
# A Tale of Two Testbeds: A Comparative Study of Attack Detection Techniques in CPS

Surabhi Athalye(✉), Chuadhry Mujeeb Ahmed, and Jianying Zhou

Singapore University of Technology and Design, Singapore, Singapore
{surabhi_athalye,mujeeb_chuadhry,jianying_zhou}@sutd.edu.sg

**Abstract.** Attack detection in cyber-physical systems (CPS) has been approached in several ways due to the complex interactions among the physical and cyber components. A comprehensive study is presented in this paper to compare different attack detection techniques and evaluate them based on a defined set of metrics. This work investigates model-based attack detectors that use mathematical system models with the sensor/actuator set as the input/output of the underlying physical processes. The detection mechanisms include statistical change monitoring (CUSUM and Bad-Data detectors) and a machine learning based-method that analyses the residual signal. This is a tale of two testbeds, a secure water treatment plant (SWaT) and a water distribution plant (WADI), which serve as case studies for the diverse range of CPS infrastructures found in cities today. The performance of the detection methods is experimentally studied by executing various types of attacks on the plants.

**Keywords:** Cyber-physical systems · Water treatment systems · Water distribution systems · Model-based attack detection

## 1 Introduction

A cyber-physical system (CPS) comprises of physical infrastructure that is controlled by computation and communication frameworks. It includes a combination of interconnected components such as Programmable Logic Controllers (PLCs), sensors, actuators, a Supervisory Control and Data Acquisition (SCADA) workstation, and a Human Machine Interface (HMI) that communicate across a network. The PLCs check the present state of the system through the SCADA and implement the corresponding control actions to facilitate the proper progress and functioning of the sub-processes.

The normal operation of a CPS requires the network and physical elements to work in tandem, for they directly influence the physical processes. Communication among such industrial IoTs is helpful but it also exposes them to malicious entities [1,2]. This makes the design of security measures for a CPS more complicated as compared to those meant for pure IT systems because attacks can occur in both the cyber and physical domain [3].

Since an inter-connected CPS also incorporates wireless communication, the infrastructure is prone to remote breaches and attacks [4]. This can be detrimental as it endangers the crucial communication links between the different nodes in a CPS, allowing


A. Rashid and P. Popov (Eds.): CRITIS 2020, LNCS 12332, pp. 17–30, 2020.
https://doi.org/10.1007/978-3-030-58295-1_2

them to be manipulated by external entities. By influencing the underlying processes in a CPS, cyber attacks could sabotage its physical infrastructure. Physical attacks can damage the sensors or other devices, which compromises the integrity of the data. This is a major risk as it results in faulty data being forwarded to the controllers, which adversely affects the control actions that are computed based on it. Conventionally, security research is focused on detecting anomalies in the communication network part of a CPS [5]. However, physical attacks can be more difficult to detect as they may not be reflected in the system network [6].

In this work, case studies are done on a water treatment testbed and a water distribution testbed, wherein model-based approaches for attack detection are considered. The sensor and control data from these plants under normal operation is used to derive Linear Time-Invariant (LTI) system models. These models are created using a control-theoretic approach, thus allowing the physical dynamics of the underlying processes to be captured analytically. The attack detection methods are then applied to the residual (the difference between the estimated and actual sensor values).

The detection performances of three attack detection techniques are evaluated in this paper. The first two methods are statistical change detectors called Cumulative Sum (CUSUM) and Bad-Data detectors that identify instances of abnormal data using empirically determined thresholds. The third technique is a machine learning-based device fingerprinting method called *NoisePrint* [3].

While gauging the performance, apart from precision, another important consideration for the attack detection techniques is their sensitivity. This refers to their tendency of raising false alarms when the plants operate normally. This is vital due to its implications in practical scenarios, wherein a system of numerous physical components needs to be checked. Hence, the detection mechanisms are evaluated under normal operating conditions as well as when the plants are under several attacks to acquire a comprehensive understanding of their performance.

The motivation for this work is to exhaustively test and compare attack detection techniques for CPS on different testbeds. The implementation of such methods on real-world systems is able to provide some useful insights to address the following issues:

1. *Impact of Noise on System Models*: The implementation and verification of theoretical models brought up some problems, one of them being the noise from the process for each different run. It can be seen that the effect of noise from the environmental disturbances on the process causes unpredictable deviations from its modelled behavior.
2. *Sensor Faults*: One of the problems was the unseen faults in sensors even during the normal operation of the plant, which hindered the creation of useful system models. This means that during the data collection under normal operation, the components must be thoroughly checked to ensure that all of them are functioning properly.
3. *Data Availability and Reliability*: Data availability plays a vital role in the design and performance of an anomaly detector. Prior to model creation, it is necessary to procure sufficient data that (a) represents the components' entire performance cycle, and (b) covers all possible modes of the operation of the Industrial Control System (ICS) in the absence of momentary glitches and outliers. In general, when a dataset is created for a study, the plant is run continuously under normal operating

conditions. The same has been done in this study for obtaining the data to create the models. However, when these models were tested on the plant when it was not running, unexpected outcomes were observed.

4. *Attack Detection Speed*: The speed with which a process anomaly is detected is of prime concern for the safety of the plant, but it is often ignored as a performance attribute [7]. Rapid detection allows for appropriate actions to be taken earlier, thereby mitigating the impact. Therefore, Time Taken for Detection (TTD) has been used as an important performance metric in this study, while highlighting its significance.

*Organization*: The remainder of this paper is organized as follows. The mathematical modelling of the two testbeds as systems is explained in Sect. 2. The attack detection framework in Sect. 3 briefly explains the working of the three detection techniques that form the focus of this paper. Following this, Sect. 4 defines the attacker profile while detailing the potential attack scenarios and their execution. The performance of the detection mechanisms is evaluated in Sect. 5, whereby the techniques are tested under normal and attack conditions. Based on the analysis of the results obtained, the conclusions that map to the contributions above, are presented in Sect. 6.

## 2 System Model

### 2.1 Two Testbeds: Our Playground

Research facilities with operational testbeds of prevalent cyber-physical systems have been utilised to implement the security strategies and test their capabilities. As mentioned earlier, these include a secure water treatment plant (SWaT) [8] and a water distribution plant (WADI) [9]. These are operational, scaled down plants that simulate the larger industrial infrastructure found in cities today. The physical process here is that of water flow, wherein it undergoes specific processes, for e.g., ultra-filtration, reverse osmosis, etc. The plants are divided into different stages, each carrying out a specific sub-process. The detailed workings of the testbeds are explained in papers [8,9].

### 2.2 System Models

Each of the two testbeds is treated as a multi-input, multi-output system, following the model-based approach. A system model represents the dynamics of a physical process using a mathematical formulation. Sub-space system identification techniques are used to obtain models of the following form, for a system with $p$ control inputs (actuators) and $m$ outputs (sensors):

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + v_k, \\ y_k = Cx_k + \eta_k. \end{cases} \tag{1}$$

where $k$ represents the time instance, $x \in \mathbb{R}^n$ is system state vector of $n$ states, $A \in \mathbb{R}^{n \times n}$ is state-space matrix, $B \in \mathbb{R}^{n \times p}$ is the control matrix, $y \in \mathbb{R}^m$ is the vector of the measured outputs, $C \in \mathbb{R}^{m \times n}$ is measurement matrix, and $u \in \mathbb{R}^p$ denotes the system control input.

The state-space matrices $A, B$ and $C$ capture the system dynamics and can be used to find a specific system state given an initial state. The sensor and process noise vectors are represented by $\eta_k$ and $v_k$, respectively.

### 2.3 Validation of the System Models

It is necessary to validate the models created for each of the systems. For this, the state-space matrices from the system identification process are applied and the estimates for the output of the system are obtained. These modelled values and real-time sensor measurements are then compared. The difference between the measured sensor values and estimates is considered using the root mean square error (RMSE). The RMSE value for $N$ readings is given as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{N}}.$$

where $y_i$ is the actual $i$-th sensor reading, and $\hat{y}_i$ is the $i$-th model estimate.

The accuracy of the system identification-based model for 6 sensors in the SWaT testbed is shown in Table 1 as an example, and it can be seen this model has high accuracy. In control theory literature, models with accuracy as high as 70% are considered a sufficiently precise approximation of real system dynamics [10,11].

**Table 1.** Validating SWAT model obtained from sub-space system identification.

| Sensor | FIT101 | LIT101 | LIT301 | FIT301 | LIT401 | FIT401 |
|---|---|---|---|---|---|---|
| RMSE | 0.0363 | 0.2867 | 0.2561 | 0.0200 | 0.2267 | 0.0014 |
| (1-RMSE) * 100% | 96.3670 | 71.3273 | 74.3869 | 98.0032 | 77.3296 | 99.8593 |

## 3 Attack Detection Framework

This work focuses on detecting attacks on sensors, primarily by validating the incoming readings. This is done by (1) estimating the sensor output using the system model, and (2) examining the residual between the actual and estimated values and verifying the source of the sensor readings. The second step is in turn done using the three different detectors (CUSUM, Bad-Data and *NoisePrint*) for comparison.

***System Model and Estimation:*** The concept of creating system models is explained in the previous section. These can be obtained either using data-based techniques or from first principles [12–14]. Using the system model, it is possible to estimate the states of the system and ultimately predict the output from a sensor applying Eq. 1. At a time instance $k$, a residual vector ($r_k$) is calculated by taking the difference between the sensor measurements ($y_k$) and estimated sensor output ($\hat{y}_k$), which is given as:

$$r_k = y_k - \hat{y}_k. \tag{2}$$

For the residual, the hypothesis testing is for $\mathcal{H}_0$, the *normal mode* (no attacks), and $\mathcal{H}_1$, the *faulty mode* (with attacks). The residuals are obtained using this data and the state estimates. The two hypotheses are stated as follows:

$$\mathcal{H}_0 : \begin{cases} E[r_k] = 0, \\ E[r_k r_k^T] = \Sigma, \end{cases} \quad \text{or} \quad \mathcal{H}_1 : \begin{cases} E[r_k] \neq 0, \\ E[r_k r_k^T] \neq \Sigma. \end{cases}$$

***Threshold-Based Detection:*** To detect the presence of an attack, the residual vector is tested against a predefined threshold designed for a particular false alarm rate. A threshold is created for the residual distribution, and while testing the model against the actual data from the plant, an attack is declared if the residual values exceed that threshold:

$$|r_k| > \tau, \; Alarm = \text{TRUE} \tag{3}$$

where $\tau$ is the threshold and $|r_k|$ is the absolute value of the residual. There have been studies on optimizing the parameters of different stateful and stateless detectors [13,14]. Next, the three attack detection techniques deployed in this study are outlined.

### 3.1 Cumulative Sum (CUSUM) Detector

The standard CUSUM [15] procedure is explained using the following equations.

---

**CUSUM:** $S_{0,i}^- = 0, \;\; S_{0,i}^+ = 0, \;\; \tilde{k}_i^+ = 0, \;\; \tilde{k}_i^- = 0,$

$$\begin{cases} S_{k,i}^+ = \max(0, S_{k-1,i}^+ + r_{k,i} - \bar{T}_i - \kappa_i), & \text{if } S_{k-1,i}^+ \leq \tau_i^+, \\ S_{k,i}^+ = 0 \;\text{ and }\; \tilde{k}_i^+ = \tilde{k}_i^+ + 1, & \text{if } S_{k-1,i}^+ > \tau_i^+. \end{cases} \tag{4}$$

$$\begin{cases} S_{k,i}^- = \min(0, S_{k-1,i}^- + r_{k,i} - \bar{T}_i + \kappa_i), & \text{if } S_{k-1,i}^- \geq \tau_i^-, \\ S_{k,i}^- = 0 \;\text{ and }\; \tilde{k}_i^- = \tilde{k}_i^- + 1, & \text{if } S_{k-1,i}^- < \tau_i^-. \end{cases} \tag{5}$$

**Design parameters:** bias $\kappa_i > 0$ and threshold $\tau_i > 0$.
**Output:** $alarm(s) = \tilde{k}_i^+ + \tilde{k}_i^-$.

---

From Eqs. 4–5, it can be observed that the CUSUM values $S_{k,i}^+$ and $S_{k,i}^-$ accumulate the distance measured $r_{k,i}$ over time to measure how far the values of the residual are from the target mean ($\bar{T}_i$). The slack variable $\kappa$ can be adjusted to tune this window for error. The parameters are chosen suitably to achieve a required false alarm rate $\mathcal{A}_i^*$.

### 3.2 Bad-Data Detector

The Bad-Data detector is widely used in the CPS security literature [16].

---

**Bad-Data Procedure:**

$$\text{If } \; |r_{k,i}| > \alpha_i, \;\; \tilde{k}_i = k, \quad i \in \mathcal{I}. \tag{6}$$

**Design parameter:** threshold $\alpha_i > 0$.

**Output:** alarm time(s) $\tilde{k}_i$.

---

Using the Bad-Data detector, an alarm is triggered if distance measure, taken as $|r_{k,i}|$, exceeds the threshold $\alpha_i$. Analogous to the CUSUM procedure, the parameter $\alpha_i$ is selected to satisfy a required false alarm rate $\mathcal{A}_i^*$.

### 3.3 *NoisePrint* (Machine Learning-based Device Fingerprinting)

*NoisePrint* is a sensor fingerprinting technique that makes use of a Support Vector Machine (SVM) [3]. It is based on the principle that when the system is in steady state [17], the residual vector of its model is a function of sensor and process noise. Therefore, it is possible to extract these sensor and process noise characteristics of the given ICS from the residual vectors. Following this, pattern recognition techniques such as machine learning are applied on the residual vectors to fingerprint the given sensor and process.

The proposed scheme begins with data collection which is then divided into smaller chunks to extract a set of time domain and frequency domain features. Features are combined and labeled with a sensor ID. A machine learning algorithm is used for classifying sensors based on their noise profiles. For more details, an interested reader is referred to [3,18].

## 4 Threat Model

Since the attacks taken into consideration for this work are on sensors, a few assumptions have been made about the attacker. These are given as follows:

1. The attacker has access to $y_{k,i} = C_i x_k + \eta_{k,i}$ (i.e., the $i$-th sensor measurements at the $k^{th}$ time instance).
2. The attacker has the knowledge about the system dynamics, the state-space matrices, the control inputs and outputs, and the implemented detection measure.

Tables 2 and 3 show the attacks carried out on SWaT and WADI. Based on their execution, these can be classified as follows:

- *Single-point Attack*—these types of attacks target a single point in the system, manipulating its value and/or disrupting its communication link.
- *Multi-point Attack*—in these types of attacks, multiple points are targeted simultaneously.
- *Stealthy Attack*—these are the attacks wherein the data value of a sensor is altered very slightly, which makes it difficult to detect the abnormality.

The single- and multi-point attacks, in turn, can be single-stage or multi-stage. In single-stage attacks, the attack points are limited to one particular stage of the plant, whereas in multi-point attacks, the target points can be spread across several stages. In real scenarios, these are dependent on the attacker's competence, extent of access and intentions.

The attacks mentioned in Tables 2 and 3 simulate data injection attacks of two kinds:

- *Bias Injection Attack*: The attacker's goal in this type of attack is to deceive the control system by sending incorrect sensor readings. The attack vector in such a scenario can be defined as:

$$\bar{y}_k = y_k + \delta_k, \tag{7}$$

where $\bar{y}_k$ is the general sensor measurement at a time instance $k$, $y_k$ is the actual sensor reading and $\delta_k$ is the biased value injected by the attacker.

For e.g., Atk-2-s in Table 2 is a simple attack wherein a bias is added to the LIT-101 reading such that the value read by the PLC is changed from the original, which is 659 mm, to a spoofed value of 850 mm. Similarly, in Atk-2-w in Table 3, the 2-FIT-001 value is changed from its original 0 $m^3/h$ to a 1.5 $m^3/h$, and the control actions taken by the PLC are based on this fake value.

– *Stealthy Attack:* In this case, the attack vector $\delta_k$ for Eq. (7) is chosen in a way that it stays inconspicuous while using statistical detectors. This happens because in these types of attacks, the residual vector may not noticeably change or exceed the thresholds, which is necessary for statistical detectors to confirm an attack.
A example of a stealthy attack is Atk-1-s from Table 2. In this attack, the reading of LIT-101 is originally 659 mm, and during the course of the attack, a small bias is repeatedly injected such that this value gradually increases by 1 mm every second.

Such attacks are operational technology (OT) attacks that aim to compromise the normal performance of the plant by manipulating sensor and/or actuator states. The SCADA system coupled with the SWaT and WADI testbeds provides an option of manually altering the sensor/actuator values that are being sent to the PLCs, and this func-

**Table 2.** List of attacks (SWaT): column 1 states the attack ID, and column 2 provides the details, wherein the '/' separates the system state before and during the attack.

| Attack ID | Description (Initial state/Attack state) |
|---|---|
| *Stage 1* | |
| Atk-1-s | LIT101 = 659 mm/change level +1 mm/s |
| Atk-2-s | LIT101 = 659 mm/LIT101 = 850 mm |
| Atk-3-s | LIT101 = 659 mm/LIT101 = 210 mm |
| Atk-4-s | LIT101 = 679 mm/LIT101 = 700 mm |
| Atk-5-s | LIT101 = 1029 mm/LIT101 = 700 mm |
| Atk-6-s | LIT101 = 789 mm/LIT101 = 789 mm |
| Atk-7-s | LIT101 = 784 mm/LIT101 = 600 mm |
| *Stage 3* | |
| Atk-8-s | L < LIT301 < H/LIT301 = HH+ |
| Atk-9-s | L < LIT301 < H/change level −1 mm/s |
| Atk-10-s | L < LIT301 < H/change level −0.5 mm/s |
| Atk-11-s | FIT301 = 0 $m^3/h$/FIT301 = 2 $m^3/h$ |
| Atk-12-s | L < LIT301 < H/water leakage attack |
| *Stage 4* | |
| Atk-13-s | FIT401 = 0.48 $m^3/h$/FIT401 = 0 $m^3/h$ |
| Atk-14-s | LIT401 < 1000 mm, P401 = ON/LIT401 = 1000 mm and P401 = ON |
| Atk-15-s | L < LIT401 < H, P301 = ON/LIT401 = 600 mm and P301 = ON |
| Atk-16-s | L < LIT401 < H/LIT401 < L |
| Atk-17-s | LIT401 = 1005 mm/LIT401 = 1005 mm |

**Table 3.** List of attacks (WADI): column 1 states the attack ID, and column 2 provides the details, wherein the '/' separates the system state before and during the attack.

| Attack ID | Description (Initial state/Attack state) |
|---|---|
| Atk-1-w | 1-FIT-001 = 1.71 $m^3/h$/1-FIT-001 = 1.5 $m^3/h$ |
| Atk-2-w | 2-FIT-001 = 0 $m^3/h$/2-FIT-001 = 1.5 $m^3/h$ |
| Atk-3-w | 2-FIT-003 = 0 $m^3/h$/2-FIT-003 = 1 $m^3/h$ |
| Atk-4-w | 1-LT-001 = 55%/1-LT-001 = 80% |
| Atk-5-w | 1-LT-001 = 40.21%/1-LT-001 = 40.21% |
| Atk-6-w | 2-LT-002 = 46%/2-LT-002 = 65% |
| Atk-7-w | 2-LT-002 = 71.2%/2-LT-002 = 71.2% |

tion has been used to simulate some of the simple bias injection attacks. For the more complicated attacks, customised Python programs have been developed that gradually change the attack vector to simulate a stealthy attack. Custom-coded modules developed at iTrust Labs [19] have been used that are able to communicate with the LabVIEW-based[1] SCADA interface in order to launch the stealthy attacks.

## 5 Performance Evaluation

### 5.1 Performance Metrics

The precision and sensitivity of the attack detection method are part of the criteria to analyse its effectiveness. The following metrics have been used to assess the three procedures:

– True Positive Rate (TPR) and False Negative Rate (FNR)—The TPR refers to the number of times the method correctly raises alarms (predicts an attack) over the duration of the attack. The FNR is an alternate way of expressing the same metric:

$$\text{FNR} = 100\ \% - \text{TPR}$$

– False Positive Rate (FPR) or False Alarm Rate (FAR)—this refers to the number of times the method incorrectly raises alarms in the absence of any attack.
– Time Taken for Detection (TTD)—this refers to the time taken by the procedure to raise an alarm in the event of an attack.

The TPR of the technique is a direct indication of its attack detection accuracy and must be as high as possible. The FPR represents the tendency of the procedure to raise false alarms, which is extremely inconvenient in practical scenarios, and should be satisfactorily small. A high TPR is not very beneficial if the mechanism takes too long to detect the attack. This is because in a realistic sense, the CPS performs critical, large-scale

[1] Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a system-design software developed by National Instruments. For attack tool see: https://gitlab.com/gyani/NiSploit.

processes that influence the surrounding economy in multiple ways. A significant delay in the detection of an attack can be detrimental not only to the system itself, but also to its end-users. Therefore, the detection mechanism must have reasonable TTD.

In practical applications, there often exists a trade-off between a high TPR and a low FPR. A detection method may have a high FPR while managing to achieve a good TPR. Likewise, it is also possible to design for a low FPR but at the cost of missing some attacks, resulting in a low TPR. Hence, the two rates must always be balanced such that a satisfactory TPR is attained while having a feasible FPR.

### 5.2 Normal Operation

As emphasized earlier, attack detection mechanisms must be designed in a way such that they do not raise too many false alarms. Hence, the detection techniques were implemented on both the plants, and their performances were observed when the plants were under normal operation.

For both the plants, the thresholds for the CUSUM and Bad-Data detectors have been designed to allow an FPR of 5% (or less). This is done to account for the temporary aberrations caused by technical glitches or external disturbances, which often occur in practical industrial plants. Each detector has thresholds and design parameters dedicated to each sensor, which are presented in Tables 4 and 5. It can be seen in these tables that, for both plants, these two attack detection methods generate false alarms within a reasonable window around the designed limit.

Figure 1 shows the residual from the system identification-based model for the level sensor (2-LT-002) in WADI. It can be seen that it mostly remains below its Bad-Data threshold during normal operation, shown in Fig. 1a. Likewise, the CUSUM values also stay within the thresholds for 2-LT-002 under normal operation, as seen in Fig. 1b. This implies that the design of the Bad-Data and CUSUM thresholds is in accordance with the requirement and it is feasible to implement these detectors on the plants under normal operating conditions.

When tested on SWaT, *NoisePrint* performed very well, with low or zero FPRs for almost all of the sensors. However, in the case of WADI, the FPRs for most of the sensors were above the desired 5%. The sensors in WADI are known to be sensitive to disturbances from the environment, thus resulting in some faults in their measurements, and this could be the reason *NoisePrint* fails to perform well.

From these figures and tables, it can be concluded that the detection methods perform satisfactorily well on both the testbeds under normal operating conditions. The $x$-axis for all the figures is the time in seconds for which sensor data is plotted. However, it is to be noted that these figures are for demonstration purposes only and do not show the complete dataset. For the normal operation of the water plants, the dataset is collected for more than a week and the attack data ranges from 5–30 min for each attack [20]. The FPR is only shown for the normal data evaluations. As for the case of the attack evaluation table in the following section, the data used was recorded only when the sensors were under attack, and hence shows FNR only. The rate (TPR) is calculated using the number of alarms raised for the whole duration of the attack.

**Table 4.** False positives under normal operation in SWaT.

| Sensor | FIT101 | LIT101 | FIT301 | LIT301 | FIT401 | LIT401 |
|---|---|---|---|---|---|---|
| *CUSUM detector* | | | | | | |
| Threshold | 0.0149 | 3.1168 | 0.2209 | 0.5529 | 0.0156 | 0.5674 |
| $\kappa$ | 0.0074 | 0.3117 | 0.0276 | 0.1382 | 0.0028 | 0.1135 |
| FAR | 5.54% | 5.19% | 5.34% | 4.65% | 4.02% | 4.03% |
| *Bad Data detector* | | | | | | |
| Threshold | 0.0205 | 1.4100 | 0.1184 | 0.4887 | 0.0108 | 0.4178 |
| FAR | 4.29% | 5.32% | 4.84% | 4.56% | 5.41% | 5.42% |
| *NoisePrint* | | | | | | |
| FAR | 0% | 1.29% | 8.3% | 2.44% | 0% | 0% |

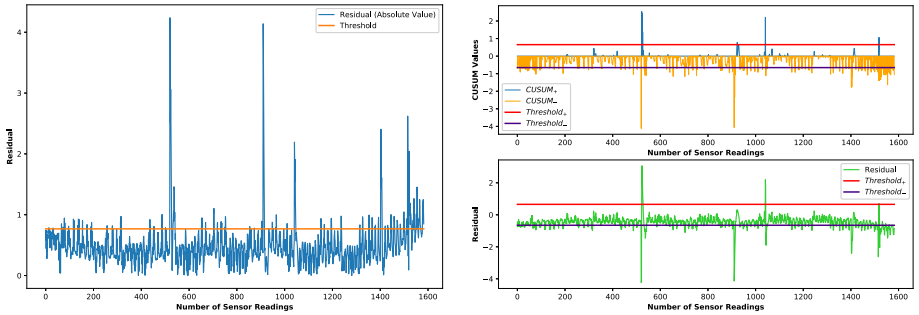
**Table 5.** False positives under normal operation in WADI.

| Sensor | 1-LT-001 | 2-LT-002 | 2-PIT-001 | 2-PIT-002 | 1-FIT-001 | 2-FIT-001 | 2-FIT-002 | 2-FIT-003 |
|---|---|---|---|---|---|---|---|---|
| *CUSUM detector* | | | | | | | | |
| Threshold | 1.109 | 0.6534 | 8.6809 | 0.2107 | 0.2964 | 0.0995 | 0.311 | 1.2972 |
| $\kappa$ | 0.3466 | 0.2042 | 0.8681 | 0.3511 | 0.0823 | 0.0829 | 0.0389 | 0.1081 |
| FAR | 4.61% | 3.76% | 5.01% | 3.47% | 4.29% | 4.13% | 4.93% | 5.01% |
| *Bad Data detector* | | | | | | | | |
| Threshold | 1.122 | 0.7674 | 3.5104 | 0.7239 | 0.2063 | 0.3018 | 0.1548 | 0.487 |
| FAR | 4.40% | 4.19% | 4.08% | 3.89% | 4.64% | 3.49% | 4.56% | 4.80% |
| *NoisePrint* | | | | | | | | |
| FAR | 13.04% | 6.95% | 21.74% | 6.95% | 6.08% | 11.30% | 4.34% | 11.30% |

### 5.3 Attack Detection

The three detection techniques were tested under different attack scenarios on both the plants. Tables 2 and 3 show the attacks carried out on SWaT and WADI, respectively. The residuals for the sensors from the system identification-based models were obtained and the detection techniques were applied while the plants were under attack. The performance metrics were computed for the different attacks on each of the testbeds and can be seen in Tables 6 and 7.

In the case of SWaT, it can be seen in Table 6 that the CUSUM and Bad-Data detectors perform well under a variety of bias injection attacks, like Atk-11-s, Atk-4-s and Atk-5-s. However, they fail to detect the stealthy attacks Atk-17-s and Atk-6-s. Whereas, *NoisePrint* is able to successfully detect the presence of all attacks, including the stealthy attacks, and demonstrates a comparable TPR for other cases. The attacks that report poor TPR while using CUSUM and Bad-Data thresholds can be detected better using *NoisePrint*. However, the superior performance of *NoisePrint* comes at the cost of speed of detection. The time taken by the CUSUM and Bad-Data detectors to confirm the occurrence of the attack is considerably less than that of *NoisePrint*, implying that they have a better TTD compared to *NoisePrint*.




(a) Bad-Data detection (b) CUSUM detection

**Fig. 1.** Statistical attack detection methods applied on the residual for level sensor (2-LT-002) estimates from WADI under normal operation. X-axis shows number of sensor reading sampled at 1 s intervals.

Figure 2 shows the residual when the level sensor (LIT-101) in SWaT is under a stealthy attack. In this attack, an attacker chooses to spoof the sensor measurement at the same value as the last known normal reading, thus deceiving the controller, while the real process state continues to progress differently. As seen in Figure 2a, the residual stays below the threshold during the stealthy attack. Similarly, in Figure 2b, it can be seen that the CUSUM values also always stay below the CUSUM thresholds. This shows that the stealthy attack could not be detected by either of the two detectors. However, as mentioned in Table 6 *NoisePrint* is able to detect this attack.

In the case of WADI, when the CUSUM detector is implemented on the residuals obtained from the system models, unsatisfactory TPRs are reported for all the attacks, as shown in Table 7. The Bad-Data detector performs reasonably well for attacks Atk-2-w and Atk-7-w, while *NoisePrint* shows a 100% TPR for attacks Atk-2-w, Atk-3-w and Atk-7-w. Both methods report poor TPRs for the other attacks. Similar to the case of SWaT, the TTD of *NoisePrint* is much higher than that of the Bad-Data detector.

These results show that while the statistical detectors, Bad-Data and CUSUM, are successfully able to confirm basic attacks such as bias injections, they fail to detect the more complicated stealthy attacks. This is expected because stealthy attacks are devised such that they do not tend to cause substantial changes to the residuals obtained from models, thereby ensuring the thresholds that determine the presence of an attack are not crossed. On the other hand, *NoisePrint* is able to identify such attacks, since the attacker may not be able to replicate the process and sensor noise, which form the basis of detection in *NoisePrint*. However, despite achieving better accuracy, *NoisePrint* falls behind in terms of detection speed.

Given the nature and performance of the detection mechanisms, the practical applicability of the methods can be challenged. The testbeds used in this work are small-scale and hence, obtaining complete system models for them was a feasible task. This might not be the case for actual industrial CPSs. A possible solution to this would be dividing the larger plants into several sub-stages (based on the processes taking place) and having multiple models corresponding for each sub-system.

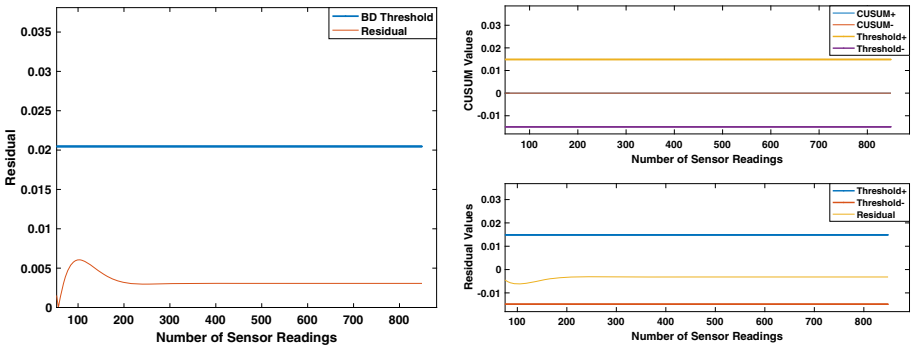
**Table 6.** Attack detection performance on SWaT testbed.

| Attack | NoisePrint | | | CUSUM | | | Bad Data | | |
|---|---|---|---|---|---|---|---|---|---|
| | TPR | FNR | TTD (s) | TPR | FNR | TTD (s) | TPR | FNR | TTD (s) |
| *Single point attacks* | | | | | | | | | |
| Atk-8-s | 85.72% | 14.28% | 121.22 | 17.46% | 82.54% | 2 | 16.75% | 83.25% | 2 |
| Atk-9-s | 14.50% | 85.50% | 179 | 88.15% | 11.85% | 2 | 93.18% | 6.82% | 2 |
| Atk-10-s | 80.64% | 19.35% | 130.09 | 56.30% | 43.70% | 5 | 58.48% | 41.52% | 3 |
| Atk-11-s | 87.50% | 12.50% | 89.59 | 100% | 0% | 1 | 100% | 0% | 1 |
| Atk-12-s | 63.63% | 36.37% | 117.83 | 95.42% | 4.58% | 6 | 96.64% | 3.36% | 6 |
| Atk-1-s | 88.88% | 11.12% | 32.48 | 91.16% | 8.83% | 2 | 91.34% | 8.66% | 1 |
| Atk-2-s | 67.56% | 32.44% | 46.90 | 85.08% | 14.92% | 1 | 78.02% | 21.98% | 1 |
| Atk-3-s | 90.91% | 9.09% | 35.25 | 98.92% | 1.08% | 1 | 99.08% | 0.92% | 1 |
| Atk-7-s | 88.24% | 11.76% | 57.35 | 77.58% | 22.42% | 1 | 60.62% | 39.38% | 1 |
| Atk-13-s | 55% | 45% | 44.43 | 32.82% | 67.18% | 2 | 13.94% | 86.06% | 2 |
| Atk-16-s | 86.21% | 13.79% | 56.26 | 6.21% | 93.79% | 1 | 6.32% | 93.68% | 1 |
| *Multi-point attacks* | | | | | | | | | |
| Atk-14-s | 81.82% | 18.18% | 125.59 | 16.32% | 83.68% | 1 | 6.76% | 93.24% | 1 |
| Atk-15-s | 77.78% | 22.22% | 105.3 | 54.68% | 45.32% | 2 | 99.64% | 0.36% | 2 |
| Atk-4-s | 94.73% | 5.26% | 35.59 | 99.66% | 0.34% | 1 | 100% | 0% | 1 |
| Atk-5-s | 90.47% | 9.53% | 44.50 | 99.68% | 0.32% | 1 | 100% | 0% | 1 |
| *Stealthy attacks* | | | | | | | | | |
| Atk-17-s | 80% | 20% | 67.03 | 0% | 100% | **ND** | 0% | 100% | **ND** |
| Atk-6-s | 75% | 25% | 174.84 | 0% | 100% | **ND** | 0% | 100% | **ND** |

**Table 7.** Attack detection performance on WADI (System identification model).

| Attack | NoisePrint | | | CUSUM | | | Bad Data | | |
|---|---|---|---|---|---|---|---|---|---|
| | TPR | FNR | TTD (s) | TPR | FNR | TTD (s) | TPR | FNR | TTD (s) |
| *Single point attacks* | | | | | | | | | |
| Atk-1-w | 25% | 75% | 100 | 7.89 % | 92.11 % | 1 | 21.74 % | 78.26 % | 1 |
| Atk-2-w | 100% | 0% | 50 | 51.28 % | 48.72 % | 2 | 91.11 % | 8.89 % | 2 |
| Atk-3-w | 100% | 0% | 50 | 22.22 % | 77.78 % | 1 | 13.16 % | 86.84 % | 1 |
| Atk-4-w | 20.51% | 79.49% | 150 | 1.81 % | 98.19 % | 1 | 3.59 % | 96.41 % | 1 |
| Atk-6-w | 56.25% | 43.75% | 100 | 17.67 % | 82.33 % | 1 | 32.49 % | 67.51 % | 1 |
| *Stealthy attacks* | | | | | | | | | |
| Atk-5-w | 19.44% | 80.56% | 200 | 1.40 % | 98.60 % | 2 | 2.51 % | 97.49 % | 1 |
| Atk-7-w | 100% | 0% | 50 | 45.79 % | 54.21 % | 3 | 94.02 % | 5.98 % | 1 |

In the case of *NoisePrint*, its longer detection time might render it less efficient when applied to some industrial CPSs, such as power grids, which require immediate response during attacks or anomalies. However, its accuracy is an important advantage when it comes to large systems with several sensors, and the method is still be applicable to CPSs wherein the attacks could take a longer time to cause any physical harm.




(a) Bad-Data detection (b) CUSUM detection

**Fig. 2.** Statistical attack detection methods (Bad-Data and CUSUM) applied on the residual for level sensor (LIT-101) estimates from SWaT under stealthy attack

## 6 Conclusions

From the model validation results, it is understood that the models generated using well-established system identification algorithms perform reasonably well. An important insight is that obtaining a normal reference system model for the plants and sensors sensitive to environmental disturbances (e.g., for the WADI testbed in this study) is a non-trivial task. It is deduced that bias injection attacks on sensors that are quite similar to faults can be easily detected using statistical techniques like Bad-Data and CUSUM detectors. However, it is observed that advanced stealthy attacks require more sophisticated detection techniques, like *NoisePrint*. From the various tests carried out on the plants, it is concluded that while detection methods must be able to demonstrate accuracy, their attack detection speed is also a crucial metric for critical CPSs.

**Acknowledgements.** This work was supported by the SUTD start-up research grant SRG-ISTD-2017-124. The authors thank the reviewers for their comments. The authors express their gratitude to the iTrust research centre at Singapore University of Technology and Design for their research facilities, which have been extensively used in this work.

## References

1. Cardenas, A., Amin, S., Lin, Z., Huang, Y., Huang, C., Sastry, S.: Attacks against process control systems: risk assessment, detection, and response. In: 6th ACM Symposium on Information. Computer and Communications Security, pp. 355–366 (2011)
2. Ahmed, C.M., Zhou, J.: Challenges and opportunities in CPS security: a physics-based perspective. IEEE Secur. Priv. (2020)
3. Ahmed, C.M., et al.: NoisePrint: attack detection using sensor and process noise fingerprint in cyber physical systems. In: AsiaCCS 18, pp. 483–497. ACM (2018)
4. Rocchetto, M., Tippenhauer, N.O.: On attacker models and profiles for cyber-physical systems. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9879, pp. 427–449. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45741-3_22

5. Krotofil, M., Gollmann, D.: Industrial control systems security: what is happening? In: 2013 11th IEEE International Conference on Industrial Informatics (INDIN), pp. 664–669, July 2013
6. Shoukry, Y., Martin, P., Yona, Y., Diggavi, S., Srivastava, M.: PyCRA: physical challenge-response authentication for active sensors under spoofing attacks. In: CCS 15, pp. 1004–1015. ACM (2015)
7. Mitchell, R., Chen, I.-R.: A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. (CSUR) **46**(4), 1–29 (2014)
8. SWaT: Secure Water Treatment Testbed (2015). https://itrust.sutd.edu.sg/wp-content/uploads/sites/3/2015/11/Brief-Introduction-to-SWaT_181115.pdf
9. Ahmed, C.M., Palleti, V.R., Mathur, A.P.: WADI: a water distribution testbed for research in the design of secure cyber physical systems. In: CPS Week. CySWATER 2017, pp. 25–28. ACM, 2017
10. Wei, X., Verhaegen, M., van Engelen, T.: Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques. Int. J. Adapt. Control Signal Process. **24**(8), 687–707 (2010). https://doi.org/10.1002/acs.1162
11. Ahmed, C.M., Murguia, C., Ruths, J.: Model-based attack detection scheme for smart water distribution networks. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ASIA CCS 2017, pp. 101–113. ACM, New York (2017). https://doi.org/10.1145/3052973.3053011
12. Qadeer, R., Murguia, C., Ahmed, C.M., Ruths, J.: Multistage downstream attack detection in a cyber physical system. In: Katsikas, S.K., et al. (eds.) CyberICPS/SECPRE -2017. LNCS, vol. 10683, pp. 177–185. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72817-9_12
13. Murguia, C., Ruths, J.: Characterization of a CUSUM model-based sensor attack detector. In: 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 1303–1309, December 2016
14. Urbina, D.I., et al.: Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1092–1105. ACM (2016)
15. Montgomery, D.: Introduction to Statistical Quality Control. Wiley, Hoboken (2009)
16. Liu, T., Gu, Y., Wang, D., Gui, Y., Guan, X.: A novel method to detect bad data injection attack in smart grid. In: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 49–54. IEEE (2013)
17. Aström, K.J., Wittenmark, B.: Computer-Controlled Systems, 3rd edn. Prentice-Hall Inc., Upper Saddle River (1997)
18. Ahmed, C.M., Zhou, J., Mathur, A.P.: Noise matters: using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In: Proceedings of the 34th Annual Computer Security Applications Conference, pp. 566–581 (2018)
19. Adepu, S., Mishra, G., Mathur, A.: Access control in water distribution networks: a case study. In: 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 184–191, July 2017
20. Palleti, V.R., Mishra, V.K., Ahmed, C.M., Mathur, A.: Can replay attacks designed to steal water from water distribution systems remain undetected? ACM Trans. Cyber Phys. Syst. (2020)