Assessing Model-free Anomaly Detection in Industrial Control Systems Against Generic Concealment Attacks

Alessandro Erba
CISPA Helmholtz Center for Information Security
and Saarbrücken Graduate School of Computer Science,
Saarland University
Saarbrücken, Germany
alessandro.erba@cispa.de

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

ABSTRACT

In recent years, a number of model-free process-based anomaly detection schemes for Industrial Control Systems (ICS) were proposed. Model-free anomaly detectors are trained directly from process data and do not require process knowledge. They are validated based on a set of public data with limited attacks present. As result, the resilience of those schemes against general concealment attacks is unclear. In addition, no structured discussion on the properties verified by the detectors exists.

In this work, we provide the first systematic analysis of such anomaly detection schemes, focusing on six model-free process-based anomaly detectors. We hypothesize that the detectors verify a combination of temporal, spatial, and statistical consistencies. To test this, we systematically analyse their resilience against generic concealment attacks. Our generic concealment attacks are designed to violate a specific consistency verified by the detector, and require no knowledge of the attacked physical process or the detector. In addition, we compare against prior work attacks that were designed to attack neural network-based detectors.

Our results demonstrate that the evaluated model-free detectors are in general susceptible to generic concealment attacks. For each evaluated detector, at least one of our generic concealment attacks performs better than prior work attacks. In particular, the results allow us to show which specific consistencies are verified by each detector. We also find that prior work attacks that target neural-network architectures transfer surprisingly well against other architectures.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; • **Computing methodologies →** Anomaly detection; Ensemble methods.

KEYWORDS

Concealment attacks, Anomaly Detection, Industrial Control

ACSAC '22, December 5–9, 2022, Austin, TX, USA

ACM ISBN 978-1-4503-9759-9/22/12…$15.00
https://doi.org/10.1145/3564625.3564633

ACM Reference Format:
Alessandro Erba and Nils Ole Tippenhauer. 2022. Assessing Model-free Anomaly Detection in Industrial Control Systems Against Generic Concealment Attacks. In *Annual Computer Security Applications Conference (ACSAC '22), December 5–9, 2022, Austin, TX, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3564625.3564633

1 INTRODUCTION

Industrial Control Systems (ICS) enable the control of physical processes by the interaction of computers, communication networks, sensors, and actuators. Examples of such systems are water distribution systems, manufacturing plants, and smart grids. ICS are threatened by malicious actors, who aim to damage or destabilize physical processes [48]. Such attacks could be conducted through local physical-layer manipulation [30, 43, 50], compromise of local controllers [1, 15], or local network traffic [37].

To address those threats in a legacy compliant way, a number of intrusion and anomaly detection schemes for ICS and in general for Cyber-Physical Systems (CPS) have been proposed in the literature [3, 4, 9, 13, 17, 27, 40, 45]. Process-based anomaly detection [13] schemes leverage actuator and sensor data to detect anomalies in the process and operations of the system. We differentiate *model-free* and *model-based* anomaly detectors, depending on whether a physical model is leveraged by the scheme. Model-based detectors leverage the physical process directly (e.g., by using a set of linear or non-linear equations describing the process physics), those anomaly detection systems are harder to be constructed as the modeling of the process requires a plant specific engineering effort [10, 36, 45]. Model-free detectors approximate the physical process indirectly and use Machine Learning [3], Deep Learning [17, 27, 40], System Identification [4, 19, 45], and Data Mining [13] techniques for the training of the classifier/predictor. Recently, model-free approaches were also introduced in commercial products for anomaly detection in ICS by major security vendors [24]. Limited security analysis of model-free schemes has been performed [11], in particular reconstruction-based detectors (using neural networks) were attacked in a white and black-box fashion in unconstrained and constrained settings. Until now, it is unclear if such attacks could also apply to other detector designs. In this work, we evaluate attacks on a wide range of state-of-the-art model-free detectors, in particular AR models [19], PASAD [4], SFIG [13] and Autoencoders [17, 27, 40].

Training and evaluation of anomaly detectors require operational and attack data [16, 23, 32, 42]. Due to the difficulty in creating

realistic datasets (requiring practical testbeds with implementations of attacks, or detailed cyber-physical simulators), the diversity of attacks represented in them is limited. In particular, the major class of concealment attacks is insufficiently presented in the datasets. As result, the evaluation would cover a subset of threats the systems are designed to protect against. This leads to unexpected vulnerabilities as shown for Deep-Learning detectors [39].

In this work, we address three research questions. **RQ1**: *Are there fundamental properties that are checked by the anomaly detectors in order to identify anomalies?* **RQ2**: *How resilient are different model-free anomaly detection approaches against generic concealment attacks?* and **RQ3**: *How do generic concealment attacks compare against prior work that targeted neural network-based detectors?*

To answer **RQ1**, we show that manipulations of the physical process will violate the consistency of the system. In particular, we differentiate between spatial, temporal, and statistical consistency. We then consider three classes of generic concealment attacks that each specifically violate one of those consistencies. For each attack primitive, the goal of the attacker is to evade the detection of anomalous system states by manipulating selected sensor values.

To answer **RQ2**, we propose a framework to test process-based anomaly detection systems against our three distinct concealment attack primitives. We apply this framework to six prior work model-free anomaly detection systems.

To answer **RQ3**, we also evaluate the six anomaly detection systems against attacks from prior work that specifically targeted neural network-based detectors [11].

We summarize our main contributions as follows:

- We provide the first systematic analysis of model-free process-based anomaly detectors.
- We introduce the concept of spatial, temporal, and statistical consistency to describe properties implicitly verified by the detectors, and how they relate to a state-space representation of the process.
- We practically implement three generic concealment attacks that are not so far represented in related public datasets, and demonstrate the attacks' efficacy against six model-free anomaly detectors from literature [4, 13, 17, 19, 27, 40]. We show that (surprisingly) even very basic attacks are effective (e.g., leading to a Recall of 0.0). We also evaluate prior work attacks [11] against the detectors.

In Appendix C, we also show how to construct an ensemble detector that reliably detects process anomalies and the evaluated concealment attacks. Our ensemble-based detector is resilient against all evaluated concealment attack primitives, while also performing well in detecting the *original process anomalies in the dataset*. Our method is inspired by Subspace-based State Space System Identification techniques [47] from the domain of control theory.

Our implementation of the concealment attacks (extending public datasets), and our ensemble countermeasure are publicly available at github.com/scy-phy/ICS_Generic_Concealment_Attacks.

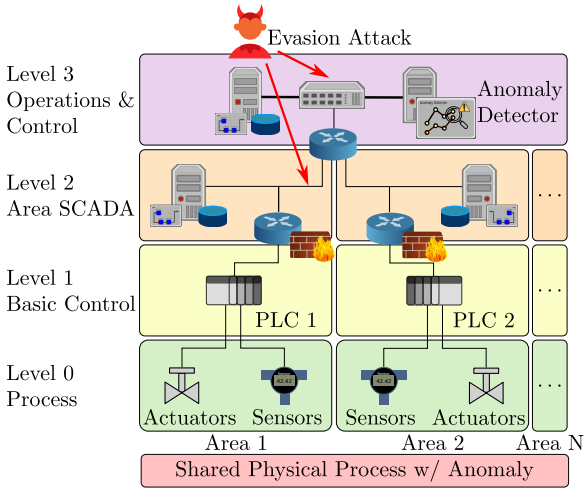

Figure 1: An Industrial Control System organized according to the Purdue Enterprise Reference Architecture (PERA). An attacker tampers with the physical process (Level 0), and performs a concealment attack to remain stealthy (Level 3).

2 BACKGROUND

Industrial Control Systems. Industrial Control Systems (ICS) are widely used to automate processes in production plants and facilities. ICSs are composed of interconnected Cyber and Physical components to interact with the physical environment. Cyber components comprise the hardware and software that are used to control the process. The Purdue Enterprise Reference Architecture (PERA) [49] is the networking architecture for ICS systems, adopted in the ANSI/ISA-95 standard. The PERA model divides the ICS system into six layers, the layers from 0 to 3 constitute the manufacturing zone, while levels 4 and 5 constitute the enterprise zone. For the purpose of this work, we focus on layers 0 to 3 (Figure 1).

ICS processes are constituted by sub-processes (e.g., electrical substations). To control the different sub-processes, we can divide the PERA architecture into areas. Each area is composed of levels 0 to 2. At level 0, the sensors and actuators are deployed to interact with the physical process. At level 1, Programmable Logic Controllers (PLC) are deployed to implement the system's control logic (e.g., Proportional Integral Derivative (PID) controller). PLCs observe sensor values and send commands to actuators. At level 2, local Supervisory Control and Data Acquisition (SCADA) systems are deployed for Area supervisory purposes. At level 3, the different areas are aggregated to perform plant monitoring. The aggregation can happen locally for co-located areas (i.e., through a router), or via the Internet/WAN for distributed areas. Monitoring can occur through Human Machine Interfaces (HMIs) and anomaly detectors.

Datasets. Datasets to evaluate detection schemes are collections of sensor readings and actuator states recorded from ICS testbed or process simulation [16, 23, 32, 42]. Data are usually organized into two data captures. The first 'train dataset' is recorded during normal operating conditions, and the second 'test dataset' is recorded while anomalies caused by an intruder occur. Batadal Dataset was

Table 1: Taxonomy of black-box attacks in CPS. For each attack we report if it was represented in the dataset ( ) or not (#) and in which research paper it was considered. † indicates the new attack that we have identified for this work.

	L0/ L2	Attack	Batadal	SWaT[32]	WaDI[23]
Overt		Random Manipul. [2, 3]			
		Linear Transform. [2, 3]			
		Stale Data L0 [29]	#	#	#
Evasion	Stealthy (L0)	Boiling Frog			
	Concealment	Full Replay [11, 33]	#	#	#
		Constr. Replay [2, 11, 40, 42]		#	#
		Random Replay†	#	#	#
		Stale Data L2	#	#	#
		Learning-based [11]		#	

released as part of the Batadal competition [42]. The dataset is generated through EpanetCPA [41]. The water distribution network simulated in the dataset is C-town [35]. Data are divided into three sets; the first ('train dataset') contains 1 year of simulation under normal operating conditions, the second [5], and the third [6] (test dataset 1 and test dataset 2) contains 14 attacks (7 attacks each). 43 variables are captured by the dataset, continuous values for sensor readings and discrete for actuators with a sampling time of 1 hour. An updated version of the dataset was made available [21] with the paper [11]. In this work, we refer to this updated version.

Classi cation of ICS Process Manipulation.

We differentiate attacks between white-box (attacker knows all details of detectors and/or process) and black-box attacks (attacker does not have details on detector and/or process). In this work we focus on generic black-box attacks. White-box attacks were explored in prior work [11, 45]. A number of black-box attacks on ICS processes have been proposed in prior work. Unfortunately, there is no systematic classification of the attacks so far. In this work, we classify those attacks as follows (see Table 1). Attacks are either of type *Overt or Evasion* (i.e., *does the attacker attempt to hide from a detector?*). We further differentiate Evasion attacks as either *stealthy or concealment*: Manipulated sensor values are identical at all receivers (stealthy), or they can differ between the receivers in the process and the SCADA/detector (concealment). For example, hidden manipulation at L0 requires stealthy attacks – as both the process and the SCADA observe the manipulation. In contrast, separate manipulation at L0 and L2 allow concealment attacks towards the SCADA/detector.

Overt Attacks. In this category we cover 'Random Manipulation' attacks in which the attacker changes a sensor/actuator to a different value without engineering of the spoofed value [2, 3]. There can be several reasons for this manipulation, for example, the attacker plans to destabilize the physical process or break some industrial equipment. The second category is the 'Linear Transformation' attacks [2, 3], where the attacker adds a constant offset or sets the sensor reading to a specific critical value to destabilize the physical process and cause the wrong control decision. The last example in this category is the 'Stale Data' attack [29], where the attacker launches a DoS attack on L0 industrial communications, leading to receivers falling back to using the last received value, which causes the system to perform erroneous control actions.

Evasion Attacks. In this category we consider two subcategories. The first category contains stealthy attacks, in particular, the 'Boiling Frog' attack in which the attacker manipulates the process characteristics slowly to drive the system to unsafe states without triggering an alarm. Examples of these attacks can be found in all the datasets that we analyzed in Table 1. The second category is concealment attacks, which attempt to hide anomalous sensor readings to the SCADA/detector by reporting the erroneous sensor readings. The first representative attack is the 'Replay' attack [33], where the attacker replays recorded sensor readings occurred in the past. If the attacker can replay all the sensor readings we have the 'Full Replay', otherwise the 'Constrained Replay' [42]. Replay attacks are very challenging to detect as the sensor readings do not present anomalies. The second category we have 'Random Replay' attack that we introduce in this work, see details in Section 3. The third category is the 'Stale Data' attack, where the attacker performs the DoS to L2 devices to make the SCADA blind w.r.t. what happens on the physical process. Finally, there are Learning-based attacks [11], in contrast to the other attacks this attack requires real-time calculations to compute the spoofing samples and was proposed to specifically evade reconstruction-based detectors.

From Table 1, it is clear that the most widely used datasets in this domain do not provide examples of all attack classes, and thus so far not been considered for the evaluation of existing model-free detectors.

3 RESEARCH QUESTIONS AND ASSUMPTIONS

3.1 Research Questions and Challenges

Limitations of Prior Work. In prior work, process-based anomaly detection schemes have been proposed to detect the effects of adversarial manipulations of an industrial process. In particular, model-free anomaly detection schemes aim to achieve this goal without explicit knowledge of the physical process. Unfortunately, the datasets used to train those schemes did not contain the important class of concealment attacks (popularized by Stuxnet [48]), in which the attacker aims to hide anomalies by manipulating the reported sensor data (see Section 2). In addition, while it was observed that different detectors appeared to be more suitable towards detecting specific attacks, no systematic analysis of the abstract data properties verified by the detectors was performed. Even when 'temporal and spatial correlations' were mentioned, they were not further analysed or specified [11].

Research Questions. To address this gap, in this work we address the three research question presented in the introduction.

Overall, answering the questions will allow us to a) provide guidance toward the design of future process-aware anomaly detectors, b) provide more complete datasets for detector design and evaluation, and c) better understand the threat of generic concealment attacks that target many different detector designs at once.

Challenges. Investigating the aforementioned research questions is challenging for a number of reasons. *i)* First, concealment attacks have not been systematically investigated in prior work. As there is no prior exhaustive enumeration of attack types, it is also unclear if datasets used to train and test detectors are comprehensive in

the types of attacks they cover, and what types of attacks are successfully detected by the resulting schemes. In addition, prior work usually only presents the high-level concepts behind attacks, and does not provide reference implementations or datasets featuring the attacks. *ii)* Second, prior work detectors are often difficult to replicate, as private datasets are used, source code is not shared (including, e.g, hyper-parameters), or custom evaluation criteria are used (see Section 5). That implies that any systematic investigation of multiple detectors will require the design and implementation of a common framework that allows to evaluate several detectors over a set of common datasets, using identical performance metrics.

3.2 System Model

We consider an Industrial Control System as depicted in Figure 1. The industrial architecture is organized according to the Purdue Enterprise Reference Architecture [49] (see Section 2). Level 0 consists of a number of sensors and actuators, connected to a controller in Level 1 (e.g., PLC). The controller reports local sensor data to Level 2, where the local area SCADA is deployed (the ICS consists of multiple different areas). At Level 3 the data gathered from the different Level 2 areas are aggregated and analyzed by a process-based anomaly detector that uses the sensor data to classify the system state as anomalous or normal.

3.3 Attacker Model

No Process and Detector Knowledge & Stealth. In contrast to assumptions in prior work [3, 4, 13, 45], our black-box attacker is weaker as they do not know process physics, i.e., they are unaware of the physical properties of the system and the impact they have on the multivariate temporal series generated by sensor readings (spatial consistency, temporal evolution, and statistical properties). Prior work discussed white-box attacks [45], and has already shown that if the attacker has detailed white-box knowledge, the best a countermeasure can do is to prolong the time until the attack succeeds (or reduce the impact, but not fully prevent it). In this work (see Section 6), we show that even black-box attacks (for which such detailed knowledge is not required) are successful for the evaluated prior work detectors. Moreover, the attacker does not know the inner working of the anomaly detector and cannot access its parameters and detection scores. Despite these constraints, the attacker aims to hide an anomaly in the physical process from the anomaly detector, which would instead lead to an alarm by the detector. The attacker attempts to perform such a concealment attack to increase the overall damage caused over time, and cover the attacker's traces (see Stuxnet [48]).

Capabilities. As in the prior work detectors summarized in Section 5, our attacker is assumed to be capable of i) drop traffic towards the detector, ii) manipulate traffic towards the detector, and iii) eavesdrop traffic towards the detector. We will also investigate attacks that do not even require eavesdropping or manipulation of traffic (i.e., stale data attacks). We note that eavesdropping and manipulating traffic can be achieved in many ways due to missing security in industrial protocols, e.g., wireless jamming, packet dropping by attackers controlling forwarding devices, routing and ARP-based attacks, etc. [34, 37, 46]. According to prior work [18, 25, 38], the detailed process knowledge of complex systems is commonly not assumed, and obtaining it is challenging (if not impossible) for many attackers. We also note that the analyzed model-free detectors were proposed for settings where even the system operators have insufficient process knowledge to simulate the process fully.

Attack duration. Attacker goal is to conceal ongoing anomalies on the system, so we assume that the attacker launches the concealment while there is an ongoing anomaly on the system. As we rely on datasets with labeled anomalies, the duration of a concealment attack is determined by the 'under attack' label in the dataset.

Manipulation Constraints. As the physical process observed by the anomaly detector can encompass large areas with multiple sites (e.g., networks, substations, plants, etc.), we will also evaluate *constraints* on the attacker. For example, the attacker might only be able to manipulate a subset of the sensors as seen by the detector.

4 CONSISTENCIES & THEIR VERIFICATION

While general properties of the physical process sensor data are often (implicitly or explicitly claimed to be) verified by ICS and in general CPS anomaly detection systems, no consistent understanding or investigation of those properties was proposed in CPS anomaly detection works. To address this gap (**RQ1**), we introduce three types of consistency that hold in the process sensor data by leveraging the notion of State-space representation to describe the deterministic behavior that characterizes a CPS. We call the consistencies *temporal*, *spatial*, and *statistical* consistency. Based on the identified consistency properties we then provide a high-level description of the three generic concealment attacks that were not part of prior work evaluation. Figure 2 provides a simplified graphical visualization of the three considered concealment attacks.

4.1 State-space representation

Physical systems behavior is deterministically modeled with the so-called State-space representation [8], Equation 1 represents a discrete-time system. This representation combines the input, the system state, and the physical properties of the system to derive the evolution of the state and the output of the system. This deterministic representation captures the relation between the system's physical properties and allows its control.

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + v_k \\ y_k = Cx_k + Du_k + w_k \end{cases} \tag{1}$$

Where $k := kT$ and T is the sampling time. $x_k \in \mathsf{R}^n$ represents the state of the system, which is defined as the set of variables (directly or indirectly measurable) that characterize the physical system at a given time. This set of variables defines a Euclidean-space i.e., the State-space, and the state of the system at time k, i.e., x_k is a vector in the State-space. $u_k \in \mathsf{R}^p$ represents the input (or control vector) to the system, it influences the state of the system x_k and its output y_k. In a feedback control loop, u_k is the output of the controller. $y_k \in \mathsf{R}^q$ represents the output of the system, and it can be measured with sensors and it is influenced by the input u_k and the state x_k. $v_k \sim WN(0, V) \in \mathsf{R}^n$ and $w_k \sim WN(0, W) \in \mathsf{R}^q$ are white noise disturbances and $V \in \mathsf{R}^{p \times p}$, $W \in \mathsf{R}^{q \times q}$ are the noise variance matrices. $A \in \mathsf{R}^{n \times n}$ is the state matrix, it contains the coefficients of the physical relationship between the state x_k and its update x_{k+1}. $B \in \mathsf{R}^{n \times p}$ is the input matrix, it contains

the coefficients of the physical relation between the system input u_k and the state update x_{k+1}. $C \in \mathsf{R}^{q \times n}$ is the output matrix, it contains the coefficients of the relation between the state x_k and the measured output y_k. $D \in \mathsf{R}^{q \times p}$ is the feed-through matrix, it contains the coefficients of the dependence between u_k and y_k.

Anomalies. In a time-invariant system, A, B, C, D are constants. In case of an attack on the physical process, at least one of those matrices is changed (as the matrices represent the physical process). In other words, the changed process becomes inconsistent with the normal process. We introduce in detail three different types of inconsistencies: spatial, temporal, and statistical.

4.2 Spatial Consistency

Spatial consistency refers to the correlation among quantities measured at the same instant in the physical process (referred to as attribute correlations by Illiano et al. [22]). This correlation depends on the physical process and control action. Considering the state-space representation of a Linear Time-Invariant system, the output y_k is observed by a set of sensors. Given the system state x_k, (i.e., $Ax_{k-1} + Bu_{k-1}$) and input u_k of the physical system, the values of the output features are correlated according to the equation:

$$\hat{y}_k = Cx_k + Du_k \tag{2}$$

An anomaly detector should correctly exploit those physical correlations among features to identify attacks occurring over the system. For example, if an attacker performs a concealment attack on a subset of sensors in a system, this can break expected correlation between unmanipulated and manipulated sensors. Even a stateless detector, that only verifies the current state of the system could potentially detect such an anomaly. We can explain it with an intuitive example: consider a public place monitored by two CCTV devices. The attacker manipulates the images of one of the cameras (e.g., by replaying old images), but not the other one. Thus, an observer is able to detect a violation of spatial consistency as both cameras do not show the same scene. The same holds in an ICS, the same process is measured with multiple correlated sensors and an attacker replaying the values of few sensors causes inconsistencies.

4.3 Temporal Consistency

Temporal consistency refers to the temporal evolution of a sensor reading and how it unfolds according to the process physics and control action. Considering the state-space representation of a Linear Time-Invariant system (see Section 4.1), the update of the state x_{k+1} captures the temporal dependence between x_{k+1}, x_k and u_k according to A and B matrices. This relation can be used to predict the output of the system at time $k+1$, as the output depends on the estimated state at time k.

$$\hat{y}_{k+1} = C(Ax_k + Bu_k) + Du_k \tag{3}$$

Anomaly detectors should check the temporal evolution of the sensed value to verify if an anomalous unfolding is occurring over the system. For example, in a sensor spoofing attack, the temporal evolution of the spoofed data within might not follow the system's physics. Using the CCTV example used before, if only one camera is monitoring the public place, the start of a replay attack can be detected due to the sudden change of scenery (e.g., if the replayed video does not match the time of day). Thus, the attacker caused a temporal inconsistency in the video stream.

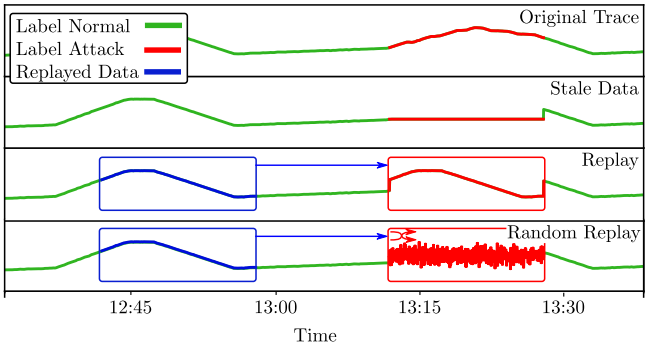

Figure 2: Visualization of attacks' effects on sensor data as seen by anomaly detector (simplified, single feature). 'Original Trace' contains anomalous data that triggers a detector.

4.4 Statistical Properties

Sensor readings in the analyzed multivariate temporal series are characterized by a statistical distribution. Those properties are derived from the process that is generating the data, i.e., the matrices A, B, C, D of the system in State-space representation, control inputs and disturbances in the process and in the sensors v_k, w_k.

Anomaly detection can leverage statistical properties to spot anomalies. For example, each sensor reading is characterized by a proper mean and standard deviation, deviation from the expected distribution can trigger alarms, as the process generating the data has changed and this can be caused by the manipulation of the physical process caused by an attacker. In [44], the authors showed that features in ICS datasets often have different statistical distributions in train and test data.

4.5 Verification of Consistencies

Given the three consistency properties, we verify if they are leveraged by anomaly detection systems. In order to do so, we identify three representative generic concealment attacks that break (one by one) the three consistency properties. If the attacks are successfully evading a detector, the detector is not correctly verifying the tested consistency property. As the attacks are designed to test the detectors, they are not necessarily optimized for unrelated metrics, e.g., minimal effort for the attacker. We note that those attacks are not present in the original datasets used for the evaluation of detectors.

Constrained Replay A ack. Constrained Replay is a variant of the full replay attack (e.g., discussed by the authors [11, 33]). Replay attacks conceal anomalies according to sensor readings observed in the past (e.g., by signal eavesdropping). This represents a relatively strong attack, as the attacker is required to record sensor readings for a certain amount of time before starting the attack. In constrained replay attacks, the attacker has a limited capability to spoof sensor readings and can replay sensor readings coming from a subset of PERA Level 2 area SCADA. Notably, also Stuxnet attack [48] resorted to a replay attack to conceal the true state of the system and avoid triggering alarms in the target industrial system.

Table 2: Attacks tested and their expected violation of consistency types. ●violates consistency.

	Consistency Tested		
Method	Spatial	Temporal	Statistical
Constrained Replay	●	-	-
Random Replay	-	●	-
Stale Data	-	-	●

While full replay attacks do not break any of the three consistency properties, constrained replay attacks are useful to understand if a detector correctly models the spatial consistency among process sensor readings. When the attacker performs the replay on a subset of sensors/actuators the attack will break the correlations among features that hold in the system. Specifically, we apply the best-case scenario constraints proposed by Erba et al. [11].

Random Replay Attack. Random replay is a second variant of the full replay attack. It is the same as a full replay attack, except that the samples in the replayed data are temporally shuffled. E.g., assuming the attackers collected four multivariate samples $[y_1, y_2, y_3, y_4]$ where $y_k \in \mathbb{R}^s$, k is the time index, and s is the number of sensors in the network. After shuffling, the attacker replays the samples in the order $[y_3, y_1, y_4, y_2]$. It requires the same attacker capabilities of the aforementioned replay attack.

This attack is useful to understand if a detection scheme correctly models the temporal evolution of the sensor values i.e., if detectors correctly consider the data coming from a Markov Sequence or consider them Independent and Identically Distributed (i.i.d.).

Stale Data Attack. This attack implements a variant of the stale data attack discussed by Krotofil et al. [29]. In the stale data attack, a Denial-of-Service (DoS) attack is launched on the receiver of sensor data (e.g., the anomaly detector). The attack effectively prevents new sensor data from arriving. As demonstrated by Krotofil et al. [29], industrial end devices commonly handle such a loss of updates by assuming the last reported value is still current (e.g., to tolerate intermittent faults). As a result, the attacker can force a sensor reading to a specific value, by starting a DoS attack when that value is currently reported. The attack is unique in the sense that it represents a weak attacker in terms of required capabilities, as the attacker does not need to be able to eavesdrop on traffic or manipulate industrial protocols. The attacker only needs to perform a DoS attack, which is less effort to achieve.

This attack is useful to understand if detectors correctly model the statistical properties of sensor readings, when the attacker performs the DoS the observed statistical properties of the signal will change (e.g., variance becomes 0).

4.6 Mapping of Consistency to Attacks

Based on the three consistency types, we classify our three attacks in Table 2. The constrained replay attack tests spatial consistency because it changes a subset of the sensor readings without breaking the consistency with the non-spoofed sensors. The random replay attack tests the temporal consistency because the value of each sensor does not evolve according to the sensor process physics. Finally, the stale data attack tests statistical consistency because it changes the statistical properties of the sensor signal producing a signal that has a different mean and no variance.

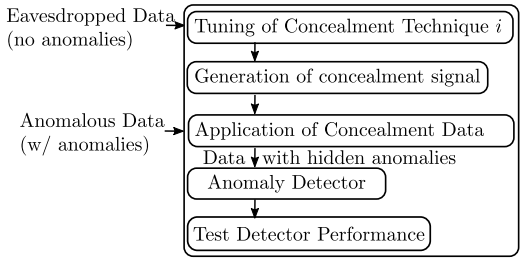

Figure 3: Overview of our framework for concealment attacks against an anomaly detector. The framework receives as input eavesdropped data (containing no anomalies) and anomalous data (containing anomalies). It then applies concealment techniques to the data. Concealment attacks are evaluated with the anomaly detector.

4.7 Abstract Framework Design

Given a target anomaly detector, our analysis requires eavesdropped data (with no anomalies) and anomalous data. We then generate concealment attack data based on the observed process features, and use this data to conceal the anomalous data. To perform the evaluation, we assess the performance of the detector over data containing anomalies. Then, we test the different concealment attacks applied to the anomalous data. If a concealment attack succeeds, the performance of the detector decreases, and the detector is observed vulnerable to that concealment technique. Concealment attacks are designed not only to attack detectors and reduce their Recall score but also they can be used to understand the properties of the physical process detectors fail to capture properly.

5 MODEL-FREE ANOMALY DETECTORS

In this section, model-free schemes considered in this work are described, for a detailed description please refer to the related papers. We start describing the univariate models followed by the multivariate models. We provide a summary of model-free process-based anomaly detectors considered in this work in Table 3.

Autoregressive Model (AR). Autoregressive models, Equation 4, are a representation used to deal with time-series data. Several prior works [4, 19, 45] considered the AR predictors to perform anomaly detection (either as a contribution or as a baseline for evaluation).

$$X_t = c + \sum_{i=1}^{p} \gamma_i X_{t-i} + \epsilon_t \tag{4}$$

The Autoregressive model AR(p) of a time series correlates the time series value (X) at time t with previous $t-p$ time series values. The parameters $\gamma_1, \ldots, \gamma_p$ can be identified with several methods (e.g., ordinary least squares, or Yule-Walker equations). The general idea to use the AR model as an anomaly detector is to fit the Autoregressive model to time series data under normal operating conditions of the CPS, compute predictions, and analyze the residuals to identify anomalies in the sensor data. Different classification functions (detection statistics) were used to identify anomalies. The

Table 3: Taxonomy of the considered model-free process-based anomaly detection. ○simulated process/data, ●real process/data, ◍simulated and real process/data.

	AR [4, 19, 45]	PASAD [4]	SFIG [13]	AE [17, 27, 40]
Evaluation				
SWaT	●	●	●	●
WADI	-	-	●	●
Batadal	-	-	-	○
Private/Other	◍	●	-	-
Code Public	7	3	~	3

work by Urbina et al. [45] shows that stateful statistics (e.g., such as Cumulative Sum (CUSUM)) provide a better detection performance.

PASAD [4]. In this work Aoudi *et al.* introduce PASAD, a model-free anomaly detector. The key motivation of this anomaly detector is to learn and obtain a mathematical representation of the regular dynamics/deterministic behavior of sensor signals in the ICS and spot any anomalous deviation. Singular Spectrum Analysis is used to analyze sensor readings and learn how the process behaves in normal conditions. Sensor readings are considered a univariate time series. Training is conducted on a set of *lagged* sensor readings, those readings are processed with Singular Value Decomposition, and projected into a Signal Subspace, where they form a cluster of points. At test time, test samples are processed with the same pipeline and compared against the centroid of the training data cluster in the Signal Subspace. If tested points exceed a distance threshold from the centroid, the system state is evaluated as anomalous. The MATLAB framework is available as open-source.

Systematic Framework for Invariants Generation (SFIG) [13]. This work by Feng et al. proposes a framework for automatic extraction of process invariants, which are used to build a model-free anomaly detector. The framework applies data mining techniques to extract invariants from frequent itemset with multiple minimum supports. The framework consists of predicate generation, and invariant mining. Predicate generation considers three kinds of predicates. Categorical predicates are generated according to actuators' states. Distribution-driven predicates, use Gaussian Mixture Models to identify the K probability distributions that describe the sensor value update (done per-sensor). Event-driven predicates are fitting Lasso regression models to capture the interplay between actuators' states and values seen by sensors. Predicates are combined to create sets of predicates that hold in a certain instant over the ICS, extracted via the CFP-growth++ [26] algorithm. Evaluation is conducted over WADI and SWaT datasets [23, 32].

Deep Autoencoders (AE). Autoencoders are a class of deep learning architectures that were successfully applied in anomaly detection tasks for industrial control systems. In particular, fully connected (FC) [40], convolutional neural network (CNN) [27] and long short term memories (LSTM) [17] architectures were proposed to detect anomalies in process data. The general idea of those anomaly detectors is to train the Autoencoder model to reproduce sensor readings occurring in the system during normal operations (i.e., without attacks) by minimizing the Mean Squared Error (MSE) between the input and output layer of the network. Then a threshold is set over the MSE observed during the system validation. At test time anomalies in the sensor data presented at the input layer will not be reproduced in the output layer, this will produce a reconstruction error higher than the threshold and will trigger an alarm. CNN and LSTM architectures are trained taking into account the temporal evolution of the signal while the FC is not.

6 EVALUATING THE ANOMALY DETECTORS

In this section, we explain how we used our framework design to test the six anomaly detectors and present the results. We apply to each of the six evaluated detectors the identified generic concealment attacks. We compare the results of the generic concealment with the detection results on the original (i.e., not enhanced with concealment attacks) dataset and with prior work Learning-based concealment attack [11], where the attacker is assumed to manipulate sensor/actuators reading using a neural-network to conceal the anomalies on the system in real-time.

For two schemes (SFIG [13] and AR [4, 19, 46]), no reference implementation was available at the time of our experiments. Details about implementation and model performance are presented in Appendix A. Our re-implementations of detectors are publicly available.

6.1 Attack Dataset Generation

We generate our attack data traces using a tool we wrote in Python 3, using the Pandas and NumPy libraries. The framework processes input training data (without anomalies) and test data (containing anomalies). Data should be organized in .csv format, where every row contains the sensor readings collected at a certain time step and every column represents a different sensor value. Test data is labeled, indicating whether the given row was anomalous or not. Optionally (in the constrained case) the framework takes the constraints on which variables can be spoofed. It then applies the presented concealment techniques to the data, and outputs the resulting augmented dataset (in .csv format). The framework can be applied to any similar dataset that meets the requirements.

Starting from the test data, the framework identifies the intervals in the dataset that are labeled as anomalous. The `evasion` function builds the dataset containing concealment attacks. It leaves the time intervals unchanged where ground truth reports 'normal' and applies the concealment attack to the time steps with ground truth 'anomalous'. The different concealment techniques are implemented as functions that apply the required spoofing to the given data. `Stale Data` attack replicates the sensor data as occurred in the last instance before the attack on the physical process started. `Replay` copies the data as found in the eavesdropped dataset. `Random Replay` copies the data as found in the eavesdropped dataset and shuffles them temporally.

Specifically, we consider Batadal 'train dataset' and 'test dataset 1' for our evaluation. We generated 6 types of additional datasets (3 for the unconstrained attacks, and 3 for the constrained attacks). As we generate a dataset for each constraint value tested, we end up with 45 datasets for constrained attacks. Runtime for the attack dataset generation is less than two minutes for all attacks in total. As this dataset augmentation only has to be performed once for

the evaluations, we find the runtime overall to be negligible. While we did not investigate real-time generation of concealment data (i.e., how an attacker might apply the concealment during an attack), we do not expect computational challenges.

6.2 Evaluation Results

We now provide the results of our experiments. Our evaluation is based on the analysis of the Recall score before and after the concealment attacks. A lower Recall indicates the attack is less likely to be detected. Note on False Positive Rate fluctuation in results: anomaly detectors classify instant t while aggregating all $t - n$ sensor readings before t. Datasets are composed of attacks interleaved by normal operations, if we spoof from instant a to instant b the classification outcome at time $b + 1$ depends on the manipulation that occurred between a, b, influencing the FPR. More details on our evaluation metrics are in Appendix B.

6.2.1 AR. We attack the autoregressive (AR) detector with concealment attacks. An AR predictor is defined as 'good' if it generates residuals (i.e., prediction errors) distributed as white noise [7]. This anomaly detector uses the CUSUM algorithm to check whether the residuals are changing their distribution w.r.t. training phase. When an anomaly occurs, the CUSUM detects a change in the distribution of the residuals (i.e., no more distributed as white noise (WN): this is caused by the predictor that is not behaving optimally because of anomalous data). To succeed in concealment, the attacker has to modify the sensor signal such that obtained residuals do not surpass CUSUM control thresholds, i.e., residuals remain WN.

In Table 4 we report the results of concealment attacks. If we consider the Recall rate, we can notice that stale data attack reduces it to 0. The stale data approach is hiding the anomalies from the detector attack changing the value of the process with a constant, this makes the AR(20) model predicting the constant value that sends to 0 the CUSUM upper and lower statistics. In the case of the random replay attack, the spoofed signal causes sudden changes in the data, causing a change in the distribution of the residuals observed by CUSUM and triggering the alarms. This observation is consistent with [19], where it was observed that sudden changes in the process trigger alarms in AR detectors. In conclusion, the detector reliably models the temporal consistency of the signal but fails to model the spatial consistency (as it is univariate) and the statistical properties (as it does not detect the stale attack).

Comparing the results of the generic attacks w.r.t. learning-based attack [11], we can observe that the neural-network based attack reduces the recall from 0.28 to 0.07 of the anomaly detector, but not as effective as the stale attack. Interestingly, an attack designed to target neural network based models, transfers to AR models.

6.2.2 PASAD. The detector treats the process data as a set of univariate time series, as in the case of the AR model this detector does not model spatial consistency. For this reason, we consider again the Batadal sensor J302, this allows direct comparison of detectors.

Table 5 summarizes the performance of PASAD when targeted with generic concealment attacks. Random Replay concealment techniques reduce the detector's performance. In two out of three proposed generic concealment attacks, we note that Recall decreases to 0.027 (from 0.243). In contrast, the detector performance

Table 4: Concealment attack results on AR model on Batadal sensor J302, unconstrained attack. † Note: technically NaN as the metric divides by 0.

Dataset	Rec.	Prec.	F1	Acc.	FPR
Original	0.28	0.79	0.41	0.91	0.01
Random Replay	0.29	0.88	0.44	0.92	0.01
Stale	**0.00**	0.00	(0)†	0.89	0.00
Learning-based [11]	0.07	0.60	0.12	0.90	0.01

Table 5: Concealment attack results on PASAD Batadal Dataset sensor J302. Threshold = 635.1057**.**

Dataset	Rec.	Prec.	F1	Acc.	FPR
Original	0.243	0.741	0.366	0.910	0.010
Random Replay	**0.027**	0.530	0.051	0.894	0.003
Stale	0.471	0.779	0.587	0.929	0.016
Learning-based [11]	0.241	0.740	0.364	0.910	0.010

increases for stale data attacks. The recall score decreases when we spoof the signal with a not physically plausible temporal evolution of the signal as in the random replay attack. This shows that the anomaly detector has learned the data distribution and not the physical process dynamics. This is also the reason why the stale data attack is detected. The stale data attack does not keep the statistical properties of the sensor signal, deviating from the expected statistical behavior. We can also note that if the stale data occurs on a value close to the process mean, the recall score decreases. Mathematically these results can be explained by analyzing Step 3 of PASAD anomaly detection scheme. PASAD projects training points in the signal subspace. Those projected points create a cluster in the projection subspace. Then, PASAD tracks the distance from the centroid of the cluster to identify anomalies. The centroid is defined as the sample mean of the lagged vectors. Our random replay attack fulfills the requirement of being projected within the cluster in the signal subspace. Despite its dynamics is not plausible, its departure score is lower than the threshold. At the same time, the stale data attack surpasses the threshold because produces a different data distribution. In conclusion, the detector reliably models the statistical consistency of the signal but fails to model the spatial (as it is univariate) and temporal consistency.

Comparing the results of the generic attacks w.r.t. learning-based attack [11], we can observe that the latter attack is not reducing the recall of the anomaly detector, so this attack does affect PASAD.

6.2.3 SFIG. We tested our attacks against SFIG anomaly detector trained on Batadal dataset (details in Appendix A). As this anomaly detector generates invariants aggregating all the sensors, we tested our framework in constrained and unconstrained settings.

Unconstrained setting. Table 6 reports the results of unconstrained attacks. Starting from the original detection Recall of 0.47, results show that the concealment attacks decrease dramatically the Recall score. When we apply the stale data attack to test the statistical consistency, Recall drops respectively to 0. This result indicates that

Table 6: Concealment attacks results on SFIG detector on Batadal test dataset 1, unconstrained attack. †Note: technically NaN as the metric divides by 0.

Dataset	Rec.	Prec.	F1	Acc.	FPR
Original	0.47	0.75	0.58	0.93	0.02
Random Replay	0.15	0.49	0.23	0.89	0.02
Stale	**0.0**	0.00	(0)†	0.88	0.02
Learning-based [11]	0.01	0.07	0.02	0.88	0.02

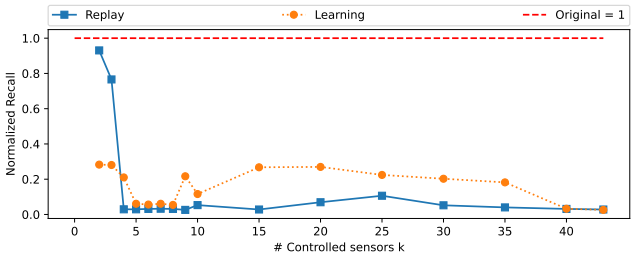

Figure 4: Impact of constrained concealment attacks on Recall score of the SFIG detector. 'Original' represents the Recall baseline of the model. Recall higher than 1 means that the detector is observing more anomalies after the concealment attack is in place.

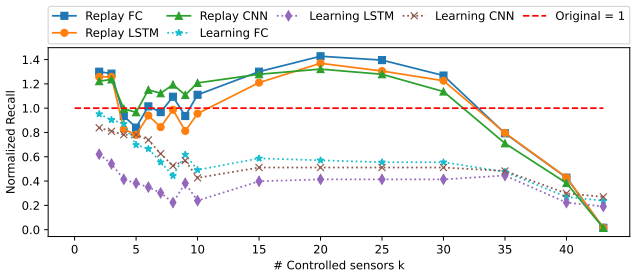

Figure 5: Impact of constrained concealment attacks on the autoencoder detectors. 'Original' represents the Recall baseline of the model. Recall higher than 1 means that the detector is observing more anomalies after the concealment attack is in place.

this concealment attack was able to conceal the instances of anomalous data. If we consider what is going on in the anomaly detector, Recall close to 0 means that there are no invariant rules violated by the attack. Indeed, Distribution Driven predicates are not violated since during training it often occurs that a sensor reading remains constant within two instants. At the same time, Event-Driven Predicates cannot be triggered. The spoofed signal reports that no events are occurring over the system. Hence, the system appears static, and the invariant-based detector stops checking invariant rules regardless of the data distribution of the samples. When we apply the random replay attack, the Recall score goes to 0.14, showing that the detector can detect (in part) the temporal inconsistencies.

Table 7: Performance of Autoencoders (Batadal).

Dataset	Rec.	Prec.	F1	Acc.	FPR
FC					
Original	0.631	0.864	0.729	0.950	0.012
Random Replay	**0**	0.000	(0)	0.883	0.012
Stale	**0**	0.000	(0)	0.883	0.012
Learning-based [11]	0.151	0.606	0.242	0.899	0.012
LSTM					
Original	0.628	0.862	0.727	0.950	0.012
Random Replay	0.366	0.784	0.499	0.922	0.012
Stale	**0.003**	0.030	0.006	0.883	0.012
Learning-based [11]	0.122	0.550	0.200	0.896	0.012
CNN					
Original	0.704	0.875	0.780	0.958	0.012
Random Replay	**0.004**	0.035	0.006	0.883	0.012
Stale	**0.003**	0.025	0.005	0.883	0.012
Learning-based [11]	0.188	0.654	0.292	0.903	0.012

Comparing the results of the generic attacks w.r.t. learning-based attack [11], we can observe that the neural-network based attack reduces the recall of the anomaly detector, almost as effectively as the stale attack. Also in this case, it is interesting to notice that the neural-network based attack transfers to invariant based methods.

Constrained attack. In the constrained case where the attacker can only spoof a constrained set of sensor readings, i.e., they have compromised a subset of the PLCs and can spoof only certain sensors. For our experiments, we consider the Batadal constraints proposed by Erba et al. [11]. As depicted in Figure 4, this detector fails to spot the constrained Replay. For example, when the attacker gains control of 4 out of 43 sensors (coming from at most 3 different PLCs/areas out of 9 in the network), the detection Recall drops to 0.0137. Comparing the results of the constrained replay attack w.r.t. constrained learning-based attack [11], we can observe that also in this case, the neural-network based attack reduces the recall of the anomaly detector, but not as much as the constrained replay.

In conclusion, this detector fails to model spatial and statistical properties while it partially detects temporal inconsistencies.

6.2.4 Deep Autoencoders. We tested the generic attacks against three different autoencoders. As in the previous case the detector considers the multivariate time series for detection and we perform unconstrained and constrained attacks. This builds upon and extends prior work experiments [11] to relate to the consistencies proposed in this work, which were not considered before.

Unconstrained sensing. Table 7 reports the results of unconstrained attacks applied to the three autoencoder architectures. Starting from the original detection Recalls (respectively 0.631 for the FC, 0.628 for the LSTM and 0.704 for the CNN), results show that the concealment attacks are capable of evading the detectors. When we apply the stale data attack to test the statistical consistency, Recall drops close to 0 for all the three architectures. These results indicate that this concealment attack was able to conceal the instances of anomalous data and the detectors are not correctly exploiting the statistical properties of the signal. When we apply the random

Table 8: Vulnerability to the attacks 3detected (not vulnerable), 7non detected (vulnerable), N.A. not applicable as the detector considers the univariate time series.

	Detected		
Detector	Constr. Replay	Random Replay	Stale
AR [4, 19, 46]	N.A.	3	7
PASAD [4]	N.A.	7	3
SFIG [13]	7	3	7
AE FC [40]	3	7	7
AE LSTM [17]	3	3	7
AE CNN [27]	3	7	7

replay attack, the Recall score goes to 0 for the FC and CNN architectures, showing that those detectors are not detecting temporal inconsistencies. On the other end, the recall of the LSTM architecture targeted with the random replay is 0.366, this shows that correlating the last two sets of sensor readings in the input layer allows detecting anomalous temporal evolution of the process.

Comparing the results of the generic attacks w.r.t. learning-based attack, we can observe that the learning-based attack reduces the recall of the anomaly detector, but not as well as the generic concealment attacks. The only exception is the random replay for the LSTM, surpassed by the learning-based attack.

Constrained a ack. As in the previous section, the second experiment studies the constrained case where the attacker can only spoof a constrained set of sensor readings i.e., they have compromised a subset of the PLCs and can spoof only certain sensors. We use the same constraints also for this model. As depicted in Figure 5, the detectors identify the spatial inconsistencies introduced by the replay attack, also when the attacker controls almost all the sensor readings (i.e., 40 out of 43) the detection recall is around 40% of the original detection score. In conclusion, all the three autoencoders model properly spatial consistency, while they fail to model statistical properties. FC and CNN also failed to capture temporal properties of the system in contrast to LSTM. Comparing the results of the generic attacks and learning-based attack [11], we observe that the neural-network based attack reduces recall of the anomaly detector almost as effectively as the stale attack.

6.3 Summary of Findings

With respect to **RQ2**, results show a varied performance of detectors w.r.t. the three considered attacks and the related consistency properties, none of the detectors is resilient to all the three considered attacks but at most to two (i.e., AE LSTM resilient to constrained replay and random replay). Those attacks break the physical properties of the system and are, in theory, easy to spot. Model-free detectors fail to exhaustively abstract Spatial, Temporal, and Statistical consistencies to perform anomaly detection.

With respect to **RQ3**, results show that neural-network based attacks [11], can effectively be used to conceal the effects of attacks on the physical process for AR, SFIG, and Autoencoders, but they fail to succeed against PASAD detector. On the other hand, the concealment performance (reduction of the recall score), is always in favor of one of the generic attacks. Moreover, learning-based attacks from prior work require real-time computation to adapt the spoofing pattern to the current sensor readings, while generic concealment attacks do not need to be adapted in real-time.

Countermeasure. Given the results for RQ2 and RQ3, accurate attack detection without detailed a priori process models remains an open issue. In Appendix C, we leverage the consistency properties to construct and test a data-driven model-based detector that reliably detects process anomalies and concealment attacks.

7 RELATED WORK

Adversarial Machine Learning (AML) is the research topic at the intersection of Machine Learning and System Security, this field investigates the security properties of machine learning algorithms when targeted by attacks. Attackers in the AML setting can be motivated to perform a different type of attack [20]. E.g., classifier concealment, model poisoning, and model stealing. Classifier evasion in the field of Cyber-Physical System is a rising research topic. So far, attacks that target Deep Learning-based classifiers have been proposed. Feng et al. [12] propose the usage of Generative Adversarial Networks to produce stealthy manipulations for ICS detectors. Erba et al. [11] proposed two real-time evasion attacks against reconstruction-based classifiers are proposed. A black-box and a white-box attack method are presented. This work is the first that models an attacker in the setting of Adversarial Machine Learning for ICS. Kravchik et al. [28] proposed an anomaly detector based on Autoencoders and PCA. The work by Zizzo et al. [51] evaluates adversarial examples in ICS by applying white-box attacks to LSTM detectors. Luo *et al.* [31], survey Deep-Learning based detectors for Cyber Physical Systems.

8 DISCUSSION AND CONCLUSIONS

In this work, we introduced the concepts of spatial, temporal, and statistical consistency for process-based anomaly detectors (RQ1). To assess which detectors verify which consistency, we leverage three general concealment attacks. We then designed and implemented a framework to add those attacks to common datasets, and evaluated six model-free detectors (RQ2). Our evaluation results show that the considered attacks are effectively evading prior work detectors, which demonstrates that detectors are not verifying all three consistencies. Although our attacks were designed to test consistencies (and were not particularly optimized for performance), we noted that they were surprisingly effective even compared to more optimized prior work. Our attacks reduced the Recall of AR models from prior work from 0.28 to 0.0, of PASAD [4] from 0.24 to 0.02, of SFIG [13] from 0.47 to 0.0 of FC autoencoder 0.631 to 0, of the LSTM autoencoder from 0.628 to 0.003 and of the CNN autoencoder from 0.704 to 0.003. The weaknesses we demonstrated in the anomaly detectors show that (despite good detection performance of the original schemes), the detectors are not able to detect adversarially manipulated physical system properties. Our results also show that generic concealment attacks are possible, in contrast to prior work that assumed to have a white-box knowledge of the target system [45]. The analysis and the results in our contribution highlight the need for more complete datasets and critical analysis of model-free detectors to evaluate their performance. As such,

we see our contribution to the discussion about the resiliency of anomaly detectors when analyzed against targeted manipulations.

We have compared the identified generic concealment attacks with prior work learning-based attacks (RQ3). The identified generic attacks are also better than prior work although they do not require real-time adaptation of the sensor readings. Identified generic attacks focus on one consistency each, and they perform better than prior work attacks. On the other hand, we show that learning based attacks transfer to other models.

REFERENCES

[1] Ali Abbasi and Majid Hashemi. 2016. Ghost in the plc designing an undetectable programmable logic controller rootkit via pin control attack. *Black Hat Europe* 2016 (2016), 1–35.

[2] Chuadhry Mujeeb Ahmed, Sridhar Adepu, and Aditya Mathur. 2016. Limitations of state estimation based cyber attack detection schemes in industrial control systems. In *2016 Smart City Security and Privacy Workshop (SCSP-W)*. IEEE, Vienna, Austria, 1–5. https://doi.org/10.1109/SCSPW.2016.7509557

[3] Chuadhry Mujeeb Ahmed, Martin Ochoa, Jianying Zhou, Aditya P. Mathur, Rizwan Qadeer, Carlos Murguia, and Justin Ruths. 2018. NoisePrint: Attack Detection Using Sensor and Process Noise Fingerprint in Cyber Physical Systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (Incheon, Republic of Korea) *(ASIACCS '18)*. ACM, New York, NY, USA, 483–497. https://doi.org/10.1145/3196494.3196532

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

[5] Batadal attacks description [n.d.]. Batadal attacks description. https://www.batadal.net/images/Attacks_TrainingDataset2.png.

[6] Batadal attacks description [n.d.]. Batadal attacks description. https://www.batadal.net/images/Attacks_TrainingDataset2.png.

[7] Peter J Brockwell, Richard A Davis, and Stephen E Fienberg. 1991. *Time series: theory and methods: theory and methods*. Springer Science & Business Media.

[8] Chi-Tsong Chen. 1998. *Linear System Theory and Design* (3rd ed.). Oxford University Press, Inc., USA.

[9] Yuqi Chen, Christopher M Poskitt, and Jun Sun. 2018. Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system. In *Proc. of the IEEE Symposium on Security and Privacy*. IEEE, San Francisco, CA, 648–660.

[10] 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.

[11] 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 *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. ACM, Austin, USA. https://doi.org/10.1145/3427228.3427660

[12] Cheng Feng, Tingting Li, Zhanxing Zhu, and Deeph Chana. 2017. A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397* (2017).

[13] 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)*.

[14] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. 2016. The SPMF open-source data mining library version 2. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 36–40.

[15] 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)*.

[16] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2016. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security (CRITIS)*. Springer, 88–99.

[17] 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.

[18] Benjamin Green, Marina Krotofil, and Ali Abbasi. 2017. On the significance of process comprehension for conducting targeted ICS attacks. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*. 57–67.

[19] 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* (New Orleans, Louisiana, USA) *(ACSAC '14)*. ACM, New York, NY, USA, 126–135. https://doi.org/10.1145/2664243.2664277

[20] 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.

[21] ICS concealment attacks repository [n.d.]. ICS concealment attacks repository. https://github.com/scy-phy/ICS-Evasion-Attacks.

[22] Vittorio P Illiano and Emil C Lupu. 2015. Detecting malicious data injections in wireless sensor networks: A survey. *ACM Computing Surveys (CSUR)* 48, 2 (2015), 24.

[23] iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design. 2017. WADI datatset. https://itrust.sutd.edu.sg/research/dataset/dataset_characteristics/#wadi, Last accessed on: 2019-01-30.

[24] Kaspersky. [n.d.]. Kaspersky Machine Learning for Anomaly Detection. https://mlad.kaspersky.com/, Last accessed on: 2022-03-30.

[25] Anastasis Keliris and Michail Maniatakos. 2018. ICSREF: A framework for automated reverse engineering of industrial control systems binaries. *arXiv preprint arXiv:1812.03478* (2018).

[26] R Uday Kiran and P Krishna Reddy. 2011. Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms. In *Proceedings of the 14th international conference on extending database technology*. ACM, 11–20.

[27] 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.

[28] Moshe Kravchik and Asaf Shabtai. 2021. Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca. *IEEE Transactions on Dependable and Secure Computing* (2021).

[29] Marina Krotofil, Alvaro A Cárdenas, Bradley Manning, and Jason Larsen. 2014. CPS: driving cyber-physical systems to unsafe operating conditions by timing DoS attacks on sensor signals. In *Proceedings of the 30th Annual Computer Security Applications Conference*. ACM, 146–155.

[30] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 145–159.

[31] Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. 2021. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[32] Aditya Mathur and Nils Ole Tippenhauer. 2016. 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)*. https://doi.org/10.1109/CySWater.2016.7469060

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

[34] Alberto Ornaghi and Marco Valleri. 2003. Man in the middle attacks. In *Blackhat Conference Europe*, Vol. 1045.

[35] Avi Ostfeld, Elad Salomons, Lindell Ormsbee, James Uber, Christopher Bros, Paul Kalungi, Richard Burd, Boguslawa Zazula-Coetzee, Teddy Belrain, Doosun Kang, Kevin Lansey, Hailiang Shen, Edward McBean, Zheng Wu, Tom Walski, Stefano Alvisi, Marco Franchini, Joshua P. Johnson, Santosh Ghimire, and Robert McKillop. 2012. Battle of the Water Calibration Networks. *JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT-ASCE* 138 (09 2012), 523–532. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000191

[36] 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

[37] Julian L Rrushi. 2012. SCADA protocol vulnerabilities. In *Critical Infrastructure Protection*. Springer, 150–176.

[38] Esha Sarkar, Hadjer Benkraouda, and Michail Maniatakos. 2020. I came, I saw, I hacked: Automated Generation of Process-independent Attacks for Industrial Control Systems. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. 744–758.

[39] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 877–894. https://www.usenix.org/conference/usenixsecurity20/presentation/sun

[40] 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

[41] 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

[42] 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

[43] Yazhou Tu, Sara Rampazzi, Bin Hao, Angel Rodriguez, Kevin Fu, and Xiali Hei. 2019. Trick or Heat? Manipulating Critical Temperature-Based Control Systems Using Rectification Attacks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2301–2315. https://doi.org/10.1145/3319535.3354195

[44] Federico Turrin, Alessandro Erba, Nils Ole Tippenhauer, and Mauro Conti. 2020. A Statistical Analysis Framework for ICS Process Datasets. In *Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy*. 25–30.

[45] 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 *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. https://doi.org/10.1145/2976749.2978388

[46] David Urbina, Jairo Giraldo, Nils Ole Tippenhauer, and Alvaro Cárdenas. 2016. Attacking Fieldbus Communications in ICS: Applications to the SWaT Testbed. In *Proceedings of Singapore Cyber Security Conference (SG-CRC)*. IOS Press, Singapore. https://doi.org/10.3233/978-1-61499-617-0-75

[47] Peter Van Overschee and Bart De Moor. 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* 30, 1 (1994), 75–93.

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

[49] Theodore J Williams. 1994. The Purdue enterprise reference architecture. *Computers in industry* 24, 2-3 (1994), 141–158.

[50] Chen Yan, Wenyuan Xu, and Jianhao Liu. 2016. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON* 24 (2016).

[51] Giulio Zizzo, Chris Hankin, Sergio Maffeis, and Kevin Jones. 2020. Adversarial Attacks on Time-Series Intrusion Detection for Industrial Control Systems. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, Guangzhou, China, 899–910. https://doi.org/10.1109/TrustCom50675.2020.00121

A DETECTOR (RE-)IMPLEMENTATION

AR. We implemented the model-free AR detector as considered in prior work [4, 19, 46] with the CUSUM test. We implemented it in MATLAB and validated it on the Batadal dataset. The detector is based on an Auto-Regressive (AR) model trained over a univariate time series. The residuals of the AR model are used to compute a Cumulative Sum (CUSUM) statistic whose objective is to reveal a change in the process generating the data, i.e., spot anomalies in the system. We implemented this detector in MATLAB, using the System Identification Toolbox. We trained the model over Batadal data, we performed our experiments on sensor 'PRESSURE J302', i.e., the sensor that (alone) was allowing to detect the highest number of attacks (8 attacks over 14) with the considered detection method. We selected the AR model of order 20 with Best Fit criteria and tuned the CUSUM parameters using grid search and selected control_limit=5.5 and min_mean_shift_detect=1 obtaining as original detection performance Accuracy = 0.92, F1-score = 0.41, Precision = 0.79, Recall (TPR) = 0.28, FPR = 0.01.

PASAD. The implementation of PASAD anomaly detector [4] is available at https://github.com/mikeliturbe/pasad. PASAD analyzes every sensor univariate temporal series independently, for every sensor PASAD requires to be trained independently. We trained PASAD on the Batadal dataset over 'PRESSURE J302' sensor, we performed parameter tuning following the instructions provided in the original paper. Specifically, we used $N = 250$, $L = 250$, $r = 18$. The resulting original detection performance with this threshold over sensor J302 is Accuracy = 0.91, F1-score = 0.37, Precision = 0.74, Recall(TPR) = 0.24, FPR = 0.010.

SFIG. We re-implemented the anomaly detector based on the paper. We used Python 3 with the following libraries: Sklearn, Pandas, NumPy, SciPy. In this section, we summarize the parameters, and the assumptions we had to make to implement the detection system. *Distribution Driven Strategy.* We normalized the data between 0 and 1. We fitted Gaussian Mixture Models with at most 4 components for every sensor and took the one with the lowest BIC score. *Event Driven Strategy.* We set the threshold for the trigger $\epsilon = 0.05$, for Lasso we set $\alpha = 0.1$. *Invariant Mining.* Invariant mining is done with the CFP-growth++ algorithm. This algorithm is only available as open-source[14] in a Java library http://www.philippe-fournier-viger.com/spmf/. We used that library from our python script. Since the library is generating all the frequent itemsets that have the allowed minimum support, we parsed the output to identify the itemsets that do not break the non-redundant condition.

After re-implementing the detection mechanism, we were able to achieve a comparable result using the Batadal dataset. The resulting original detection performance is as follows. Accuracy = 0.93 F1-score= 0.58 , Precision = 0.75, Recall (TPR) = 0.47, FPR = 0.02.

AE. The implementation of the Autoencoder Based mechanisms using the three deep architectures (FC, LSTM, CNN) is available at the repository https://github.com/scy-phy/ICS-Evasion-Attacks. We leverage this implementation in this work. The input of the FC architecture is one set of sensor readings (i.e., 43 sensors), while for the LSTM and CNN the input is represented by the last two sampled set of sensor readings (i.e., 2x43 sensors). The performances of the three architectures are detailed in Table 9.

Table 9: Performance of Autoencoders trained on Batadal.

Dataset	Acc.	F1	Prec.	Rec.	FPR
FC	0.950	0.729	0.864	0.631	0.012
LSTM	0.950	0.727	0.862	0.628	0.012
CNN	0.958	0.780	0.875	0.704	0.012

B METHODOLOGY

In order to evaluate the performance of the anomaly detector, we observe how Accuracy Eq. 5, Precision Eq. 6, Recall Eq. 7, and False Positive Rate Eq. 9 scores change when the spoofing technique is applied to the data.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

Algorithm 1: Rolling window detector training

```
function RollingDetector(trainData, window)
    features ← []
    percentiles ← []
    for sensor in listOfSensors do
        hMatrix ← hankel(trainingData[sensor], window)
        // mean of rows
        slidingMean ← mean(hMatrix)
        // std of rows
        slidingStd ← std(hMatrix)
        // indexes of windows where variance is zero
        indexesZeroStd ← slidingMean[slidingStd==0]
        // find the mean of the sliding windows means with zero variance
        meanZeroStd ← mean(indexesZeroStd)
        if meanZeroStd in [0,NaN] then
            // find minimum non-null percentile of the slidingStd vector
            percentile ← findPercentile(slidingStd)
            features.append(sensor)
            percentiles.append(percentile)
    return features, percentiles
```

Algorithm 2: Rolling window detector testing

```
function anomalyDetection(features, percentiles, window, testData)
    predictions ← []
    for sensor in features do
        hMatrix ← hankel(testData[sensor], window)
        // mean of rows
        slidingMean ← mean(hMatrix)
        // std of rows
        slidingStd ← std(hMatrix) // for each window verify if the variance is lower than the lowest non-null percentile observed during training
        for i in len(slidingMean) do
            if slidingMean[i] > 0 and slidingStd[i] < percentiles[sensor] then
                predictions[i] ← 1
            else
                predictions[i] ← 0
    return predictions
```

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{9}$$

Given the original classification scores (e.g., when no spoofing is applied to data), concealment is effective if the Precision and Recall score reduces substantially. When those two scores reduce, towards 0, it means that the instances where the concealment was applied were misclassified moving them from being True Positives to False Negatives. Looking at the False Positive Rate (FPR) score we can also verify if the attacks are introducing False Positive in the Classification. If the FPR remains almost like the original, it means that the concealment did not induce any wrong classification (as expected since we are not spoofing data outside the boundaries of the attacks present in the dataset). Finally, since the datasets are unbalanced, with more samples of the negative class, the Accuracy score will not reach zero but at most the baseline where all the instances are labeled as the negative class.

C DATA DRIVEN MODEL-BASED DETECTOR

Given our results in the Section 6, accurate attack detection without detailed a priori process models remains an open research question. In the following, we construct and test a data-driven model-based anomaly detector that leverages the physical properties of the process and reliably detects both the process anomalies in the data and concealment attacks. Like our assessment framework, the proposed ensemble detector code is publicly available. Following our attacker model and capabilities (Section 3), we consider white-box attackers (i.e., with process knowledge and detector knowledge) out of scope in our evaluation. As shown in prior work [11, 45], that in white-box setting attacks cannot be fully-prevented can be at most delayed or reduce their impact. Moreover getting this white-box process/detection knowledge is challenging (if not impossible) [18, 25, 38].

Detector Architecture. The detector is based on an ensemble of two complementary detectors trained on process data collected on the system in normal operation condition (i.e., without anomalies). The detectors in the ensemble are designed to detect process anomalies and concealment attacks. Figure 6 gives an overview of the ensemble architecture. The first model in the ensemble is using an identified Linear Time Invariant (LTI) model of the process to predict future system behavior, which enables us to compute residuals which we use for a stateful (CUSUM) detector. This part of the ensemble allows us to identify spatial and temporal inconsistencies among features. The second model is a sliding-window based statistical outlier detector, required to detect remaining attacks that manipulate the signal without inducing physical inconsistencies (e.g., Stale Data attacks). The predictions are combined using the OR operator. The anomaly detector benefits from the ensemble

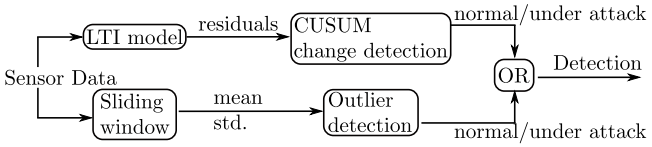

Figure 6: Overview of the ensemble of detector, LTI model and sliding window statistics are used to build a detector resilient to concealment attacks.

as each component is trained to abstract certain properties of the physical process (see Figure 7).

No A-Priori Process Model. In contrast to other model-based schemes (e.g. [10, 36]), our data-driven model-based detector does not require explicit a priori process models or templates to be constructed and trained. Instead, our two models leverage Subspace-based State Space System Identification techniques and statistical analysis, respectively.

C.1 Data-driven Stateful LTI Detector

Formulating the precise system equation for a complex CPS is challenging, requires in-depth process knowledge, and might lead to deal with complex non-linear equations. This part of the ensemble uses an LDS-model based stateful detector (see [45]), which effectively requires a system characterization in form of a set of LTI equations. While such designs were discussed in prior work [45], no concrete implementations have been released, there was no evaluation w.r.t. the attacks of the datasets in this work, or our newly contributed concealment attacks for those datasets.

In order to not rely on a-priori process characterizations, we derive an approximation of the LTI representation leveraging the Subspace-based State Space System Identification techniques (n4sid algorithm [47]). Through this technique, we approximate the coefficients of the system model (i.e., matrices A, B, C, D and disturbances K) without explicit knowledge of the system equations. Specifically, we consider the water tank levels as output values of the system, while all the other continuous sensor readings as input data.

Once the model is identified, a classification function of the one-step-ahead prediction residuals (CUSUM, SVM, etc.), can be used to identify attacks to the spatial and temporal properties. Specifically, for each output of the LTI model, we use the CUSUM algorithm with change detection as a classification function for residuals.

C.2 Statistical Outlier Detection

In order to detect the attacks that violate the statistical properties of the system, we propose a sliding window-based outlier detection method that identifies changes in sensor statistics. Algorithm 1 shows how the detector is trained, while Algorithm 2 shows how the detector is used to perform anomaly detection. The intuition is to detect changes in the process variance (e.g., the changes introduced by the stale data attack). Using l samples of training data, we apply a sliding window of length w to each sensor reading in the dataset, obtaining $k = l - w$ traces of data (per sensor). For each trace in k we compute its mean and standard deviation. For sensors that have a variance greater than 0 (or variance 0 when the mean value is 0) in all the k traces (i.e., the sensor updates its value at least once in w timesteps), we use an approximate binary search algorithm to find the minimum non-null percentile of the sliding window standard deviation distribution and use it as a threshold. At test time, we compute the sliding window statistics for the sensors that satisfy the conditions in the training set. If the variance of a window is greater than 0 but lower than the sensor's threshold an alarm is raised.

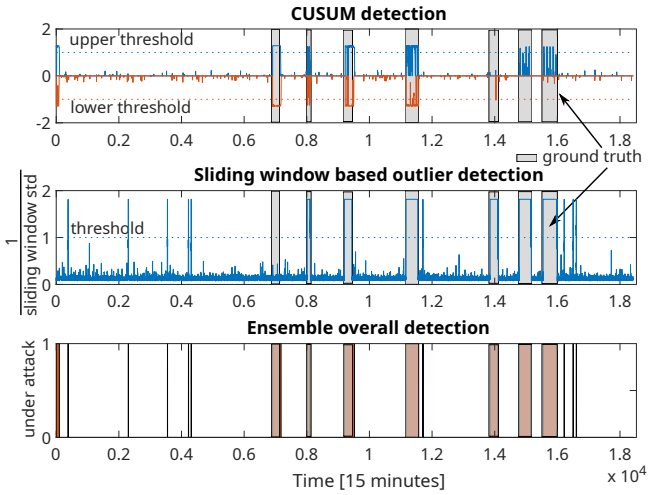

Figure 7: Contribution of classifiers in the ensemble. Detection of constrained stale data attack (normalized thresholds for readability). Some of the attacks are detected exclusively by one of the two detectors in the ensemble. The OR condition between the predicted labels of the two models allows to detect attacks with high precision and recall.

C.3 Classifier Contributions in the Ensemble

Figure 7 shows an example on how the two detectors in the ensemble complement each other. The attacker launches a Stale Data attack (i.e, the attack breaks statistical properties of the sensor readings). We note most of the attacks are detected by both the models in the ensemble, but others are detected by either one of the two as the spoofed signal might trigger exclusively spatial/temporal or statistical inconsistencies.

C.4 Results

We train the predictions of the two methods presented in C.1, C.2 tested on the Batadal dataset. The LTI system was modeled using the 7 water level sensors as output data while all the other 24 continuous sensor readings in the Batadal dataset were used as input. We identified the model of order 11 using n4sid MATLAB implementation. The residuals of the model are classified using CUSUM. The Statistical based detector was trained using all the 31 continuous sensor readings. Table 10 reports the results of the ensemble of methods on the Batadal dataset attacked with the unconstrained concealment attacks. Stale and random replay attacks are detected with a high recall rate. We can observe how the two components of the ensemble contribute to the detection, for example in the random replay attack the temporal inconsistency is triggering the

Table 10: Contribution of classifiers in the Ensemble tested on Batadal dataset. For each experiment, the table reports the performance of each classifier in the ensemble (LTI, Sliding) and their overall performance (Ensemble)

Dataset	Acc.	F1	Prec.	Rec.	FPR
Original Attacks					
LTI	0.91	0.29	0.74	0.18	0.01
Sliding	0.94	0.62	0.93	0.46	0.00
Ensemble	0.95	0.71	0.86	0.61	0.01
Stale					
LTI	0.98	0.91	0.91	0.91	0.01
Sliding	0.99	0.97	0.96	0.99	0.00
Ensemble	0.99	0.94	0.89	1.00	0.01
Random Replay					
LTI	0.99	0.96	0.93	1.00	0.01
Sliding	0.89	0.00	0.04	0.00	0.00
Ensemble	0.99	0.95	0.91	1.00	0.01
Learning-based Attack [11]					
LTI	0.97	0.88	0.81	0.96	0.03
Sliding	0.89	0.01	0.12	0.00	0.00

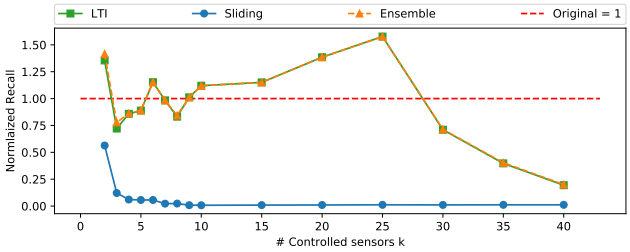

Figure 8: Constrained replay attacks results on our countermeasure. This plot shows the impact of constrained concealment attacks over the Recall score. The plot shows the contribution of the classifies in the ensemble

LTI detector but not the Sliding window detector (as expected since there is no statistical change in the data). Figure 8 reports the results of the constrained replay attack in Batadal data (same constraints used in Figure 5), also constrained replay is detected by our countermeasure. The main contribution to the detection, in this case, is given by the LTI model. The performance of the ensemble in this scenario is comparable to the autoencoder models (Figure 5), although the recall of the ensemble model decreases faster when the attacker controls 30 sensors or more.

C.5 Summary of Findings

We built an ensemble detector that is constructed without leveraging process knowledge and outperforms prior model-free detectors. Our proposed ensemble can detect both the original process anomalies contained in the datasets and the concealment attacks

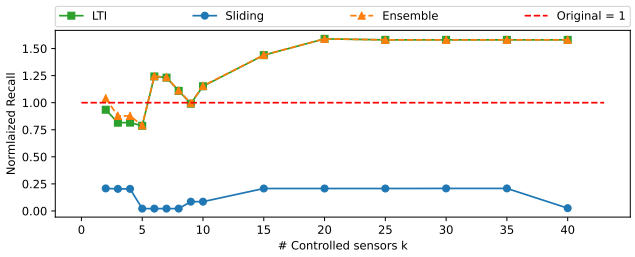

Figure 9: Constrained Learning-based [11] attacks results on our countermeasure. This plot shows the impact of constrained concealment attacks over the Recall score. The plot shows the contribution of the classifies in the ensemble

considered in the work (i.e., both generic concealment attacks and Learning-based [11]). Our model outperforms prior work model-free approaches as it is capable of detecting spatial, temporal, and statistical inconsistencies in the data.