Constrained Concealment Attacks against Reconstruction-based Anomaly Detectors in Industrial Control Systems

Alessandro Erba[*][†]
alessandro.erba@cispa.saarland
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

Riccardo Taormina
r.taormina@tudelft.nl
Delft University of Technology
Delft, Netherlands

Stefano Galelli
stefano_galelli@sutd.edu.sg
Singapore University of Technology and Design
Singapore

Marcello Pogliani
marcello.pogliani@polimi.it
Politecnico di Milano
Milan, Italy

Michele Carminati
michele.carminati@polimi.it
Politecnico di Milano
Milan, Italy

Stefano Zanero
stefano.zanero@polimi.it
Politecnico di Milano
Milan, Italy

Nils Ole Tippenhauer
tippenhauer@cispa.saarland
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany

ABSTRACT

Recently, reconstruction-based anomaly detection was proposed as an effective technique to detect attacks in dynamic industrial control networks. Unlike classical network anomaly detectors that observe the network traffic, reconstruction-based detectors operate on the measured sensor data, leveraging physical process models learned a priori.

In this work, we investigate different approaches to evade prior-work reconstruction-based anomaly detectors by manipulating sensor data so that the attack is concealed. We find that replay attacks (commonly assumed to be very strong) show bad performance (i.e., increasing the number of alarms) if the attacker is constrained to manipulate less than 95% of all features in the system, as hidden correlations between the features are not replicated well. To address this, we propose two novel attacks that manipulate a subset of the sensor readings, leveraging learned physical constraints of the system. Our attacks feature two different attacker models: A white box attacker, which uses an optimization approach with a detection oracle, and a black box attacker, which uses an autoencoder to translate anomalous data into normal data. We evaluate our implementation on two different datasets from the water distribution domain, showing that the detector's Recall drops from 0.68 to 0.12 by manipulating 4 sensors out of 82 in WADI dataset. In addition, we show that our black box attacks are transferable to different detectors: They work against autoencoder-, LSTM-, and CNN-based detectors. Finally, we implement and demonstrate our attacks on a real industrial testbed to demonstrate their feasibility in real-time.

[*]Also with Saarbrücken Graduate School of Computer Science, Saarland University.
[†]A part of this work was done while Alessandro Erba was student at Politecnico di Milano, visiting SUTD.

CCS CONCEPTS

• Security and privacy → Intrusion detection systems; **• Computer systems organization → Embedded and cyber-physical systems**; **• Computing methodologies → Anomaly detection**; **Neural networks**.

KEYWORDS

Industrial Control System, Intrusion Detection, Deep Learning, Adversarial Machine Learning, Evasion Attack, Classifier Evasion, Mean Squared Error, Autoencoder, Multivariate Time Series

ACM Reference Format:
Alessandro Erba, Riccardo Taormina, Stefano Galelli, Marcello Pogliani, Michele Carminati, Stefano Zanero, and Nils Ole Tippenhauer. 2020. Constrained Concealment Attacks against Reconstruction-based Anomaly Detectors in Industrial Control Systems. In *Annual Computer Security Applications Conference (ACSAC 2020), December 7–11, 2020, Austin, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3427228.3427660

1 INTRODUCTION

Computational and physical infrastructures are nowadays interconnected. Computers, communication networks, sensors, and actuators allow to control physical processes, resulting in what is known as cyber-physical systems (CPS). Examples of such systems are interconnected critical industrial control systems (ICS) like power grids [36], water supply systems [3], and autonomous vehicles [8].

The integration of modern security features into existing ICS is challenging, as solutions have to be backward compatible with decades-old devices in the field, which do not support authentication or encryption. As a result, attackers with the goal of damaging the process and local access to the network are usually assumed to be able to eavesdrop on traffic, send malicious commands to actuators, and spoof sensor values to hide problems in the physical process [17, 27, 59]. Such activities produce anomalies in the physical sensor data that can be successfully leveraged for attack detection. Hence, attackers can attempt to conceal the physical anomalies through replay attacks [39] or through stealthy attacks based on solving models of the (known) physical processes [54, 57]. A promising anomaly detection technique in industrial control systems involves the use of machine learning-based classifiers and, in particular, reconstruction-based classifiers as proposed in [20, 32, 51]. An attacker who wants to conceal the physical anomalies from this detector will modify a sample to induce a wrong classification outcome: This can be framed as an Adversarial Machine Learning (AML) evasion attack. So far, no systematic analysis of evasion attack against reconstruction-based detectors has been proposed.

Evasion attacks in the context of ICS (which we call *Concealment Attacks* to distinguish them from the general case) face novel and specific challenges, which make standard AML techniques [26] not directly applicable. In particular, adversarial examples[1] obtained with a concealment attack must meet four requirements. **R1**: Due to the distributed nature of the system, the attacker is constrained to manipulate only a subset of features. **R2**: Adversarial examples must meet the temporal and spatial correlations expected from the observed physical processes [24, 53]. Particularly, adversarial examples must not introduce contextual anomalies (i.e., observations classified as abnormal only when viewed against other variables that characterize the behavior of the physical process [24]). **R3**: Most AML attacks in other domains target end-to-end Neural Network Classifiers instead of reconstruction-based classifiers. We realized that this requires the attacker to optimize the Mean Squared Error loss instead of optimizing the cross-entropy loss (Section 4.5). **R4**: To the best of our knowledge, previous work either assumes unlimited computational power to compute ideal pertubations [9] or assumes static systems that allow universal adversarial perturbations [35, 41]. For real-time attacks[2] on dynamic ICS, neither approach is feasible, and new solutions are required.

In this work, *we propose and evaluate constrained concealment attacks against reconstruction-based anomaly detectors.* To meet R1, we formalize a detailed attacker model and evaluate different settings relating to attacker constraints, i.e., the number of features under the control of the attacker. To meet R2, the attacker leverages passive observation of system behavior to approximate how realistic examples should behave. Based on that, we consider a white box attacker that can leverage knowledge on the system to perform iterative attacks on general reconstruction-based detectors (meeting R3). Moreover, we consider a black box attacker and show that it is possible to craft effective adversarial samples in milliseconds (meeting R3 and R4). Our implementation meets all four requirements for Reconstruction-based detectors.

We summarize our main contributions as follows:

- We propose a detailed attacker model that formalizes implicit models in prior work, introduce constraints on the attacker motivated by real-world ICS, and provide an AML taxonomy for the attacker.
- We show that replay attacks do not conceal anomalies when the attacker is constrained to manipulate less than 95% of the ICS features, due to physical correlations that are not exploited by such attacks.
- We propose and design concealment attacks on ICS process-based anomaly detectors which produce examples that do not violate correlations (outperforming replay attacks in constrained scenarios). A white box attacker exploits the knowledge of the Anomaly Detection System launching an iterative attack based on coordinate descent algorithm. A black box attacker without the knowledge of the Anomaly Detection System uses learning-based attack leveraging adversarially trained autoencoders[3].
- We evaluate and discuss the proposed attacks, and compare their performance against replay attacks. The evaluation is conducted over a simulated ICS process dataset and a real ICS process dataset, both containing data of water distribution systems[4].
- We practically implement and demonstrate the attacks in real-word Industrial Control System testbed, and show that they are possible in real-time.

The remainder of this work is structured as follows. Background is introduced in Section 2. We present the problem of adversarial learning attacks on ML-based detectors in Section 3. Our design is proposed in Section 4, and its implementation and evaluation is presented in Section 5. We summarize related work in Section 6. The paper is concluded in Section 7.

2 BACKGROUND

In this section, we provide a brief overview on Industrial Control Systems and Evasion Attacks. A review of related work is presented in Section 6.

2.1 Industrial Control Systems

Industrial Control Systems are universally employed to control an industrial process [18]. In an ICS, *Physical* components include the hardware equipment required to execute the process; among these, actuators and sensors represent the junction point between *Cyber* and *Physical* components. *Cyber* components comprise the computer hardware and software that is deployed to execute the plant control logic and monitoring. Typical industrial control hardware contain Programmable Logic Controllers (PLCs) and a Supervisory Control and Data Acquisition (SCADA) system. In an ICS, one or more PLCs implement the process control logic by monitoring sensor values and sending commands to actuators. Sensor values and actuator states are reported from PLC to the SCADA system. In a distributed ICS, several PLCs control the system; taking control of a sub-process governed by a single PLC can disrupt the system. Attacks targeting both the cyber and physical components of industrial processes have occurred during the past decades. A notable example is Stuxnet [59], an attack that targeted the physical part of an ICS to reduce rotation frequencies of nuclear

[1]We differentiate between *sample* (original set of sensor readings), and *adversarial example* (manipulated set of sensor readings).

[2]With real-time, we mean examples are crafted w.r.t. the current dynamic state of the system, in less time than the sampling rate (e.g., 10ms).

[3]Note: Not to be confused with Adversarial Autoencoders [38].

[4]Implementation available at https://github.com/scy-phy/ICS-Evasion-Attacks

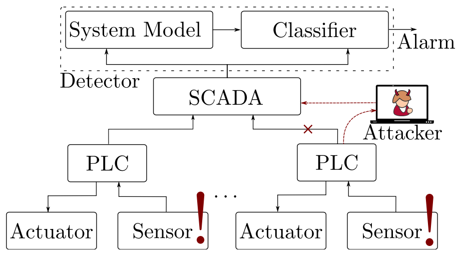


Figure 1: High level system and attacker model. The PLCs report sensor data about the anomalous process to the SCADA. The attacker can eavesdrop and manipulate a subset of the data provided to the SCADA. The reconstruction-based detector attempts to detect attacks based on a learned model of the system's benign operations.

centrifuges, and the cyber component to spoof reported sensor readings (via a man-in-the-middle attack) thus avoiding anomaly detection.

2.2 Evasion Attacks

In AML, an evasion attack is launched by an adversary to control the output behavior of a machine learning model through crafted inputs i.e., *adversarial examples*. Several evasion attack and defense mechanisms have been proposed in the context of image processing [50], speech recognition [11] and malware detection [22].

The authors of [6] characterize attacks on machine learning models using a 4-tuple representation of the attackers' knowledge of the system under attack, the training dataset $\mathcal{D}$, the feature set $\mathcal{X}$, the learning algorithm f, and the trained parameters w. In an adversarial setting, an attacker has complete or partial knowledge of components; partial knowledge of a component is denoted with the symbols $\hat{\mathcal{D}}$, $\hat{\mathcal{X}}$, $\hat{f}$ and $\hat{w}$ respectively. In particular, the authors characterize three types of attack scenarios: Perfect-knowledge white box attackers $(\mathcal{D}, \mathcal{X}, f, w)$; Limited-knowledge gray box attacks $(\hat{\mathcal{D}}, \mathcal{X}, f, \hat{w})$; Zero-knowledge black box attacks $(\hat{\mathcal{D}}, \hat{\mathcal{X}}, \hat{f}, \hat{w})$. Attacks are achieved by solving an optimization problem that minimizes distance between the sample and the adversarial example e.g. by minimizing norms: L_0, L_2, L_∞. In Section 3, we use this notation to introduce our proposed solution and position it within the related literature.

3 CONCEALMENT ATTACKS ON RECONSTRUCTION-BASED ANOMALY DETECTION

In this section, we introduce our system and attacker model, and our general problem statement for constrained concealment attacks. Then, we present our abstract white and black box attacker model.

3.1 System Model

We consider a system under attack (Figure 1) consisting of several sensors and actuators connected to one or more PLCs, which are in turn connected to a SCADA system that gathers data from the PLCs. In our work, we assume that the SCADA is passive, so it does not send control commands to the PLCs (e.g., to actively probe for manipulations), or uses steganographic approaches to authenticate sensor readings [40]. The SCADA feeds an attack detection system, whose goal is to accurately identify the instances in which the attacker manipulates the physical process while minimizing the number of false detections. The attack detection system generally consists of two main components: a *system model*, which is used to generate additional features, and a *classifier*, which (for each time step) classifies the system as either under attack or under normal operating conditions (see Section 6 for more details on classifiers in this context). During the attack, the physical process may be in an anomalous state, which will be detected unless the attacker manages to conceal it. The anomalies themselves are out of the scope of this work; we use prior-work datasets [28, 53].

To the best of our knowledge, our work is the first one that enables the use of constraints on the number of sensors that can be manipulated by an attacker (see Section 3.2). As we will show, the performance of the attack degrades when lowering the number of channels that are under the attacker's control. Fully authenticated sensor signals would eventually prevent the attack to occur at the process level, but would impose cost on the normal system operations. Since our attacks exploit sensor signals received by the detector, they can be deployed somewhere else w.r.t. the industrial plant. Software exploits on the machine running the detector or historian server could also offer the attack surface to mount our proposed concealment attacks (those alternative attacker models are not modeled here for the sake of simplicity).

3.2 Attacker Model

3.2.1 Attacker Goal and Capabilities. The goal of the attacker is to launch a concealment attack on an ICS to hide the real state of the process from an anomaly detector. The modeled attacker is assumed to have access to the ICS network, e.g., by physically attaching malicious devices to the network, intercepting communications to selected remote substations, or performing a Man-in-the-PLC [17] attack. The attacker is thus assumed to control a subset of the communication between PLCs and the SCADA system, and as a result, able to eavesdrop on traffic and send manipulated sensor readings to the detector. PLC communication with the SCADA system can be exploited to hide the real state of the system as practically demonstrated in [17]. In contrast to [17], our attacker does not require explicit knowledge of the physical model equations to conceal the anomalies.

In particular, we assume that the anomalous physical process results in a feature vector $\vec{x}$, which triggers the detection system. The attacker thus needs to find an alternative vector $\vec{x}'$, which prevents detection of the attack. We formalize the *concelament attack* as follows: given a feature vector $\vec{x}$ and a classification function $y()$ s.t. the detector correctly classifies $y(\vec{x})$ = 'under attack', the attacker is looking for a perturbation $\vec{x}+\vec{\delta}$ s.t. $y(\vec{x}+\vec{\delta})$ ='safe'. Since the attacker's goal is to evade Mean Squared Error-based classifiers, the adversarial attack has to find a perturbation $\vec{\delta}$ to minimize the reconstruction error between the input $\vec{x}+\delta$ and output $\hat{\vec{x}}$ of the Reconstruction-based detector. Please refer to Section 4.1 for further details on the target model. In a mathematical notation, it can be written as the following constrained optimization problem in Equation 1:

Table 1: Classification of our attacker models based on training data and features, Data tuple $(\mathcal{D}, \mathcal{X})$ and algorithm knowledge and parameters. Access: ● =full, ◐ =partial. For all white box attacks, the Defense tuple is (f, w), for all black box attacks, ($\cancel{f}$, $\cancel{w}$).

Attacker's Constraints		$\mathcal{D}$	$\mathcal{X}$ Read	$\mathcal{X}$ Write
Unconstrained	§ 5.4	●	●	●
$\mathcal{X}$ Partially	§ 5.5	●	●	◐
$\mathcal{X}$ Fully	§ 5.5	●	◐	◐
$\mathcal{D}$	§ 5.5	◐	●	●

$$\begin{aligned} \text{minimize} \quad & MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{x_i} - (x_i + \delta_i))^2 \\ \text{s.t.} \quad & \vec{\delta} \in \text{constraint space (Section 3.2.2)} \qquad (1) \\ & \text{real-time constraints imposed by CPS} \\ & y(\vec{x} + \vec{\delta}) = \text{`safe'} \end{aligned}$$

We note that the attacks we demonstrate are not necessarily optimal, as the constraints are satisfied with non unique solutions.The attacks are conducted in real-time (i.e., in milliseconds per time step), not *a posteriori* (i.e., applied retrospectively to a longer sequence of sensor readings after the attacker fully receives them).

3.2.2 Attacker Knowledge. Using the adversarial learning notation introduced in Section 2, a concealment attack is characterized by the knowledge of the attacker about the training dataset $\mathcal{D}$, feature set $\mathcal{X}$, learning algorithm f, and trained parameters w. In the ICS setting, the attacker can be characterized differently according to his knowledge of the attacked system. In order to explain our attacker model, we split the tuple $(\mathcal{D}, \mathcal{X}, f, w)$ into two: the Data tuple $(\mathcal{D}, \mathcal{X})$ and Defense tuple (f, w). We assume the attacker to be *unconstrained* or *constrained* w.r.t. the Data tuple, i.e., the sensor readings $\mathcal{X}$ that she can observe and manipulate and the data $\mathcal{D}$ that she eavesdrops. Moreover, we classify attackers as *white box*, *black box*, w.r.t. the Defense tuple, i.e., the knowledge of learning algorithm f, and trained parameters w. Table 1, provides an overview of the attacker's constraints considered in this work.

Constraints over Data Tuple. According to the Data tuple $(\mathcal{D}, \mathcal{X})$, we classify the attacker as:

- *Unconstrained* $(\mathcal{D}, \mathcal{X})$, in which the attacker can manipulate all n features in $\vec{x}$, and her perturbations are limited in terms of L_0 distance to be at most n.
- *Features Partially Constrained* $(\mathcal{D}, \hat{\mathcal{X}})$, we assume that the attacker is constrained to perturb a subset of k out of n variables in $\vec{x}$, and her perturbations are limited in terms of L_0 distance to not exceed distance k.
- *Features Fully-Constrained* $(\mathcal{D}, \hat{\mathcal{X}})$, we assume that the attacker is constrained to observe and perturb a subset of k out of n variables in $\vec{x}$, and her perturbations are limited in terms of L_0 distance to not exceed distance k.
- *Data Constrained* $(\hat{\mathcal{D}}, \mathcal{X})$, we assume that the attacker is constrained to eavesdrop a limited quantity of process data that are used for training its attacks.

Selection of Constrained Features The subset of features that can be modified is highly use-case dependent (i.e., which link is attacked, which device was compromised). To demonstrate the generality of our findings, we explored two types of constraints: a best-case scenario and a topology-based scenario.

For the best-case scenario, we assume that the selection of the k out of n manipulated features can be made by the attacker to maximize the attack impact. This arguably represents a best-case scenario for the constrained attacker (i.e., an attacker constrained to features that happen to be relatively ideal for the attacker). For the second scenario, constraints are derived from the network topology. We assume that the attacker can compromise a single substation (or PLC) in the network, and the selection of k out of n features is based on which sensors are interconnected to the compromised substation.

Knowledge of Defense Tuple. We classify the attacker according to their knowledge of the Defense tuple (f, w), as:

- *White box* (f, w), the attacker knows the exact system model and its variables (such as the currently estimated system state), and the exact thresholds of the classification system. Thus, the white box attacker is characterized by the knowledge of (f, w). With that information, the attacker could either run a basic exhaustive search, basic optimization strategies, or more sophisticated approaches (especially solutions that use the gradient signal from the attacked model).
- *Black box* ($\cancel{f}$, $\cancel{w}$), the attacker is aware of the general detection scheme, but unaware of internal variables, architecture and exact thresholds used in the classification. We note that our black box attacker is different from the one defined in [6], $(\hat{f}, \hat{w})$. Our attacker does not require the knowledge of f or its approximation $\hat{f}$. In our case, the nature of the environment imposes that the attacker cannot query the system even in a black box manner to get feedback on the provided labels or confidence scores (this is done for example in [12, 14, 56, 61]), as this would mean potentially raising the alarm. Thus, we consider that the only assumption of the attacker concerning f is that Deep Learning techniques are used for detection.

Given this taxonomy, the attacker can be classified for example, as *unconstrained white box*.

3.3 Example Constraint Scenarios

We argue our *Constrained* and *Unconstrained* attacks represent a realistic threat model in the ICS setting and fit the taxonomy of attacks in the AML literature. In particular, practical ICS are typically composed of multiple stages, and each stage is controlled by a different PLC (i.e., different brands/models). Moreover, the ICS can be deployed in a physically distributed manner. For example, in the case of water distribution networks, pumping stations are typically located kilometers away from the water reservoir. In this heterogeneous setting, an attacker can either gain control over a limited set of resources as practically demonstrated in [17], or the whole plant (by compromising the network or central SCADA). We capture those different capabilities of the attacker in the following three scenarios:

- The *Unconstrained Attacker* can read and write any features arbitrarily, e.g., by compromising the SCADA system, as modeled by Mo et al. [39].
- The *Partially Constrained attacker* can read all traffic received by the SCADA system (for example by a passive wiretap [55] or by leveraging access control misconfigurations), but she is only able to spoof sensor readings from a specific substation, exploiting specific vulnerabilities of the substation or its protocol to the SCADA (e.g., lack of authentication).
- The *Fully Constrained attacker* relates to a scenario where an attacker compromised a specific substation, giving him read and write access to features from this substation only [1, 17].

Similar assumptions were also considered in the BATADAL dataset [53], where the attacker was assumed to perform constrained replay attacks to reduce the confidence of anomaly detectors. In our contribution, we perform a concealment attack in a systematic way to assess their impact over the anomaly detector.

A similar intuition is also used in the FAIL attacker model [49] for AML. In particular, FAIL proposes to characterize attacker knowledge over 4 dimension: **F**eature, **A**lgorithm, **I**nstance, **L**everage. While the first three have a counterpart in systematization by Biggio et al. [6], **L**everage stands for the subset of features that the adversary can modify (just like our constrained attacker). Thus, our *Unconstrained* and *Constrained* attacks represent attacks with full and limited **L**everage.

3.4 Our Framework for Attack Computation

For both the white box and black box case, the attacker is assumed to intercept and manipulate a Constrained or Unconstrained set of sensor readings in real-time.

In the white box setting, we propose an iterative attack, able to interactively query a classification oracle to determine which features to manipulate, and the value to assign to those features. We propose to compute the manipulations using an iterative algorithm. This algorithm calculates solutions that are 'safe' from the detector perspective. The algorithm is tunable, i.e., the attacker can act on some algorithm parameters that impact over time the computation and, consequently, the concealment efficacy. Again, this speeds up computation but can impact the solution quality.

In the black box setting, we propose the use of a learning based attack, specifically a Deep Neural Network that is capable of outputting concealed sensor readings, without the oracle's feedback. The attacker is adversarially training the neural network to learn how the detector expects the ICS to behave. This trained neural network then receives the traffic coming from the PLC. While the attacker creates an anomaly over the physical process, the neural network adjusts the anomalous data to resemble 'safe' data. This manipulated version of sensor data is sent to the SCADA.

In order to avoid confusion, we point out explicitly that our iterative attack can operate under the attacker white box assumption as it requires to query an oracle of the anomaly detector, while the learning based attack can operate under the black box assumption as no query access is required to compute the adversarial sensor readings. We compare iterative and learning based approaches with replay attacks, as proposed in literature [39]. The attacker that performs a replay attack can be categorized as black box.

4 DESIGN OF CONCEALMENT ATTACKS

We now present a detailed design for the three attacks that we consider. We start with details on the (prior work) Reconstruction-based attack detector, then introduce the replay attack (proposed by [39]). We provide details on the iterative attack (white box knowledge). We then conclude with the learning based approach (black box knowledge), which leverages an online concealment method without any prior knowledge about the physical process that generates the sensor readings and the detection scheme. Given these premises, we note that, while adversarial examples found using the iterative approach depend on the internal structure of the attacked anomaly detector, examples crafted through the learning based approach are independent from the addressed detection scheme (see Section 5.6).

4.1 Background: Reconstruction-based Attack Detector

In this work, we target anomaly detection systems proposed in prior works [20, 32, 51], which share the same underlying idea, reconstruction-based anomaly detection. The anomaly detector consists of two parts, namely a Deep Learning autoencoder model (with $m \times n$ features as input and n output) trained over the normal operation sensor readings of an ICS to optimize Mean Squared Error Loss, and a classifier function. The idea is that the deep model has learned to reproduce the system behavior under normal operating conditions with a low reconstruction error, so it reproduces a higher reconstruction error when fed with anomalous sensor readings. The comparison between the input and output of the deep model is used to decide if the system is 'safe' or 'under attack'. Reconstruction based classifiers represent the state of the art for anomaly detection in ICS on a multi-fold basis. First, they can overcome the problem of shortage of 'under attack' samples that are hard to be gathered from the system without damaging the plant. Second, they can capture interdependence between sensor signals that helps localization of anomalies. Third, they guarantee a low time of detection, which is a fundamental property for ICS anomaly detectors.

As reference implementation, we use the general Autoencoder-based anomaly detector framework proposed in [51] and available as open source [46]. Moreover, we explore transferability of our black box attack between different Deep Architectures (DA) in Section 5. In particular we tested Long Short Term Memory (LSTM) [25] as proposed in [20] and Convolutional Neural Networks (CNN) [33] as proposed in [32]. The input to the DA is $X = [\vec{s_{t-m}}, \dots, \vec{s_t}]$, representing $m + 1$ time-steps of $\vec{s} = [r_1, r_2, \dots, r_n]$, which is an n-dimensional vector of sensor readings. DA's goal is to find ϕ parameters that minimize following the Mean-Squared Error optimization problem:

$$\min_{\phi} \mathbb{E}_{(X, \vec{s_t}) \sim \mathcal{D}} \left[\| DA(\phi, X), \vec{s_t} \|_2^2 \right] \tag{2}$$

where DA outputs an n-dimensional vector $\vec{o} = [v_1, v_2, ..., v_n]$, and $v_i \; s.t. \; i \in \{1, ..., n\}$ represents the reconstructed value w.r.t. the input reading r_i^t . In order to decide if the system is under attack, the mean squared reconstruction error between observed and predicted features are computed. If the mean squared reconstruction error exceeds a threshold θ, the system is classified as under attack. The authors of [51] chose θ as 99.5 percentile (Q99.5) of the average reconstruction error over the training set.

We formalize this as follows. Given a target vector $\vec{s_t}$, we define: $\vec{e} = \vec{s_t} - \vec{o} = [d_1, \ldots, d_n]$ as the reconstruction error n-dimensional vector, $\varepsilon(\vec{e})$ as the corresponding average reconstruction error:

$$\varepsilon(\vec{e}) = \frac{1}{n} \sum_{i=1}^{n} {d_i}^2, \tag{3}$$

and $y(X)$ as the classified state of the water distribution system out of Reconstruction-based Intrusion Detection System. Given an input X, y is 'under attack' if ε greater than θ:

$$y(X) = \begin{cases} \text{'under attack' if } \varepsilon(\vec{e}) > \theta \\ \text{'safe' otherwise} \end{cases} \tag{4}$$

Moreover, the authors propose a *window* parameter that takes into consideration the mean of $\varepsilon(\vec{e})$ of the last *window* time steps to decide if the current tuple is 'safe'. This helps diminishing the amount of false positives, since an alarm is raised only if in the last *window* time steps the mean of $\varepsilon(\vec{e})$ is above θ.

4.2 Baseline: Replay Attack

In the replay attack setting (prior work, used here as baseline), the attacker does not know how detection is performed. In order to avoid detection, the attacker can replay sensor readings that have been recorded while no anomalies were occurring in the system. In particular, we assume that the attacker could record selected data occurring exactly n days before—i.e., if the concealment attack starts at 10 a.m., the attacker starts replaying data from 10 a.m. one day before.

4.3 Iterative Attack

In the iterative attack, the white box attacker knows how detection is performed, all thresholds and parameters of the detector, as well as the normal operation range for each one of the model features. For example, the attacker knows which sensor readings are common during normal operation of the physical process. As a result, the attacker essentially has access to an *oracle* of the Deep Architecture, where the attacker can provide arbitrary $\vec{x}$ features and gets the individual values of the reconstruction error vector $\vec{e}$. The attacker then computes $\max_i \vec{e}$ and finds the sensor reading r_i with the highest reconstruction error from $\vec{x}$.

In order to satisfy the constraint $\varepsilon(\vec{e'}) < \theta$ (i.e. $y(\vec{x}+\vec{\delta})$ = 'safe' in Equation 1), the attacker performs a coordinate descent algorithm to decrease the reconstruction error related to r_i (As we rely on coordinate descent algorithm we do not use gradient estimation methods). At each iteration of the algorithm, a coordinate of the feature vector is modified until a solution is found or the computational budgets are exceeded.

Two computational budgets are put in place: *patience* and *budget*. If no lower reconstruction error is found by descending a coordinate, the algorithm tries descending other coordinates. If no improved solutions are found in *patience* iterations, the input is no longer optimized. *budget* is the maximum number of iterations for coordinate descent. After *budget* attempts without satisfying $\varepsilon(\vec{e'}) < \theta$, the input is no longer optimized, and no solution is found. Additional details are found in Appendix A.

Sensor readings r_i are modified in the range of normal operating values; this guides the computation to a solution that is consistent with the physical process learned by the detector. For example, if normal operations of sensor r_i are in the range $[0, 5]$, the attacker tries to substitute the corresponding value of r_i according to its range to see if the related reconstruction error decreases. The algorithm 1 can be found in Appendix A as a reference.

4.4 Learning Based attack

In the learning based setting, the black box attacker does not know anything about the detection mechanism except the fact that it relies on a Reconstruction-based model: the attacker can only intercept and manipulate the communication between the PLCs and SCADA. However, the nature of the ICS environment allows us to assume that a detection mechanism trained over a specific CPS should represent its physical rules in order to spot anomalies.

In this case, a reasonable attack scheme can be divided into five steps. The attacker first intercepts traffic from PLCs to SCADA in order to collect information on how the ICS behaves under normal conditions. Second, collected data are used to learn how the system behaves normally and train a Deep Learning model, implicitly training a model to solve the minimization problem in Equation 1. Third, the attacker manipulates the physical process; anomalous data are generated as a consequence. Fourth, the adversarially trained model is used to conceal anomalous readings by morphing them into concealed data that will be classified as 'safe'. Fifth, the concealed data is forwarded to the SCADA.

Autoencoder-based Generator. We implement the learning based attack using an autoencoder network to generate concealed data (the word generator is used with a different meaning than the usual one. In our case the input is not random noise that is going to be crafted by the network). The autoencoder is trained while intercepting normal traffic; the network learns to output tuples that are classified as being normal with high confidence. Forwarding the output of the adversarial network—regardless of how the detector is built—forces it to misprediction, because the adversarial examples have been adjusted to resemble normal operations. Note that the autoencoder does not completely change the tuple; intuitively, only the part that contains the anomaly is reconstructed to match the learned physical behavior. We explored different architectures, activation, and loss functions for the autoencoder network. Experimentally, we verified that by using hidden layers to increase the data dimensionality (i.e., with more units than the input and output layers), rather than decreasing it, we achieved a higher concealment capacity than canonical 'compressing' models. Particularly, we implemented an autoencoder network with three hidden layers, with input and output dimensions equal to the number of sensors and actuators in the network. We used mean squared error as a loss function and sigmoid as an activation function. To train the network, we use the ADAM [31] optimizer with a learning rate set to 0.001.

Post-processing. In order to generate feasible inputs for the anomaly detector, we need to consider that not all the sensor readings assume continuous values—some are categorical integers that represent the status of actuators. Since the output of a neural network is continuous, we need to post-process all the readings that are supposed to be integers. For example, if a pump status assumes a value 0 when it is turned *off* and 1 when it is turned *on*, post-processing approximates the corresponding output value to the nearest allowed integer. According to this post-processing, some other values should be adjusted in order to match the physical rules. This is the case, for example, for speed sensors that must read 0 if their related pump is *off*.

4.5 Positioning with respect to State-of-The-Art AML attacks

In this work, we consider attacks on a (prior work) reconstruction based classifier. This represents a substantial difference from classifiers considered in AML attacks in other domains. In related work [10, 37, 43], attacks on end-to-end Neural Network classifiers are considered (i.e., those classifiers are trained to optimize the cross-entropy loss). In particular, target misclassification is achieved, diminishing the predicted probability of the true class in the output layer. Our problem setting is different: To evade the classifier, we need to diminish the residual between input and output. Our target Neural Networks are trained to optimize the Mean Squared Error loss. This can be achieved by reconstructing the sensor signal in a way that matches the learned physical properties of the ICS. Those differences motivated our novel white-box and black box approaches to evade Deep Learning-based Anomaly Detectors in ICS.

Model Robustness. Adversarial Robustness [37] is achieved by a) adversarial training b) classifier capacity. Following the definition, adversarial training is obtained embedding adversarial examples in the training set. In our context, the Neural Network is trained to approximate the system behavior during normal operating conditions, with no samples for the 'under attack' class. Thus, adversarial training here does not apply: indeed, we cannot train the system to be resilient to adversarial attacks since samples from the class 'under attack' is unknown to the defender.

5 EVALUATION

In this section, we experimentally evaluate the proposed attacks. We start by introducing the datasets we used for our experiments: the BATADAL dataset and data coming from a real industrial process (WADI dataset). We start the evaluation targeting an Autoencoder-based detector and explore the performance of replay, iterative, and learning based attacks in constrained and unconstrained conditions. Then, we show that our learning based attack generalizes to other schemes based on LSTM and CNN. Finally, we show the concealment attack results obtained in a real industrial testbed.

5.1 Dataset 1: BATADAL

The first dataset was generated with epanetCPA [52], an open-source, object-oriented Matlab toolbox for modeling the hydraulic response of water distribution systems to cyber-physical attacks. The dataset was originally generated for the BATADAL [53] competition, which ran between 2016 and 2017. The BATADAL competition was based on three datasets: the first contains data coming from the simulation of 365 days of normal operations, while the second and third contains 14 attacks (7 attacks each). The details of the attacks can be found in [53]. These datasets contain readings from 43 sensors: tank water levels (7 variables), inlet and outlet pressure for one actuated valve and all pumping stations (12 variables), as well as their flow and status (24 variables). All variables are continuous, except for the status of valve and pumps, represented by binary variables.

The original attack dataset (from http://www.batadal.net/data.html) contained sensor data readings that were manually concealed. For that reason, we could not use the original attack dataset directly (as we wanted to add concealment ourselves). Instead, we re-created the attacks (and resulting sensor data) from the BATADAL dataset for this work using the original setup, without any manual concealment. In our new version, the data are collected from sensors every 15 minutes.

5.2 Dataset 2: WADI

Our second dataset is based on the Water Distribution (WADI) testbed, a real-world ICS testbed located at the Singapore University of Technology and Design [2]. It is composed of two elevated reservoir tanks, six consumer tanks, two raw water tanks, and a return tank. It contains chemical dosing systems, booster pumps and valves, instrumentation, and analyzers. WADI is controlled by 3 PLCs that operate over 103 network sensors. Moreover, the testbed is equipped with a SCADA system. WADI consists of three main processes: P1 (Primary supply and analysis), P2 (Elevated reservoir with Domestic grid and leak detection), and P3 (Return process). For anomaly detection purposes, we consider sensor data from P1 and P2, since the return process is only implemented for recycling water. Considering stages P1 and P2, we have data coming every second from 82 sensors. In this work, we use two WADI datasets. The first dataset contains data of 14 days of normal operations. The second contains 15 attacks on physical processes spanned over two days of operations. This dataset is available on request [28].

We primarily use the WADI dataset for two reasons: i) to show that the discussed detection mechanism applies to real-world ICS data, and ii) to see whether our attack methodology is transferable from a scenario in which simulated data are used to another scenario in which real data are used.

5.3 Evaluation Setup

We evaluate the detection Recall over datasets under original conditions (i.e., no concealment attacks), replay, iterative, and learning-based concealment attacks. The *Recall* (True Positive Rate) is defined as:

$$Recall = \frac{TP}{TP + FN}, \tag{5}$$

where TP stands for True Positive and FN for False Negative. Recall measures the rate of correctly classified positive instances. When the anomaly detector is tuned, a higher Recall means that the anomaly detector is correctly retrieving anomalies. The attacker's concealment goal can be expressed in terms of $Recall$: the concealment attack is successful if the detector Recall over the concealed tuples goes to 0. The closer the Recall comes to 0, the higher the amount of misclassified tuples. Note that we launch our concealment attacks over the instances of anomalous data, i.e., data reporting ground truth 'under attack'.

Both iterative and learning based attacks are implemented using Python 3.7.1; neural networks are implemented and trained using Keras 2.3.1 with TensorFlow 1.11.0 backend. Experiments use a laptop equipped with Intel i7-7500U CPU, 16GB of RAM, and NVIDIA GeForce 940MX GPU 4GB.

Training of Attack Detector. For both BATADAL and WADI, we trained the third party attack detector [51] on sensor readings occurring during normal operational data.

For the BATADAL dataset (where sensor readings are sampled every 15 minutes), we found that parameter $window = 3$ quarter of hours is a reasonable decision boundary to flag correctly attacks and do not raise False alarms. This gives a $Accuracy = 0.93$, $Precision = 0.90$, $Recall = 0.60$, $FPR = 0.01$. Changing the *window* parameter, we can increase the *Recall* at the price of decreasing *Precision* that means raising a higher number of False Alarms.

For the WADI dataset, we found that parameter $window = 60$ seconds is a reasonable decision boundary to flag correctly attacks and distinguish them from False Positives. This gives a $Accuracy = 0.97$, $Precision = 0.77$, $Recall = 0.68$, $FPR = 0.01$.

Results are in line with the current state of the art detection over the BATADAL dataset [51] and WADI [16].

Replay attack. In this attack, the attacker replays for the whole duration of the physical manipulation, using the sensor readings as recorded at the same hour s days before (assuming that process operations are often periodic within 24h). s is chosen to let the replay contain only normal operations data. For example, given a physical manipulation that lasts 50 hours, we replay sensor readings as happened 72 hours earlier.

Iterative attack. The attacker manipulates variables required to find a solution (according to the two stopping criteria introduced in Section 4.3 and constraints over modifiable sensor readings). For BATADAL dataset, we tuned the two stopping criteria via grid search to guarantee a trade-off between the decrease of detection accuracy and computational time. Specifically we selected $patience = 15$ and the $budget = 200$. For WADI dataset, the iterative parameters (following the same rational as in BATADAL case) we choose are $patience = 40$ and the $budget = 300$. The result of this experiment depends on the detection mechanism. The attacker is using the oracle to determine if the concealment is successful.

Learning based attack. For the learning based attack, the attacker uses an autoencoder (*AE*) as the generator and sends predicted readings to the SCADA. According to the attacker's constraints, we train an autoencoder over the readable features. We used sigmoid as activation function, Gorlot initialization [19] as weights initializer and mean squared error as loss function. Moreover, we split the data in train $\frac{2}{3}$ and validation $\frac{1}{3}$, use early stopping [29] to avoid overfitting and reduce learning rate on plateaus [30]. Depending on the constrained scenario (i.e., the features that the attacker can read $\mathcal{X}$ or the amount of data that she spoofed $\hat{\mathcal{D}}$), the adversarially trained autoencoder has a different number of input neurons. Given n as input/output dimension, the autoencoder is composed of 3 hidden-layers with respectively $2n$, $4n$, $2n$ neurons. To perform and evaluate the learning based attacks, we trained 83 models with BATADAL data and 63 for WADI. In the unconstrained case, for the BATADAL dataset (43 variables), we train an autoencoder with 64 and 128 units for the first/third and second hidden layers, respectively, training requires 18 epochs (2 seconds/epoch), for a total of 36 seconds to train the model; for the WADI dataset (82 variables), we use 128 and 256 units, training required 7 epochs (64 seconds/epoch), for a total of 488 seconds to train the model.

Table 2: Detector Recall (BATADAL (B) and WADI (W) datasets), before and after unconstrained concealment attacks. The column 'Original' refers to the detection Recall over the data without concealment; 'Replay', reports the Recall after replay attack, while 'Iterative' and 'Learning based' columns report the Recall after our proposed adversarial concealment attacks.

	Detection Recall			
Data	Original	Replay	Iterative	Learning based
B	0.60	0	0.14	0.14
W	0.68	0.07	0.07	0.31

Table 3: Average required time (in seconds) to manipulate sensor readings. 'Replay' column is empty as replay attacks do not require computation. 'Iterative' and 'Learning based' columns report the mean and std deviation required to compute the manipulation sensor readings at a given time step.

	Computational time, mean($\mu_{\bar{x}}$) and std($\sigma_{\bar{x}}$)				
	Replay	Iterative		Learning based	
Data		$\mu_{\bar{x}}$[s]	$\sigma_{\bar{x}}$	$\mu_{\bar{x}}$[s]	$\sigma_{\bar{x}}$
B	-	2.28	2.46	0.002	0.005
W	-	0.60	0.41	0.005	0.002

5.4 Unconstrained Concealment Attack

In this experiments, we assume the Unconstrained attacker $(\mathcal{D}, \mathcal{X})$ that is able to read and control all the reported sensor readings, in the White box (f, w) and Black Box($\cancel{\hat{f}}$, $\cancel{\hat{w}}$) scenarios. We discuss the results of our evaluation of the detector for both datasets in several scenarios (see Table 2). We evaluated the performance of our concealment attacks over the time steps with ground truth 'under attack' labels only, i.e., we exclude normal operation data time steps from the computation of Recall for this attack evaluation.

The first row of Table 2 reports the average results obtained with the three different attack strategies. In this setting, the replay attack is giving 0 Recall over the replayed sensor readings. This means that when the attacker can manipulate all the sensor readings, the anomaly detector is no more able to spot the attack occurring over the physical process. Considering the iterative and learning based approaches, we notice that the Recall is 0.14, this represents a significant drop in detector performance, but not as effective as the replay of all sensors.

The second row of Table 2 refers to concealment attacks over the WADI dataset. The result over this dataset shows that the replay attack can hide the anomaly occurring over the CPS. The performance of the iterative equals the one of the replay attack. Finally, the learning based approach is underperforming the other methods. Despite this, the detector's Recall reduces more than 50% after learning based manipulation.

Computational Time. Table 3 reports the average time required to compute the adversarial examples. In contrast to iterative and learning based, the replay attack does not require computation. The iterative approach requires an amount of time that depends on the algorithm computational budgets. The black box approach requires a constant amount of time since it consists of a neural network prediction. Given our real-time constraints of adversarial examples computation (i.e., target time within milliseconds), we can conclude that learning based approach easily meets the requirements. In the BATADAL case (where the sampling time is 15 minutes), we do not require more than $2ms$ on average to compute an adversarial example. In the WADI case (where sampling time is 1 second), on average, we do not require more than $5ms$ to compute an adversarial example. The iterative attack is slower, but on average, still below the sampling intervals.

Summary of Unconstrained Attacks findings. When the attacker is free to manipulate all the sensor readings, results show that replay attacks hide anomalies occurring over the physical process. First, a replay attack does not require computation to find the manipulated set of sensor readings; second, the attacker does not need to be aware of the detection mechanism; and third, the considered anomaly detector Recall goes to zero since the replayed sensor readings do not contain (additional) anomalies. White box, even though it achieves valuable results, requires computation, and the attacker needs to be omniscient w.r.t. the defense mechanism. We note that the learning based attack can decrease the detector's Recall without having access to detector's oracle, with low computational effort (after training) and the same knowledge w.r.t. the attacked model as the replay attack.

5.5 Constrained Concealment Attack

In the previous subsection, we found that full replay attacks can be a powerful and low-cost way to evade anomaly detectors if all features can be replayed. In this section, we demonstrate the impact of constraints on the attacker, e.g., if the attacker can only control a subset of the reported sensor values. Specifically, we perform Partially, Fully, and Data constrained attacks as modeled in Section 3.2 and show how our proposed iterative and learning based outperform replay attacks.

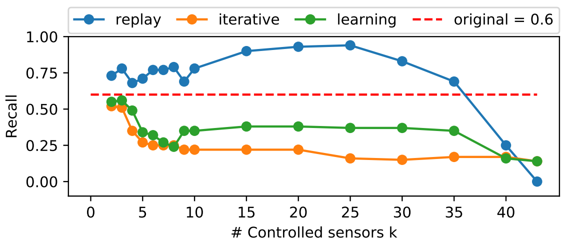
Partially Feature-Constrained attack, $(\mathcal{D}, \hat{\mathcal{X}})$. Figure 2 reports the average result of the constrained attacks over BATADAL and WADI datasets with an *best-case* selection of constraints. Due to space limitations, the constraint selection can be found in Appendix B. In the case of the BATADAL dataset, we note that the replay attack does not cope well with constraints. Since the anomaly detector can spot the presence of contextual anomalies, the replay of only k features results in alarms, with an average detection Recall higher than in the benign case (i.e., no replay of sensors applied), the value decreases when 40 out of 43 sensors are replayed. In the case of iterative and learning based attacks, we can notice that the detection Recall is always lower than the original Recall. In the iterative case, Recall decreases with the number of features that can be modified. Learning-based attack Recall is not monotonically decreasing with the number of features that can be modified. Specific constraint sets better match the physical rules learned by the detector and allow the creation of more effective adversarial examples. In the case of WADI, we can observe that the replay attack can diminish the detector's Recall, especially when the attacker manipulates 3 or more features. The iterative based attack can achieve the same Recall as if in the Unconstrained Attack case when manipulating 15 out of 82 features. In the case of learning based attack, results show that for 3 manipulated (best-case) features, the attack performs slightly better than in the unconstrained case.

In the case of *topology-based* constraints, the attacker controls the sensors connected to 1 PLC in the network. We found that in the BATADAL case, Recall is reduced to 0.36 with the replay attack and 0.34 with the iterative and learning-based attack. In the WADI case, Recall increases to 0.64 with the replay attack while it is reduced to 0.12 in the iterative attack and 0.36 in the learning-based attack. In addition, in this case the iterative and learning-based approach overcomes the limitations of constrained replay attacks, especially in the case of the WADI dataset (Table 7 in the Appendix reports the numerical Recall scores found in the different constrained settings).

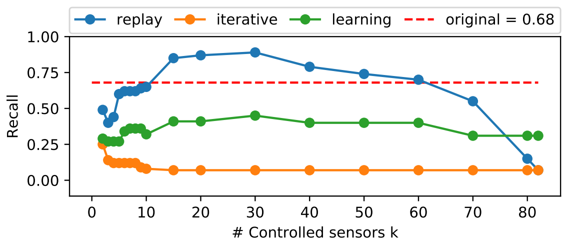
Fully Feature-Constrained Attacker, $(\mathcal{D}, \hat{\mathcal{X}})$. In the case of the fully constrained attacker, Replay and Iterative attack approach do not change, since those two methods do not exploit correlations among features to output the perturbations. The learning-based attack is the only one affected by these constraints, i.e., the adversarial network is trained on the constrained set of sensor readings. We launched this attack with the topology based constraints, and we found that also in this case, when the attacker gains control of a PLC over the network, the detection Recall can be compromised by the attacker. In the BATADAL case, Recall is reduced to 0.39. In the WADI case is reduced to 0.45.

Data Constrained Attacker, $(\hat{\mathcal{D}}, \mathcal{X})$. We also investigated the impact of less available normal data (i.e., a fraction of $\hat{\mathcal{D}}$) on the achieved reduction in detection Recall for the black box attacker. Due to space constraints, the detailed results are presented in Appendix C. For BATADAL, the resulting mean detection Recall ranges from 0.14 for 100% of $\hat{\mathcal{D}}$ available for AE training to 0.22 for 5% of $\hat{\mathcal{D}}$ available. For WADI, we found that the attacker can leverage on 5% of normal operations (i.e., only 16 hours) to decrease detector Recall to 0.24.

Summary of Constrained Attacks findings. Our results demonstrate that replay attacks perform worse if a limited set of sensors can be manipulated. In particular, if the replay attack is constrained to manipulate less the 95% of the features the detector's Recall *increases* due to contextual anomalies that are created. For our iterative and learning based approaches, this effect does not occur, as the two attacks reduce the detector's Recall without introducing contextual anomalies. In the WADI, the attacker that controls 4% of the features succeeds in the evasion using the iterative attack. In addition, an attacker that collects a little amount of data $\hat{\mathcal{D}}$ can train the adversarial autoencoder and perform the black box attack.

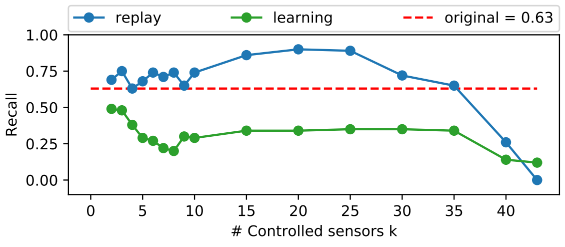


(a) BATADAL

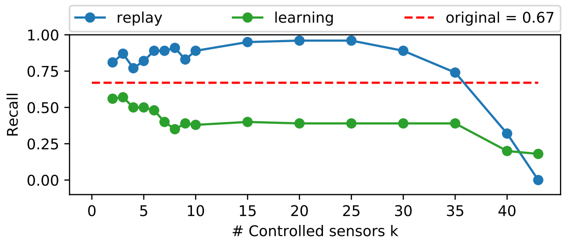


(b) WADI

Figure 2: Impact of Partially Constrained attacker (best-case scenario), comparison between replay attack and our proposed concealment attacks. In the constrained scenario, we notice that replay attack performs bad increasing detector's Recall and raising more alarms than the original data without concealment. This is due to contextual anomalies introduced as part of large-scale replay. Both learning based and iterative approaches outperform the replay attack as they do not introduce contextual anomalies and reduce detector's Recall manipulating few features.



(a) LSTM



(b) CNN

Figure 3: Generizability of our proposed Learning based attack compared with replay attack. Attack to LSTM (a) and CNN (b) based defenses on BATADAL dataset.

5.6 Generalizability of Learning Based Attack

In this section, we evaluate the generalizability of our proposed learning based attack. We consider different reconstruction-based anomaly detectors trained on BATADAL dataset and apply the concealment attack computed with the adversarially trained autoencoder as proposed in our learning based attack. Our target Deep Architecture is an LSTM reconstruction-based anomaly detector as proposed in [20] and a CNN reconstruction-based anomaly detector as proposed in [32]. Since those detectors are not available as open source, we implemented their architecture with Keras following the details found in the paper. The LSTM detector was trained to minimize the MSE loss. The network takes 8 timesteps of the multivariate time series as input (input size $8x43$). The input is processed by one LSTM layer with output size 43, followed by a fully connected layer with 43 neurons as output dimension. Performance of the LSTM based anomaly detector resulted in *Accuracy* = 0.94, *Precision* = 0.89, *Recall* = 0.63, *FPR* = 0.01. The CNN detector was trained to minimize the MSE loss. The network takes 2 timesteps of the multivariate time series as input (input size $2x43$). The input is processed by three stacked 1D convolutional layers (respectively with 64, 128 and 256 neurons), each convolutional layer is followed by a 1D Max Pooling Layer layer. The last Pooling layer is followed by a flatten and a dropout layer. Dropout layer connects to the fully connected output layer with dimension 43. CNN based performance after training resulted in *Accuracy* = 0.95, *Precision* = 0.90, *Recall* = 0.67, *FPR* = 0.01.

Results in Figure 3 show how the detection Recall diminishes when targeted with replay and learning based attacks. In particular, a replay attack can evade detection only in the case in which at least 40 out of 43 sensors are controlled by the attacker (following previous results). Results over learning based attack, despite the different architectures for the offense (Autoencoder) and defense (LSTM and CNN), show that the learning based concealment attack is transferable. In particular, the LSTM architecture appears more vulnerable to concealment attacks, since the learning based attack is achieving higher concealment efficacy than AE and CNN based defenses. Concealment efficacy over CNN defense is comparable to Autoencoder defense, notwithstanding the original Recall of CNN detector is higher than the other considered defense. These results allow us to conclude that the proposed learning based attack efficacy is independent from the Reconstruction-based anomaly detector.

5.7 Real-time Concealment Attacks

In order to test the real-time feasibility of our attacks, we deployed the anomaly detector [51] in WADI Testbed, and then attacked it

Table 4: Real-time detection of process manipulations in WADI Testbed. We replicated anomalies in WADI dataset.

Attack Identifier	Duration (minutes)	Attack detected	Detected concealment Iterative	Detected concealment Learning
W1	22	✓	✗	✗
W7	4	✓	✗	✗
W8	10	✗	✗	✗
W9	1	✓	✗	✗
W14	2	✓	✗	✗

in real-time. We collected 15 hours of normal operations occurring over the ICS. We recorded 62 sensors sampled every 10 seconds.

In this case, we tuned the *window* parameter to 30, which means that the detector is considering the sensor readings occurring in the last 5 minutes. First, we tested the reliability of the system; we left the system running 7 hours without anomalies occurring. We obtained 2 false positives instances occurring for 10 minutes each. We then performed actuators manipulation in the system. In Table 4 we report the summary of the tested anomalies. We replicated anomalies reported in the WADI dataset.

While anomalies were occurring, we also launched our Unconstrained concealment attacks in WADI Testbed to assess their feasibility and efficacy. We tested both the iterative and learning based approach in real-time by simulating the sensors value manipulation done by the attacker, all the instances of the anomalies occurring in the system were successfully misclassified. In Table 4, the last two columns summarize the results of iterative and learning based attacks carried out in real-time. In particular, our learning based modifications took the same time to compute examples as during the earlier experiments (on average, 5*ms*), which is much faster than the sampling rate in the system.

6 RELATED WORK

We now discuss important related work in the area of anomaly detection in CPS, and evasion attacks on classifiers.

Anomaly Detection in CPS. Detecting stealthy attacks in CPS through the identification of process-based anomalies without requiring a detailed physical model is an active research topic. Hadžiosmanović et al. [23] use an autoregressive model on time series extracted from modbus PLC traffic, evaluating their approach on data from two water treatment plants; Krotofil et al. [34] use a theoretical information approach to detect sensor spoofing attacks; Aoudi et al. [4] use model-free techniques rooted on singular spectrum analysis to detect structural changes in the process behavior. More recently, in Autonomous Vehicles setting, control-based techniques used for anomaly detection such as Control Invariant [13] and Extended Kalman Filters [7, 45], were found vulnerable to several stealthy attacks [15, 45, 48].

In addition, various proposals in this space use deep learning techniques (usually by training a learning-based model on data gathered during the normal operation of the process) and statistically comparing the sensor readings with the model's prediction at runtime. Wickramasinghe et al. [60] provide an overview of how

Table 5: Recent adversarial learning techniques for evasion, according to the attacker's knowledge and the domain of application. The setting for our attacks is marked with ⋆.

	White Box	Grey Box	Black Box	
	$(\mathcal{D}, \mathcal{X}, f, w)$	$(\hat{\mathcal{D}}, \mathcal{X}, f, \hat{w})$	$(\hat{\mathcal{D}}, \hat{\mathcal{X}}, \hat{f}, \hat{w})$	
			oracle	samples
Malware	[58]	[22]	[14, 61]	-
Image	[10, 21, 47]	[43]	[42]	-
ICS	⋆	-	-	⋆

Deep Learning techniques can be used in the context of CPS security. Goh et al. [20] propose an architecture to detect anomalies over a water treatment testbed with a Recurrent neural network (LSTM-RNN) used to predict sensor readings, and CUSUM to compute the difference between the predicted outputs and the actual sensor readings. Starting from this approach and using the same dataset for evaluation, Kravchik et al.[32] suggest the use of a convolutional neural network to perform one-step prediction, while Taormina et al. [51] propose the autoencoder-based detector (the target of our attacks). With respect to this former category of deep learning-based detectors, our proposed approach is the first to propose a systematic constrained attacker model and identifies vulnerabilities that affect detector performance. This enables an attacker to hide the physical anomalies induced on the system, that would be otherwise detected. Our experiments show how those attacks can be applied in constrained settings and in real time, making the prior work anomaly detectors ineffective.

Adversarial Learning for Classifier Evasion. The effectiveness of Adversarial Machine Learning to evade ML-based classifiers has been demonstrated in a wide range of applications, ranging from face recognition [47] to voice recognition [62] and malware detection [61]. Table 5 classifies recent techniques in this domain according to the adversary's knowledge on the classifier's algorithm and training dataset. In the iterative scenario (i.e., the adversary knows the internals of the trained model and the training set entirely), Rndic and Laskov [58] present a case study on the evasion of PDFRate, a malicious PDF detector based on random forests, using an white box gradient-based evasion method [5], comparing it to a black box mimicry attack, and discussing the attack effectiveness according to different attacker models. After the seminal paper that demonstrated the existence of adversarial examples for neural networks [50], work has shifted to Deep Learning. Goodfellow et al. [21] study the cause of adversarial examples and devise a fast gradient method to perform adversarial perturbations, demonstrating their results in the image classification context under a perfect-knowledge iterative scenario. More recently, Carlini and Wagner [10] defeat a defensive technique known as defensive distillation [44]. White box techniques have also been applied to defeat face recognition, also through perturbations in physical objects [47].

In more restrictive scenarios, the adversary is only aware of the general structure of the model and how features are extracted. Papernot et al. [43] use this imperfect knowledge to build a surrogate model and demonstrate the effectiveness in source-target misclassification (image recognition). Grosse et al. [22] generalize the adversarial example crafting algorithm presented in [43] to malware detection systems. In other cases, the adversary attacks a classifier while querying the system under attack as an oracle. This is the case of attacks against proprietary online learning systems: to evade an online malware classifier, Xu et al. [61] leverage the fact that the target systems output the classification score to build a genetic algorithm that morphs the adversarial examples into being undetected. More recently, Dang et al. [14] lifted the assumption of knowing the classification score, attacking oracle-like black box classifiers that only output a binary label; Papernot et al. [42] work similarly in the context of multi-class image classification.

W.r.t. prior work, in our learning based attack the adversary does not rely on querying the classifier as an oracle, or on building a surrogate learner; instead, we exploit the characteristics of the CPS domain to lift this requirement.

7 CONCLUSIONS

In this work, we started by formalizing an attacker model for real-world ICS and provided AML taxonomy for it. We then presented the first real-time concealment attacks on reconstruction-based anomaly detectors in the context of Industrial Control Systems. We argued that such attacks present four unique challenges, and addressed them proposing iterative and learning based attacks. Our white box attacker uses the iterative approach with a detection oracle, while the black box attacker uses an autoencoder to hide anomalies.

Using data from two water distribution systems, we demonstrated that our attacks are feasible in general, and outperform replay attacks when the attacker is constrained to control of less the 95% of the features. Moreover, we show that for the BATADAL dataset, our novel learning based attack using autoencoder was able to reduce detection Recall as efficiently as the iterative attack (Recall dropped from 0.60 to 0.14 in both cases). Our results demonstrate that the proposed autoencoder based attack achieves successful concealment without knowledge of the targeted Reconstruction-based anomaly detector (only using normal operational data), without knowledge of the physical model equations and is computationally cheap (after training).

We implemented our attacks in a real testbed and showed that malicious data could be generated on-the-fly, i.e., in between each sampling step (every 10s, actual example generation took on average 5ms for iterative). That demonstrates that the proposed attacks are allowing attackers to perform constrained concealment attacks on dynamic systems in real-time. In prior work, manipulations are usually performed offline against a dataset or assume that data to be manipulated can be precisely predicted. Our results show that reconstruction-based attack detectors proposed in prior work are vulnerable to manipulation despite the unique challenges in this setting, and such attacks need to be considered when designing future attack detection schemes. Implementation is available at our Online Repository[5].

ACKNOWLEDGMENTS

Several authors were supported by the National Research Foundation (NRF), Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-40). Politecnico di Milano received funding for this project from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement nr. 690972.

[5]https://github.com/scy-phy/ICS-Evasion-Attacks

REFERENCES

[1] Marshall Abrams. 2008. Malicious control system cyber security attack case study–Maroochy Water Services, Australia. (2008).

[2] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya Mathur. 2017. WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In *Proceedings of the Workshop on Cyber-Physical Systems for Smart Water Networks)*. ACM, 25–28. https://doi.org/10.1145/3055366.3055375

[3] S.M. Amin, X. Litrico, S. Sastry, and A.M. Bayen. 2013. Cyber Security of Water SCADA Systems; Part I: Analysis and Experimentation of Stealthy Deception Attacks. *Control Systems Technology, IEEE Transactions on* 21, 5 (2013), 1963–1970.

[4] Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. 2018. Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. ACM, 817–831. https://doi.org/10.1145/3243734.3243781

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (Eds.). 387–402.

[6] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331.

[7] P-J Bristeau, Eric Dorveaux, David Vissière, and Nicolas Petit. 2010. Hardware and software architecture for state estimation on an experimental low-cost small-scaled helicopter. *Control Engineering Practice* 18, 7 (2010), 733–746.

[8] A.A. Cárdnas, S.M. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. 2011. Attacks against process control systems: Risk assessment, detection, and response. In *ACM Symp. Inf. Comput. Commun. Security*.

[9] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On Evaluating Adversarial Robustness. *arXiv preprint arXiv:1902.06705* (2019).

[10] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. of the IEEE Symposium on Security and Privacy*. 39–57. https://doi.org/10.1109/SP.2017.49

[11] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 1–7.

[12] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of ACM Workshop on Artificial Intelligence and Security*. ACM, 15–26.

[13] Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Xinyan. 2018. Detecting attacks against robotic vehicles: A control invariant approach. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. ACM, 801–816.

[14] Hung Dang, Yue Huang, and Ee-Chien Chang. 2017. Evading classifiers by morphing in the dark. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. ACM, 119–133.

[15] Pritam Dash, Mehdi Karimibiuki, and Karthik Pattabiraman. 2019. Out of control: stealthy attacks against robotic vehicles protected by control-based techniques. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 660–672.

[16] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems.. In *Proc. Network and Distributed System Security Symp. (NDSS)*.

[17] Luis Garcia, Ferdinand Brasser, Mehmet H. Cintuglu, Ahmad-Reza Sadeghi, Osama Mohammed, and Saman A. Zonouz. 2017. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit. In *Proceedings of the Annual Network & Distributed System Security Symposium (NDSS)*.

[18] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. 2018. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Computing Surveys (CSUR)* 51, 4, Article 76 (July 2018), 36 pages. https://doi.org/10.1145/3203245

[19] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[20] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. 2017. Anomaly detection in cyber physical systems using recurrent neural networks. In *High*

Assurance Systems Engineering (HASE), 2017 IEEE 18th International Symposium on. IEEE, 140–145.

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *CoRR* abs/1412.6572 (2014).

[22] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2017. Adversarial Examples for Malware Detection. In *Proc. of the European Symposium on Research in Computer Security.* Springer International Publishing, 62–79.

[23] Dina Hadžiosmanović, Robin Sommer, Emmanuele Zambon, and Pieter H. Hartel. 2014. Through the Eye of the PLC: Semantic Security Monitoring for Industrial Processes. In *Proceedings of the 30th Annual Computer Security Applications Conference (ACSAC '14).* ACM, 126–135. https://doi.org/10.1145/2664243.2664277

[24] Michael A Hayes and Miriam AM Capretz. 2015. Contextual anomaly detection framework for big sensor data. *Journal of Big Data* 2, 1 (2015), 2.

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[26] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence.* ACM, 43–58.

[27] Peter Huitsing, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi. 2008. Attack taxonomies for the Modbus protocols. *International Journal of Critical Infrastructure Protection* 1 (2008), 37–44.

[28] iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design. 2017. WADI datatset. (2017). https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/, Last accessed on: 2020-06-15.

[29] Keras EarlyStopping callback [n. d.]. EarlyStopping. https://keras.io/api/callbacks/early_stopping. ([n. d.]).

[30] Keras ReduceLROnPlateau callback [n. d.]. ReduceLROnPlateau. https://keras.io/api/callbacks/reduce_lr_on_plateau/. ([n. d.]).

[31] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[32] Moshe Kravchik and Asaf Shabtai. 2018. Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy.* ACM, 72–83.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[34] Marina Krotofil, Jason Larsen, and Dieter Gollmann. 2015. The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '15).* ACM, 133–144. https://doi.org/10.1145/2714576.2714599

[35] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy Chowdhury, and Ananthram Swami. 2019. Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems. *Proceedings of the Annual Network & Distributed System Security Symposium (NDSS)* (2019).

[36] Yao Liu, Peng Ning, and Michael K Reiter. 2011. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)* 14, 1 (2011), 13.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.*

[38] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).

[39] Yilin Mo and Bruno Sinopoli. 2009. Secure control against replay attacks. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on.* IEEE, 911–918.

[40] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. 2015. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine* 35, 1 (2015), 93–109.

[41] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1765–1773.

[42] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks Against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17).* ACM, New York, NY, USA, 506–519. https://doi.org/10.1145/3052973.3053009

[43] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. In *2016 IEEE European Symposium on Security and Privacy (EuroSP).* 372–387. https://doi.org/10.1109/EuroSP.2016.36

[44] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *Proc. of the IEEE Symposium on Security and Privacy.* 582–597. https://doi.org/10.1109/SP.2016.41

[45] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. 2020. SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants. In *Proc. of the USENIX Security Symposium.* Boston, MA. https://www.usenix.org/conference/usenixsecurity20/presentation/quinonez

[46] R. Taormina. 2018. AutoEncoders for Event Detection (AEED): a Keras-based class for anomaly detection in water sensor networks. (2018). https://github.com/rtaormina/aeed, Last accessed on: 2020-06-15.

[47] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16).* ACM, 1528–1540. https://doi.org/10.1145/2976749.2978392

[48] Junjie Shen, Jun Yeon Won, Shinan Liu, Qi Alfred Chen, and Alexander Veidenbaum. 2020. Poster: Security Analysis of Multi-Sensor Fusion based Localization in Autonomous Vehicles. (2020).

[49] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. 2018. When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks. In *27th USENIX Security Symposium (USENIX Security 18).* 1299–1316.

[50] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR* abs/1312.6199 (2013). arXiv:1312.6199

[51] Riccardo Taormina and Stefano Galelli. 2018. A Deep Learning approach for the detection and localization of cyber-physical attacks on water distribution systems. *Journal of Water Resources Planning Management* 144, 10 (2018), 04018065. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000983

[52] R. Taormina, S. Galelli, H.C. Douglas, N. O. Tippenhauer, E. Salomons, and A. Ostfeld. 2019. A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. Environmental Modelling Software. *Environmental Modelling Software* 112 (02 2019), 46–51. https://doi.org/10.1016/j.envsoft.2018.11.008

[53] Riccardo Taormina, Stefano Galelli, Nils Ole Tippenhauer, Elad Salomons, Avi Ostfeld, Demetrios G. Eliades, Mohsen Aghashahi, Raanju Sundararajan, Mohsen Pourahmadi, M. Katherine Banks, B. M. Brentan, Enrique Campbell, G. Lima, D. Manzi, D. Ayala-Cabrera, M. Herrera, I. Montalvo, J. Izquierdo, E. Luvizotto, Jr, Sarin E. Chandy, Amin Rasekh, Zachary A. Barker, Bruce Campbell, M. Ehsan Shafiee, Marcio Giacomoni, Nikolaos Gatsis, Ahmad Taha, Ahmed A. Abokifa, Kelsey Haddad, Cynthia S. Lo, Pratim Biswas, Bijay Pasha, M. Fayzul K.and Kc, Saravanakumar Lakshmanan Somasundaram, Mashor Housh, and Ziv Ohar. 2018. The Battle Of The Attack Detection Algorithms: Disclosing Cyber Attacks On Water Distribution Networks. *Journal of Water Resources Planning and Management* 144, 8 (Aug. 2018). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000969

[54] André Teixeira, Iman Shames, Henrik Sandberg, and Karl H Johansson. 2012. Revealing stealthy attacks in control systems. In *Proceedings of Annual Allerton Conference on Communication, Control, and Computing (Allerton).* IEEE, 1806–1813.

[55] Throwing Star LAN Tap [n. d.]. Throwing Star LAN Tap. https://greatscottgadgets.com/throwingstar/. ([n. d.]).

[56] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs.. In *USENIX Security Symposium.* 601–618.

[57] David Urbina, Jairo Giraldo, Alvaro A. Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. 2016. Limiting The Impact of Stealthy Attacks on Industrial Control Systems. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS).* https://doi.org/10.1145/2976749.2978388

[58] Nedim Šrndić and Pavel Laskov. 2014. Practical Evasion of a Learning-Based Classifier: A Case Study. In *Proceedings of IEEE Symposium on Security and Privacy.* 197–211. https://doi.org/10.1109/SP.2014.20

[59] Sharon Weinberger. 2011. Computer security: Is this the start of cyberwarfare? *Nature* 174 (June 2011), 142–145.

[60] Chathurika S Wickramasinghe, Daniel L Marino, Kasun Amarasinghe, and Milos Manic. 2018. Generalization of Deep Learning for Cyber-Physical System Security: A Survey. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society.* IEEE, 745–751.

[61] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically evading classifiers. In *Proceedings of the Network and Distributed Systems Symposium.*

[62] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible voice commands. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 103–117.

A DETAILS OF ITERATIVE ATTACK

The attacker essentially has access to an *oracle* of the Deep Architecture, where the attacker can provide arbitrary $\vec{x}$ features and gets the individual values of the reconstruction error vector $\vec{e}$. The attacker then computes $\max_i \vec{e}$ and finds the sensor reading r_i with the highest reconstruction error from $\vec{x}$. In order to satisfy $\varepsilon(\vec{e'}) < \theta$, the attacker attempts to decrease the reconstruction error d_i error by changing r_i. Sensor readings r_i are modified in the range of normal operating values; this guides the computation to a solution that is consistent with the physical process learned by the detector. For example, if normal operations of sensor r_i are in the range $[0, 5]$, the attacker tries to substitute the corresponding value of r_i according to its range to see if the related reconstruction error decreases. This results in $\vec{x}' = [r_1, \dots, r_i', \dots, r_n]$, where $d_i' < d_i$ and, accordingly, $\varepsilon(\vec{e}') < \varepsilon(\vec{e})$. Algorithm 1 is the pseudo-code applied to compute sensor readings modifications.

In order to find the value of r_i that decreases $\varepsilon(\vec{e})$ the most, we can introduce X as the matrix containing the mutations of $\vec{x}$ w.r.t. r_i.

$$X = \begin{bmatrix} r_1 & \dots & r_i^1 & \dots & r_n \\ r_1 & \dots & r_i^2 & \dots & r_n \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_1 & \dots & r_i^m & \dots & r_n \end{bmatrix}$$

were $r_i^k \in$ *normal operations values for senor i*. Among the all mutations, we select the one that generates the lower reconstruction error $\varepsilon(\vec{e})$. After choosing the best value over the variable r_i the algorithm repeats until a solution with average reconstruction error lower than θ is found.

Two stopping criteria are put in place: *patience* and *budget*. It could happen that no lower reconstruction errors d_i are found by changing the value of a chosen reading r_i. In this case, we try to change the other readings in descending order of reconstruction error. *patience* mechanism is put in place to avoid wasting of computation. If no improved solutions are found in *patience* iterations, the input is no more optimized.

According to the communication mechanism between PLCs and SCADA, the attacker may be constrained to send the data in a certain amount of time. *budget* is the maximum amount of times that loop at Line 8 (Algorithm 1) can be performed. After *budget* attempts without finding a set of modified readings that satisfies $\varepsilon(\vec{e}') < \theta$, the input is no more optimized, and no solution is found.

Exiting the loop at Line 8 due to a stopping criterion is not providing a misclassified example. Even though a solution such that $\varepsilon(\vec{e}') < \theta$ is not found, the resulting tuple is likely to have a lower $\varepsilon(\vec{e})$, i.e., $\varepsilon(\vec{e}) > \varepsilon(\vec{e}') > \theta$.

B DEFINITION OF CONSTRAINTS

In order to study the impact of this best-case constraints, we selected k features for every attack that can be modified. Then we studied how replay, iterative, and learning based attacks perform when these constraint are applied. We defined the constrains as follows: starting from the results of iterative and learning based attacks we determined the k features that were changed most frequently (over the course of each attack). The intuition behind this is as follows: features that are modified most often in the unconstrained case are assumed to have the highest impact on the performance of the attack. We are assuming a best case scenario for the attacker, in which she was able to choose the k our of n features to maximise her efficiency. Then, we created 11 sets of k features that can be modified by the attacker (with different counts k of features to be manipulated, with the maximal number determined by the dataset used). In the case of iterative and learning based attack, we limited the adversarial example exploration to the k features extracted for the considered approach. Effectively, that implies that we only used the allowed k features out of the learning based model, while the iterative model was able to learn modifications to the k allowed features that would minimize the detector accuracy. In the case of replay attack we applied the same replay strategy introduced before but we replayed only the selected k features extracted from the iterative approach. We note that this choice (replay the features extracted from the iterative approach) was made to reflect worst case scenario, i.e., an attacker that is able to replay exactly the k features that an iterative attacker would replay.

Algorithm 1 White Box concealment attack

```
1:  procedure CONCEAL($\vec{x}$)
2:      $c \leftarrow 0$                                   ▷ number of changes
3:      $i \leftarrow 0$                                   ▷ last optimization
4:      $solved \leftarrow False$
5:      $\vec{e} \leftarrow compute\_reconstruction\_errors(\ \vec{x}\ )$
6:      $previous\_best\_error \leftarrow \varepsilon(\vec{e})$          ▷ access oracle
7:      $\vec{e} \leftarrow sort\_descending(\vec{e})$
8:      while $!(solved)$ && $(c - i) < patience$ && $c < budget$ do
9:          $f \leftarrow choose\_feature\_to\_optimize\ (\vec{e})$
10:         $X \leftarrow compute\_matrix\_of\_mutations(\vec{x}, f)$
11:         $x', \vec{e}' \leftarrow find\_best\_mutation(\vec{X})$
12:         if $\varepsilon(\vec{e}') < previous\_best\_error$ then
13:             $previous\_best\_error \leftarrow \varepsilon(\vec{e}')$
14:             $new\_best \leftarrow \vec{x}'$
15:         else
16:             $i \leftarrow c$
17:         end if
18:         if $\varepsilon(\vec{e}') < \theta$ then
19:             $solved \leftarrow True$
20:         end if
21:         $c \leftarrow c + 1$
22:         $\vec{e} \leftarrow sort\_descending(\vec{e}')$
23:     end while
24:     return $new\_best$
25: end procedure
```

C LEARNING BASED ATTACK: IMPACT OF $\mathcal{D}$ DIMENSION

Another aspect that we investigated is the impact of $\hat{\mathcal{D}}$ on the applicability of learning based attack. Especially, we are interested in understanding how much normal operational data the attacker needs to conduct the proposed learning based attack. We investigated the impact of less available normal data (i.e., a fraction of $\hat{\mathcal{D}}$) on the achieved reduction in detection Recall for the learning based attacker. We performed a sensitivity analysis by random sampling normal operations data 10 times for each one of the considered percentages of data. Then, we trained an adversarial network for each sampling of the data percentage (50 adversarial networks trained for each dataset). As result we computed the sample mean ($\mu_{\bar{x}}$) and

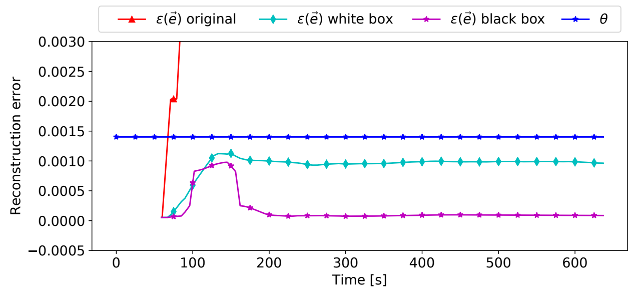


Figure 4: Comparison of concealment results. While the Recall after concealment in both white and learning based goes to 0, we can see how the two approaches are behaving differently. We plot the average reconstruction error over time ($\varepsilon(\vec{e})$) and the threshold θ.

sample standard deviation ($\sigma_{\bar{x}}$) of the resulted detection Recall by using the different learning based networks.

For BATADAL, the resulting mean detection Recall ranges from 0.14 for 100% of $\hat{\mathcal{D}}$ available for AE training to 0.22 for 5% of $\hat{\mathcal{D}}$ available. For WADI, the resulting mean detection Recall ranges from 0.31 for 100% of $\hat{\mathcal{D}}$ available for AE training to 0.50 for 5% of $\hat{\mathcal{D}}$ available (compared to 0.68 without concealment). Results over BATADAL dataset show, performance of the attacker's adversarial network is performing almost the same if trained with 100% to 25% of data. Lower than 25% of the data we notice substantial performance degradation. Looking at standard deviation, we notice that less data (10%. 5%) causes high model variance. To perform the learning based attack the attacker needs 25% of data to guarantee evasion success.

In the case of WADI dataset performance of the adversarial network increases diminishing the number of data available to the attacker, this means that with less data the attacker's Autoencoder generalizes better. WADI water distribution network is small and the three stage are repetitive. Information contained in 5% of the data (16 hours of recordings) could be enough to model the system behavior.

D DISCUSSION

We showed that replay attacks (while not requiring machine learning algorithms) is only efficient when all sensor readings replayed. Thus, replay attacks do not represent a viable solution for hiding anomalies when the attacker can act on a limited set of sensor readings. In particular, replay attacks introduce contextual anomalies since sensor readings will not be consistent any longer.

We now discuss the quality of results coming from the proposed approaches. Figure 4, represents the comparison between trend of $\varepsilon(\vec{e})$ wrt. the threshold θ during the whole actuators' manipulation done in one attack from WADI dataset. Comparing the white and learning based $\varepsilon(\vec{e})$ results, we notice that the solution provided by the iterative algorithm is closer to θ than the learning based solution. This is because the iterative algorithm is looking at θ value to decide whether to stop the computation. Black box is not performing any optimization wrt. the attacked detector, so it is providing a solution that is matching the learned physical behavior, and what the detector expects from a non-anomalous sample. After second 200, the magenta line shows that the $\varepsilon(\vec{e})$ is around 0, meaning that we are sending inputs that are in line with the detector expected behavior.

Table 6: Impact of fraction of $\hat{\mathcal{D}}$ on concealing capacity.

	Original	Recall for Black Box % of $\hat{\mathcal{D}}$										
		100%	75%		50%		25%		10%		5%	
Data	Recall		$\mu_{\bar{x}}$	$\sigma_{\bar{x}}$	$\mu_{\bar{x}}$	$\sigma_{\bar{x}}$	$\mu_{\bar{x}}$	$\sigma_{\bar{x}}$	$\mu_{\bar{x}}$	$\sigma_{\bar{x}}$	$\mu_{\bar{x}}$	$\sigma_{\bar{x}}$
B	0.60	0.14	0.15	0.02	0.15	0.02	0.16	0.03	0.25	0.09	0.22	0.09
W	0.68	0.31	0.27	0.06	0.26	0.01	0.24	0.03	0.24	0.04	0.24	0.06

Table 7: Impact of Partially Constrained attacker (best-case scenario and topology based scenario). By decreasing the number of features that the attacker can control, we notice that replay attack performance decreases drastically. This is due to contextual anomalies introduced as part of large-scale replay, while both black and iterative approaches avoid this problem.

	Original		Recall vs. # of Controlled sensors k (43 features BATADAL)																Topology
Data	Recall	Experiment	43	40	35	30	25	20	15	10	9	8	7	6	5	4	3	2	1 PLC
B	0.60	replay	0	0.25	0.69	0.83	0.94	0.93	0.9	0.78	0.69	0.79	0.77	0.77	0.71	0.68	0.78	0.73	0.36
		iterative	0.14	0.17	0.17	0.15	0.16	0.22	0.22	0.22	0.22	0.25	0.25	0.25	0.27	0.35	0.51	0.52	0.34
		learning	0.14	0.16	0.35	0.37	0.37	0.38	0.38	0.35	0.35	0.24	0.27	0.32	0.34	0.49	0.56	0.55	0.34

	Original		Recall vs. # of Controlled sensors k (82 features WADI)																		Topology
Data	Recall	Experiment	82	80	70	60	50	40	30	20	15	10	9	8	7	6	5	4	3	2	1 PLC
W	0.68	replay	0.07	0.15	0.55	0.7	0.74	0.79	0.89	0.87	0.85	0.65	0.64	0.62	0.62	0.62	0.6	0.44	0.4	0.49	0.64
		iterative	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.09	0.12	0.12	0.12	0.12	0.12	0.14	0.25	0.12
		learning	0.31	0.31	0.31	0.4	0.4	0.4	0.45	0.41	0.41	0.32	0.36	0.36	0.36	0.34	0.27	0.27	0.27	0.29	0.36

Table 8: Generizability of Learning based attack. Attack to LSTM and CNN based defenses on BATADAL dataset.

	Original		Recall vs. # of Controlled sensors k (43 features BATADAL)															
Data	Recall	Experiment	43	40	35	30	25	20	15	10	9	8	7	6	5	4	3	2
LSTM	0.63	replay	0	0.26	0.65	0.72	0.89	0.9	0.86	0.74	0.65	0.74	0.71	0.74	0.68	0.63	0.75	0.69
		learning	0.12	0.14	0.34	0.35	0.35	0.34	0.34	0.29	0.3	0.2	0.22	0.27	0.29	0.38	0.48	0.49
CNN	0.67	replay	0	0.32	0.74	0.89	0.96	0.96	0.95	0.89	0.83	0.91	0.89	0.89	0.82	0.77	0.87	0.81
		learning	0.18	0.2	0.39	0.39	0.39	0.39	0.4	0.38	0.39	0.35	0.4	0.48	0.5	0.5	0.57	0.56