IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 September 2022, accepted 1 November 2022, date of publication 28 November 2022,
date of current version 1 December 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3224930*

RESEARCH ARTICLE

# A Methodology for Evaluating the Robustness of Anomaly Detectors to Adversarial Attacks in Industrial Scenarios

**ÁNGEL LUIS PERALES GÓMEZ[1], LORENZO FERNÁNDEZ MAIMÓ[1], FÉLIX J. GARCÍA CLEMENTE[1], JAVIER ALEJANDRO MAROTO MORALES[2], ALBERTO HUERTAS CELDRÁN[3], (Member, IEEE), AND GÉRÔME BOVET[4]**

[1]Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, Espinardo, 30003 Murcia, Spain
[2]École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland
[3]Communication Systems Group CSG, Department of Informatics (IfI), University of Zurich, 8006 Zürich, Switzerland
[4]Cyber-Defence Campus within armasuisse Science & Technology, 3602 Thun, Switzerland

Corresponding author: Ángel Luis Perales Gómez (angelluis.perales@um.es)

This work was supported by the Spanish Ministry of Science, Innovation and Universities, State Research Agency, FEDER Funds, under Grant RTI2018-095855-B-I00; by the Swiss Federal Office for Defense Procurement (Armasuisse) with the CyberSpec under Grant CYD-C-2020003; and by the European Commission Horizon 2020 Programme under grant agreement number H2020-SU-DS-2019/883335 - PALANTIR (Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises), and the European Commission (FEDER/ERDF).

**ABSTRACT** Anomaly Detection systems based on Machine and Deep learning are the most promising solutions to detect cyberattacks in the industry. However, these techniques are vulnerable to adversarial attacks that downgrade prediction performance. Several techniques have been proposed to measure the robustness of Anomaly Detection in the literature. However, they do not consider that, although a small perturbation in an anomalous sample belonging to an attack, i.e., Denial of Service, could cause it to be misclassified as normal while retaining its ability to damage, an excessive perturbation might also transform it into a truly normal sample, with no real impact on the industrial system. This paper presents a methodology to calculate the robustness of Anomaly Detection models in industrial scenarios. The methodology comprises four steps and uses a set of additional models called support models to determine if an adversarial sample remains anomalous. We carried out the validation using the Tennessee Eastman process, a simulated testbed of a chemical process. In such a scenario, we applied the methodology to both a Long-Short Term Memory (LSTM) neural network and 1-dimensional Convolutional Neural Network (1D-CNN) focused on detecting anomalies produced by different cyberattacks. The experiments showed that 1D-CNN is significantly more robust than LSTM for our testbed. Specifically, a perturbation of 60% (empirical robustness of 0.6) of the original sample is needed to generate adversarial samples for LSTM, whereas in 1D-CNN the perturbation required increases up to 111% (empirical robustness of 1.11).

**INDEX TERMS** Adversarial attacks, evasion attacks, industrial control systems, machine learning, deep learning, robustness.

## I. INTRODUCTION

The industry is experiencing its fourth revolution, also known as Industry 4.0, which is mainly driven by the adaptation of industrial processes to new technologies and computational paradigms. Among the most relevant changes affecting current industries we highlight the integration of the fifth generation of mobile networks (5G), bringing to reality minimum latency and high bandwidth in communications; the big data, optimizing the analysis of large amounts of data; and the Industrial Internet-of-Things (IIoT), connecting large amounts of heterogeneous and resource-constrained devices to the Internet [1]. However, despite the benefits of Industry 4.0, it is also opening the door to new cyberattacks affecting devices and critical industrial processes [2]. Every year, the number and variety of cyberattacks are

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

growing, making traditional security approaches outdated in short time windows. In this context and due to the number of highly specialized and new (zero-day) attacks affecting heterogeneous industries, the research community is evolving towards the use of semi-supervised or unsupervised Machine Learning (ML) and Deep Learning (DL) techniques to detect cyberattacks [3].

In such a scenario, the current Anomaly Detection (AD) systems relying on ML and DL are the most promising and effective solutions to detect unseen attacks [4]. In contrast to traditional approaches, these systems discriminate between the normal and abnormal behavior of the industrial processes without relying on existing databases that store the cyberattacks patterns. However, the current AD solutions based on ML/DL are vulnerable to adversarial attacks, making them inappropriate for real systems. Adversarial attacks consist of manipulative actions to ML/DL models intending to cause model misbehavior or acquire protected information. Among the existing adversarial attacks, evasion attacks are some of the most relevant as they are performed during the evaluation phase of the system, once the model is trained. In industrial scenarios affected by malware, the main goal of evasion attacks is to craft samples modeling the malware behavior (anomalous samples) to misclassify them (as normal samples) and allow the malware to affect industrial devices or processes without being detected.

Adversarial attacks raise new trust and security challenges affecting ML/DL in general, and AD-based solutions to detect cyberattacks in particular. In this context, data scientists are already making efforts to provide highly precise and trustworthy AI-based solutions in different application scenarios [5]. Recently, IBM has identified a set of pillars needed to achieve trusted AI [6]. One of these pillars is robustness, whose main goal is to measure how resilient ML/DL models are against adversarial attacks. Once the robustness level is calculated, it can be notified to end-users, in conjunction with classical performance metrics, or even be used to improve the model's robustness using adversarial training, where the network is fine-tuned with adversarial samples.

The literature has offered different metrics to measure model's robustness. The three most widespread are Empirical Robustness (ER) [7], Local Loss Sensitivity (LLS) [8], and Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER) [9]. These metrics are highly effective in different application fields such as computer vision. However, they present limitations when used to evaluate the robustness of AD in industrial scenarios. One of the most relevant limitations is the impossibility of distinguishing between an adversarial sample that deceives the anomaly detector and an adversarial sample converted into a normal sample by an excessive alteration. For example, consider a water distribution process where a denial of service (DoS) cyberattack is launched. This cyberattack aims to stop the water supply for a certain geographical area. The water supply is controlled by valves that can take values between 0 (completely closed) and 1 (completely open). Therefore, the DoS cyberattack can modify such features to close the valves and stop the supply. Besides, an attacker who wants to launch a DoS cyberattack that goes unnoticed by the AD system could modify the DoS samples to make them adversarial. However, these features could take the value 1 (completely open) due to excessive disturbance, leaving the DoS cyberattack without effect. In both cases, the adversarial attack is considered successful, but in the second case, it does not have not a negative impact on the industrial device. For this reason, a mechanism is needed to differentiate these two adversarial versions, thus providing a reliable measurement of the model's robustness. An additional drawback is the heterogeneity of data types used in industrial environments. Unlike image recognition and other domains such as audio signals, where values are usually floats, there are discrete values, continuous values or even timestamps, usually with internal consistency constraints, which makes it not always possible to calculate a gradient or generate a valid adversarial sample [10], [11], [12].

In order to face the previous limitations affecting ER, LLS, and CLEVER, the current paper presents the following contributions:

- A methodology for estimating the robustness of an AD model based on ML and DL techniques in industrial scenarios, using a set of additional ML models (support models) to determine if an adverse sample remains anomalous. This methodology considers four fundamental steps and proposes a robustness metric that is a modification of the ER metric. It is worth mentioning that the proposed methodology does not focus on training a robust AD model, but on measuring the robustness of AD model already trained.
- Validation of the proposed methodology using a dataset generated from the Tennessee Eastman Process [13], an industrial scenario that, although simulated, is realistic. Specifically, we show the robustness calculation for Long Short-Term Memory (LSTM) and 1-Dimensional Convolutional Neural Network (1D-CNN) models, which are well suited to deal with time-series data. The model that achieves the highest robustness should be considered to be deployed in a real scenario. Our experiments show that the 1D-CNN model achieves a robustness of 1.1, approximately twice that of the LSTM model (0.6).

The remainder of this paper is structured as follows. Section II reviews the state of the art. Section III shows a motivating example explaining the difficulty of generating adversarial samples in industrial scenarios. In Section IV, we detail the four-step methodology proposed to measure the model's robustness when using the AD paradigm with ML and DL techniques in an industrial scenario. The methodology implementation using the Tennessee dataset and its validation are detailed in Section V. Finally, the conclusions and future work are included in Section VI.

## II. RELATED WORK

In this section, we present a brief review focusing on robustness to adversarial attacks in AD. In addition, we introduce different solutions in the context of AD in industrial environments to fully understand the proposed methodology.

### A. ANOMALY DETECTION IN INDUSTRIAL SETTINGS

Cybersecurity in industrial environments is a field of great interest to the research community. In this context, a wide variety of approaches have been proposed. For example, the authors of [14] propose a collaborative trust-based unbiased control mechanism that performs a dynamic assignment of industrial control to avoid malicious nodes attacking industrial devices. However, the most widely used techniques are those in charge of detecting anomalies. These techniques can be categorized into DL techniques specially designed to work with time-series data and classical ML techniques.

In the first category, we highlight the models LSTM and 1D-CNN, which are especially designed to deal with time-series data. For example, the authors of [15] presented a scalable and efficient solution for real-time AD in industrial settings. In particular, the authors proposed a hybrid statistical-ML model that integrated a SARIMA (seasonal autoregressive integrated moving average)-based dynamic threshold model and an LSTM model to identify the abnormal behavior in a joint way with a low false-positive rate. Another example of LSTM usage can be found in [16] where the authors proposed a Variational LSTM learning model for AD based on reconstructed feature representation. The authors designed an encoder-decoder architecture associated with the Variational LSTM in order to learn low-dimensional representation from high-dimensional raw data. Then, the transformed data was fed into a lightweight estimation network to identify anomalies.

As an example of using 1D-CNN in AD, we highlight [17]. In this study, the authors proposed an AD method based on measuring the statistical deviation of the predicted value from the observed value. Besides, the authors tested different configurations of 1D-CNN. After detecting 32 out of 36 attacks, the authors claimed the effectiveness of 1D-CNN in AD problems. Another example is presented in [18], where the authors introduced the 1D-CNN to diagnose anomalies from 1D time-series data generated by industrial sensors. To reduce the number of parameters, a 1D global average polling (1D-GAP) layer was designed to replace the fully connected layers. Furthermore, the authors replaced the usual final softmax layer with a nonlinear multi-class Support Vector Machine (SVM). The authors of [19] presented a novel approach that combined 1D-CNN and Gated Recurrent Units (GRU) to learn the spatiotemporal correlation between parameters.

In the category of classical ML approaches, we highlight the solution proposed in [20], where the authors presented an adaptive approach for defense against cyber-attacks in the context of industrial systems. In particular, the solutions combined several algorithms such as Artificial Neural Networks (ANN), LSTM, Isolation Forest (IF), and One-Class Support Vector Machine (OCSVM). In [21], the authors performed a study to compare different ML and DL models to detect anomalies in industrial settings. In particular, the models compared were Random Forest (RF), SVM, DNN, OCSVM, and IF. The authors conducted experiments with the traffic of Modbus TCP and S7comm protocols, concluding that SVM and RF were the models with a higher F1-score in both scenarios. Despite the fact that classical ML models cannot deal with time-series data out of the box, several modifications can be made to use such models in time-series data. The most popular approach is to preprocess the dataset to create a lagged dataset as shown in [22].

### B. ROBUSTNESS TO ADVERSARIAL ATTACKS IN AD

The contributions in the context of adversarial attacks to AD models in industrial systems are relatively recent. In [12], the authors briefly describe the techniques used in the generation of adversarial samples, illustrating the main differences between the cyber-physical domain and the traditional image domain (constraints in the sample perturbation, system knowledge of the attacker, the timing of the attack, and the existence of a human detector). They demonstrate this by performing an attack on the SWaT testbed. Similarly, the authors of [23] describe how to slightly modify sensor values in the Tennessee-Eastman Process Control System so that they remain unnoticed by an anomaly detector. In addition, the authors of [24] present a new adversarial attack especially designed for industrial scenarios. They compare adversarial samples generated with the proposed technique and those generated with existing methods such as Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM).

Since an AD model can suffer an adversarial attack, it is necessary to improve its adversarial robustness. In this context, a variety of defense mechanisms have been proposed to make the model more robust [25]. However, to the best of our knowledge, there are no proposed methodologies for determining the robustness of an AD model in industrial environments. The most similar approach -although applied to images- is the one presented in [25]. Additionally, when new adversarial defense techniques are presented, the authors tend not to measure the robustness achieved with a metric, but on the contrary, they use indirect methods like plotting the loss of accuracy with each technique. By way of illustration, in [26] the determination of the robustness of a model is based on the plotting of the drop in accuracy experienced in the presence of each adversarial attack. Nevertheless, we can find several robustness metrics [27] that can be used as a starting point to develop a suitable metric for industrial environments.

Learning from the limitations of existing approaches, our proposal consists in a methodology to estimate the robustness of a model taking into account the exposed constraints of industrial environments when determining the adversarialness of a sample.

## III. MOTIVATING EXAMPLE

In scenarios such as computer vision, an adversarial attack is successful if it simply causes the modified sample to be misclassified. In industrial scenarios, it is often further required that the adversarial sample be classified as harmless while preserving its anomalous nature. Misclassification can easily be achieved by altering the original anomalous sample until a clearly normal sample is obtained. However, this is not the goal of an adversarial attack. The modification of the adversarial sample has to be such that the model considers it as normal, but without belonging to the distribution of normal samples to have a negative impact on the target industrial process or system. This is achieved by techniques that take advantage of the peculiarities of the class separation boundary established by the trained model. However, there is an additional difficulty, which is how to determine when an adversarial sample has been altered so much that it has become a true and innocuous normal sample.

Fig. 1 illustrates a simplified example in an industrial setting where adversarial attacks are applied on a binary classification model. Fig. 1 (left) shows the probability density function (p.d.f) of two classes, from which a set of samples has been extracted. We assume that class 2 is the normal class and take a sample from class 1 that is considered hazardous to the industrial system. After altering the sample, one of the three situations shown could happen. Sample *a* is clearly adversarial, because it belongs to the anomalous distribution, but the model classifies it as normal. Sample *b*, on the other hand, has been classified as normal by the model but, actually, it does belong to neither the anomalous nor normal classes. Finally, sample *c* has become a harmless sample, because it clearly belongs to the normal class.

Fig. 1 (right) illustrates the real situation, where the p.d.f. of the classes are unknown and the boundary of the trained model serves as an estimate. Unfeasible areas, whose samples, would be considered corrupt and the industrial system would discard them, are also depicted (*d* sample). Additionally, the boundaries of two different models trained with the same dataset (support models) have also been plotted.

The boundary of each model has a different shape, and, therefore, we can distinguish three zones: the region where all the models agree on classifying the samples as class 1, i.e. (*a*); the region where they agree on classifying the samples as class 2, i.e. (*c*); and finally the remaining region, where there are discrepancies in the classification, i.e. (*b*). Our proposal is based on how to use these support models to estimate whether an adversarial sample has reached the p.d.f of the normal class ceasing to be adversarial, e.g. (*c*). However, some samples retain their adversarial nature after alteration, e.g. (*b*) and (*a*), and we call them truly adversarial samples.

## IV. METHODOLOGY

This section describes the proposed methodology to evaluate the robustness of anomaly detectors against adversarial attacks in industrial scenarios. A graphical representation of

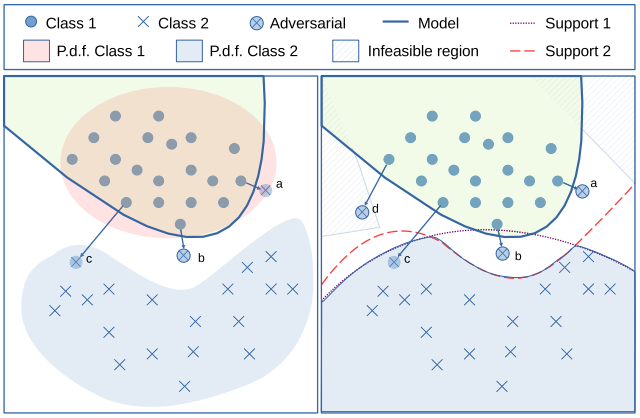


**FIGURE 1. A Simplified example of binary classification to illustrate the potential evolution of an adversarial sample in the context of industrial environments. A given sample from class 1 can go to different zones when altered adversarially, sometimes even falling into the actual p.d.f. of class 2 if the modification is excessive. Left: P.d.f of each class and the boundaries of the trained model. Right: boundaries of each trained model and the infeasible regions. The intersection of the support models gives us an estimate of the core of the actual p.d.f. of class 2.**

the methodology can be seen in Fig. 2. It can be divided into the following four steps:

1) *Models Preparation*: This step guides through the process of selecting and training models. In particular, it considers two types of models: the AD model employed to detect anomalies and whose robustness needs to be evaluated, and the support models that will help to discriminate between non-adversarial and truly adversarial samples. The support models are the core of the methodology and its main novelty. All these models need to be selected considering their suitability to be used with time-series data since most industrial systems produce this type of data. Once the models are selected, they need to be trained following a methodology focused on AD.
2) *Adversarial Samples Generation*: This step guides through the generation of adversarial samples. In this context, different approaches to performing an adversarial attack based on the AD model selected in the previous step are discussed. Besides, the methodology makes some recommendations about the parameters used together with the adversarial attack selected.
3) *Adversarial Dataset Generation*: This step draws the guidelines to generate a truly adversarial dataset that will be used later to evaluate robustness. Firstly, this step uses the support models trained in step 1 to discriminate between truly adversarial samples and non-adversarial ones. Finally, the adversarial dataset is generated considering only the truly adversarial samples.
4) *Robustness Considerations*: This step recommends using a specific metric to evaluate the model's robustness and discuss the considerations that must be taken into account. Specifically, the proposed metric is a

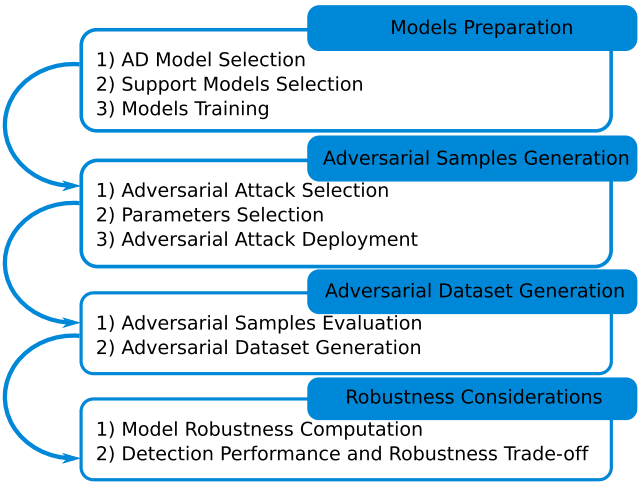


**FIGURE 2.** Graphical representation of the methodology proposed.

slight modification of the original ER metric because it does not depend on the model. In other words, it can be applied to any model whether it is based on a gradient or not. Finally, once the robustness is evaluated, it is necessary to consider if the AD model achieves the desirable robustness.

### A. MODELS PREPARATION

We define the next three tasks to prepare models.

#### 1) AD MODEL SELECTION

The first task is to select the proper model that will be implemented in the AD system. When selecting the AD model, it is essential to pay attention to the properties of the dataset used. Different model architectures have different implicit biases. For example, for tabular datasets with temporal dependencies, as in industrial systems, LSTM models might be the best option since they process features over the temporal dimension. In contrast, for datasets with tabular data but without temporal dependencies, Dense Neural Networks (DNN) models should be considered. Since, in industrial environments, most of the data have temporal dependence, the methodology recommends the use of models that can deal with time series out of the box, such as LSTM or 1D-CNN models.

#### 2) SUPPORT MODELS SELECTION

The second task is to select the proper support models used to identify the truly adversarial samples and avoid the problem explained in Section III. The support models need to be trained using the same dataset employed to train the AD model, and they will be in charge of evaluating each adversarial sample. In further steps, when a particular adversarial sample is evaluated as normal by a majority of the support models, the sample will be considered as belonging to the p.d.f of the normal class and, hence, non-adversarial. However, it is important to highlight that the support models need to be selected following a specific criterion. In particular, we defined three criteria to select such models.

- Support models need to be as much deterministic as possible. Otherwise, each time they are trained they may result in a different boundary and, therefore, the robustness of a particular model may vary. With this restriction, those DL models with a large number of hyper-parameters should be discarded. Nevertheless, there are ML models with interesting properties which make them suitable as support models.
- Supporting models need to achieve sufficient generalization ability to ensure that the p.d.f. of the normal class lies inside the intersection of their boundaries. This could cause support models to underperform the chosen AD models.
- Support models do not need to evaluate samples as quickly as the AD model, and therefore, we can select models that do not achieve a high degree of parallelism, such as classical ML models.

In particular, the methodology recommends using ensemble models like RF or gradient boosting models like XGBoost. These models have lesser hyper-parameters than DL models and, since their results are based on the average decision of many estimators, they achieve a high degree of determinism. Besides, both models can deal with time series data [28], [29] which is predominant in industrial scenarios.

#### 3) MODELS TRAINING

In this task, all the previously selected models are trained. A difference in the training process of both AD and support models is that AD in industrial scenarios is typically based on a multi-class classifier in order to discriminate between the different types of anomalies. In contrast, support models should be binary classifiers, since we are only interested in detecting if the sample is normal or abnormal. Considering this particularity, both models should be trained using the same dataset, but in the case of support models, the classes should be reduced to normal and abnormal. To train all the models, we recommend the methodology presented in [30].

Besides, to reduce the complexity of the AD model, we propose training such models using as few parameters as possible. We also recommend including regularization techniques such as dropout. Regularization smooths the decision boundary, improving the ability of the support models to distinguish between non-adversarial and adversarial samples. This recommendation is supported by the fact that industrial systems carry out repetitive actions, and therefore, the behavior of such systems should be able to be modeled with less complex models.

In case the introduction of regularization techniques is not possible, either because we must use a pre-trained AD model or because they reduce the performance of the AD model, it is advisable to increase the number of models in the ensemble to capture the complexity of the AD model.

Finally, one more circumstance that can arise in model training is that the support models do not achieve

sufficient performance to distinguish between non-adversarial and adversarial samples. In this case, the methodology recommends carrying out an exhaustive grid-search strategy to find the optimal hyper-parameters for RF and XGBoost, paying special attention to the number of estimators, the maximum depth of the trees and the maximum number of leaf nodes.

### B. ADVERSARIAL SAMPLES GENERATION

In this step, we identify three tasks to generate adversarial samples.

#### 1) ADVERSARIAL ATTACK SELECTION

The first task consists in selecting the proper adversarial attack to generate adversarial samples. In this task, the AD model previously chosen needs to be taken into account because not all attacks apply to all models. On the one hand, we need to consider what information we have concerning the model and the dataset. If we have full access to the model and the dataset, the methodology suggests using an adversarial attack based on the white-box approach. Similarly, if we do not have access to the model, but we have the dataset, the methodology suggests training a substitute model and using a white-box approach. Finally, if we do not have access to either dataset or model, the methodology recommends a black-box approach.

In addition, the data types present in the dataset and the AD model need to be considered. In DL and other differentiable models, we can use attacks exploiting the model gradient. The drawback of these attacks is that they are specifically designed to work on continuous data. However, a slight modification of the adversarial attack can be made to become compatible with categorical data [24]. On the contrary, if the target model is based on certain ML techniques, such as RF, whose gradient cannot be computed, an adversarial attack based on the black-box approach needs to be adopted.

Another consideration to keep in mind is whether to use a targeted or untargeted adversarial attack. A targeted attack attempts to deceive the model into predicting a particular class that is specified beforehand. In contrast, untargeted attacks perturb the sample to maximize the model loss, giving no special preference towards a particular class as long as it is not the original label. If an untargeted attack is used, the samples could change their class between different abnormal classes, but not necessarily to the normal class. A sample that changes between the different abnormal classes is not a problem as long as the AD system detects it as abnormal and it cannot reach the industrial target device. However, an abnormal sample classified as a normal sample could affect industrial devices. All in all, this methodology recommends using targeted adversarial attacks whenever possible.

The methodology presented in this paper is designed to evaluate the robustness of AD models against a single and several adversarial attacks. In the second case, it would be necessary to select all those attacks we are interested in. In addition, in the case of being interested in estimating the robustness against a large number of adversarial attacks, the methodology recommends including adversarial samples generated by different adversarial attacks based on the gradient since they are the most common.

#### 2) PARAMETERS SELECTION

This task makes some recommendations when selecting the adversarial attack parameters. These parameters fundamentally influence the model's robustness and the time required to generate the adversarial samples. Let us suppose an attack based on the gradients, which has two parameters frequently shared with other attacks of the same type. These two parameters are the maximum magnitude of the final perturbation, $\varepsilon$, and the number of iterations.

If an extremely large $\varepsilon$ is chosen, the reported robustness will be wrongly high. This is due to the high distortion introduced in each iteration, greatly increasing the difference between adversarial and original samples. However, this leads to a misleading measure, since introducing such a large distortion may cause the samples to cease to have a physical meaning and, therefore, have no effect on the physical world. Likewise, if a small $\varepsilon$ is chosen, the adversarial perturbations will be less prone to leave the anomalous class, and the model will seem overly robust.

Concerning the number of iterations, a trade-off is observed. The greater the number of iterations specified, which allows for lower values of $\varepsilon$, the greater the quality of the adversarial samples. However, it comes at the cost of taking a long time to generate them.

#### 3) ADVERSARIAL ATTACK DEPLOYMENT

Once the adversarial attack and its parameters are selected, the third task involves its deployment. During this task, the attack will convert the original samples into adversarial ones. To do so, the methodology proposes employing the test dataset used to validate the performance of the AD model. Although it is also possible to use the training or validation dataset, it is more realistic to use the test dataset. Note that in a real adversarial attack, the attacker may not have access to the training dataset, and needs to use samples not previously seen by the AD model.

### C. ADVERSARIAL DATASET GENERATION

The following tasks are suggested to generate a dataset with truly adversarial samples.

#### 1) ADVERSARIAL SAMPLES EVALUATION

The first task is to evaluate the adversarial samples previously generated to decide which are actual adversarial samples. Let $X$ be a set of anomalous samples for the model $M$, and $X^{adv}$ the set of adversarial samples obtained from $X$. Then, the truly adversarial samples are determined using the support models $M_1, \ldots, M_n$. A sample $x_j^{adv} \in X^{adv}$ is considered as non-adversarial if $(M_i(x_j^{adv}) == \text{Normal})$ for the majority of $i$ values. Otherwise, it will be considered as a truly adversarial sample. There are two reasons why support models can

evaluate an adversarial sample as normal. The first is the transferability of the adversarial perturbation between models. This methodology minimizes this possibility by recommending the use of support models whose architecture varies substantially from the model to be evaluated. The second reason is that the variations introduced by the adversarial attack can convert an abnormal sample into a truly normal sample as illustrated in Fig. 1.

#### 2) ADVERSARIAL DATASET GENERATION

### D. ROBUSTNESS CONSIDERATIONS

In this step, we establish the following tasks.

#### 1) MODEL's ROBUSTNESS COMPUTATION

This task consists in quantifying the model's robustness. For this purpose, the original sample dataset and the truly adversarial dataset are required. With these two datasets, the methodology uses the ER metric for the computation of robustness because it is independent of the chosen model. Many of the robustness metrics in the literature are targeted at specific models, e.g., differentiable models. However, the methodology allows the use of any other suitable metric.

The formal definition of ER metric is presented in Equation 1.

$$ER = \frac{1}{|X^{true}|} \sum_{i=1}^{|X^{true}|} \frac{\|o(x_i^{true}) - x_i^{true}\|}{\|o(x_i^{true})\|} \tag{1}$$

where $o(x_i^{true}) \in X$ is the original anomalous sample from which $x_i^{true} \in X^{true}$ was generated. This gives a measure of how robust the evaluated model is. The higher the ER, the larger the disturbance necessary to convert original samples into truly adversarial samples, and therefore, the greater the robustness of the model.

#### 2) DETECTION PERFORMANCE AND ROBUSTNESS TRADE-OFF

In this task, the methodology suggests evaluating both detection performance and robustness to decide which model needs to be deployed in the industrial scenario. Both aspects are crucial in industrial scenarios. On the one hand, if the detection performance is low, the possibility that a non-adversarial attack impacts the physical world is high. On the other hand, if the model's robustness is significantly low, it is easy to deploy adversarial attacks to generate adversarial samples. Therefore, the possibility of adversarial attacks impacting the physical world is high. In general, the methodology recommends selecting a model with the highest robustness as long as the detection performance does not differ significantly from the other candidate AD models.

## V. METHODOLOGY VALIDATION

In this section, the methodology proposed in Section IV is applied to validate it in an industrial scenario. To validate the methodology, we used the dataset generated by Rieth et al. [31] using the Tennessee Eastman (TE) process, which is a simulated testbed of a chemical process where the authors introduced 20 anomalies. The authors published four files: training and test files with anomalies and training and test files free of anomalies. Each file contains 500 simulations for each anomaly type. The training files contain 500 samples in each of these simulations, while the test files contain 960 samples. These files contain 52 features, including 41 measurement variables and 11 manipulated variables sampled every 3 minutes. This amounts to 25 hours for the training datasets and 48 hours for the test datasets.

The experiments performed in this work were carried out in a workstation with 94 GB of RAM, an Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz, and an NVIDIA GEFORCE GTX 1080.

### A. TESTBED

The TE process is shown in Fig. 3, where five main modules can be observed: the reactor, the condenser, the liquid-vapor separator, the compressor, and the stripper. The process produces two products through the reaction of eight components: A, B, C, D, E, F, G, and H. These reactions are defined by the following equations:

$$\begin{aligned} A(g) + C(g) + D(g) &\rightarrow G(liq) && Product\,1 \\ A(g) + C(g) + E(g) &\rightarrow H(liq) && Product\ 2 \\ A(g) + E(g) &\rightarrow F(liq) && Byproduct \\ 3D(g) &\rightarrow 2F(liq) && Byproduct \end{aligned}$$

During the process, reactants A, D, E, and C are injected into the reactor, where parts of the reactions described above occur. This results in products in the form of vapor and unreacted components that pass to the condenser, where they change from gas to liquid through a cooler. These products and unreacted components pass through the liquid-gas separator, where the unreacted components are recycled and reinjected at the inlet through the compressor. Conversely, the products condensed go to a product stripping module where the remaining reactants are removed. Finally, products G and H are generated from the output of the stripper.

Furthermore, all the reactions that take place are irreversible and exothermic, and they are approximately first-order with respect to the reactant concentrations. The reaction rates are a function of the temperature over an Arrhenius expression. Among all components, G is the one with the highest sensitivity to the temperature since it has more activation energy.

The control objectives for this process are the following:

- Maintain process variables at desired values.
- Keep process operating conditions within equipment constraints.
- Minimize variability of product rate and product quality during disturbances.
- Minimize movement of valves which affect other processes.
- Recover quickly and smoothly from disturbances, production rate, or product mix changes.

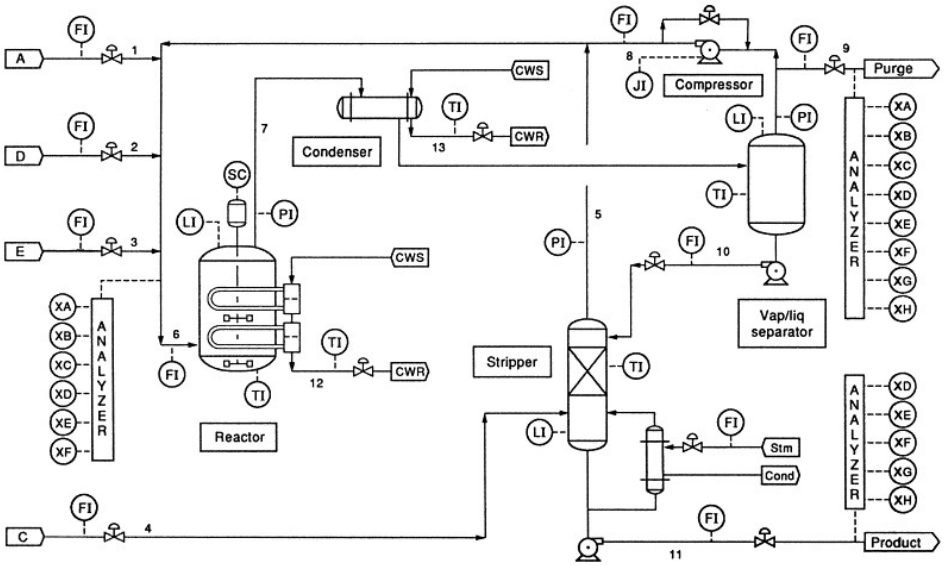


**FIGURE 3.** Tennessee eastman process.

### B. MODELS PREPARATION

This section details the implementation of the first step of the methodology where the AD and support models are trained.

#### 1) AD MODEL SELECTION

In this task, and considering the properties of the TE process, we chose two models that can deal with continuous data and manage time series. On the one hand, we selected an LSTM model that accepts time-series sequences as input and predicts the target class as output. On the other hand, we selected a 1D-CNN model that also accepts time-series sequences as input but whose internal architecture substantially differs from the LSTM model.

#### 2) SUPPORT MODELS SELECTION

Following the second task, we chose the support models used to determine truly adversarial samples. As suggested by the methodology, we selected two ensemble models that typically achieve a high degree of determinism. In particular, we selected RF and XGBoost models.

#### 3) MODELS TRAINING

Following the third task, we carried out the training of both AD and support models. With this aim, we created our training, validation, and test datasets from the training and test files provided by [31], considering the 52 features included in the dataset. To generate our training dataset, we selected all the samples of the first 400 simulations from the original training files. Our validation dataset was created by taking all the samples from the training files corresponding to the last 100 simulations. In both datasets, for each simulation, we ignored the initial 20 samples of the files containing anomalies because these samples were mislabeled as anomalous. Finally, our test dataset was generated by selecting all the simulations included in the original test files but considering only 500 samples of each anomaly, starting from sample 160 because the previous 159 samples were mislabeled as anomalous.

Since our proposal does not focus on training the AD model, but measuring the robustness of such models, we did not carry out any feature engineering step except grouping samples, and therefore, the data used were identical to the ones provided by the authors of the dataset in [31]. As the last step, we created sequences with 5 timesteps, resulting in samples of shape (5, 52) to train both AD and support models.

Subsequently, the samples referring to anomalies 3, 9, and 15 were eliminated since they did not suffer enough variation to be considered as anomalies, as pointed out by [32]. In conclusion, taking into account the remaining 17 anomaly types, the size of the training, validation and testing datasets were 3 264 000, 816 000, and 4 250 000 samples, respectively. Additionally, we scaled the three datasets by using the mean and standard deviation of the training dataset.

As an additional step to train, validate, and test the support models, we generated a two-class version of the dataset with only the normal and anomalous labels. The number of anomalous samples becomes much higher than the number of normal samples. Therefore, we obtained a balanced version of the dataset by taking all the normal samples and the same number of samples randomly chosen among all the anomalies.

Before training the 1D-CNN and LSTM models, we performed a hyper-parameters optimization with training and validation datasets, resulting in the architectures shown in Table 1. A similar procedure was performed with the support models using the balanced binary dataset, and the results are listed in Table 2.

**TABLE 1. Architectures of 1D-CNN and LSTM models.**

| | Layers | Hyper-parameters |
|---|---|---|
| 1D-CNN | 1-D Convolutional Layer | Filters: 32 |
| | | Kernel size: 3 |
| | Dense Layer | Neurons: 64 |
| | Dense Layer | Neurons: 21 |
| LSTM | LSTM Layer | Neurons: 8 |
| | Dense Layer | Neurons: 16 |
| | Dropout | Rate: 0.5 |
| | Dense Layer | Neurons: 21 |

**TABLE 2. Hyper-parameters used in support models.**

| | Hyper-parameter | Value |
|---|---|---|
| RF | Maximum Depth | 35 |
| | Number of estimators | 10 |
| XGBoost | Maximum Depth | 35 |
| | Number of estimators | 10 |
| | Alpha | 10 |

**TABLE 3. Detection performance achieved by each model.**

| | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| XGBoost | 0.788 | 0.835 | 0.788 | 0.781 |
| RF | 0.847 | 0.873 | 0.847 | 0.844 |
| 1D-CNN | **0.976** | **0.978** | **0.976** | **0.976** |
| LSTM | 0.928 | 0.935 | 0.928 | 0.929 |

To objectively compare the performance of these models, we used accuracy, precision, recall, and F1-score metrics on the test. Since the last three metrics are designed to be used with binary classifiers, we used for multiclass the weighted version of these metrics provided by the library Scikit-Learn. As can be observed in Table 3, the model that achieved the best F1-score was 1D-CNN (0.976) followed by LSTM (0.929). In contrast, the supported models achieved the worst F1-scores (0.781 for XGBoost and 0.844 for RF).

### C. ADVERSARIAL SAMPLES GENERATION

In this step, we selected an adversarial attack to be deployed and to generate adversarial samples from LSTM and 1D-CNN models.

#### 1) ADVERSARIAL ATTACK SELECTION

Following the first task, we chose an adversarial attack according to the characteristics of the model and the testbed. To be specific, we selected an attack that handles continuous data and targets DL models. We also assumed that the attacker had access to both the model and the dataset, and, therefore, we applied a white-box approach.

The adversarial attack selected was a slight modification of BIM, targeted and unclipped. The method is the same as proposed in [24], except for the mask for categorical features. We did not use such a mask because we wanted to modify all the features and the dataset does not have categorical features. The formal definitions of this method can be seen in Equation 2, where $X$ is the original array of samples, $X'_0$ is the first iteration where the original samples are considered, and $X'_{n+1}$ are the successive iterations. In this step, the gradient, $\nabla$, of the cost function, $J$, of the previous samples, $X_n$ with respect to the target label, $y_{target}$, is computed and added to the samples, modulated by the perturbation parameter, $\varepsilon$.

$$X'_0 = X; X'_{n+1} = X'_n + \varepsilon \cdot sign(\nabla J(X_n, y_{target})) \quad (2)$$

This method generates a batch of adversarial samples from a batch of original samples. Since it is a gradient-based attack, the fundamental parameters are the number of iterations and epsilon, $\varepsilon$, which indicates the magnitude of the disturbance introduced in each iteration. In the original BIM, all samples are altered in every iteration. This means that when a sample in the batch is converted to adversarial, i.e., its class changes from abnormal to normal according to the model, it will continue being modified until the algorithm reaches the last iteration. The main consequence is that a significant number of samples will undergo a great and unnecessary change.

Conversely, our version considers, in each iteration, only those samples that are not adversarial. In other words, when a sample changes from abnormal to normal class, it is excluded from the following iterations. This means that only the minimum disturbance will be applied to change the class of the sample.

#### 2) PARAMETERS SELECTION

Following the second task, we selected the parameters used with the adversarial attack. The main goal of this task in our specific attack is to select the right parameter to craft adversarial samples as similar as possible to the original ones. To this end, we chose an $\varepsilon$ of 0.005 since we considered this value small enough so that a large disturbance is not introduced at each iteration. Concerning the number of iterations, we selected 100, which may seem to be an excessive value. However, our version of BIM stops modifying the samples that have become adversarial regardless of the number of iterations.

#### 3) ADVERSARIAL ATTACK DEPLOYMENT

Following the third task, we generated the adversarial samples. To accomplish this goal, we employed all the anomalies of the test dataset. Then, we applied our version of BIM using the gradients of the LSTM model and the original abnormal samples in the test dataset (4 000 000), creating a new dataset of adversarial samples. This dataset was used to evaluate the robustness of the LSTM model. The same process was performed with the 1D-CNN model. After executing the adversarial attacks we obtained 3 007 479 and 2 652 192 adversarial samples for 1D-CNN and LSTM (see Table 4), respectively. As can be seen in the row labeled *non-adversarial samples*, the AD models (LSTM and 1D-CNN) generated samples that were subsequently classified as anomalous samples by the support models (RF and XGBoost), considering them as non-adversarial samples. The number of samples considered to compute robustness is presented in the row labeled *truly adversarial samples*, i.e., 2 545 456 and 2 382 082 for 1D-CNN and LSTM, respectively.

**TABLE 4. Number of truly adversarial and non-adversarial samples generated by 1D-CNN and LSTM.**

| | 1D-CNN | LSTM |
|---|---|---|
| Truly adversarial samples | 2 545 456 | 2 382 082 |
| Non-adversarial samples | 462 023 | 270 110 |
| Total adversarial samples | 3 007 479 | 2 652 192 |

### D. ADVERSARIAL DATASET GENERATION

In the third step, we used the support models to decide which samples were truly adversarial and create the adversarial dataset employed to measure the robustness of the models.

#### 1) ADVERSARIAL SAMPLES EVALUATION

Since we were evaluating the robustness of LSTM and 1D-CNN models, we carried out this first task twice. When both support models classified an adversarial sample as normal, it was removed from the dataset in the next task. Specifically, Table 4 shows the number of truly adversarial samples that were preserved, the number of non-adversarial samples that were removed, and the total adversarial samples generated after executing the adversarial attack. As can be seen, the adversarial attack managed to generate more truly adversarial samples using 1D-CNN than LSTM. Similarly, the number of non-adversarial samples was also greater for 1D-CNN than for LSTM. In particular, considering the 1D-CNN model, around 15% of samples were non-adversarial, whereas considering the LSTM model, this number decreases up to 10 %.

The final output of this task is two sets. One set contains those adversarial samples that are not truly adversarial but are generated by the LSTM model. Similarly, the second set contains those adversarial samples not truly adversarial by the 1D-CNN model.

#### 2) ADVERSARIAL DATASET GENERATION

Following the second task, we generated the final adversarial dataset for each model, which was used to evaluate their respective robustness. To be specific, the final dataset is equal to the original dataset but removing the non-adversarial samples. Therefore, this dataset only contained truly adversarial samples, resulting in a dataset with 2 545 456 and 2 382 082 samples for 1D-CNN and LSTM robustness evaluation, respectively.

### E. ROBUSTNESS CONSIDERATIONS

In this step, we followed the two tasks proposed in the methodology. In particular, we computed the robustness using Equation 1 and discuss the robustness and detection performance trade-off.

#### 1) MODEL's ROBUSTNESS COMPUTATION

Following the first task, we computed the ER of the LSTM and the 1D-CNN models previously trained. Fig. 4 shows the distribution of the ER (horizontal axis) for each of the considered models after 100 iterations. To clearly show the ER distribution, extremely large values were removed. As the figure also shows, the 1D-CNN model is more robust. To be specific, if we compute the median of ER (the white point in Fig. 4), we see that the 1D-CNN model achieved a robustness of 1.110, while the LSTM achieved a robustness of 0.601. This means that for generating adversarial samples for the LSTM model, it is necessary to modify the sample introducing a perturbation greater than 60.1% of the original sample for more than 50% of samples. In contrast, to generate adversarial samples for 1D-CNN, the perturbation needed is around 111% of the original sample. The model's robustness is related to the perturbation needed to convert original samples into adversarial ones. The larger the perturbation, the more robust the model. In this specific case, it seems that a more complex model (1D-CNN) is more robust than a simpler one (LSTM). In particular, the 1D-CNN model has 12 597 trainable parameters, while the LSTM model has 2 453 trainable parameters. Besides, the 1D-CNN model used the dropout regularization technique during the training phase, while the LSTM did not apply that technique. This implies that the boundary decision of the 1D-CNN model is smoother and, therefore, simpler than the LSTM model.

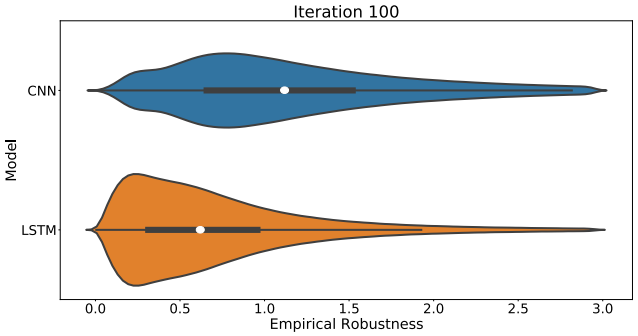


**FIGURE 4. Empirical robustness for iteration number 100. The blue violin plot shows the ER distribution for the CNN model, while the orange violin plot displays the ER for the LSTM model. Each plot represents the probability density and they are vertically symmetric about the horizontal gray bar. The thick part of the bar shows the interquartile range, and the white point represents the median.**

#### 2) DETECTION PERFORMANCE AND ROBUSTNESS TRADE-OFF

Following the second task, we discuss next the trade-off between detection performance and robustness. Table 5 shows a summary of the detection performance and both the ER using our methodology and without using it. Besides, although different metrics are not comparable, we decided to include the CLEVER score since it can tell us if one model is more robust than another. The CLEVER score was computed using the default parameters specified in the ART library. In addition, we set the maximum ball distortion to 10 and used $l_2$ as the norm. Finally, we selected 5 000 uniformly random samples from the original dataset and computed the median of their CLEVER scores.

**TABLE 5.** Detection performance and empirical robustness achieved by each model.

| | Accuracy | Precision | Recall | F1-Score | ER using our methodology | ER without using our methodology | CLEVER score |
|---|---|---|---|---|---|---|---|
| 1D-CNN | **0.976** | **0.978** | **0.976** | **0.976** | **1.110** | **2.488** | **0.046** |
| LSTM | 0.928 | 0.935 | 0.928 | 0.929 | 0.601 | 1.356 | 0.023 |

In this specific case, the 1D-CNN model achieved the best results in both detection performance and robustness. Therefore, this model should be used in the core of an AD system. Furthermore, the robustness varies substantially depending on whether our methodology is used or not. As previously mentioned, this is because calculating robustness without applying our methodology will include non-adversarial samples and may lead to wrong results. For example, in this particular case, both models are apparently twice as robust, and actually they are not. In general, the selection of a model depends on the particular scenario. In our case, due to the malicious intention that attackers could have, it is better to choose the model with the highest robustness. Otherwise, the model with the highest evaluation performance should be selected.

As can be seen in Table 5, the median of the CLEVER scores also shows that the 1D-CNN model is more robust than the LSTM model. In this case, the value tells us the lower bound $l_2$ minimum distortion needed to convert the samples into normal ones. In particular, as shown by our approach, the distance required to craft an adversarial sample using the 1D-CNN (0.046) model is twice that using the LSTM (0.023).

### F. DISCUSSION

Our work establishes a clear four-step methodology for computing the robustness of ML and DL models specially designed for AD in industrial scenarios in relation to adversarial attacks. Although we presented a general methodology, we applied it to a specific scenario. In particular, we validated it using the TE process [31], a simulated testbed of a chemical process widely used in works related to AD. The results of the experiments demonstrated that it is not only necessary to take into account the detection performance but also the robustness of the model against adversarial attacks. Thus, our work fills in a gap in the literature regarding methodologies to evaluate the robustness of ML and DL models. As we observed in Section I, there are several metrics to compute the robustness of a model. However, these metrics have important limitations. On the one hand, most metrics are specific to differentiable models, e.g., CLEVER [9] and LLS [8]. On the other hand, other proposals focus on computing the robustness in terms of the minimal perturbation needed to change the sample class [7]. However, in AD, the robustness needs to be computed considering the change from one of the abnormal classes to the normal class.

One relevant aspect of the methodology that we propose is that only the truly adversarial samples are considered when computing the robustness. When an adversarial attack is performed, some samples can change their actual class to the normal class. As we discussed in Section III, in industrial scenarios, abnormal samples that change to normal classes are harmless. Furthermore, in contrast to other fields such as computer vision, identifying if an adversarial sample presents a potential threat to the industrial system requires expert knowledge. Therefore, when computing the robustness, we need to discard these harmless samples and only consider the adversarial samples that are misclassified by the AD system but continue being anomalous. In order to filter these samples, which we call non-adversarial samples, we propose using support models. Specifically, to discard non-adversarial samples, we propose a voting process carried out by these support models.

Another relevant aspect of our methodology is that it can be applied to all ML and DL models irrespective of whether the model is not differentiable. This is because the metric that we propose computing the robustness is ER, which considers the original samples and the adversarial samples generated. However, unlike the original ER metric proposed in [7], our methodology allows computing the metrics using targeted adversarial attacks.

Finally, the most critical limitation of our methodology is the selection of the support models. On the one hand, these models allow discriminating between truly adversarial samples and non-adversarial samples. However, these models also introduce a degree of uncertainty. In fact, the selection of different support models can lead different authors to obtain different robustness results. Therefore, as proposed in the methodology, these models need to be selected following specific criteria. For example, those models with a relevant number of hyper-parameters, such as DL models, need to be avoided. In contrast, an ensemble based on a voting process between different models achieves a high degree of certainty. Therefore, they are a convenient choice to be selected as support models.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new methodology to measure the robustness of AD models to adversarial attacks in industrial scenarios. Its novelty is the consideration of the possibility that, after applying adversarial attacks, some adversarial samples become truly normal and do not need to be taken into account in the robustness computation. The methodology comprises four steps: models preparation, adversarial samples generation, adversarial dataset generation, and robustness consideration. To be precise, the methodology uses a set of models called support models to discriminate between truly adversarial and non-adversarial samples, and robustness is computed considering only the truly adversarial

samples. Besides, we applied this methodology to the TE process, which is a realistic industrial scenario. In this scenario, we evaluated the robustness of two AD models: 1D-CNN and LSTM. The experiments showed that, in this specific scenario, 1D-CNN model achieved higher robustness (1.110) than LSTM (0.601). This means that to generate adversarial samples, the perturbation required in the LSTM is equals 60.1% of the original samples, while the perturbation needed in the 1D-CNN is about double, 110%.

As future work, we plan to continue evaluating the robustness of AD systems in other industrial scenarios using different industrial datasets. In addition, we plan to study the properties of different models to be used as support models. Besides, we also plan to study the relationship between robustness, adversarial samples, and interpretability method. One application of this study can be the improvement of the robustness of the AD model by detecting adversarial samples using interpretability techniques.

## REFERENCES


[1] T. Stock and G. Seliger, "Opportunities of sustainable manufacturing in industry 4.0," *Proc. CIRP*, vol. 40, pp. 536–541, Jan. 2016.

[2] T. H. Szymanski, "Security and privacy for a green Internet of Things," *IT Prof.*, vol. 19, no. 5, pp. 34–41, Oct. 2017.

[3] P. M. S. Sanchez, J. M. J. Valero, A. H. Celdran, G. Bovet, M. G. Perez, and G. M. Perez, "A survey on device behavior fingerprinting: Data sources, techniques, application scenarios, and datasets," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1048–1077, Mar. 2021.

[4] J. Men, Z. Lv, X. Zhou, Z. Han, H. Xian, and Y.-N. Song, "Machine learning methods for industrial protocol security analysis: Issues, taxonomy, and directions," *IEEE Access*, vol. 8, pp. 83842–83857, 2020.

[5] S. Thiebes, S. Lins, and A. Sunyaev, "Trustworthy artificial intelligence," *Electron. Markets*, vol. 31, pp. 1–18, Jun. 2020.

[6] (2021). *Trusted AI*. Accessed: May 12, 2021. [Online]. Available: https://www.research.ibm.com/artificial-intelligence/trusted-ai/

[7] 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.

[8] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 233–242.

[9] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–18.

[10] V. Ballet, X. Renard, J. Aigrain, T. Laugel, P. Frossard, and M. Detyniecki, "Imperceptible adversarial attacks on tabular data," in *Proc. NIPS Workshop Robust AI Financial Services, Data, Fairness, Explainability, Trustworthiness Privacy (Robust AI FS)*, Vancouver, BC, Canada, Dec. 2019, pp. 1–9.

[11] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ML attacks in the problem space," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1308–1325.

[12] G. Zizzo, C. Hankin, S. Maffeis, and K. Jones, "Adversarial machine learning beyond the image domain," in *Proc. 56th Annu. Design Autom. Conf.*, New York, NY, USA, Jun. 2019, pp. 1–4.

[13] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[14] J. Chen, J. Wu, H. Liang, S. Mumtaz, J. Li, K. Konstantin, A. K. Bashir, and R. Nawaz, "Collaborative trust blockchain based unbiased control transfer mechanism for industrial automation," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4478–4488, Aug. 2019.

[15] W. Hao, T. Yang, and Q. Yang, "Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber-physical systems," *IEEE Trans. Autom. Sci. Eng.*, early access, May 6, 2021, doi: 10.1109/TASE.2021.3073396.

[16] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3469–3477, May 2020.

[17] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proc. Workshop Cyber-Physical Syst. Secur. (PrivaCy)*, Jan. 2018, pp. 72–83.

[18] W. Gong, Y. Wang, M. Zhang, E. Mihankhah, H. Chen, and D. Wang, "A fast anomaly diagnosis approach based on modified CNN and multisensor data fusion," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13636–13646, Dec. 2021.

[19] X. Xie, B. Wang, T. Wan, and W. Tang, "Multivariate abnormal detection for industrial control systems using 1D CNN and GRU," *IEEE Access*, vol. 8, pp. 88348–88359, 2020.

[20] J. Vávra, M. Hromada, L. Lukáš, and J. Dworzecki, "Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment," *Int. J. Crit. Infrastructure Protection*, vol. 34, Sep. 2021, Art. no. 100446.

[21] A. L. P. Gomez, L. F. Maimo, A. H. Celdran, F. J. G. Clemente, C. C. Sarmiento, C. J. Del Canto Masa, and R. M. Nistal, "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019.

[22] D. Lekkas, G. D. Price, and N. C. Jacobson, "Using smartphone app use and lagged-ensemble machine learning for the prediction of work fatigue and boredom," *Comput. Hum. Behav.*, vol. 127, Feb. 2022, Art. no. 107029.

[23] A. Ghafouri, Y. Vorobeychik, and X. Koutsoukos, "Adversarial regression for detecting attacks in cyber-physical systems," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3769–3775.

[24] Á. L. P. Gómez, L. F. Maimó, A. H. Celdrán, F. J. G. Clemente, and F. Cleary, "Crafting adversarial samples for anomaly detectors in industrial control systems," in *Proc. 4th Int. Conf. Emerg. Data Ind. 4.0 (EDI)*, 2021, pp. 1–8.

[25] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 321–331.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[27] 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*.

[28] W. Z. Wang, "Electricity consumption prediction using XGBoost based on discrete wavelet transform," in *Proc. 2nd Int. Conf. Artif. Intell. Eng. Appl. (AIEA)*, 2017, pp. 716–729.

[29] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, "Comparison of ARIMA and random forest time series models for prediction of avian influenza $H_5N_1$ outbreaks," *BMC Bioinf.*, vol. 15, no. 1, pp. 1–9, Dec. 2014.

[30] Á. L. P. Gómez, L. F. Maimó, A. H. Celdrán, and F. J. G. Clemente, "MADICS: A methodology for anomaly detection in industrial control systems," *Symmetry*, vol. 12, no. 10, p. 1583, Sep. 2020.

[31] C. A. Rieth, B. D. Amsel, R. Tran, and M. B. Cook, "Issues and advances in anomaly detection evaluation for joint human-automated systems," in *Proc. Int. Conf. Appl. Hum. Factors Ergonom.* New York, NY, USA: Springer, 2017, pp. 52–63.

[32] M. Onel, C. A. Kieslich, and E. N. Pistikopoulos, "A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process," *AIChE J.*, vol. 65, no. 3, pp. 992–1005, Mar. 2019.


**ÁNGEL LUIS PERALES GÓMEZ** received the M.S. and Ph.D. degrees in computer science from the University of Murcia, Spain. He is currently a Postdoctoral Researcher with the Department of Computer Engineering, University of Murcia. His research interests include deep learning, machine learning, interpretability, robustness, and cybersecurity of industrial control systems.

**LORENZO FERNÁNDEZ MAIMÓ** received the M.Sc. and Ph.D. degrees in computer science from the University of Murcia. He is currently an Associate Professor with the Department of Computer Engineering, University of Murcia. His research interests include machine learning and deep learning applied to cybersecurity and computer vision.

**FÉLIX J. GARCÍA CLEMENTE** received the Ph.D. degree in computer science from the University of Murcia, in 2006. He is currently an Associate Professor with the Department of Computer Engineering, University of Murcia. His research interests include cybersecurity and management of distributed communication networks. He is the coauthor of over 100 scientific publications and an active member of different national and international research projects.

**JAVIER ALEJANDRO MAROTO MORALES** received the dual bachelor's degree in engineering telecommunications and engineering physics from the Universitat Politecnica de Catalunya. He is currently pursuing the Ph.D. degree with EPFL working under the supervision of Pascal Frossard. His research interests include understanding the factors that influence neural networks robustness to adversarial perturbations, with a particular interest in the effect of the data used for training the networks.

**ALBERTO HUERTAS CELDRÁN** (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Murcia, Spain. He is currently a Senior Researcher at the Communication Systems Group CSG, Department of Informatics (IfI), University of Zurich (UZH). His research interests include cybersecurity, machine and deep learning, continuous authentication, and computer networks.

**GÉRÔME BOVET** received the Ph.D. degree in networks and computer systems from Telecom ParisTech, France. He is currently the Head of Data Science for the Swiss Department of Defense. His work focuses on machine and deep learning, with an emphasis on anomaly detection, adversarial, and collaborative learning in IoT sensors.