White-Box Concealment Attacks Against Anomaly Detectors for Cyber-Physical Systems

Alessandro Erba[1,2](✉) and Nils Ole Tippenhauer[1]

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{alessandro.erba,tippenhauer}@cispa.de

[2] Saarbrücken Graduate School of Computer Science, Saarland University, Saarbrücken, Germany

Abstract. Anomaly detection for cyber-physical systems is an effective method to detect ongoing process anomalies caused by an attacker. Recently, a number of anomaly detection techniques were proposed (e.g., ML based, invariant rule based, control theoretical). Little is known about the resilience of those anomaly detectors against attackers that conceal their attacks to evade detection. In particular, their resilience against white-box concealment attacks has so far only been investigated for the subset of neural network-based detectors. In this work, we demonstrate for the first time that white-box concealment attacks can also be applied to detectors that are not based on neural network solutions. In order to achieve this, we propose a generic white-box attack that evades anomaly detectors and can be adapted even if the target detection technique does not optimize a loss function. We design and implement a framework to perform our attacks, and test it on several detectors from related work. Our results show that it is possible to completely evade a wide range of detectors (based on diverse detection techniques) while reducing the number of samples that need to be manipulated (compared to prior black-box concealment attacks).

1 Introduction

Cyber-Physical Systems (CPS) interact with the physical environment to accomplish a task by using sensors and actuators while applying a control strategy. Examples of such systems are Industrial Control Systems (ICS), Critical Infrastructures (such as power and water systems), and Autonomous Vehicles (AV).

The security and reliability of those systems are crucial in our society. For example, the water reaches houses through water treatment and distribution systems, which are critical infrastructures, consisting of pipes, pump stations, industrial controllers, etc. Attacks targeting those infrastructures can cause disruption (e.g., no water to houses), or harm people (e.g., contaminants in water).

Recently, anomaly detection techniques for CPS gained popularity as they allow the identification of process anomalies caused by cyber-attacks while

D. Gruss et al. (Eds.): DIMVA 2023, LNCS 13959, pp. 111–131, 2023.
https://doi.org/10.1007/978-3-031-35504-2_6

remaining legacy compliant. Different techniques were proposed in the literature to detect anomalies in CPS, system identification [3,9,19,29], Kalman filtering [2], Support Vector Machines [8], Deep learning [16,22,28] and control invariants [1,13]. Little is known about the resilience of those anomaly detection techniques against targeted manipulation, especially regarding classifier evasion [4]. If an attacker evades the anomaly detection system to conceal the true state and avoid or delay detection, can cause severe hardware damage or harm human beings. Concealment attacks are a variant of evasion attacks, in which evasion by sensor manipulation will not have a direct effect on the process [11], and can be performed in white-box and black-box settings. We refer to white-box and black-box to differentiate the knowledge of the attacker. A white-box attacker has access to a copy of the anomaly detector, which can be queried to get detection scores for a sample. A black-box attacker can not access this information.

Prior work demonstrated that generic black-box concealment attacks on general anomaly detectors are possible [12], but those limitations lead to attacks that manipulate a large number of sensors, over many samples. It is unclear how optimal those attacks are—we need a baseline to compare against. White-box concealment attacks by a less constrained, more knowledgeable attacker could provide such a baseline, but those attacks were only investigated for the specific subclass of Deep Learning based anomaly detectors [11]. Thus, the threat posed by white-box concealment attacks on general anomaly detectors is unclear, and in particular, the minimal perturbation required to achieve misclassification (by strong attackers) is unknown for each detector.

In this work, we bridge this research gap by addressing three research questions: **R1** How resilient are anomaly detectors for cyber-physical systems against white-box concealment attacks? **R2** Can white-box attacks efficiently compute manipulations at runtime? **R3** How do the white-box attacks perform compared to prior work black-box attacks?

To address the aforementioned research questions we tackle two research challenges: **C1** The attacker manipulates dynamic streaming data, i.e., the attacker cannot retroactively change past values, or predict future process sensor values. **C2** General detectors are not guaranteed to optimize a differentiable loss function for detection (in contrast to Deep Learning-based detectors). We address C1 by implementing and evaluating a method that manipulates only the current sensors' observations and show that it is still possible to minimize the detection function loss. We address C2 by proposing a method to re-write non-differentiable classification functions as differentiable and hence allow concealment attacks.

List of Contributions. The main contributions of the paper are:

- Designing an effective general purpose white-box concealment framework for anomaly detection systems.
- Formulation of loss-free detectors (i.e. process invariants), as loss-based.
- Evaluation of proposed white-box attacks with real testbed data against five state-of-the-art anomaly detection systems.
- Comparison of the proposed white-box concealment with prior attacks.

Table 1. Summary of anomaly detection families proposed in prior work in the context of CPS. The table reports the approach used for detection and the detectors that we analyze in our evaluation (○ = no, ● = yes). We skip DNN as it was analyzed before.

	[19]	[3]	[9,29]	[2,8]	[16,22,28]	[1,13]
Approach type	AR	SVD	LTI	SVM	DNN	Invariants
Classification Differentiability	D	D	D	D	D	N
Prior WBC analysis	○	○	○	○	●	○
Analyzed in this work	●	●	●	●	○	●

2 CPS: Background and Related Work

CPS Architecture. Cyber-physical systems encompass a wide spectrum of applications [23]. The general CPS architecture consists of three main components. Sensors: measure the physical environment; controllers: use the information received from sensors and decide which actions to take; and actuators: execute those commands.

CPS Security. Given the high degree of interconnections in a CPS, the overall security of CPS deployments relies on trustworthy communication. In practice, CPS systems are often deployed relying on protocols that do not implement security features (such as authentication or encryption) e.g., Fieldbus [14], CAN [14], or mavlink [21]. Communication protocols that promise security were introduced for ICS, but in practice, there are challenges in deploying secure CPS [10].

Attacks to CPS. CPS are important for our society and they are a valuable target of attacks [6]. Attacks on CPS occurred in the past. For example, attacks to ICS and critical infrastructures e.g., Stuxnet [31] targeting nuclear plants, the Colonial Pipeline attack [32] targeting gasoline pipeline and Oldsmar's water treatment attack [7] targeting a water facility. The common goal of attacks is to physically or remotely exploit the CPS to cause process disruption.

Anomaly Detection for CPS. A number of process-based anomaly detection techniques were proposed in the literature. They leverage the characteristics of the physical process to detect deviations in the process data caused by attacks [6]. *i) Residual-based approaches* are trained to minimize a loss function (usually Mean Squared Error), between the expected and observed sensor readings. To detect anomalies, the loss between input and output is monitored, if it exceeds a threshold an alarm is raised. In this category, we find control theoretic approaches e.g., Auto Regressive (AR) models [19] and Linear Time-Invariant (LTI) models [9,29], and machine learning approaches e.g., Support Vector Machines [2,8] and Deep Neural Networks [16,22,28]. *ii) Invariant-based approaches* consist of rules that describe conditions that always hold in a given state on the CPS [1]. Those rules are often written based on detailed process knowledge [1,13].

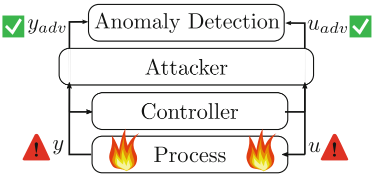

Fig. 1. System and attacker model, we assume a physical process that is controlled by a controller and monitored by an anomaly detection system. The attacker wants to hide an ongoing anomaly on the CPS. The attacker is aware that anomaly detection is deployed and wishes to conceal the true state and evade detection.

Evasion Attacks Against CPS. In Adversarial Machine Learning, evasion attack refers to the setting in which an attacker modifies a sample to induce misclassification in a classifier [4]. In the context of the Advanced Driver-Assistance System (ADAS), several attacks were proposed e.g., against LIDAR [5], location estimation [27]. In the context of CPS anomaly detection, white-box attacks against Deep Learning models [11,33] were demonstrated. Also, generic black-box evasion techniques were proposed [12]. Table 1 summarizes prior work in the field of anomaly detection for CPS, and reports which models were analyzed before for white-box concealment attacks. In this work, we focus on models proposed in prior work but not analyzed so far against white-box concealment.

3 System and Attacker Model

We assume a Cyber-Physical System that is monitored by an anomaly detection system to detect anomalies (Fig. 1). The physical process is controlled by one or multiple controllers, control commands u and sensor readings y are observed by the anomaly detector and used for the detection. Consistent with related work [11], we assume an attacker that has physical access to the CPS e.g., the attacker can attach malicious hardware to the network, and perform sensor spoofing exploiting communication protocol vulnerabilities (e.g., unauthenticated industrial protocols [14]) or performing attacks such as Man-in-the-PLC [15] attack. The attacker has knowledge of the system and can query the anomaly detector to obtain the predictions/classifications w.r.t. the current y and u. The attacker's goal is to launch a concealment attack to hide an ongoing process anomaly in the system (i.e., conceal the anomalies caused by the attacker on the process from the anomaly detection system).

The attacker can modify exchanged industrial traffic in transit, or compromise intermediate hosts to change values being forwarded (y_{adv} and u_{adv}), in Fig. 1. For example, in the Stuxnet attack [31] a compromised PLC was changing the rotation frequency of centrifuges of a nuclear process while reporting the correct frequency value to the anomaly detection to hide the anomaly. We measure the cost of the attack with respect to the number of features that are manipulated using the L0 norm (independent of the modification amount, i.e. L2 norm), as the effort is in compromising the communication channel, and at that point, arbitrary values can be set [11]. In practice, we allow any perturbations within the operational limits of the respective sensor or actuator [26].

3.1 Research Goals and Challenges

We address the three open research questions presented in the introduction. While addressing the three research questions we tackle the following research challenges: **C1** The attacker manipulates dynamic streaming data on the fly, which means that the attacker i) iteratively manipulates each value sequentially without knowing future values in advance; ii) adapts the strategy according to previous values stored in data logs without altering them. This is imposed by the Cyber-Physical Systems, where the attacker is assumed to perform sensor spoofing exploiting communication channels vulnerabilities. **C2** Not all general detectors are guaranteed to have differentiable loss functions (in contrast to Deep Learning based detectors). Thus, we need a general technique to attack different detectors even in absence of a loss function. For example, the detector [13] represents the current sample as a boolean vector (each element representing whether a specific invariant was violated). For this reason, we cannot use gradient-based methods (for example) to find optimal evasion samples.

Our main goal is to assess whether additional knowledge on detection mechanisms (i.e. white-box attacks) allows the attacker to perform better compared to black/grey-box attacks discussed in prior work [11,12]. This allows us to assess the robustness of CPS anomaly detectors, i.e., the minimal number of communication channels (features) that need to be controlled by the attacker to avoid detection.

3.2 Formal Definition of Concealment Attack

We now summarize the formal definition of the attack based on prior work [11]. Sensor and actuator values from a CPS are logged and used for anomaly detection. Given an anomalous feature vector $x = (y, u)$ (i.e., sensors and actuators readings) collected at a certain instant in time, a binary classification function $f(x)$ that classifies system state as 'anomalous' or 'safe', the concealment attack looks for a feature perturbation δ that added to x produces target misclassification (Eq. 1).

$$\begin{aligned} \text{Given} \quad & x = (y, u) \\ \text{s.t.} \quad & f(x) = \text{`anomalous'} \\ \text{Find} \quad & x_{adv} = x + \delta \\ \text{s.t.} \quad & f(x_{adv}) = \text{`safe'} \end{aligned} \tag{1}$$

where $y \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $x \in \mathbb{R}^{n+m}$, $x_{adv} = (y_{adv}, u_{adv})$, $y_{adv} = y + \delta_y$, $\delta_y \in \mathbb{R}^n$, $u_{adv} = u + \delta_u$, and $\delta_u \in \mathbb{R}^m$.

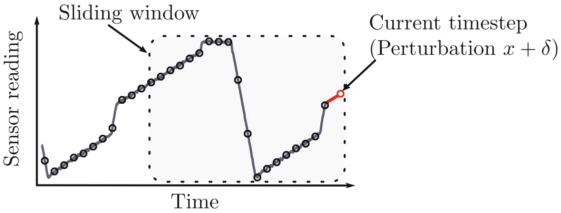

Fig. 2. Challenge **C1**. For each time slot, the attacker can only manipulate the latest sensor reading without knowing future values. We note the attacker cannot retroactively modify previous (manipulated or original) values. Eventually, the data considered in the sliding window will exclusively process values that were manipulated before.

4 Proposed Approach

We design a generic white-box concealment attack for CPS anomaly detectors. In this section, we start proposing the general framework that can be applied to attack prior work anomaly detectors.

4.1 White-Box Concealment Attacks (WBC)

We translate the white-box concealment attack (WBC) objective (Eq. 1) into an error minimization problem (Eq. 2)

$$\begin{aligned} \text{minimize} \quad & Loss_{x_{adv}}(x_{adv}, tc) \\ \text{where} \quad & tc = \text{target class} \end{aligned} \tag{2}$$

Then, we induce targeted misclassification (to achieve the goal in Eq. 1) inspired by the Fast Gradient Signal Method (FGSM) [18] proposed originally for the domain of image manipulation.

$$\delta = -\epsilon * \text{sign}(\nabla_x Loss(x, tc)) \tag{3}$$

Every anomaly detection method has a different classification function, and consequently a different loss (if explicitly present), for this reason, this generic method is suitable to be applied to different categories of anomaly detectors.

The perturbation in Eq. 3 is iteratively applied until the concealment attack is successful and the detector no longer flags the anomaly (Eq. 1). Two attackers can be considered in this setting (we will compare them in Sect. 6). The first continue iterating until the objective (Eq. 2) is minimized, and the second continues until the classification label is changed, but the objective is not necessarily minimized.

4.2 Attacking Detectors with Differentiable Classifiers

We now address the research challenge **C1**: on-the-fly manipulation of streaming data (see Fig. 2). Residual-based anomaly detectors classify anomalies based

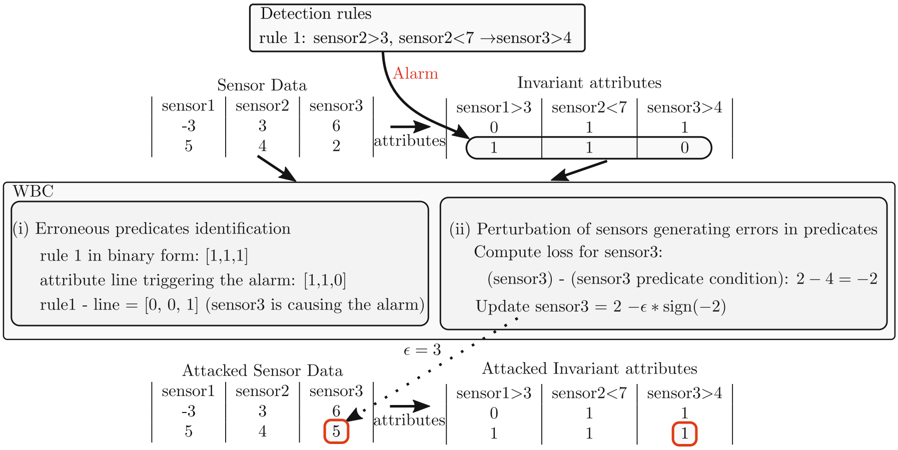
Detection rules
rule 1: sensor2>3, sensor2<7 →sensor3>4
Alarm
Sensor Data
sensor1 sensor2 sensor3
-3 3 6
5 4 2
attributes
Invariant attributes
sensor1>3 sensor2<7 sensor3>4
0 1 1
1 1 0
WBC
(i) Erroneous predicates identification
rule 1 in binary form: [1,1,1]
attribute line triggering the alarm: [1,1,0]
rule1 - line = [0, 0, 1] (sensor3 is causing the alarm)
(ii) Perturbation of sensors generating errors in predicates
Compute loss for sensor3:
(sensor3) - (sensor3 predicate condition): $2 - 4 = -2$
Update sensor3 = $2 - \epsilon * \text{sign}(-2)$
$\epsilon = 3$
Attacked Sensor Data
sensor1 sensor2 sensor3
-3 3 6
5 4 5
attributes
Attacked Invariant attributes
sensor1>3 sensor2<7 sensor3>4
0 1 1
1 1 1

Fig. 3. Challenge **C2**. WBC concealment against detectors without loss function. Invariant-based detection works by checking whether the sensor readings satisfy the invariant rules, without relying on predictive models (no loss function). To apply our WBC attack, we re-formulate invariant-based methods as loss-based. We manipulate the sensors and actuators readings according to the difference (loss) between the desired state specified by the rules and the current state.

on the residual error between the sensors and actuators readings x and a predicted output value o from the anomaly detection classifier. That classification is performed over a sliding window of past observed values (i.e., $[x_{t-n}, \ldots, x_t]$).

In our scenario, the attacker can not simultaneously manipulate each value in this sliding window (as it would require post-hoc change of data), only the current sensor reading x_t can be manipulated. This introduces a novel constraint on the attack as the attacker has to minimize the residual loss acting on the last observed sample, and cannot globally minimize the loss function. We account for this additional constraint in our evaluation.

For example, to perform the WBC attack for residual-based detectors, we model the residuals by using the Mean Squared Error loss (Eq. 4). Then, we compute the partial derivative of the mean squared error w.r.t. x (Eq. 5) and apply directly FGSM to it (Eq. 3).

$$Loss(x, o) = \frac{1}{2}(x - o)^2 \quad (4) \qquad \nabla_x Loss(x, o) = x - o \quad (5)$$

4.3 Attacking Detectors with Non-differentiable Classifiers

Invariant-based anomaly detectors [1, 13] classify anomalies based on the coherence of the system sensors and actuators w.r.t. a set of process invariant rules. When invariants are used, detectors check if some invariant rules are not fulfilled and raise an alarm consequently.

$$\text{Given an invariant rule } R\text{:} \quad A \rightarrow B \qquad \text{(read as: if A then B)} \tag{6}$$

where A is the antecedent and B is the consequent of the invariant rule. Antecedent and consequent of a rule, consist of a set of predicates over certain sensors and actuators (e.g., valve_status = 1 and sensor_value < 4). An anomaly is identified if predicates in the antecedent A are all satisfied but not all predicates in consequent B are satisfied.

This method does not employ a loss function. In order to evade such detectors we need to consider the research challenge **C2**, i.e., we need to formulate the invariant-based approach as a loss-based method. Specifically, to evade the detector an attacker is required to modify the sensor readings in such a way that the predicates in B are fulfilled[1]. In order to do so, we decompose the attack in two steps (Fig. 3 provides a toy example of the method).

(i) Erroneous Predicates Identification. In the first step we identify which predicates trigger the anomaly in B. To do so we perform the set difference between the predicates in the rule R and the predicates observed in the system P (Eq. 7).

$$R \setminus P \tag{7}$$

Practically predicates are represented by Boolean conditions (i.e., boolean vectors where the position represents a certain invariant and the value 1 or 0 represents if the invariant condition holds). We identify the predicate that does not match the triggered rule performing the difference of such vectors.

(ii) Perturbation of Sensors Generating Errors in Predicates. In the second step, for the predicates that are erroneous we need to perturb the data related to that predicate to induce the change in the generated predicates. To guide sensor reading perturbation we can consider the desired value (i.e., the condition required by the predicate) of the erroneous predicates as our target value. This step can be performed by substituting the desired value directly in the sensor reading if the predicate is a direct equality or inequality over the sensor value (e.g., sensor = 3). Otherwise, if the predicate aggregates more information about a sensor reading (e.g., Gaussian Mixture Models over sensor value updates), we formulate the problem as a Mean Squared Error minimization as in Sect. 4.2, and compute the perturbation using Eq. 3 and using the loss as in Eq. 4.

5 Implementation and Evaluation Setup

In this section, we provide details about the implementation setup, the target anomaly detection systems, and the dataset used for evaluation. Based on the

[1] Alternatively the attacker can deactivate a rule by violating one condition in A, but this does not give guarantees about other rules that might be triggered by the modification.

categories of detectors identified in Sect. 2 and the analysis of prior work white-box concealment attacks in Table 1 we selected the target detectors according to three main criteria: (i) diversity of the detection technique (ii) not covered by prior work studies on white-box concealment attacks (iii) code availability for the detector. Our selection covers the research gap in the field of white-box concealment attacks on CPS anomaly detection. We consider five different anomaly detectors proposed in relevant prior peer-reviewed publications; namely, Auto Regressive model [19], Linear Time Invariants [29], Support Vector Machines [3,8], Process Invariants [13], and for each, we apply our proposed approach to achieve misclassification.

5.1 Attack Implementation and Hardware Setup

All experiments were performed on a laptop, equipped with Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, and 16GB of RAM. Experiments were performed either using Matlab 2019a, or Python 3.8.10 (depending on detector sources).

Implementation of the attack required: 201 lines of Matlab code for the AR model [19], 249 lines of Matlab code for the LTI model [29], 287 lines of Python code for the SVM [8] in this case we relied on the secml [25] library for gradients calculation by creating a wrapper for sklearn OneClassSVM, 324 lines of code for the PASAD detector [3], and 490 lines of code for the SFIG detector [13]. The code of our attacks is available at https://github.com/scy-phy/whiteboxDimva23.

5.2 Auto Regressive Models

AR models are a popular method used to model time series processes using linear equations starting from process data. Specifically, an Auto Regressive model (Eq. 8), tries to minimize the prediction error of sample X_t given the previous values $(X_0 \dots X_{t-1})$.

$$X_t = c + \sum_{i=1}^{p} \gamma_i X_{t-i} + \epsilon_t \tag{8}$$

where c is a constant, $\gamma_i, \dots, \gamma_p$ indicates the parameters of the model and ϵ_t is white noise. The parameters of the model are fitted using Yule-Walker equations [20]. AR models were applied to perform anomaly detection in cyber-physical systems [19,29]. The AR model is fitted starting from normal operations data, consequently, residuals observed during training are used to identify some thresholds or to tune Cumulative Sum (CUSUM) statistics. At test time the residuals are monitored to detect some deviations from expected behavior. **Availability**. We relied on the re-implementation by Erba et al. [12] and adapted it to work with SWaT dataset.

5.3 Linear Time Invariant Models

Linear Time Invariant (see Eq. 9) models were applied [9,29].

$$\begin{cases} s_{k+1} = As_k + Bq_k \\ x_k = Cs_k + Dq_k \end{cases} \tag{9}$$

where $k := kT$ and T is the sampling time. $s_k \in \mathbb{R}^n$ is the state of the system, i.e., the variables (directly or indirectly observable) of the process. $q_k \in \mathbb{R}^p$ is the input to the system. $x_k \in \mathbb{R}^q$ is the output of the system. $A \in \mathbb{R}^{n\times n}$ is the state matrix, relates the state s_k and its update s_{k+1}. $B \in \mathbb{R}^{n\times p}$ is the input matrix, relates the system input q_k and the state update s_{k+1}. $C \in \mathbb{R}^{q\times n}$ is the output matrix, relates the state s_k and the measured output x_k. $D \in \mathbb{R}^{q\times p}$ is the feed-through matrix, relates q_k and x_k.

Similarly to the AR model system identification (n4sid algorithm [30]) is applied to identify the LTI model parameters. Then the CUSUM algorithm performs anomaly detection. We identified an order 4 LTI model for the SWaT dataset. We use 22 sensors as input of the system, and the 3 tank level sensors as the output of the model. **Availability.** We relied on the re-implementation by Erba et al. [12] and we adapted it to work with the SWaT dataset.

5.4 SVM

We implement the SVM model proposed by Chen et al. [8], the proposed SVM is trained on the water tank sensor readings (π,π') measured at d timesteps from each other. To apply their proposed method to the SWaT dataset which contains exclusively benign samples in the training set, we switched to one class SVM classifier. Following the guideline in the paper we performed a grid search to tune the parameters of the SVM. The resulting model is OneClassSVM with linear kernel, γ=0.01, ν=0.02. We also tuned the parameter d. With our experiments, we tested d = 1, 10, 100, and 1000 s and found the best performance at 1 s. We note that the simulator used in the original paper has a faster sampling rate (5 ms) than the actual SWaT testbed sampling rate (1 s). **Availability.** This detector was made available to us by the authors of [8] upon request. We adapted it to work with the SWaT testbed dataset (originally it was proposed for the SWaT simulator).

5.5 PASAD

The PASAD model proposed in the work by Aoudi et al. [3] is based on the idea of Singular Value Decomposition (SVD) [17]. PASAD uses the time series data and applies a sliding window to them. Using the sliding window data samples, PASAD identifies a projection subspace where normal operations (i.e., the training data) sensor readings are projected to. Normal operations form a cluster in the projection subspace. At test time, if anomalous sensor readings occur on the system, the data points will be projected far away from the cluster obtained during training. The distance from the center of the cluster is used as a criterion to detect anomalies. **Availability.** This detector is available online on GitHub[2].

5.6 SFIG

The Systematic Framework for Invariant Generation (SFIG) method proposed by Feng et al. [13], based on the idea of process invariants (see Sect. 2), proposes a method to automatically find invariant rules starting from process data. The rules are generated based on three sets of predicates: distribution driven predicates, event driven predicates, and categorical predicates. Distribution driven predicates are generated fitting a Gaussian mixture model of the system, while categorical predicates are generated according to actuator states. Finally, event driven predicates are generated by fitting some linear models to capture critical values that trigger changes in actuator states. To perform anomaly detection, at each time step, sensor readings are tested against all the rules in the collection of identified rules. If a rule is not fulfilled an alarm is raised. **Availability.** This detector is available online on GitHub[3].

5.7 SWaT Dataset

SWaT [24] is a water treatment testbed located at the Singapore University of Technology and Design. It consists of a six-stage process for water treatment. Those six stages are controlled by interconnected PLCs, connected to Human Machine Interfaces (HMIs), Supervisory Control and Data Acquisition (SCADA) workstation, and a Historian. The SWaT dataset is a collection of data from 11 days of operations; 7 days were collected during the system in normal operation while 4 days were collected while 41 attacks were launched on the system. We rely on this dataset as it is commonly used in related research, notably, it was used to evaluate all the detectors from prior work that we test in this work against WBC.

6 Evaluation Results

In this section, we present the results of our evaluation. To answer to **R1**, we applied the five aforementioned detection mechanisms to the SWaT dataset [24] and attacked them with the proposed WBC. To answer **R2**, we verify the computational runtime of the proposed approach and the cost of the perturbations. Finally, to answer to **R3**, the results of the WBC attack methodology are compared against the performance when no concealment was applied to the data, and against the black-box attacks for CPS detectors [12].

For our proposed WBC attack we consider three variants. Namely, *WBC baseline*, where the WBC attack is applied to every set of sensor readings labeled

[2] https://github.com/mikeliturbe/pasad.

[3] https://github.com/cfeng783/NDSS19_InvariantRuleAD.

Table 2. WBC Attack on the AR model trained over SWaT sensor LIT301 (used as reference in prior work [3]). The WBC attacks evade the anomaly detection system (see original recall vs. WBC recall). μ indicates the mean, and σ the standard deviation. N indicates how many rows were modified by the attack †Note: technically NaN as the metric divides by 0.

						Elapsed (ms)		Euclidean D.		
Data	Acc	F1	Prec	Rec	FPR	μ	σ	μ	σ	N
Original	0.797	0.254	0.227	0.288	0.134	–	–	–	–	–
Prior Work [12]										
Replay	0.775	0.088	0.086	0.091	0.131	–	–	13.592	48.322	541
Random R.	0.832	0.501	0.389	0.702	0.151	–	–	12.832	45.903	541
Stale	0.788	0.186	0.173	0.201	0.131	–	–	17.341	55.804	522
Our WBC										
baseline	0.858	$(0)^{\dagger}$	0.000	0.000	0.024	0.004	0.014	11.046	40.578	515
NTP	0.860	$(0)^{\dagger}$	0.000	0.000	0.022	249.36	148.77	2.081	24.563	74
NA	0.879	$(0)^{\dagger}$	0.000	0.000	0.001	171.11	71.7	5.092	30.485	258

as 'anomalous' as ground truth (i.e., the attacker is manipulating the physical process), regardless if they are detected as anomalous or not. In this setting, the attacker iterates until the objective (Eq. 2) is minimized. This is the same setting considered by the attacks proposed by Erba et al. [12], and we use it for comparison. *No True Positives (WBC NTP)*, in this setting the WBC is applied to every set of sensor readings labeled as 'anomalous', which is also detected as anomalous by the anomaly detection system (i.e., physical anomaly correctly detected by the anomaly detection system). Finally, we consider the *No Alarms (WBC NA)*, in this setting the WBC is applied to every set of sensor readings that are detected as anomalous by the anomaly detection system (i.e., conceal also false positives). In WBC NTP and WBC NA settings, the attacker iterates until the label is changed.

We note that since there is the white-box assumption on the target detector, the attacker is assumed to access the prediction of the detector. Moreover, since the physical process manipulations are under the control of the attacker, the attacker knows when the physical process anomaly is occurring on the system (i.e., 'anomalous' ground truth in the SWaT dataset).

Evaluation Metrics. To assess the impact of the attack on the detection capability of the classifier we consider the following metrics: Accuracy, F1 score, Precision, Recall, and False Positive Rate. In particular, the Recall score gives us information on how the attack is capable of concealing the true state of the system from the anomaly detector. Elapsed time is measured to assess the mean computational overhead required by the WBC attack. Specifically, we measure average the time required to compute an adversarial example. Finally, we measure the Euclidean distance (L2) between the original sample p and the

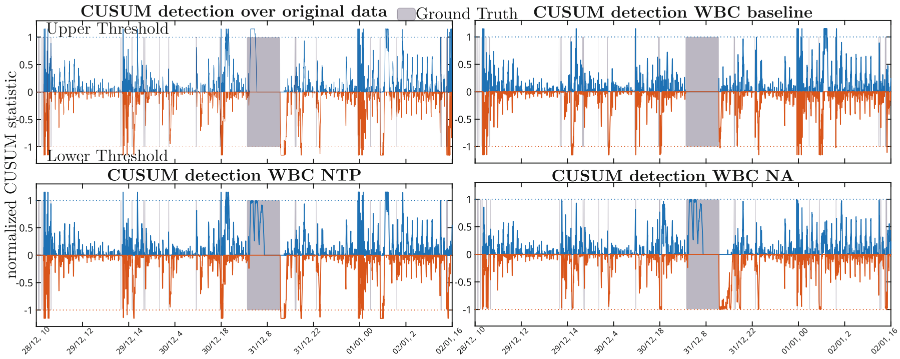

Fig. 4. Comparison of AR detection before and after the WBC attack. The concealment attack hides the anomalies in the process data. In the bottom figure (WBC NA) WBC is applied to all the readings even if no physical attack is present, this removes not only the True Positives but also the False Positives.

perturbed sample q to assess the perturbation required on the features by the attack. Moreover, to evaluate the minimal number of features under the control of the attacker we compute the Hamming distance (L0), as the number of sensors/actuators that were changed by the attack.

6.1 Auto Regressive

We apply the proposed approach to the AR detection model. In Table 2 we present the results of the WBC attack and compare them with the result from prior work black-box attacks [12], while Fig. 4 shows the impact of the WBC over the CUSUM statistics.

The AR detector precision and recall drop to 0 after the attack, this means that no more true positives are detected, and consequently, the F1 score becomes not defined as we have a division by zero. This result means that the detector is no longer capable of recognizing anomalies in the system. Looking at Fig. 4 we can also observe the difference between the three attack approaches (baseline, NTP, NA). WBC baseline brings the CUSUM error to 0 when the ground truth label reports 'anomalous', this happens because the attacker iterates until the loss is minimized. This is in contrast to WBC NTP and WBC NA for which the attacker stops iterating as soon as the alarm threshold is not surpassed anymore. Finally, we can notice the difference between the WBC NTP and WBC NA, the WBC NTP (as the name suggests) brings the True Positives to zero, while the WBC NA hides all the positives (both True Positive and False Positive).

Regarding the computational time, we observe that the WBC concealment attacks required hundreds of milliseconds to compute (while SWaT sampling time is 1 s). WBC baseline is sensibly faster because code optimization was used. As in the WBC baseline, we care of loss minimization, and we attack the AR model, we can achieve loss minimization in one step by selecting

Table 3. WBC Attack on the LTI model trained over SWaT.

						Elapsed (ms)		Euclidean D.			Ham. D.	
Data	Acc	F1	Prec	Rec	FPR	μ	σ	μ	σ	N	μ	σ
Original	0.962	0.815	0.987	0.694	0.001	–	–	–	–	–	–	–
Prior Work [12]												
Replay	0.879	0.008	0.233	0.004	0.002	–	–	69.60	210.7	53863	21.4	1.8
Random R	0.998	0.992	0.987	0.996	0.002	–	–	69.58	210.53	53863	21.4	1.8
Stale	0.887	0.126	0.845	0.068	0.002	–	–	66.05	211.53	53862	19.8	4.1
Our WBC												
baseline	0.884	0.081	0.785	0.043	0.002	121.0	326.1	67.25	218.92	53863	2.9	0.3
NTP	0.885	0.087	0.831	0.046	0.001	32.9	22.9	59.41	208.37	37385	2.9	0.5
NA	0.885	0.087	0.944	0.046	0.000	87.8	46.4	59.89	208.95	37881	2.9	0.5

$\epsilon = ||\nabla_x Loss(x, o)||_2$ in Eq. 3. For clarity, this is equivalent to changing Eq. 3 to $\delta = -\nabla_x Loss(x, o)$.

We compare the white-box concealment technique w.r.t. the black-box attacks proposed by Erba et al. [12] (See Table 2). As we can observe the white-box attacks outperform the black-box attacks in terms of concealment capability, as the black-box attacks never conceal all the True Positives (i.e. recall greater than 0). Finally, we can compare the Euclidean distances between the attacks. As we can observe in Table 2, the average perturbation is always lower for the WBC attacks w.r.t. prior work black-box attacks. This is because the white-box setting optimizes the samples to be optimal w.r.t. the past observed process data. This is instead impossible for black-box attacks. This can be observed by looking at the number of modified values (N) in Table 2, which is always in favor of the WBC NTP ad NA attacks. Since the AR model is univariate, we do not report the hamming distance (it would be 1 in any case).

6.2 Linear Time Invariant

We apply the WBC concealment attacks to the LTI model. Table 3 reports the results of our evaluation. The WBC concealment attacks evade the LTI detector, and the detector recall drops from 0.69 to 0. We can observe the impact of the NA attack that reduces also the number of False Positive Rate.

The required computational time of the WBC attacks is at most 120 ms, which is lower than the sampling time of the SWaT system (1 s).

Also in this case the Euclidean Distance of the perturbed samples is lower than in prior work attacks. Moreover, when looking at the Hamming Distance, we can observe how the number of features to be manipulated decreases (2.93 vs 28.4). This happens because our WBC is constrained to manipulate the output of the model x_k but cannot operate on the input q_k (see Eq. 9). This number tells us that an attacker which controls 3 out of the 25 features used by the model, can significantly reduce the classifier recall by reducing it from 0.694 to 0.046.

Table 4. WBC Attack on the SVM model trained over SWaT sensor LIT101, LIT301, LIT 401.

						Elapsed (ms)		Euclidean D.			Ham. D.	
Data	Acc	F1	Prec	Rec	FPR	μ	σ	μ	σ	N	μ	σ
Original	0.931	0.689	0.754	0.634	0.028	–	–	–	–	–	–	–
Prior Work [12]												
Replay	0.855	0.0	0.0	0.0	0.028	–	–	87.13	266.40	53897	5.99	0.12
Random R	0.855	0.0	0.0	0.0	0.028	–	–	87.45	266.03	53897	5.99	0.12
Stale	0.855	0.0	0.0	0.0	0.028	–	–	84.52	269.66	53896	5.84	0.55
Our WBC												
baseline	0.855	0.0	0.0	0.0	0.028	65.56	56.25	7.54	27.33	53934	5.99	0.07
NTP	0.855	0.0	0.0	0.0	0.028	103.84	33.13	7.54	27.33	34187	5.99	0.07
NA	0.880	0.0	0.0	0.0	0.000	107.22	61.49	10.71	36.74	45361	5.96	0.33

The number N is lower for WBC NTP and NA attacks when compared to prior work, i.e. 37881 vs 53863. When we compare the evasion performance of the attacks, we can observe that the WBC approach has comparable performance to the Replay and Stale attacks in terms of reduction of the model recall.

6.3 SVM

Table 4 reports the results of the evaluation of the proposed attacks on the SVM model. The proposed WBC concealment attack evades the SVM model and the recall drops from 0.63 to 0 in all the considered settings. We can observe how the NA approach differs by bringing the FPR to 0. The average computational time is at most 107 ms, which is lower than the sampling rate of the SWaT testbed. Since the detector is using the water tank levels measured at d timesteps of distance (π,π'), we constraint the adversarial example to modify only π' (3 features), this is consistent with our challenge C1.

When comparing the WBC attacks to prior work generic concealment attacks, we can observe that the Euclidean distance required by the WBC attack is lower, as well as the number of perturbed samples by the NTP and NA approaches. The Hamming distance remains almost the same, after d timestep of continuous attack (in our case 1 step) all the features in (π,π') are under the control of the attacker (6 features).

6.4 PASAD

In this section, we attack PASAD with our WBC approach. The results of the attack are summarized in Table 5. Also, in this case, the attacks are successful and the performance of the detector is compromised, as the recall drops close to 0 in all the three considered attacks. Differently from the previous case, the recall does not reach exactly 0, this is because there are a few instances in which the WBC is not reducing enough the distance from the PASAD cluster

Table 5. WBC results on PASAD trained on SWaT Dataset sensor LIT301 (used in the paper [3]). The WBC attacks evade PASAD. The WBC requires less than 4ms to compute. The Euclidean distance is smaller when compared to prior work attacks. Threshold 3×10^6.

						Elapsed (ms)		Euclidean D.		
Data	Acc	F1	Prec	Rec	FPR	μ	σ	μ	σ	N
Original	0.878	0.557	0.492	0.641	0.090	–	–	–	–	–
Prior Work [12]										
Replay	0.822	0.118	0.145	0.100	0.080	–	–	13.664	48.693	53859
Random R	0.819	0.083	0.106	0.069	0.079	–	–	13.341	47.829	53852
Stale	0.899	0.617	0.563	0.681	0.072	–	–	17.356	56.065	52013
Our WBC										
baseline	0.825	0.039	0.057	0.03	0.067	2.31	1.84	12.92	48.716	40962
NTP	0.818	0.008	0.011	0.006	0.072	3.91	7.95	8.265	94.526	24842
NA	0.870	0.004	0.025	0.002	0.012	2.6	2.44	10.431	48.302	33369

center. Similar to the previous experiment, we can see the difference between the baseline, NTP, and NA approaches. Again we can observe how the FPR rate reduces in the case of the NA setting. This time it reaches 0.012 meaning that there are few false positives.

Looking at the computational time required, the WBC algorithm finds the adversarial examples in 2.3 ms which is lower that the SWaT sampling time of 1 s. In this case, optimizations cannot be performed in the baseline setting. PASAD projects the univariate sensor readings into a subspace and tracks the distance of the projected time series form the centroid of the normal operations cluster. As explained with the research challenge C1, we assume we cannot change the whole time series sliding window but we manipulate just the last observation from the coming from the physical process. For this reason, the attack evades the detector by changing one sample at a time. Eventually, if the attack continues, all the samples in the sliding window are under the control of the attacker.

Finally, if we compare the performance of the white-box attacks w.r.t. black-box attack from prior work [12], we can observe that also in this case the WBC attacks are more effective than the black-box attacks in terms of concealment performance as the WBC recall score is always lower than in the case of the three attacks black box attacks from prior work. Looking at the Euclidean distance (Table 5), we can observe that the WBC attacks are on average less expensive than the black-box attack. Looking instead at the number of modified rows (N) we can observe that the WBC attacks are always less expensive than prior work.

6.5 SFIG

We then apply our attack method to the SFIG detector (see Table 6). In this setting, the WBC baseline and NTP coincide, because the invariant-based detector

Table 6. Attack against the SFIG detector on the SWaT dataset. The WBC baseline and NTP coincide because in invariant-based detectors alarms can be triggered only if rules are contradicted.

						Elapsed (ms)		Euclidean D.			Ham. D.	
Data	Acc	F1	Prec	Rec	FPR	μ	σ	μ	σ	N	μ	σ
Original	0.958	0.793	0.950	0.681	0.005	–	-	–	–	–	–	–
Prior Work [12]												
Replay	0.876	0.000	0.004	0.000	0.005	–	–	69.60	210.70	53863	28.4	5.3
Random R	0.893	0.240	0.797	0.141	0.005	–	–	69.58	210.53	53863	28.4	5.3
Stale	0.881	0.080	0.544	0.043	0.005	–	–	66.05	211.53	53862	27.4	8.4
Our WBC												
base./NTP	0.876	0.003	0.040	0.002	0.005	256.2	34.5	0.136	0.459	36704	2.8	0.5
NA	0.880	(0)†	0.0	0.0	0.0	354.3	264.6	0.141	0.465	38643	2.8	0.6

triggers only when rules are contradicted (i.e., there is no loss to minimize). In this experiment, we consider attacks that deal with the 51 features of the SWaT dataset, as the detector considers them all together.

Also in this setting, the detector was evaded by the attacks reducing the performance of the detector from 0.68 to 0 in both cases. Also here we can appreciate the difference induced in the false positive rate in the two attack settings, the NA setting leaves no false positives.

In the WBC baseline/NTP, we notice that the recall is not 0.000, this is because we noticed that there is an artifact in the detection rules which causes a contradictory set of rules. This means that applying our attack to fix the data to turn off the alarms, triggers another rule in contradiction. This makes it impossible to turn off the alarm in a row of data.

Looking at the computational time required by the attacks in this case, we are in the order of 200/300 ms which is lower than the SWaT sampling time. Regarding the Euclidean distance, from Table 6 we can observe that the WBC attacks are less expensive than the attacks from the black box attacks from prior work, the proposed WBC attacks are always 2 orders of magnitude closer to the original values, meaning that features need to be slightly modified to achieve the goal. Also, the number of modified rows N (as in the previous experiments) is smaller. In this multivariate setting, we can also measure the number of features that were modified by the attack (i.e., the Hamming distance). As we can observe in Table 6, out of the 51 features in the SWaT dataset, WBC attacks modify on average 2.8 features (maximum 7 features out of 51), while prior work attacks modify on average ~30 features (maximum 37 features out of 51).

Finally, if we compare the performance of the WBC w.r.t. attacks from prior work [12], we can observe that on one hand, the WBC NTP have a similar performance to Replay and Stale attacks from prior work, but on the other hand, as we pointed out before the WBC NTP is overall cheaper in terms of features that are modified by the attack.

Table 7. Summary of findings on our white-box concealment attacks. '# Manipulated' refers to the number of features that needed to be manipulated by the attacker.

Method	Attack works	# Manipulated	Computational Cost $\leq$ 1 s
AR	✓	1/1	249 ms
LTI	✓	3/25	120 ms
SVM	✓	3/6	107 ms
PASAD	✓	1/1	4 ms
SFIG	✓	3/51	360 ms

7 Discussion and Conclusion

In this section, we discuss the answers to our research questions. In Table 7 we summarize our findings. With respect to question **R1**, we tested three variations of the proposed WBC attacks, over five different anomaly detection systems. To do so the attacker has to deal with challenge **C1** (i.e., manipulate only the last sensor value) and with challenge **C2** (i.e., transform to differentiable detectors which do not use a loss function). As a result, we found that the evaluated detectors are vulnerable to white-box concealment attacks, i.e., for all the tested detectors, the recall score drops to 0 or very close to it. This result demonstrates that the proposed attack methodology can affect a wide range of anomaly detectors for cyber-physical systems, affecting their detection performance with often little perturbation of the sensor data (in terms of Hamming and Euclidean distance). Our analysis reveals that only a low number of resources need to be under the control of an attacker to subvert the classification outcome of the target anomaly detector. For example, for the LTI and the SFIG, our results show that is enough to control ∼3 features of the multivariate detector to conceal attacks.

With respect to research question **R2**, we measured the time to compute the adversarial examples (worst case ∼350 ms), and we found that runtime manipulations are possible, as it is possible to compute manipulations faster than the system's sampling rate of the SWaT system (1 sample per second). We note that temporal constraints for adversarial examples are not generally investigated by related adversarial machine learning literature, as in other domains adversarial examples can be pre-computed (for example in the image classification domain) and do not need to be adapted based on the context.

Concerning research question **R3**, we compared the proposed attacks with black-box attacks from prior work [12], in particular in terms of concealment performance and Euclidean distance. We found that our proposed WBC attacks are more effective (e.g., F1 score of the PASAD model is always lower in the WBC attack 0.039 vs 0.083 from prior work). Moreover, in general, our attacks require less manipulation than prior work attacks, (e.g., the Euclidean Distance in the SFIG case is 0.136 vs 66.05 from prior work, same the holds for the Hamming distance WBC 2.81 vs 27.41 from prior work).

Our results demonstrate that it is possible to evade a wide range of detectors while reducing the number of samples that need to be manipulated (compared to prior black-box concealment attacks). Those findings highlight the need for further research and constructive discussion about guarantees for CPS anomaly detectors against adversarial manipulation. As such we see our contribution toward the robustness and reliability of CPS detectors against adversarial examples.

References

1. Adepu, S., Mathur, A.: Distributed detection of single-stage multipoint cyber attacks in a water treatment plant. In: Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS) (2016)
2. Ahmed, C.M., et al.: Noiseprint: attack detection using sensor and process noise fingerprint in cyber physical systems. In: Proceedings of the Asia Conference on Computer and Communications Security (AsiaCCS) (2018)
3. Aoudi, W., Iturbe, M., Almgren, M.: Truth will out: departure-based process-level detection of stealthy attacks on control systems. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS). ACM (2018)
4. Biggio, B., et al.: Evasion attacks against machine learning at test time. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) Machine Learning and Knowledge Discovery in Databases, pp. 387–402 (2013)
5. Cao, Y., et al.: Adversarial sensor attack on lidar-based perception in autonomous driving. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS), p. 2267–2281. ACM, New York, NY, USA (2019)
6. Cárdenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., Sastry, S.S.: Challenges for securing cyber physical systems. In: Workshop on Future Directions in Cyber-physical Systems Security. DHS, July 2009
7. Cervini, J., Rubin, A., Watkins, L.: Don't drink the cyber: extrapolating the possibilities of Oldsmar's water treatment cyberattack. In: International Conference on Cyber Warfare and Security, vol. 17, pp. 19–25 (2022)
8. Chen, Y., Poskitt, C.M., Sun, J.: Learning from mutants: using code mutation to learn and monitor invariants of a cyber-physical system. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 648–660. IEEE (2018)
9. Choi, H., et al.: Detecting attacks against robotic vehicles: a control invariant approach. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS) (2018)
10. Dahlmanns, M., Lohmöller, J., Fink, I.B., Pennekamp, J., Wehrle, K., Henze, M.: Easing the conscience with OPC UA: an internet-wide study on insecure deployments. In: Proceedings of the ACM Internet Measurement Conference (2020)
11. Erba, A., et al.: Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC), December 2020
12. Erba, A., Tippenhauer, N.O.: Assessing model-free anomaly detection in industrial control systems against generic concealment attacks. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC). Austin, USA, December 2022

13. Feng, C., Palleti, V.R., Mathur, A., Chana, D.: A systematic framework to generate invariants for anomaly detection in industrial control systems. In: Proceedings of Network and Distributed System Security Symposium (NDSS) (2019)
14. Galloway, B., Hancke, G.P., et al.: Introduction to industrial control networks. IEEE Commun. Surv. Tutor. **15**(2), 860–880 (2013)
15. Garcia, L., Brasser, F., Cintuglu, M.H., Sadeghi, A.R., Mohammed, O., Zonouz, S.A.: Hey, my malware knows physics! attacking PLCs with physical model aware rootkit. In: Proceedings of Network and Distributed System Security Symposium (NDSS), February 2017
16. Goh, J., Adepu, S., Tan, M., Lee, Z.S.: Anomaly detection in cyber physical systems using recurrent neural networks. In: 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), pp. 140–145. IEEE (2017)
17. Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. In: Bauer, F.L. (eds.) Linear Algebra, vol. 2, pp. 134–151. Springer, Heidelberg (1971). https://doi.org/10.1007/978-3-662-39778-7_10
18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR abs/1412.6572 (2014)
19. Hadžiosmanović, D., Sommer, R., Zambon, E., Hartel, P.H.: Through the eye of the plc: semantic security monitoring for industrial processes. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC), pp. 126–135. ACM, New York, NY, USA (2014)
20. Hayes, M.H.: Statistical Digital Signal Processing and Modeling. Wiley, Hoboken (2009)
21. Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., Khalgui, M.: Micro air vehicle link (MAVlink) in a nutshell: a survey. IEEE Access **7** (2019)
22. Kravchik, M., Shabtai, A.: Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy, pp. 72–83. ACM (2018)
23. Lee, E.A.: Cyber physical systems: Design challenges. Technical report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, January 2008
24. Mathur, A., Tippenhauer, N.O.: SWaT: a water treatment testbed for research and training on ICS security. In: Proceedings of Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), April 2016
25. Melis, M., Demontis, A., Pintor, M., Sotgiu, A., Biggio, B.: secML: a Python library for secure and explainable machine learning. arXiv:1912.10013 (2019)
26. Pierazzi, F., Pendlebury, F., Cortellazzi, J., Cavallaro, L.: Intriguing properties of adversarial ML attacks in the problem space. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 1332–1349. IEEE (2020)
27. Shen, J., Won, J.Y., Chen, Z., Chen, Q.A.: Drift with devil: security of multi-sensor fusion based localization in high-level autonomous driving under GPS spoofing. In: Proceedings of the USENIX Security Symposium, pp. 931–948, August 2020
28. Taormina, R., Galelli, S.: A deep learning approach for the detection and localization of cyber-physical attacks on water distribution systems. J. Water Resourc. Plann. Manag. **144**(10) (2018)
29. Urbina, D., et al.: Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS), October 2016
30. Van Overschee, P., De Moor, B.: N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica **30**(1), 75–93 (1994)

31. Weinberger, S.: Computer security: is this the start of cyberwarfare? Nature **174**, 142–145 (2011)
32. Wikipedia, t.f.e.: Colonial pipeline ransomware attack. https://en.wikipedia.org/wiki/Colonial_Pipeline_ransomware_attack. Accessed 21 May 2022
33. Zizzo, G., Hankin, C., Maffeis, S., Jones, K.: Adversarial attacks on time-series intrusion detection for industrial control systems. In: IEEE TrustCom (2020)