborated the data by reducing its dimensions to the following features:

$$\mathbf{I} = [s_{os_2}, m_{N_c}, v_{N_c}, v_{phi}, m_{htBleed}, s_{htBleed}, m_{W31}]$$

The choice on these variables was done by evaluating the LLM feature ranking on the FD001 dataset V-A. Moreover, due the high under-sampling of faulty class and for consistency with the other two applications, we decided to merge the critical and faulty samples into a single faulty class. Hence, the problem becomes a binary classification between healthy (RUL$>$150) and faulty (RUL$\leq$150). After the attacks generation, following the approaches summarized in Figures 2 and 3, each described dataset is be merged (as legitimate points) to the malicious (attacked) test set.

B. CANONICAL SUPERVISED LEARNING AND HYPERPARAMETER OPTIMIZATION

In order to provide a first possible protection from the adversarial machine learning attacks, we focused on the adoption of classic ML algorithms. This approach is adopted to validate if classic ML algorithms are able to correctly classify possible adversarial attacks. For this reason, we implemented different algorithms such as decision tree, gradient boost, K-nearest neighbors, logistic regression, random forest and a support vector machine. The dataset, composed by legitimate and malicious rows, is splitted in 70% of training and 30% of testing. The algorithms were implemented through the Sklearn [55] library, an open source ML library for the Python programming language. The tests were performed with the same dataset and on the same machine to avoid differences in obtained results to guarantee consistency on the tests and results. Moreover, the number of rows of the legitimate and adversarial datasets are of the same order of measurement in order to have a balanced dataset since an unbalanced dataset could reports high values of metrics.

As presented in Section VI-A, we tested the ML algorithms on three different datasets: DNS tunneling, platooning and RUL estimation. The results obtained are reported by using metrics extracted from confusion matrices, in particular we decided to report false positive rate (FPR), true positive rate (TPR), false negative rate (FNR) and true negative rate (TNR). All results are shown in Table 1, divided by algorithm, attack and dataset.

For the results on the DNS dataset, it is possible to note that, with default configurations of the algorithms, the detection of an adversarial machine learning attack is not achieved since most of the algorithms are not able to correct classify the malicious payload. Also analyzing the results for the platooning dataset, we can note that the confusion matrices report low values of correct classification. In particular,

algorithms are not able to classify the attack as demonstrated by the high number of false positive rate and false negative rate. By focusing only on the correct classification of the attack, most of the algorithms classify the attack as legitimate. In particular, for the CW attack, only the SVM algorithm manages to classify it well enough, at the expense of many incorrect classifications of legitimate data. These results are also validated by the FPR table reported in Table 1 where all the algorithms have high value of FPR except for the SVM which has 0.03 but related to a wrong classification of the legitimate data, so this value is not considerable as good result. While considering the JSMA attack, all the algorithms are not able to classify with good performance the attack. Finally, regarding the FGSM, the random forest and decision tree obtained a good FPR value but the other algorithms are not able to perform a correct classification. These results actually demonstrate that the adversarial attacks are complex to identify since with minimum perturbations of the dataset, the behaviour of a ML algorithm is totally confused.

The performance of canonical ML methods on RUL dataset differs from DNS and platooning, since they perform bad on CW attack, managing to lower the FPR but at the cost of very high FNR values. In contrast, on JSMA and FGSM they generally perform well, with the surprising exception of SVM on FGSM attack, whose performance is the same as on CW (FPR=0, TPR=0).

As possible to note from the above discussions, with default configurations of the algorithms, the detection of an adversarial machine learning attack is not achieved since most of the algorithms are not able to correctly classify the malicious payload. However, to carry out a correct classification, a detailed and specific configurations of the models must be tested and validated. For this reason, we decided to perform a hyperparameter optimization of the ML algorithms in order to validate if the adversarial attacks can be correctly identified by tuning the models adopted to detect them. The hyperparameter optimization challenge is to select a set of optimal parameters for a ML algorithm to improve the evaluation metrics and the precision of a model.

In order to achieve these results, we adopted the hyperparameters optimization tool called Optuna [56]. Optuna formulates the hyperparameter optimization as a process of minimizing/maximizing an objective function that takes a set of parameters as input and returns its score. Once the parameters to be optimized have been defined, Optuna starts combining the different parameters and evaluating the algorithm to validate if a combination of the parameters leads to an algorithm improvement in terms of metrics. At the end of the parameter testing process, the optimal ML model returns the parameters that calculate the higher metrics. In our tests, to obtain efficient results and to test a good number of parameter combinations, 1000 parameter combinations with different values (chosen by Optuna according to a tool logic) were performed for each algorithm.

Once the optimal parameters for the algorithms were obtained, confusion matrices of each algorithms were calculated to validate if, with the hyperparameter optimization, the ML model is able to detect the adversarial attacks. Obtained results are reported in Table 2.

Regarding the DNS dataset, by analyzing the obtained results, the hyperparameter optimization improved the metrics. In particular, in the FGSM and the JSMA, most of the algorithms are able to correct classify the adversarial attacks while in the CW the metrics are still low since the CW attack is more complex than the others.

After the implementation of the hyperparameter optimization, an improvement of the metrics on the platooning dataset is obtained. For the CW attack, the metrics value for each algorithm is again high, so a classification of the attack is not suitable for a real environment. In particular, the SVM is near the 63% of FPR due to a correct classification of the legitimate data. For the JSMA attack instead, the FPR metric is near to the 25%-30% which is again not good for classifying the attack. Finally, the FGSM attack obtained good results for the decision tree, gradient boost and the random forest with a FPR rate lower than 17% due to the fact that this attack is more simple to detect (since it requires minor time to be executed so it is not accurate).

The hyper-parameters optimization on RUL dataset was effective for the cases where the detection was already satisfactory, i.e. on the JSMA and FGSM; CW attack remains hardly detectable with any canonical method instead.

C. DETECTION THROUGH XAI-DRIVEN RELIABLE AI

As described in Sections V-A2, V-A3 and V-A4, another method to detect the adversarial attacks consists in the search of adversarial regions denoted by zero FPR starting from an explainable AI model (LLM).

We applied the methods on the vehicle platooning, DNS tunneling and RUL estimation datasets, as described in Section VI-A.

For all test cases, the first step was the training of the default Logic Learning Machine (with 5% maximum error allowed for each rule) on a 70% training set with a 30% test set (the same sets are used for all the detection algorithms.)

Before entering the details of the detection, as an example, we provide a visual comparison of the obtained LLM rules for the three attacks in DNS tunneling case. Figure 5 shows the rules via a special kind of visualization, called the *rule viewer*. Each circle represents a rule: the larger the former is, the more the respective rule covers a larger number of points. The size of the central hole represents the error of that rule: the larger the hole, the greater the error. In the plots, green color refers to legitimate class (no attack) and red to attack class. In the outer circle, the input features are shown. A large number of rules is an indication of the complexity of the system, that means a larger number of rectangles (rules, in two dimensions) is needed to best approximate the complicated shape of the boundary of the classes. Just comparing the rule viewers under CW, FGSM or JSMA attacks gives immediately the perception of the complexity of the adversarial problem and its variability under different attacks.

TABLE 1. Canonical machine learning. The table shows the performance statistics of canonical machine learning algorithms, divided by attack and dataset.

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Decision Tree	CW	0.13	0.29	0.87	0.71	0.43	0.55	0.57	0.45	0.23	0.03	0.77	0.97
	JSMA	0.48	0.80	0.52	0.20	0.45	0.87	0.55	0.13	0.00	1.00	1.00	0.00
	FGSM	0.04	0.27	0.96	0.73	0.18	0.86	0.82	0.14	0.00	1.00	1.00	0.00
Gradient boost	CW	0.49	0.99	0.51	0.01	0.58	0.69	0.41	0.31	0.001	0.00	0.999	1.00
	JSMA	0.50	0.99	0.50	0.01	0.35	0.88	0.65	0.12	0.00	1.00	1.00	0.00
	FGSM	0.02	0.16	0.98	0.84	0.23	0.91	0.77	0.09	0.0002	0.998	0.9998	0.002
KNN	CW	0.85	0.80	0.15	0.20	0.46	0.49	0.53	0.51	0.05	0.02	0.95	0.98
	JSMA	0.96	1.00	0.04	0.00	0.62	0.82	0.38	0.18	0.00	1.00	1.00	0.00
	FGSM	0.62	0.82	0.38	0.18	0.49	0.52	0.51	0.48	0.008	0.4105	0.99	0.59
Logistic regression	CW	0.50	1.00	0.50	0.00	0.59	0.64	0.4	0.36	0.00	0.00	1.00	1.00
	JSMA	0.36	1.00	0.64	0.00	0.51	0.91	0.49	0.09	0.00	1.00	1.00	0.00
	FGSM	0.03	0.93	0.97	0.07	0.48	0.63	0.52	0.37	0.01	0.53	0.99	0.47
Random forest	CW	0.32	0.70	0.68	0.30	0.48	0.62	0.51	0.38	0.17	0.002	0.83	0.998
	JSMA	0.50	0.99	0.50	0.01	0.41	0.87	0.58	0.13	0.00	1.00	1.00	0.00
	FGSM	0.03	0.13	0.97	0.87	0.18	0.91	0.82	0.09	0.00	1.00	1.00	0.00
SVM	CW	0.98	0.98	0.02	0.02	0.03	0.11	0.97	0.89	0.00	0.00	1.00	1.00
	JSMA	0.10	0.59	0.90	0.41	0.81	0.92	0.19	0.08	0.00	1.00	1.00	0.00
	FGSM	0.25	0.92	0.75	0.08	1	1	0	0	1.00	1.00	0.00	0.00

TABLE 2. Canonical machine learning optimized. The table is the optimized version of algorithms presented in Table 1 in which the performance statistics of the optimized canonical machine learning algorithms divided by attack and dataset are shown.

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Decision Tree	CW	0.50	1.00	0.50	0.00	0.43	0.55	0.57	0.45	0.00	0.00	1.00	1.00
	JSMA	0.25	1.00	0.75	0.00	0.23	0.84	0.77	0.16	0.00	1.00	1.00	0.00
	FGSM	0.50	1.00	0.50	0.00	0.13	0.85	0.87	0.15	0.0006	1.00	0.9994	0.00
Gradient boost	CW	0.48	1.00	0.52	0.00	0.44	0.59	0.56	0.41	0.00	0.00	1.00	1.00
	JSMA	0.50	1.00	0.50	0.00	0.33	0.88	0.67	0.12	0.00	1.00	1.00	0.00
	FGSM	0.03	0.36	0.97	0.64	0.12	0.92	0.88	0.08	0.00	1.00	1.00	0.00
KNN	CW	0.97	1.00	0.03	0.00	0.62	0.69	0.37	0.31	0.00	0.00	1.00	1.00
	JSMA	0.89	1.00	0.11	0.00	0.53	0.84	0.46	0.16	0.00	1.00	1.00	0.00
	FGSM	0.11	0.28	0.89	0.72	0.47	0.57	0.53	0.43	0.00	1.00	1.00	0.00
Logistic regression	CW	0.49	0.99	0.51	0.01	0.51	0.6	0.49	0.4	0.00	0.00	1.00	1.00
	JSMA	0.09	0.98	0.91	0.02	0.34	0.79	0.66	0.21	0.00	1.00	1.00	0.00
	FGSM	0.03	0.99	0.97	0.01	0.56	0.78	0.44	0.22	0.00	0.81	1.00	0.19
Random forest	CW	0.49	1.00	0.51	0.00	0.46	0.66	0.54	0.34	0.00	0.00	1.00	1.00
	JSMA	0.50	1.00	0.50	0.00	0.33	0.87	0.67	0.13	0.00	1.00	1.00	0.00
	FGSM	0.03	0.32	0.97	0.68	0.17	0.92	0.83	0.08	0.00	0.9992	1.00	0.0008
SVM	CW	0.39	0.65	0.61	0.35	0.63	0.85	0.37	0.15	0.00	0.00	1.00	1.00
	JSMA	0.09	0.98	0.91	0.02	0.24	0.69	0.76	0.31	0.00	1.00	1.00	0.00
	FGSM	0.15	0.95	0.85	0.05	0.69	0.89	0.31	0.11	0.00	0.00	1.00	1.00

Keeping in mind the non trivial nature of the adversarial problem, we now present the obtained results through LLM-based reliable AI.

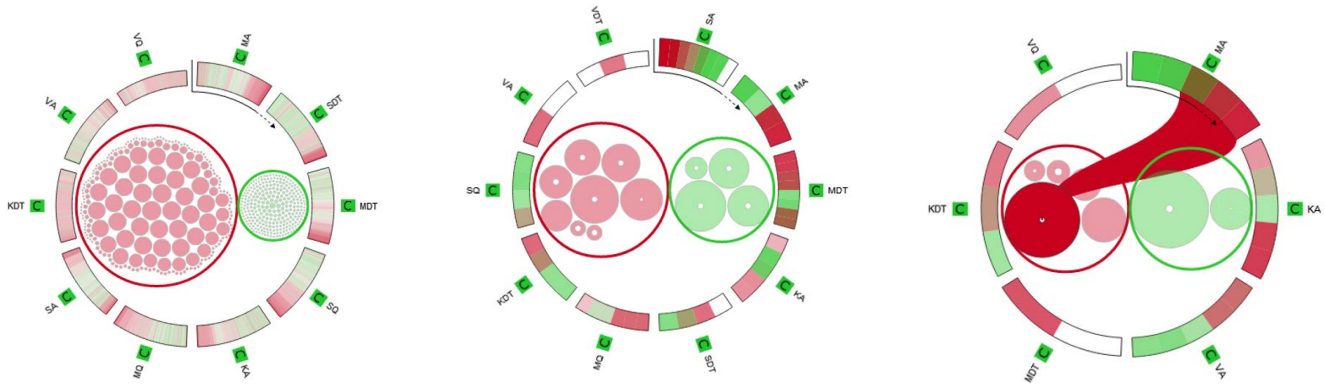
Table 3 reports all the obtained performance metrics for the considered applications.

Concerning `reliability from outside` and `inside` methods (V-A2,V-A3), we first investigated the value ranking obtained from the LLM for the first $N^{FR} = 2$ most relevant features for classes $y = 0$ (legitimate) and $y = 1$ (attack) respectively. The resulting intervals were then joined in `OR` ($\vee$), as reported in the "original intervals" columns in Table 4.

Starting from such joined intervals, we performed the perturbation approaches (Step 5 in Algorithms 1 and 2) and obtained new optimized intervals (Steps 6-7 in Algorithms 1 and 2), i.e. the adversarial regions, which we report, for each application case, in the corresponding columns of Table 4.

By looking at the results, we can observe that the overall performance of both `inside` and `outside` methods is better for DNS tunneling than for vehicle platooning and RUL dataset. In fact, TPR is higher than 0.60 in DNS tunneling for JSMA with both methods and FGSM with `inside`: this means that more than 60% of attacks is detected in these cases. In particular, for DNS tunneling, it is worth underlying the surprising result for JSMA, that can be detected very well by using the `inside` method (a plot of this region is provided in Figure 6a).

As it is possible to observe in Figure 6a, the good performance that can be achieved by the `inside` method on JSMA may also be partially influenced by the outlier attack data injected for high values of both m_A and k_A.



(a) Rule viewer of CW attack (b) Rule viewer of FGSM attack (c) Rule viewer of JSMA attack

FIGURE 5. LLM rules comparison via rule viewers for DNS tunneling problem.

TABLE 3. Inside, Outside, LLM0%. Obtained performance metrics for the detection of adversarial attacks based on eXplainable AI.

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Inside	CW	0.03	0.45	0.97	0.55	0.03	0.02	0.97	0.98	0.02	0.01	0.98	0.99
	JSMA	0.03	0.93	0.97	0.07	0.07	0.56	0.93	0.44	0.02	1.00	0.98	0.00
	FGSM	0.04	0.62	0.96	0.38	0.26	0.29	0.74	0.71	0.03	0.81	0.97	0.19
Outside	CW	0.00	0.01	1.00	0.99	0.01	0.00	0.99	1.00	0.00	0.00	1.00	1.00
	JSMA	0	0.72	1.00	0.28	0.01	0.26	0.99	0.74	0.00	0.06	1.00	0.94
	FGSM	0.00	0.25	1.00	0.75	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00
LLM0%	CW	0.04	0.44	0.96	0.56	-	-	-	-	-	-	-	-
	JSMA	0.47	0.50	0.53	0.50	-	-	-	-	0.00	0.81	1.00	0.19
	FGSM	0.39	0.42	0.61	0.58	-	-	-	-	0.00	0.77	1.00	0.23

These features represent, respectively, the mean and the kurtosis of the size of the answer packets generated by the DNS server to reply DNS address resolution requests received from clients [4]. By zooming on the lower values as depicted in Figure 6b, the portion of attack data contained inside the adversarial region is still acceptable. Nevertheless, the superposition of some attack points on legitimate, observable outside the adversarial region, gives an idea about the difficulty of the problem.

For this JSMA case, which is particularly suitable for analysis with native XAI, we report other interesting knowledge emerged from visual analytics tools, available in Rulex platform: feature and value ranking bar charts (Fig. 7a and 7b).

Figure 7a shows a predominant role of m_A in detecting the JSMA attacks through LLM rules. However, k_A, v_A (this latter being the variance of the DNS server answer packets size) and the other attributes have important ranking scores too. This results in a greater complexity of the problem, as all the variables participate in inferring the attacks through complex LLM rulesets. Also, value ranking (in Fig. 7b for m_A) is able to individuate specific ranges in which the presence of the attack is more likely.

The inside and outside methods find the regions under zero FPR by starting from the value rankings of the most meaningful variables, thus minimizing the complexity of the problem, which otherwise would require a search space over all the variables involved.

In vehicle platooning dataset, `outside` and `inside` methods are not suitable for any accurate attack detection: as we can see from the corresponding entries in Table 3, there is no case with TPR higher than 0.60.

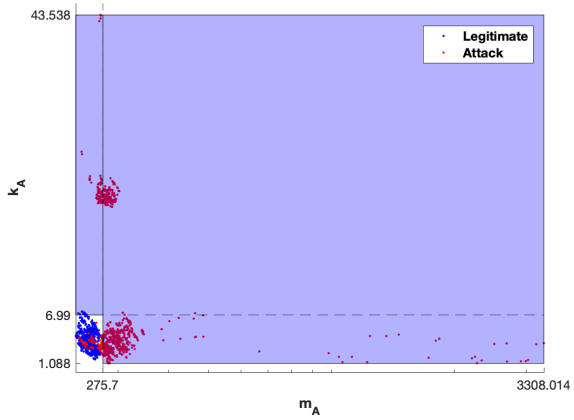
This behavior may be related to the sharp boundaries defined by the union (with `inside` method) or intersection (for `outside`) of the two intervals considered until now. A way to look for more refined regions is offered by our third method: `LLM0%` (Section V-A4). First of all, for both vehicle platooning and DNS tunneling, we applied the LLM model trained by lowering to 0% the maximum error allowed in the rules. Focusing on the adversarial class ($y = 1$), this procedure resulted in the following results:

- Vehicle platooning: 483 rules for CW attack with maximum covering of 1.5%; 542 rules for FGSM, with 6.7% as maximum covering; 308 rules for JSMA with up to 37% of covering.
- DNS tunneling: 364 rules for CW attack with covering up to 38%; 40 rules with covering up to 47% for FGSM; 7 rules with covering up to 79% for JSMA.
- RUL: 2373 rules for CW attack with covering up to 0.14%; 6 rules with covering up to 72% for FGSM; 2 rules with covering up to 78% for JSMA.

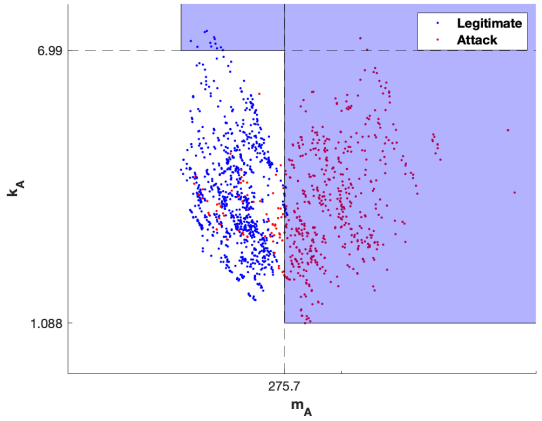
As the maximum covering was too low in CW and FGSM rules for the vehicle platooning case, we were not able to test the `LLM0%` approach on this dataset, since it would have

TABLE 4. Inside, Outside. Obtained adversarial regions.

		DNS		Platooning		RUL	
		Original Intervals	Adversarial Region	Original Intervals	Adversarial Regions	Original Intervals	Adversarial Regions
Inside	CW	m_A>270.23∨ s_{Dt}>70.65	m_A>275.7∨ s_{Dt}>70.65	PER≤0.08 ∨ v_0>82	PER<0 ∨ v_0>89	$s_htBleed$<0.39 ∨ s_os2<0.41	$s_htBleed$<−0.817∨ s_os2<−0.558
	JSMA	d_0>8.97 ∨ F_0>-1	m_A>275.7∨ k_A>6.99	d_0>8.97 ∨ F_0>-1	d_0>8.99 ∨ F_0>-1	v_phi>10.22 ∨ $s_htBleed$>19.11	v_phi>0.262 ∨ $s_htBleed$>0.875
	FGSM	s_A≤1.71 ∨ m_A>269.56	s_A≤1.63 ∨ m_A>270.9	N≥6 ∨ F_0<-8	N≥10 ∨ F_0<-8	v_phi>0.34 ∨ m_Nc<9048.38	v_phi>0.22 ∨ m_Nc<9038.35
Outside	CW	m_{Dt}>0.31∨ v_A>24503	m_{Dt}<0.34∧ v_A<25923	v_0≥82 ∨d_0>8.67	v_0<43∧d_0≤4.013	$m_htBleed$<391.76∨ $s_htBleed$<0.64	$m_htBleed$>395.52∧ $s_htBleed$< − 0.504
	JSMA	m_A≤270.34 ∨ v_A>33393	m_A>276.58∧ v_A<39286	F_0<-7 ∨ d_0≤6	F_0≥-4 ∧ d_0≥8.99	v_phi<10.22 ∨ $s_htBleed$<19.11	v_phi>0.144 ∧ $s_htBleed$>1.172
	FGSM	s_A>1.82 ∨ m_A<267.92	s_A≤1.68 ∧ m_A>275.02	N≥9 ∨ PER<0.16	N≤3 ∧ PER>1	v_phi<0.34 ∨ $m_htBleed$<395.76	v_phi>0.0796 ∧ $m_htBleed$>395.62



(a) 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).



(b) Zoom on the part of adversarial region, obtained for JSMA attack in DNS tunneling dataset with `inside` method, corresponding to low values of m_A and k_A

FIGURE 6. Adversarial region with `inside` for JSMA in DNS tunneling.

required higher covering rules as a starting point. Hence, we only applied the `LLM0%` optimization technique for DNS tunneling and RUL datasets, where the covering percentages were satisfactory in all kinds of adversarial attack, except for the case of CW in RUL dataset.

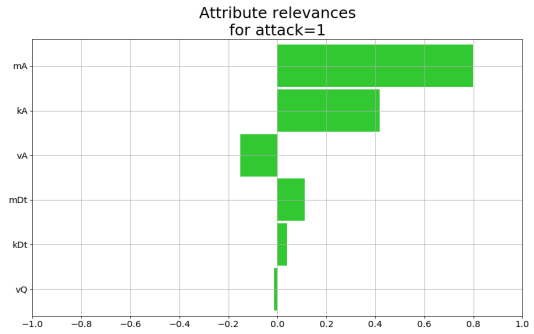
For each attack case, we decided to select the first 5 highest covering rules for the adversarial class and merge them in logical `OR`. Then, we perturbed the thresholds of the conditions involving the first two most important features, according to LLM feature ranking, as expressed by Step 5 in **Algorithm 3**. The obtained results are reported in the bottom part of Table 3.

In our adversarial application, such method does not perform as expected from other, non-adversarial, applications [18], where the error on the resulting regions was usually ≤0.05 but with significant increase in the number of points contained into their boundaries with respect to the `inside` and `outside` methods. However, it is interesting to underline that `LLM0%` is the only method that works better on CW attack than on JSMA or FGSM. For CW attack, we report more details about `LLM0%` in the following.

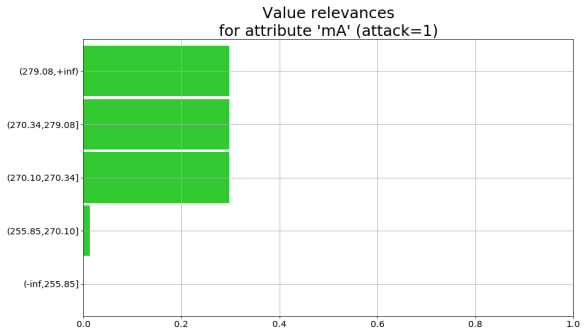
The joining in `OR` (∨) of the 5 highest-covering LLM rules with 0% error, before any perturbation, results in the following predictor:

if$((\boldsymbol{m_A > 277.94})$
$\vee(m_A > 274.55 \wedge 26257 < v_A \leq 39245)$
$\vee(m_A > 271.67 \wedge \boldsymbol{s_{Dt} > 7.61})$
$\vee(m_A > 269.84 \wedge 8.98 < v_{Dt} \leq 11179 \wedge k_{Dt} > 55.19)$
$\vee(m_{Dt} > 0.95 \wedge m_A > 265.03 \wedge 223.15$
$< k_Q \leq 543101255))$**then attack**

The feature ranking for the attack class indicated features m_A and s_{Dt} being the most relevant. As already mentioned, the first attribute is the average size of the answer packets from the DNS server; s_{Dt} is the skewness of the time interval between queries and answers [4]. Starting by the predictor shown above, we perturbed the most stringent threshold values corresponding to such features (namely, 277.94 for m_A and 7.61 for s_{Dt}, as previously highlighted in bold). This led us to obtain a new optimized predictor characterized by



(a) LLM feature ranking for JSMA attack in DNS tunneling, showing the most important variables in inferring the attack presence.



(b) LLM value ranking for the most powerful variable in JSMA attack for DNS tunneling, m_A; the chart shows the intervals in which the largest part of the attacks take place.

FIGURE 7. Feature and value ranking for JSMA attack in DNS tunneling.

new thresholds:

$$\begin{aligned}&\textbf{if}((\boldsymbol{m_A} > \mathbf{291.83})\\&\quad \vee(m_A > 274.55 \wedge 26257 < v_A \leq 39245)\\&\quad \vee(m_A > 271.67 \wedge \boldsymbol{s_{Dt}} > \mathbf{8.14})\\&\quad \vee(m_A > 269.84 \wedge 8.98 < v_{Dt} \leq 11179 \wedge k_{Dt} > 55.19)\\&\quad \vee(m_{Dt} > 0.95 \wedge m_A > 265.03 \wedge 223.15\\&\quad < k_Q \leq 543101255))\textbf{then attack}\end{aligned}$$

As far as concerns the results for the RUL dataset application (right side in Table 3), the inside and outside methods confirm an overall behavior similar to their previous application for the DNS tunneling and vehicle platooning scenarios; the first method overcomes the latter in the detection of all the three adversarial attacks. Despite the geometrical simplicity of the inside method, which looks at 2D adversarial regions by perturbing the thresholds of $N^{FR} = 2$ features only, we can point out the good balance between FPR and FNR obtained on JSMA attack (an analog result has been obtained for the DNS tunneling too) and FGSM. The obtained region for FGSM with inside method is shown in Fig. 8.

With respect to JSMA attack, an interesting result is the distribution of attack points in the space of the first $N^{FR} = 2$ Differently from the DNS tunneling case, the LLM0% algorithm works very well on JSMA and FGSM attacks, but it could not be applied to CW attack, since we obtained a huge number (2373) of very low-covering ($\leq$0.5%) rules after training the LLM with error forced to 0%. Hence, in applying Algorithm 3, we stumbled upon the same criticity we already experienced with the platooning problem. Together with the low metrics obtained through inside and outside on CW, this result corroborates the overall difficulty in detecting such an attack through the proposed XAI-based algorithms. Indeed, we observed very intricate overlapping of points in the two classes.

1) PERFORMANCE OF XAI-DRIVEN RELIABLE AI

The presented results corroborate the effort towards the detection of adversarial attacks. In particular, the simplification

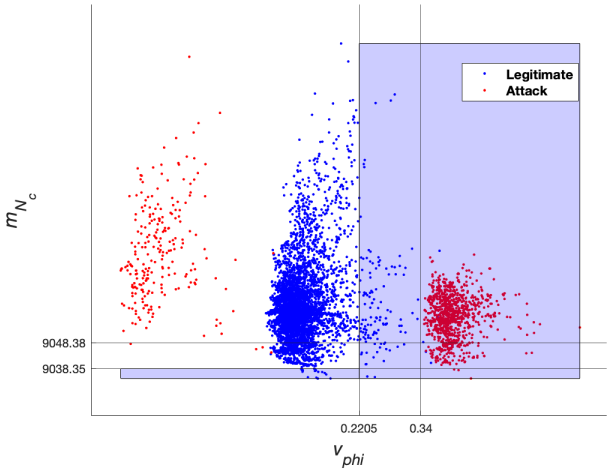


FIGURE 8. Adversarial Region obtained for FGSM attack in RUL dataset by perturbing the intervals thresholds for features v_{phi} and m_{N_c} with `inside` method (TPR=0.81, FPR=0.03, TNR=0.97, FNR=0.19).

brought by hyper-rectangles in outside and inside methods, allows to check the current state of the system: in case any of the variable is approaching the threshold found in the adversarial region, an alarm may be triggered to alert the proximity to unsafe conditions. The explainability of the region thus may shed new light into the way to guarantee safety and can be evaluated by the expert in the field. The worse performance obtained on platooning and RUL datasets indicates that the effectiveness of the defense depends on the quality of the underlying dataset. Moreover, the time required for the application of XAI-based methods is in the order of tens of seconds for rules generation via LLM, while the optimization processes take up to tens of minutes. However, the low TPR obtained with `inside` and `outside` methods in the platooning, the RUL and in some cases of DNS tunneling may highlight possible explanations about the attacks behaviors, which seem to act within specific small regions of the feature space. This hypothesis is also supported by the rules obtained by the LLM with 0% error: especially in the RUL and the platooning cases, we got a high number of

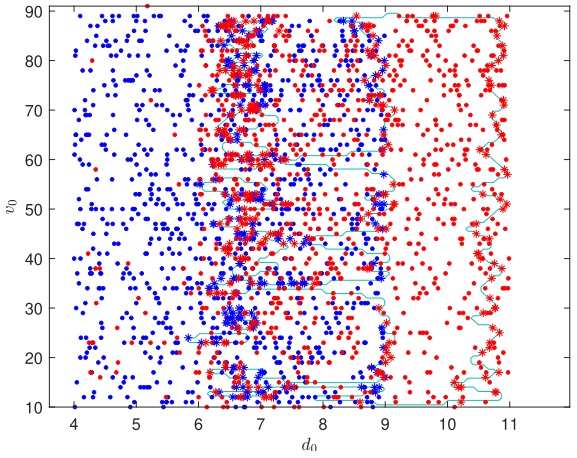


FIGURE 9. 2D graph of the "adversarial region" (the red points are the attacked ones) with d_0 (distance between cars) and v_0 (initial platooning speed) as input features of the JSMA-platooning dataset. The star points are the SVs of the description, coloured referring their specific label.

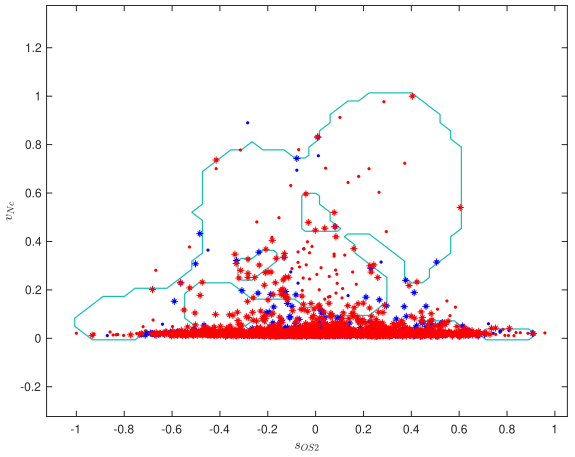


FIGURE 10. 2D graph of the "adversarial region" (the red points are the attacked ones) with s_{OS2} (Skewness of operational setting 2) and v_{Nc} (Variance of physical core speed) as input features of the CW-RUL dataset. The star points are the SVs of the description, coloured referring their specific label.

rules with very low covering, thus individuating many small areas where the attack tends to be localized. This result is consistent with the nature of the adversarial attacks, which generate "minimal" distortions of the attacked datasets, thus producing very intricate overlaps between the classes. Hence, this is an example of how the usage of XAI plays an essential role in discovering unknown and unexpected information about the data.

D. OBTAINED RESULTS WITH SAFE SVDD

In order to improve the results obtained with classical ML algorithms showed in Section VI-B, our goal is to determine the largest region of parameters with no false positives (i.e. prediction of attack, but no attack in reality) using the algorithms proposed in Section V-B.

For the `zeroFPRSVDD` algorithm we set $C_1 = 1/(\nu_1 N_1)$, 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 SVDD region) and we used the RBF kernel with σ determined with cross-validation for all the three datasets and attacks. The results are shown in Table 5. Let's pay more attention on the FPR index since it is the one that explains most the aim of the Safe SVDD: recall you that the purpose of the Safe SVDD is to find the largest region with the lowest rate of negative points within it, so we are less interested in what happens outside the Safe SVDD region. The performances on the three datasets show results totally in line with the other methodology. In particular we can notice that the CW attack is the most difficult to detect, emphasizing the hypothesis that the CW attack is the attack that most distorts the output of the algorithm under attack.

Although for some attacks the safety regions determined are not very large, with this algorithm we are sure to find areas with very low misclassification error (tending to zero)

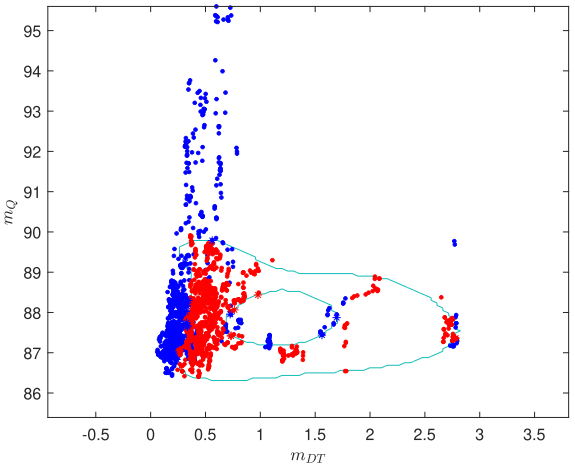


FIGURE 11. 2D graph of the "adversarial region" (the red points are the attacked ones) with m_{Dt} (average interarrival time between query and answer packet over 1000 sample) and m_Q (average size of query packet) as input features of the JSMA-DNS dataset. The star points are the SVs of the description, coloured referring their specific label.

in relation to the target class. In particular, when compared to SVM algorithm (to which the SVDD is closely related [15]) we can observe that the results have been improved.

Regarding the `eXplainable SVDD` algorithm, for each classification made via `zeroFPRSVDD` algorithm we extracted the set of intelligible rules and performed again the classification of the datasets. Results are reported in Table 5.

Not surprisingly, the performance of `eXplainable-SVDD` is inferior to that of the other algorithm: it is the price to pay for extracting explainability from a black-box algorithm. With the explainable version of the safe SVDD, we try to approximate complex decision boundaries with rectangles, i.e., rules. Thus, to avoid exponential generation of rules to exactly describe decision boundaries (which would not be as explainable and useful to a potential user), it is preferable to

TABLE 5. FPR, TPR, TNR and FNR for each dataset and attack with the Safe SVDD methods.

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
zeroFPRSVDD	CW	0.04	0.35	0.95	0.64	0.11	0.21	0.89	0.78	0.03	0.03	0.97	0.97
	JSMA	0.15	0.85	0.84	0.14	0.09	0.36	0.90	0.63	0	0.99	1	0.01
	FGSM	0.03	0.77	0.96	0.22	0.13	0.14	0.86	0.85	0.01	0.99	0.99	0.01
eXplainableSVDD	CW	0.23	0.35	0.76	0.64	0.34	0.34	0.65	0.65	0.44	0.73	0.55	0.26
	JSMA	0.28	0.53	0.71	0.46	0.34	0.30	0.65	0.69	0.50	0.57	0.50	0.43
	FGSM	0.28	0.28	0.71	0.71	0.35	0.33	0.64	0.66	0.57	0.55	0.43	0.45

admit a larger margin of error. This way you get fewer but more understandable rules.

In the case where safety regions are not operationally representative (as is also the case with canonical machine learning), it is necessary to admit that it is not possible to obtain zero statistical error. Therefore, it is better to allow the algorithms to have a higher probability of error (i.e., set the threshold on the number of FPRs higher) to still obtain the possibility of having a sufficiently significant data region in which to apply appropriate countermeasures.

As an example of the kind of rules generated by `eXplainable SVDD`, in the CW-DNS dataset we set $\varepsilon = 0.1432$, in the JSMA-platooning dataset we set $\varepsilon = 0.0184$ and in the FGSM-RUL dataset we set $\varepsilon = 0.0167$. Always referring to the three datasets in example, the first one highest-covering rule (i.e. the rule involving the largest number of data points, (3)) for the class *attack* for CW-DNS dataset is

if $(30931149 < v_A \leq 166588766)$
and $(211 < v_Q \leq 2604)$ and $(3779 < v_{Dt} \leq 155832)$
and $(360 < s_{Dt} \leq 392)$ and $(52 < s_A \leq 326)$
and $(368 < k_{Dt} \leq 4874$ and $(29 < k_A \leq 328)$
then attack

the first three rules for JSMA-platooning dataset are

1. **if** $0.37 < PER \leq 0.89$ and $7.38 < d_0 \leq 7.59$
and $54 < v_0 \leq 66$ **then attack**
2. **if** $N < 5$ and $0.36 < PER \leq 0.78$
and $9.81 < d_0 <= 9.94$ **then attack**
3. **if** $N < 5$ and $F_0 < -2$ and $0.37 < PER \leq 0.53$
and $7.26 < d_0 \leq 9.43$ and $29 < v_0 \leq 78$ **then attack**

and the first rule for FGSM-RUL dataset is

if $s_{os2} \leq 0.33$ and $m_{Nc} < 9060.24$
and $132.14 < v_{Nc} \leq 526.57$ and $0.04 < v_{phi} \leq 0.12$
and $38.22 < m_{W31} \leq 39.33$ **then attack**

As we were saying above, 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, i.e. rules. To ensure acceptable prediction confidence with these rules, a large amount of them is required: for the cases in example, JSMA-platooning and CW-DNS, the total number of rules generated are 751, 146 and 102 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.

These results show how SVDD can improve LLM algorithm for the detection of the attacked points in the datasets. Moreover, this procedure offers a simple and clear method for making SVDD explainable which is quite innovative with respect the well known methods for extracting rules from SVM [57], [58].

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 the following results. Platooning: `zeroFPRSVDD` on CW (0.11 FPR, 0.79 FNR), `inside` on JSMA (0.07, 0.44), `zeroFPRSVDD` (0.13, 0.86) and `inside` (0.26, 0.71) on FGSM; DNS: `inside` on CW (0.03, 0.55) and JSMA (0.03, 0.07), `zeroFPRSVDD` on FGSM (0.03, 0.22); RUL: `zeroFPRSVDD` (0.03, 0.97) and `inside` (0.02, 0.99) on CW; on JSMA, all optimized canonical algorithms (0, 0), `inside` (0.02, 0) and `zeroFPRSVDD` (0, 0.01) present comparable performance; similarly, all optimized canonical algorithms (0, 0), except for logistic regression and SVM, and `zeroFPRSVDD` (0.01, 0.01) present very good metrics on FGSM. Overall, `zeroFPRSVDD` and `inside` thus arise as the most competitive reliable AI solutions and should be considered jointly, along with canonical solutions, if one wants to build the right firewall in front of unknown adversarial threats. The rationale behind why one should perform better than the other in specific circumstances deserves further investigation. It is however worth noting that, despite SVDD is the more powerful and therefore the preferred choice in principle, it may lead to non-optimal performance in case of non-optimal setting of its parameters (which is a non-trivial task on its own) and, in that case, `inside`, though built on the more complex `LLM0%`, may exploit its simplicity and surprisingly achieve good results.

As a final remark, results obtained through canonical SVM with hyper-parameters optimization for DNS tunneling, under JSMA and FGSM, also reveal good detection ability, as well as all the canonical methods on RUL dataset. However, in contrast with Reliable AI approaches, the optimization is

TABLE 6. Performance metrics (mean ± st.dev.) for canonical algorithms, over 100 test sets of increasing size.

		DNS		PLATOONING		RUL	
		FPR	TPR	FPR	TPR	FPR	TPR
Decision Tree	CW	0.10±0.01	0.55±0.03	0.51±0.02	0.70±0.02	0.00±0.00	0.00±0.00
	JSMA	0.08±0.01	0.99±0.00	0.28±0.03	0.91±0.01	0.00±0.00	1.00±0.00
	FGSM	0.52±0.03	0.92±0.03	0.24±0.02	0.86±0.02	0.00±0.00	1.00±0.00
Random Forest	CW	0.04±0.01	0.50±0.03	0.58±0.03	0.78±0.02	0.00±0.00	0.00±0.00
	JSMA	0.00±0.01	0.98±0.01	0.19±0.03	0.87±0.02	0.00±0.00	1.00±0.00
	FGSM	0.04±0.01	0.83±0.02	0.26±0.02	0.92±0.02	0.00±0.00	1.00±0.00
KNN	CW	0.17±0.02	0.55±0.03	0.54±0.03	0.64±0.03	0.00±0.00	0.00±0.00
	JSMA	0.14±0.02	0.74±0.03	0.23±0.03	0.77±0.02	0.00±0.00	1.00±0.00
	FGSM	0.16±0.03	0.38±0.04	0.41±0.03	0.74±0.03	0.00±0.00	1.00±0.00
Logistic Regression	CW	1.00±0.00	1.00±0.00	0.47±0.03	0.52±0.04	0.00±0.00	0.00±0.00
	JSMA	0.68±0.03	0.77±0.02	0.23±0.03	0.78±0.03	0.00±0.00	1.00±0.00
	FGSM	1.00±0.00	1.00±0.00	0.49±0.04	0.50±0.03	0.00±0.00	0.81±0.02
Gradient Boost	CW	0.11±0.02	0.57±0.03	0.32±0.02	0.71±0.03	0.00±0.00	0.00±0.00
	JSMA	0.00±0.01	1.00±0.00	0.20±0.03	0.88±0.02	0.00±0.00	1.00±0.00
	FGSM	0.01±0.00	0.97±0.01	0.19±0.02	0.91±0.02	0.00±0.00	1.00±0.00
SVM	CW	0.94±0.01	0.94±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
	JSMA	0.05±0.01	0.15±0.02	0.29±0.03	0.77±0.03	0.00±0.00	1.00±0.00
	FGSM	0.04±0.01	0.03±0.01	0.01±0.01	0.01±0.00	0.00±0.00	0.00±0.00

computationally costly and does not provide any form of error control and explainability.

1) EXPLAINED BLACK-BOX VS NATIVE XAI

In this section, we briefly report some considerations about our two approaches to XAI.

Given the complex nature of the adversarial problem (see e.g. Fig.9), the logic derived from native XAI cannot be expected to be very simple and intuitive. Hence, besides adopting reliable AI algorithms based on LLM only, we also developed an explained black-box (`eXplainableSVDD`) approach. It is able to track the complex non-linear boundaries between the classes, but it requires two steps (interrogation of the SVDD and rule extraction), thus introducing two error factors. From this point of view, although reduced to profiling the separation through hyper-rectangles only (i.e., the geometrical shape of the LLM rules), native XAI generates the rules in a single step and may achieve good performance as well. This is corroborated by the reported experiments, where we had good results from both methodologies, in terms of balance between FPR and FNR. Hence, again, our final recommendation is to consider both approaches.

F. ON THE SCALABILITY OF THE DETECTION MODELS

To strengthen the results obtained so far, further analysis was carried out on the scalability of the proposed method. Scalability is here intended as the capability of the obtained detection models to perform adequately despite the size of the input attacked dataset.

For each application scenario (DNS tunneling, vehicle platooning, RUL estimation), we divided the original test sets (used for the previous detection experiments) into 100 subsets of increasing size (from 1% to 100% of the original size, in 1% increments) on which statistical, runtime, and memory analyses were performed.

Such experiments were all executed using a host equipped with Intel Core i5 dual-core processor at 2.6Ghz and 8GB RAM memory. The host runs macOS version 10.15.7.

1) STATISTICAL ANALYSIS

Tables 8, 6, 7 show the FPR and TPR performance of each method for all datasets and attacks. We can see that the results are in line with those obtained in previous experiments. In particular, the reliable AI methods based on XAI confirm their aptitude to provide better classification results in general than canonical machine learning algorithms. The performance of `zeroFPRSVDD` remains fairly unchanged.

2) RUNTIME AND MEMORY

Tables 9, 10, 11 report the mean values (over the 100 test sets) obtained for the processing time (in tables, denoted as CPU) and memory (Resident Set Side - or RSS memory, hence, related to the memory allocated to the process in RAM), along with the corresponding standard deviations, for all datasets and attacks. The following considerations can be derived. As a general remark, a difference on the measures can be observed between canonical machine learning and XAI-based reliable AI (inside, outside and LLM0%), on the one side, and zeroFPRSVDD on the other. The first two approaches exhibit faster processing time and more stable memory consumption over size increases, while the latter is less fast and presents higher and less stable RSS memory values than for python-based algorithms. However, processing time increases with the size of the test set for all the three detection approaches (canonical, XAI-based and SVDD-based).

The plots in Fig. 12 show this behavior for all datasets and both reliable AI approaches (XAI-based and SVDD-based). The trend of the CPU time seems to be nearly the same for all the three attacks, with the only exception of zeroFPRSVDD on RUL dataset (Fig.12m). Also, some spikes can be observed in the plots, especially for the CW attack,

TABLE 7. Performance metrics (mean ± st.dev.) for XAI-based reliable AI methods, over 100 test sets of increasing size.

		DNS		PLATOONING		RUL	
		FPR	TPR	FPR	TPR	FPR	TPR
Inside	CW	0.05±0.01	0.46±0.03	0.02±0.01	0.01±0.01	0.02±0.00	0.01±0.00
	JSMA	0.02±0.01	0.92±0.02	0.07±0.02	0.56±0.03	0.00±0.00	0.00±0.00
	FGSM	0.04±0.01	0.62±0.03	0.26±0.02	0.28±0.02	0.02±0.00	1.00±0.00
Outside	CW	0.00±0.00	0.00±0.01	0.00±0.00	0.00±0.00	0.00±0.00	0.06±0.01
	JSMA	0.00±0.00	0.24±0.03	0.00±0.00	0.26±0.03	0.03±0.00	0.81±0.02
	FGSM	0.00±0.00	0.24±0.03	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
LLM0%	CW	0.04±0.01	0.44±0.02	–	–	–	–
	JSMA	0.26±0.02	0.95±0.01	–	–	0.00±0.00	0.81±0.02
	FGSM	0.00±0.00	0.78±0.03	–	–	0.00±0.00	0.77±0.02

TABLE 8. Performance metrics (mean ± st.dev.) with the Safe SVDD methods over 100 test sets of increasing size.

		DNS		PLATOONING		RUL	
		FPR	TPR	FPR	TPR	FPR	TPR
zeroFPRSVDD	CW	0.00±0.00	0.34±0.05	0.09±0.02	0.15±0.03	0.10±0.05	0.10±0.04
	JSMA	0.10±0.01	0.80±0.02	0.09±0.02	0.55±0.08	0.00±0.00	0.99±0.01
	FGSM	0.14±0.02	0.28±0.01	0.10±0.01	0.15±0.02	0.00±0.00	0.99±0.02
eXplainableSVDD	CW	0.00±0.00	0.01±0.00	0.08±0.00	0.01±0.03	0.00±0.04	0.33±0.04
	JSMA	0.00±0.00	0.00±0.00	0.01±0.01	0.08±0.00	0.12±0.01	0.01±0.01
	FGSM	0.28±0.04	0.58±0.03	0.00±0.01	0.14±0.02	0.07±0.00	0.01±0.02

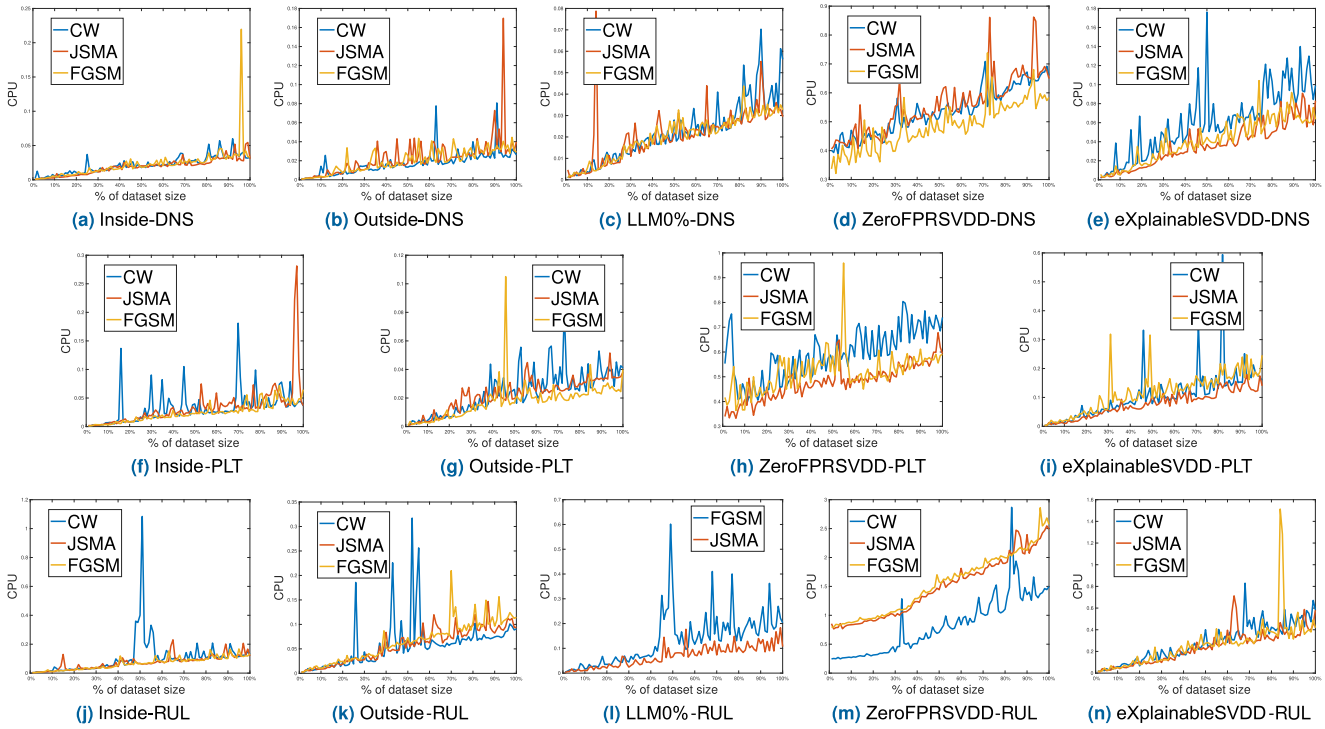


FIGURE 12. Graphs of the processing time (denoted as CPU in the figure) as the dataset and algorithm change: first row DNS dataset, second row Platooning dataset (denoted with PLT in the plot captions), third row RUL dataset.

but their presence may be related to external causes and not to the detection algorithms. Considering that CPU usage is in direct relationship with the computational complexity of an algorithm, [59] final remark that emerges from the results about processing time is that, for all methods and attacks, the complexity of the detection models increases with the dataset size. We did not observe the same behavior for RSS memory, which oscillates and increases with test set size only for zeroFPRSVDD method, while for the other methods it follows a nearly constant trend. These differences may be associated to the software code of the detection algorithms, being python 3.9 for canonical machine learning and the XAI-based reliable AI methods and Matlab R2021 for zeroFPRSVDD.

For the optimized canonical algorithms, processing time is very short (<0.1s) for all the datasets, except for

TABLE 9. Processing time (CPU) and memory consumption (RSS) for canonical models tested on 100 test sets with increasing sizes (results reported as mean ± standard deviation).

		DNS		PLATOONING		RUL	
		CPU	RSS	CPU	RSS	CPU	RSS
Decision Tree	CW	0.00±0.00	9174.40±6.20	0.00±0.00	9176.24±1.36	0.00±0.00	7556.56±2.04
	JSMA	0.00±0.00	9180.00±0.00	0.00±0.00	9180.00±0.00	0.00±0.00	7556.00±0.00
	FGSM	0.00±0.00	9180.00±0.00	0.00±0.00	9178.04±2.00	0.00±0.00	7556.00±0.00
Random Forest	CW	0.05±0.02	9180.00±0.00	0.08±0.04	9180.00±0.00	0.05±0.04	7553.00±1.73
	JSMA	0.07±0.02	9180.76±1.57	0.02±0.01	9180.04±0.40	0.07±0.03	7552.00±0.00
	FGSM	0.05±0.02	9180.00±0.00	0.04±0.02	9180.00±0.00	0.06±0.04	7552.00±0.00
KNN	CW	0.09±0.05	9184.00±0.00	0.16±0.08	9184.00±0.00	0.64±0.36	7552.00±0.00
	JSMA	0.05±0.03	9184.00±0.00	0.15±0.08	9164.00±0.00	0.39±0.22	7552.00±0.00
	FGSM	0.05±0.03	9184.00±0.00	0.02±0.01	9178.52±8.87	0.16±0.09	7554.56±3.73
Logistic Regression	CW	0.00±0.00	9184.00±0.00	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00
	JSMA	0.00±0.00	9184.00±0.00	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00
	FGSM	0.00±0.00	9184.00±0.00	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00
Gradient Boost	CW	0.01±0.01	9184.00±0.00	0.02±0.01	9164.00±0.00	0.02±0.02	7560.00±0.00
	JSMA	0.00±0.00	9184.00±0.00	0.00±0.00	9164.00±0.00	0.01±0.04	7560.00±0.00
	FGSM	0.01±0.00	9184.00±0.00	0.01±0.00	9164.00±0.00	0.02±0.01	7560.00±0.00
SVM	CW	0.00±0.00	9186.28±1.98	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00
	JSMA	0.00±0.00	9188.00±0.00	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00
	FGSM	0.00±0.01	9188.00±0.00	0.00±0.00	9164.00±0.00	0.00±0.00	7560.00±0.00

TABLE 10. Processing time (CPU) and memory consumption (RSS) for XAI-based models tested on 100 test sets with increasing sizes (results reported as mean ± standard deviation).

		DNS		PLATOONING		RUL	
		CPU	RSS	CPU	RSS	CPU	RSS
Inside	CW	0.02±0.01	9124.96±6.20	0.03±0.03	9112.72±4.68	0.10±0.13	9147.04±6.43
	JSMA	0.02±0.01	9131.60±1.20	0.03±0.04	9116.48±1.90	0.07±0.05	9152.00±0.00
	FGSM	0.02±0.02	9128.00±0.00	0.02±0.02	9116.00±0.00	0.07±0.04	9152.00±0.00
Outside	CW	0.02±0.01	9131.72±1.02	0.02±0.02	9992.64±8710.40	0.06±0.05	9152.00±0.00
	JSMA	0.02±0.02	9116.00±0.00	0.02±0.01	9116.00±0.00	0.06±0.03	9152.00±0.00
	FGSM	0.02±0.01	9118.84±4.01	0.02±0.01	9116.00±0.00	0.06±0.04	9152.00±0.00
LLM0%	CW	0.04±0.03	18577.96±16.63	–	–	–	–
	JSMA	0.03±0.02	18596.00±0.00	–	–	0.07±0.04	8265.20±8.45
	FGSM	0.04±0.02	18596.00±0.00	–	–	0.12±0.09	8273.04±13.48

TABLE 11. Processing time (CPU) and memory consumption (RSS) for Safe SVDD methods tested on 100 test sets with increasing sizes (results reported as mean ± standard deviation).

		DNS		PLATOONING		RUL	
		CPU	RSS	CPU	RSS	CPU	RSS
zeroFPRSVDD	CW	0.54±0.08	981114.16±49663.31	0.59±0.10	991341.28±44006.02	0.82±0.49	1168571.40±233561.18
	JSMA	0.56±0.10	981126.20±50738.10	0.48±0.07	991242.68±40774.24	1.48±0.52	1430018.56±223946.24
	FGSM	0.49±0.08	977234.60±54004.73	0.52±0.08	991892.68±45294.48	1.54±0.53	1449057.04±223909.95
eXplainableSVDD	CW	0.06 ± 0.03	9151±2.05	0.10±0.08	9156±1.90	0.28± 0.17	9112±1.12
	JSMA	0.03±0.02	9152±1.37	0.07±0.04	9159±0.00	0.26±0.21	9112±0.00
	FGSM	0.04±0.02	9152±0.0	0.11±0.06	9156±0.79	0.24±0.15	9112±0.00

K-Nearest-Neighbors (KNN), which seems to be slower (>0.1s) on platooning and RUL datasets. The RSS memory reaches higher values for platooning and DNS than for RUL, but the very low standard deviations (<0.01, approximated with 0.00 in the table) denote very high stability despite the increase of test set size (only a few exceptions are present, e.g. for KNN on FGSM-attacked platooning dataset).

XAI-based reliable AI (inside, outside and LLM0%) are very quick too (CPU < 0.1s, except for inside method on CW-attacked RUL dataset and LLM0% on FGSM-attacked RUL dataset). Unlike canonical algorithms, RSS memory has similar mean values for all the three datasets, with the notable exception of LLM0% on DNS, whose RSS mean value doubles with respect to the other methods. Again, the standard deviation is very low (<0.05) in most cases, with just a few exceptions (e.g., CW-attacked platooning dataset)

As mentioned, zeroFPRSVDD is not as quick as the previous methods (CPU > 0.4s in all cases) and the RSS shows higher mean values than python-based algorithms, with greater variability (high standard deviations) in all cases. Moreover, higher mean RSS values are obtained for RUL scenario than for platooning and DNS.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we investigated an innovative approach to detect adversarial machine learning attacks by comparing canonical ML algorithms with two innovative Reliable AI approaches focused on eXplainable AI and on Support Vector Data

Description (SVDD). In particular, we investigated three possible adversarial attacks, namely the Carlini-Wagner, the Fast Gradient Sign Method and the Jacobian based saliency map. The proposed approach plans to generate malicious datasets (i.e. under attack by adversarial algorithms) on the defensive side to train the algorithms by combining a malicious dataset with the legitimate one. In this way, the algorithm is able to identify a possible attack certainly sacrificing legitimate data but, the basic idea of the work, provides for a classification of adversarial machine learning attacks.

Initially, we decided to test canonical algorithms to detect the adversarial attacks also trained with a hyperparameters optimization. Then, since the results were not convincing at all, we addressed the problem via Reliable AI. The first innovative method we proposed was the application of three techniques (`reliability from outside`, `reliability from inside` and `LLM 0%`) to individuate the largest portions of attacks with zero statistical error. Such techniques were all completely based on the LLM, which is explainable: hence, the obtained results, although not always satisfactory in terms of detection performance, allowed us to understand useful knowledge about the adversarial behaviour. Indeed, our resulting adversarial regions are defined in the form of interpretable ranges of values in the space of the considered features and represent a warranty to detect, at least, a small region where the attack infiltrates.

The second reliable AI algorithm, the `zeroFPRSVDD`, built from the SVDD, allowed to profile with sufficient care and for each of the three attacks on all example datasets an adversarial region with low false positive values and acceptable indices of accuracy on the target class. However, being a black box algorithm it was necessary to translate it into an explainable language through the `eXplainableSVDD` algorithm: compared to the previous algorithm the performances are slightly decreased but however remained acceptable considering the difficulty of the problem. It will be certainly a good starting point for future work to try to improve the part related to the extraction of rules, taking advantage of results already known in the extraction of rules from SVM [57].

To evaluate the results, we tested the entire process on three datasets with different classification scopes, each referring to a particular real-world scenario (DNS tunneling, vehicle platooning and RUL estimation). As reported in the related Section VI, the results of the variety of performed tests do not indicate a predominant algorithm in terms of performance (not even SVDD which is in closed form); rather, they must all be used to find the best combination between FPR and FNR. As discussed in Section VI-C1, we evaluated the more interesting combination in terms of FPR and FNR for each dataset. For the platooning dataset, the `zeroFPRSVDD` approach is a good countermeasure against a CW and FGSM attacks while, against the JSMA attack, the `inside` is more efficient. Regarding the DNS dataset instead, the `inside` algorithm reports more balance for the selected metrics against the CW while the `zeroFPRSVDD` is more performant for the other two considered adversarial attacks. Surprisingly, on RUL estimation dataset, the optimized canonical algorithms (with few exceptions) share, or overcome, the same performance of reliable AI solutions, working very well on FGSM and JSMA attacks, while being ineffective for CW attacks detection. Again, this proves the need to consider all the methodologies in order to perform a satisfactory detection. However, we point out that reliable AI solutions allow for a better control on the error due to their specific design.

To complete the analysis, we also performed a statistical validation by measuring the FPR and TPR when the different proposed models were used for inference on multiple test sets of increasing size, showing that all the models are stable as the size gets larger. Also, we evaluated the CPU usage time and memory consumption of the detection models: all models resulted to be fast enough during inference, while the memory resulted to depend on the algorithms implementation instead.

Regarding the shortcomings of our methodology, the proposed detection framework is not yet complete for preventing adversarial attacks. The method works a posteriori, after the attack has been carried out, so it is not possible to apply countermeasures to prevent the attack. However, the information obtained from the detection phase can be exploited to identify or prevent subsequent attacks, since the method clearly defines the adversarial regions. It will be future work to study how to use our method to prevent the attacks or to apply suitable countermeasures.

As far as future work is concerned, we plan to evaluate the proposed approach in the image field where adversarial machine learning is mainly known and tested. Moreover, 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 [60]. Also, another interesting direction would demand to test the applicability of the detection models at production stage.

REFERENCES

[1] M. Pak and S. Kim, "A review of deep learning in image recognition," in *Proc. 4th Int. Conf. Comput. Appl. Inf. Process. Technol. (CAIPT)*, Aug. 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, p. 3726, May 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, Nov. 2020.

[4] M. Aiello, M. Mongelli, and G. Papaleo, "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning," *Int. J. Commun. Syst.*, vol. 28, no. 14, pp. 1987–2002, Sep. 2015.

[5] I. S. Candanedo, E. H. Nieves, S. R. González, M. T. S. Martín, and G. A. Briones, "Machine learning predictive model for industry 4.0," in *Proc. Int. Conf. Knowl. Manage. Organizations*. Cham, Switzerland: 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 *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 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," *Int. J. Autom. Comput.*, vol. 17, no. 2, pp. 151–178, Apr. 2020.

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

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[10] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[11] (Mar. 2020). *Concepts of Design Assurance for Neural Networks CoDANN*. European Union Aviation Safety Angency. Daedalean, AG, Standard. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/EASA-DDLN-Concepts-of-Design-Assurance-for-Neural-Networks-CoDANN.pdf

[12] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[13] *EASA Concept Paper: First Usable Guidance for Level 1 Machine Learning Applications, a Deliverable of the EASA AI Roadmap*, Eur. Union Aviation Saf. Agency, Daedalean, AG, Standard, Cologne, Germany, Apr. 2021.

[14] V. N. Balasubramanian, S. Ho, and V. Vovk, *Conformal Prediction for Reliable Machine Learning*, 1st ed. Waltham, MA, USA: Morgan Kaufmann, 2014.

[15] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.

[16] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Mach. Learn.*, vol. 20, pp. 45–66, Nov. 2004.

[17] A. Carlevaro and M. Mongelli, "Reliable ai trough SVDD and rule extraction," in *Proc. Int. IFIP Cross Domain (CD) Conf. Mach. Learn. Knowl. Extraction (MAKE)*, 2021, pp. 153–171.

[18] S. Narteni, M. Ferretti, V. Orani, I. Vaccari, E. Cambiaso, and M. Mongelli, "From explainable to reliable artificial intelligence," in *Proc. Int. IFIP Cross Domain (CD) Conf. Mach. Learn. Knowl. Extraction (MAKE)*, 2021, pp. 255–273.

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

[20] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Defence Sci. J.*, vol. 68, no. 4, p. 356, Jun. 2018.

[21] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Appl. Sci.*, vol. 9, no. 5, p. 909, Mar. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/5/909

[22] 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.

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

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

[25] 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," *Comput. Secur.*, vol. 73, pp. 326–344, Mar. 2018.

[26] 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 *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 533–537.

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

[28] L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Functionality-preserving black-box optimization of adversarial Windows malware," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3469–3478, 2021.

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

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

[31] Y. E. Sagduyu, Y. Shi, and T. Erpek, "IoT network security from the perspective of adversarial deep learning," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[32] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[33] Z. Luo, S. Zhao, Z. Lu, Y. E. Sagduyu, and J. Xu, "Adversarial machine learning based partial-model attack in IoT," in *Proc. 2nd ACM Workshop Wireless Secur. Mach. Learn.*, 2020, pp. 13–18.

[34] E. Anthi, L. Williams, A. Javed, and P. Burnap, "Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102352.

[35] M. Mongelli, E. Ferrari, M. Muselli, and A. Fermi, "Performance validation of vehicle platooning through intelligible analytics," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 4, no. 2, pp. 120–127, Jun. 2019.

[36] A. Fermi, M. Mongelli, M. Muselli, and E. Ferrari, "Identification of safety regions in vehicle platooning via machine learning," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2018, pp. 1–4.

[37] J. M. Faria, "Machine learning safety: An overview," Safety-Critical Syst. Club, York, U.K., Tech. Rep. SCSC-140, 2018.

[38] K. Czarnecki and R. Salay, "Towards a framework to manage perceptual uncertainty for safe automated driving," in *Proc. Int. Workshop Artif. Intell. Saf. Eng. (WAISE)*, 2018, pp. 1–7.

[39] Y. Wiener and R. El-Yaniv, "Agnostic pointwise-competitive selective classification," *J. Artif. Intell. Res.*, vol. 52, pp. 171–201, Jan. 2015.

[40] A. Campagner, F. Cabitza, and D. Ciucci, "Three-way decision for handling uncertainty in machine learning: A narrative review," in *Proc. Int. Joint Conf. Rough Sets*, 2020, pp. 137–152.

[41] L. Demetrio, A. Valenza, G. Costa, and G. Lagorio, "WAF-A-MoLE: Evading web application firewalls through adversarial machine learning," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 1745–1752.

[42] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, pp. 1–21, May 2021.

[43] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," 2018, *arXiv:1807.01069*.

[44] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[45] M. A. A. Milton, "Evaluation of momentum diverse input iterative fast gradient sign method (M-DI2-FGSM) based attack method on MCS 2018 adversarial attacks on black box face recognition system," 2018, *arXiv:1806.08970*.

[46] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[47] A. Carlevaro and M. Mongelli, "A new SVDD approach to reliable and explainable AI," *IEEE Intell. Syst.*, vol. 37, no. 2, pp. 55–68, Mar. 2022.

[48] M. Muselli, "Switching neural networks: A new connectionist model for classification," in *Neural Nets*. Berlin, Germany: Springer, 2005, pp. 23–30.

[49] M. Muselli and A. Quarati, "Reconstructing positive Boolean functions with shadow clustering," in *Proc. Eur. Conf. Circuit Theory Design*, vol. 3, Sep. 2005, pp. 377–380.

[50] (Nov. 2012). *KEEL. Ebsite: Keel (Knowledge Extraction Based on Evolutionary Learning)*. [Online]. Available: http://sci2s.ugr.es/keel/datasets.php

[51] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2489, pp. 145–152, Jan. 2015.

[52] *Turbofan Engine Degradation Simulation Data Set*. Accessed: May 2022. [Online]. Available: https://data.nasa.gov/Aerospace/Turbofan-engine-degradation-simulation-data-set/vrks-gjie/

[53] *Remaining Useful Life Estimation Using Convolutional Neural Network*. [Online]. Available: https://it.mathworks.com/help/predmaint/ug/remaining-useful-life-estimation-using-convolutional-neural-network.html#mw_rtc_RULEstimationUsingCNNExample_98C35430
[54] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–9.
[55] 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," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[56] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.
[57] H. Nuñez, C. Angulo, and A. Catala, "Rule-based learning systems for support vector machines," *Neural Process. Lett.*, vol. 24, no. 1, pp. 1–18, Aug. 2006.
[58] N. Barakat and A. P. Bradley, "Rule extraction from support vector machines: A review," *Neurocomputing*, vol. 74, nos. 1–3, pp. 178–190, Dec. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231210001591
[59] L. Caviglione, M. Gaggero, E. Cambiaso, and M. Aiello, "Measuring the energy consumption of cyber security," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 58–63, Jul. 2017.
[60] M. Mongelli, "Design of countermeasure to packet falsification in vehicle platooning by explainable artificial intelligence," *Comput. Commun.*, vol. 179, pp. 166–174, Nov. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421002504

IVAN VACCARI received the degree *(laude)* in computer engineering and the Ph.D. degree in computer science from the University of Genoa, in 2017 and 2021, respectively. During his research activities, he worked in different European projects focused on security in healthcare data, the IoT, and financial infrastructures. He is currently a Research Fellow at the IEIIT Institute, Consiglio Nazionale delle Ricerche, working on the IoT and network security focused on identification of vulnerabilities and developed of innovative cyber threats. Regarding detection and mitigation systems, he is working on machine learning and artificial intelligence approaches.

ALBERTO CARLEVARO received the master's degree *(cum laude)* in applied mathematics from the University of Genoa, in March 2020, with a physics-mathematics thesis on the behavior of liquid crystals under electromagnetic fields, where he is currently pursuing the Ph.D. degree with the Department of Electrical, Electronic and Telecommunications Engineering and Naval Architecture (DITEN) in the research topic "Traffic Analysis in the Smart City," in collaboration with CNR and S.M.E. Aitek. He was a Research Fellow at the Institute of Electronic, Computer and Telecommunications Engineering (IEIIT), National Research Council (CNR), where he worked on machine learning and explainable AI in collaboration with Rulex Inc. His current research interests include machine learning, deep learning, statistical learning, and explainable AI.

SARA NARTENI received the M.Sc. degree in bioengineering from the University of Genoa, in March 2020, with a thesis entitled "Pleural Line Ultrasound Videos Analysis for Computer Aided Diagnosis in Acute Pulmonary Failure." She is currently pursuing the Ph.D. degree with the Politecnico di Torino, working with the IEIIT Institute of Consiglio Nazionale delle Ricerche. She works on data analytics and machine learning topics from different fields, such as industry, healthcare, and automotive. Her research interest includes computer security topics, including covert channels and the Internet of Things.

ENRICO CAMBIASO received the Ph.D. degree in computer science from the University of Genoa. He is working for Ansaldo STS and Selex ES, both companies are part of the Finmeccanica Group. He has a strong background as a Computer Scientist and he is currently employed at the IEIIT Institute, Consiglio Nazionale delle Ricerche, as a Technologist, working on cyber-security topics and focusing on the design of last generation threats and related protection.

MAURIZIO MONGELLI (Member, IEEE) received the Ph.D. degree in electronics and computer engineering from the University of Genoa (UNIGE), in 2004. The Ph.D. was funded by Selex Communications S.p.A. (Selex). He worked with both Selex and the Italian Telecommunications Consortium (CNIT), from 2001 to 2010. During his Ph.D. and in the following years, he worked on the quality of service for military networks with Selex. He was the CNIT Technical Coordinator of a research project concerning satellite emulation systems, funded by the European Space Agency; and he spent three months working on the project at the German Aerospace Center in Munich. He is currently a Researcher at the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), National Research Council (CNR), where he deals with machine learning applied to bioinformatics and cyber-physical systems. He is the coauthor of over 100 international scientific articles, two patents, and is participating in the SAE G-34/EUROCAE WG-114 AI in Aviation Committee.