NoisePrint: Attack Detection Using Sensor and Process Noise Fingerprint in Cyber Physical Systems

Chuadhry Mujeeb Ahmed
SUTD, Singapore
chuadhry@mymail.sutd.edu.sg

Martín Ochoa
SUTD, Singapore AND Universidad del Rosario, Bogotá, Colombia

Jianying Zhou, Aditya P. Mathur
SUTD, Singapore
jianying_zhou@sutd.edu.sg,aditya_mathur@sutd.edu.sg

Rizwan Qadeer
SUTD, Singapore
rizwan_qadeer@sutd.edu.sg

Carlos Murguia
Melbourne University, Australia
carlos.murguia@unimelb.edu.au

Justin Ruths
UT Dallas, USA
jruths@utdallas.edu

ABSTRACT

An attack detection scheme is proposed to detect data integrity attacks on sensors in Cyber-Physical Systems (CPSs). A combined fingerprint for sensor and process noise is created during the normal operation of the system. Under sensor spoofing attack, noise pattern deviates from the fingerprinted pattern enabling the proposed scheme to detect attacks. To extract the noise (difference between expected and observed value) a representative model of the system is derived. A Kalman filter is used for the purpose of state estimation. By subtracting the state estimates from the real system states, a residual vector is obtained. It is shown that in steady state the residual vector is a function of process and sensor noise. A set of time domain and frequency domain features is extracted from the residual vector. Feature set is provided to a machine learning algorithm to identify the sensor and process. Experiments are performed on two testbeds, a real-world water treatment (SWaT) facility and a water distribution (WADI) testbed. A class of *zero-alarm* attacks, designed for statistical detectors on SWaT are detected by the proposed scheme. It is shown that a multitude of sensors can be uniquely identified with accuracy higher than 90% based on the noise fingerprint.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection**; • **Computer systems organization → Sensors and actuators**; **Embedded systems**; *Dependable and fault-tolerant systems and networks*;

KEYWORDS

Cyber Physical Systems, Security, CPS/ICS Security, Sensors and Actuators, Device Fingerprinting, Physical Attacks

ASIA CCS '18, June 4–8, 2018, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5576-6/18/06…$15.00
https://doi.org/10.1145/3196494.3196532

ACM Reference Format:
Chuadhry Mujeeb Ahmed, Martín 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 *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security, June 4–8, 2018, Incheon, Republic of Korea.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3196494.3196532

1 INTRODUCTION

A Cyber Physical System (CPS) is a combination of computing elements and physical phenomenon [8, 39]. In particular we will consider examples of water treatment and distribution plants in this paper, also known as Industrial Control Systems (ICS) [29]. An ICS consists of cyber components such as Programmable Logic Controllers (PLCs), sensors, actuators, Supervisory Control and Data Acquisition (SCADA) workstation, and Human Machine Interface (HMI) elements interconnected via a communications network. The PLCs control a physical process based on the sensor data via a SCADA workstation. The advances in communication technologies resulted in widespread of such system to better monitor and operate ICS, but this connectivity also exposes physical processes to malicious entities on the cyber domain. Recent incidents of sabotage on these systems [13, 20, 48], have raised concerns on the security of CPS [12].

Challenges in CPS security are different as compared with conventional IT systems, especially in terms of consequences in case of a security lapse. Attacks on CPS might result in damage to the physical property, as a result of an explosion [16, 56] or severely affecting people who depend on a critical infrastructure as was the case of recent power cutoff in Ukraine [13]. Data integrity is an important security requirement for CPS [24] therefore, integrity of sensor data should be ensured. Sensor data can either be spoofed in cyber (digital) domain [52] or in physical (analog) domain [47, 49]. Sensors are a bridge between the physical and cyber domains in a CPS. Traditionally, an intrusion detection system (IDS) monitors a communication network or a computing host to detect attacks. However, physical tampering with sensors or sensor spoofing in physical/analog domain, may go undetected by the legacy IDS [47].

Data integrity attacks on sensor measurement and impact of such attacks have been studied in theory, including false data injection [34], replay attacks [33], and stealthy attacks [17]. These

previous studies proposed attack detection methods based on system model and statistical fault detectors [3, 5, 37, 38] and also point out the limitations of such fault detectors against an adversarial manipulation of the sensor data. In practice attacks on sensor measurement can be launched by analog spoofing attacks [28, 47, 57]), or by tampering with the communication channel between a sensor and a controller by means of a classical *Man-in-The-Middle (MiTM)* attack [52].

The proposed scheme serves as a device identification framework and it can also detect a range of attacks on sensors. The proposed attack detection framework improves on the limitations of model based attack detection schemes. In general for a complex CPS there can be many possible attack scenarios. However *zero-alarm* attack is a worst case scenario for a model based attack detection method employing a threshold based detector. A *zero-alarm* attack exposes the limitations of threshold based statistical attack detection methods. To be fair while making comparison, we choose the same attack vector namely *zero-alarm* attack. Another important thing is that the input to *NoisePrint* and reference methods is the same, i.e. a residual vector. We also executed bias attack as an example of an attack which can be detected using CUSUM and Bad-Data detectors. The proposed scheme is a non-intrusive sensor and process fingerprinting method to authenticate sensors transmitting measurements to one or more PLCs. To apply this method we need to extract noise pattern, for which system model of an ICS is used. This scheme intelligently uses model of the system in a novel way to extract noise pattern and then input that noise to *NoisePrint* as shown in Figure 1. The input to *NoisePrint* block is a function of sensor and process noise. Sensor noise is due to construction of the sensor and process noise due to variations in the process e.g. fluid sloshing in a storage tank in a process plant. Sensor noise is different from one sensor to another because of hardware imperfections during the manufacturing process [19]. Process noise is unique among different processes essentially because of different process dynamics. Sensor and process noise can be captured using a real system state (from sensor measurements) and system state estimate (from system model). These noise variations affect each device and process differently and thus are hard to control or reproduce [23] making physical or digital spoofing of sensor noise profiles challenging.

A technique, referred to as *NoisePrint*, is designed to fingerprint sensor and process found in ICS. *NoisePrint* creates a noise fingerprint based on a set of time domain and frequency domain features that are extracted from the sensor and process noise. To extract noise pattern a system model based method is used. A two-class Support Vector Machine (SVM) is used to identify each sensor from a dataset, comprising of a multitude of industrial sensors. According to the ground truth one class is labeled as legitimate sensor/process and other class of illegitimate data (including attacks and data from rest of the sensors in the plant). Experiments are performed on two operational water treatment and distribution facilities accessible for research [6, 31]. A class of attacks as explained in threat model are launched on a real water treatment testbed and results are compared with reference statistical methods. Sensor identification accuracy is observed to be as high as 96%, and at least 90% for a range of sensors.

The major contributions of this work are thus:

- A novel fingerprinting framework that is based on sensor and process noise, and is a function of hardware characteristics of a device and Physics of the process.
- A detailed evaluation of the proposed *NoisePrint* as attack detection method, for a class of sensor spoofing attacks.
- Extensive empirical performance evaluation on realistic testbeds.
- A comparison of the performance of the proposed scheme with the reference statistical detectors.
- A detailed evaluation of the proposed *NoisePrint* as a device identification method in a complex CPS.

This work evaluates *NoisePrint* in the context of water treatment and water distribution testbeds [6, 31]. Commonly found industrial sensors are studied, but without loss of generality, the analysis is applicable to other industrial applications.

2 SYSTEM DESCRIPTION AND ATTACK DETECTION

In this section we will explain the overview of the proposed scheme. Figure 1 shows the block diagram of the proposed scheme.

2.1 System Dynamics

In Figure 1, the first block represents data collection step from the real water testbeds. A linear time invariant system model is obtained using either first principles (laws of Physics) or subspace system identification techniques. Then, we construct a Kalman filter which is used to obtain estimates for the system states and to find the residual vector. We studied the system design and functionality of the water treatment (SWaT) testbed [31] to obtain the system model. For the water distribution (WADI) testbed, we used data collected under regular operation (no attacks) and subspace system identification techniques [40] to obtain a system model. For both testbeds, resulting system models are Linear Time Invariant (LTI) discrete time state space model of the form:

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

At the time-instants $k \in \mathbb{N}$, the output of the process y_k is sampled and transmitted over a communication channel as shown in Figure 2. The control action u_k is computed based on the received sensor measurement $\bar{y}_k$. Data is exchanged between different entities of this control loop and it is transmitted via communication channels. There are many potential points where an attacker can hack into the system. For instance, *Man-in-The-Middle (MiTM)* attacks at the communication channels and physical attacks directly on the infrastructure. In this paper, we focus on sensor spoofing attacks, which could be accomplished through a *Man-in-The-Middle (MiTM)* scheme [52] or a replacement of on board PLC software [15, 22, 25]. After each transmission and reception, the attacked output $\bar{y}_k$ takes the form:

$$\bar{y}_k := y_k + \delta_k = Cx_k + \eta_k + \delta_k, \tag{2}$$

where $\delta_k \in \mathbb{R}^m$ denotes sensor attacks. Throughout this paper, we reserve the variable k as the discrete-time index of various sequences; where clear, we omit reminding the reader that $k \in \mathbb{N}$.

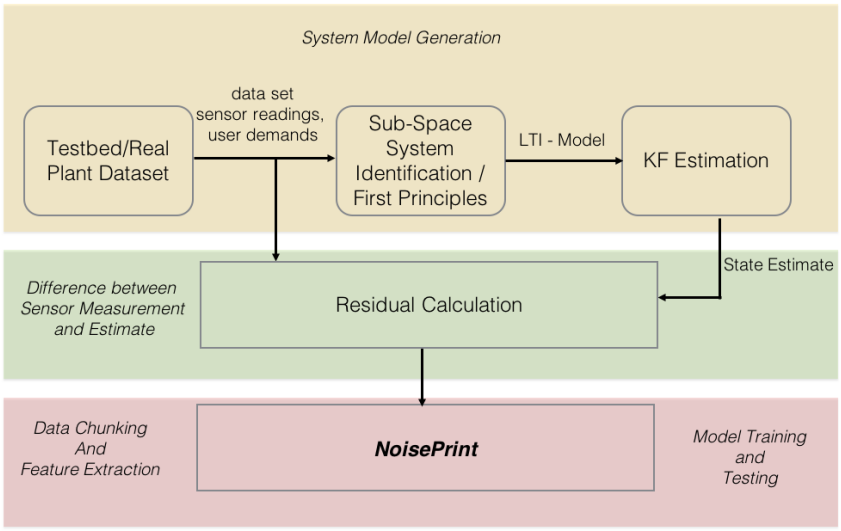

Figure 1: Block diagram explaining an overview of the proposed attack detection scheme.

2.2 Kalman Filter

We used Kalman filter to estimate the state of the system based on the available output y_k,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L_k(\bar{y}_k - C\hat{x}_k), \tag{3}$$

with estimated state $\hat{x}_k \in \mathbb{R}^n$, $\hat{x}_1 = E[x(t_1)]$, where $E[\cdot]$ denotes expectation, and gain matrix $L_k \in \mathbb{R}^{n\times m}$. Define the estimation error $e_k := x_k - \hat{x}_k$. In the Kalman filter, the matrix L_k is designed to minimize the covariance matrix $P_k := E[e_k e_k^T]$ (in the absence of attacks). Given the system model (1),(2) and the estimator (3), the estimation error is governed by the following difference equation

$$e_{k+1} = (A - L_k C)e_k - L_k\eta_k - L_k\delta_k + \upsilon_k. \tag{4}$$

If the pair (A, C) is detectable, the covariance matrix converges to steady state in the sense that $\lim_{k\to\infty} P_k = P$ exists [10]. We assume that the system has reached steady state before an attack occurs. Then, the estimation of the random sequence $x_k, k \in \mathbb{N}$ can be obtained by the estimator (3) with P_k and L_k in steady state. It can be verified that, if $R_2 + CPC^T$ is positive definite, the following estimator gain

$$L_k = L := (APC^T)(R_2 + CPC^T)^{-1}, \tag{5}$$

leads to the minimal steady state covariance matrix P, with P given by the solution of the algebraic Riccati equation:

$$APA^T - P + R_1 = APC^T(R_2 + CPC^T)^{-1}CPA^T. \tag{6}$$

The reconstruction method given by (3)-(6) is referred to as the steady state Kalman Filter, cf. [10].

2.3 Attack Detection Framework

In this section, we explain the details of the proposed attack detection scheme. First, we discuss the Kalman filter based state estimation and residual generation. Then, we present the design of our residual-based fingerprinting method (namely *NoisePrint*).

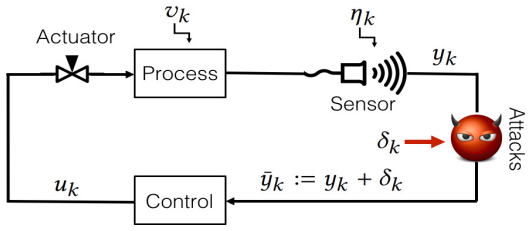

Figure 2: A general CPS under sensor attacks.

2.3.1 Residual and Noise Fingerprint. **Proposition 1.** In steady state [10], residual vector is a function of sensor and process noise. Consider the process (1), the Kalman filter (3)-(6). The residual vector is given as, $r_k = Ce_k + \eta_k$ and $e_k = \sum_{i=0}^{k-2}(A - LC)^i(\upsilon_{k-i-1} - L\eta_{k-i-1})$, where $\upsilon_k \in \mathbb{R}^n$ is the process noise and $\eta_k \in \mathbb{R}^m$ is the sensor noise.

Proof: Due to space limitations the proof is given in Appendix A. This is an important intuition behind the idea of *NoisePrint* as it can be seen that the residual vector obtained from the system model, is a function of process and sensor noise. Using system model and system state estimates it is possible to extract the sensor and process noise. Once we have obtained these residual vectors capturing sensor and process noise characteristics of the given ICS, we can proceed with pattern recognition techniques (e.g. machine learning) to fingerprint the given sensor and process.

2.3.2 Design of NoisePrint. Figure 3 shows the steps involved in composing a sensor and process noise fingerprint. The proposed scheme begins with data collection and then divides data 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 sensor classification.

Table 1: List of features used. Vector x is time domain data from the sensor for N elements in the data chunk. Vector y is the frequency domain feature of sensor data. y_f is the vector of bin frequencies and y_m is the magnitude of the frequency coefficients.

Feature	Description
Mean	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
Std-Dev	$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)^2}$
Mean Avg. Dev	$D_{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N}\lvert x_i - \bar{x}\rvert$
Skewness	$\gamma = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^3$
Kurtosis	$\beta = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^4 - 3$
Spec. Std-Dev	$\sigma_s = \sqrt{\frac{\sum_{i=1}^{N}(y_f(i)^2)*y_m(i)}{\sum_{i=1}^{N} y_m(i)}}$
Spec. Centroid	$C_s = \frac{\sum_{i=1}^{N}(y_f(i))*y_m(i)}{\sum_{i=1}^{N} y_m(i)}$
DC Component	$y_m(0)$

***Residual Collection*:** The next step after obtaining a system model for an ICS is to calculate the residual vector as explained in previous section. Residual is collected for different types of industrial sensors present in SWaT and WADI testbeds.We collect residual for the level sensors in SWaT testbed and a multitude of sensors in WADI testbed. The objective of residual collection step is to extract sensor and process noise by analyzing the residual vector. When the plant is running, an error in sensor reading is a combination of sensor noise and process noise (water sloshing etc.). The collected residual is analyzed, in time and frequency domains, to examine the noise patterns, which are found to follow Gaussian distribution. Sensors and processes are profiled using variance and other statistical features in the noise vector. The experiment is run, to obtain sensor and process profile, so that it can be used for later testing. A machine learning algorithm is used to profile sensors from fresh readings (test-data). Noise fingerprints can be generated over time or at the commissioning phase of the plant. Since these noise fingerprints are extracted from the system model, it does not matter if the process is dynamic or static.

***Feature Extraction*:** Data is collected from sensors at a sampling rate of one second. Since data is collected over time, we can use raw data to extract time domain features. We used the Fast Fourier Transform (FFT) algorithm [55] to convert data to frequency domain and extract the spectral features. In total, as in Table 1, eight features are used to construct the fingerprint.

***Data Chunking*:** After residual collection, the next step is to create chunks of dataset. In following sections, it will be seen that we have performed experiments on a dataset collected over 14 days in WADI testbed. An important purpose of data chunking is to find out, *how much is the sample size to train a well-performing machine learning model? and How much data is required to make a decision about presence or absence of an attacker?* The whole residual dataset (total of N readings) is divided into m chunks (each chunk of $\lfloor \frac{N}{m} \rfloor$), we calculate the feature set $< F(C_i) >$ for each data chunk i. For each sensor, we have m sets of features $< F(C_i) >_{i \in [1,m]}$. For n sensors

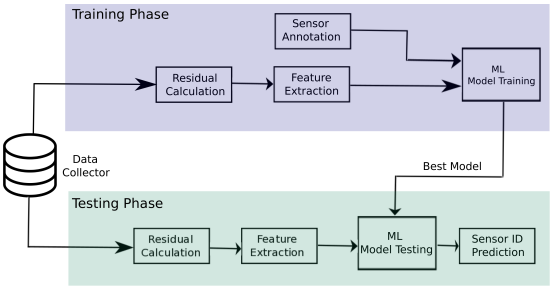

Figure 3: *NoisePrint* Framework.

we can use $n \times m$ sets of features to train the multi-class SVM. We use supervised learning method for sensor identification which has two phases– training and testing. For both phases, we create chunks in a similar way as explained above.

***Size of Training and Testing Dataset*:** It is found empirically that 2-class SVM produced highest accuracy for the chunk size of $\lfloor \frac{N}{m} \rfloor = 120$. For a total of m feature sets for each sensor, at first we used half ($\frac{m}{2}$) for training and half ($\frac{m}{2}$) for testing. To analyze the accuracy of the classifier for smaller feature sets during training phase, we began to reduce number of feature sets starting with $\frac{m}{2}$. Classification is then carried out for the following corresponding range of feature sets for Training : $\{\frac{m}{2}, \frac{m}{3}, \frac{m}{4}, \frac{m}{5}, \frac{m}{10}\}$, and for Testing : $\{\frac{m}{2}, \frac{2m}{3}, \frac{3m}{4}, \frac{4m}{5}, \frac{9m}{10}\}$, respectively. In section 5, empirical results are presented for such feature sets and the one with best performance is chosen, for further analysis of the proposed scheme. For the classifier we have used a multi-class SVM library [14], as briefly described in Appendix C.

3 ATTACKER AND ATTACK MODEL

In this work, we consider specific cyber and physical attacks on sensor measurements in an ICS, as shown in Figure 2. First, we lay down our assumptions about the attacker, followed by justification for such assumptions. In this section, we introduce the types of attacks launched on our secure water treatment testbed (SWaT). Essentially, the attacker model encompasses the attacker's intentions and its capabilities. The attacker may choose its goals from a set of intentions [50], including performance degradation, disturbing a physical property of the system, or damaging a component. In our experiments, three classes of attacks from literature [5, 11, 34, 37] are designed and executed.

3.1 Attacker Model

Assumptions on Attacker: It is assumed that the attacker has access to $y_{k,i} = C_i x_k + \eta_{k,i}$ (i.e., the opponent has access to i^{th} sensor measurements). Also, the attacker knows the system dynamics, the state space matrices, the control inputs and outputs, and the implemented detection procedure. An attacker can not arbitrarily change sensor measurements by learning and adding the sensor and process noise to a modified sensor value. We do not consider replay attack in this article because noise profile for process and sensor is preserved during a replay attack.

We consider a strong adversary who is able to launch cyber and physical attacks. In an ICS, sensors, actuators, and PLCs communicate with each other via communication networks. An attacker can compromise these communication links in a classic *Man-in-The-Middle (MiTM)* attack [2, 9, 52], for example, by breaking into the link between sensors and PLCs. Besides false data injection in sensor readings via cyber domain, an adversary can also physically tamper a sensor, to drive a CPS into an unstable state. Therefore, we need to authenticate sensor measurements, which are transmitted to a controller. A *malicious insider* is an attacker with physical access to the plant and thus to its devices such as level sensors. However, an attacker who can physically replace or tamper sensors may not necessarily be an *insider*, because critical infrastructures, e.g., for water and power, are generally distributed across large areas [21, 51]. An *outsider*, e.g., end user, can also carry out a physical attack on sensors such as smart energy monitors.

3.2 Attack Scenarios

Data Injection Attacks: For data injection attacks, it is considered that an attacker injects or modifies the real sensor measurement. In general for a complex CPS there can be many possible attack scenarios. We consider a simple bias attack to show a comparison between reference and proposed methods. However *zero-alarm* attack is a worst case scenario for a model based attack detection methods employing a threshold based detector and exposes the limits of such detectors while *NoisePrint* can detect those attacks. In our experiments, we consider the following two types of data injection attacks:

- *Bias Injection Attack*: First, a failure-like attack is designed. The attacker's goal is to deceive the control system by sending incorrect sensor measurements. In this scenario, the level sensor measurements are increased while the actual tank level is invariant. This makes the controller think that the attacked values are true sensor readings, and hence, the water pump keeps working until the tank is empty and cause the pump to burn out. The attack vector can be defined as,

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

  where δ_k is the bias injected by the attacker.
- *Zero-Alarm Attack for Statistical Detectors:* This attack is designed so as not to be detected by statistical detectors e.g. Bad-Data or CUmmulative SUM (CUSUM) change detectors. We implemented these two detectors in SWaT testbed to compare the performance of the proposed *NoisePrint* with these reference schemes. An attack is detected by analyzing the statistics of the residual vector based on certain thresholds. Design of such attacks is presented in section 5, after giving a brief description of these statistical detectors and how it works. Essentially an attacker chooses attack vector δ_k in (7) in a way that it stays stealthy against statistical detectors. In literature [5, 37] impact of such attacks has already been studied. We call these attacks as *zero-alarm* as the statistical detectors will not raise any alarms even the system was under attack, enabling the attacker to conceal its data injection while still impacting the system.

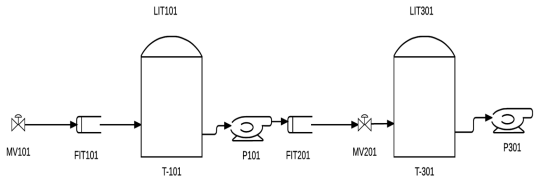

Figure 4: Experimentation setup in SWaT testbed. *LIT* represents a level sensor in tank *T*, along with flow meters *FIT* and pump *P*.

3.3 Attack Execution

Cyber Domain: Data traffic from sensors to PLCs is intercepted in a *Man-in-The-Middle (MiTM)* manner and packets are inspected to change the payload (sensor measurement). Depending on the attacker's strategy, a false reading is injected to either execute a bias injection attack or a *zero-alarm* attack.

Physical Domain: Sensor measurements can be spoofed in physical domain by bringing a malicious device near the sensing environment [47]. Hence both *bias injection attack and* zero-alarm *attack* can be executed in the physical domain. An attacker with the physical access to the plant can physically tamper with the sensors. It is demonstrated in the evaluation section that the sensor noise is a function of hardware characteristics of the device and possesses a unique fingerprint. Therefore, any physical tampering will result in the deviation from the reference noise pattern.

4 EXPERIMENTATION SETUP

The experiments are carried out in a state-of-the-art water treatment and distribution facility [6, 31]. The proposed method is tested on these two testbeds to demonstrate its viability on different cyber physical systems. To further diversify the study, system model for a portion of SWaT testbed is obtained using laws of Physics and system model for WADI is produced using sub-space system identification technique [40]. We executed attacks on the water storage tanks (via level sensors therein) in two different stages of SWaT testbed. For sensor identification based on the system model, we collected data from WADI testbed and used the proposed method to identify a sensor against adversarial physical manipulations of a sensor. we give a detailed explanation of both testbeds in Appendix D, for an interested reader.

4.1 SWaT Two-Tank System Model

We performed sensor spoofing attacks on two different processes (water tanks), as a *Man-in-The-Middle (MiTM)* manner [52] in SWaT testbed [31]. In Figure 4 an illustration of the two stages used in experiment are shown.

The intuition behind this step in the proposed scheme, is that if a system model is carefully designed by considering physical principles and system dynamics, we can calculate residual vector for *NoisePrint*. It could detect the fault or raise an alarm if there is an anomaly in noise dynamics of the system. A joint model for both tanks is derived [44] to demonstrate a system wide scalability of

the proposed scheme. The rate of change for water level in a tank is equal to the difference between water flowing in and flowing out over time. Inflow and outflow rates are controlled by actuator actions. We can represent this flow of fluid using mass-balance equation such as,

$$\frac{dV}{dt} = Q_{in} - Q_{out}$$
$$\frac{dh}{dt} = \frac{Q_{in} - Q_{out}}{A} \text{ since } V = A \times h, \quad (8)$$

where V represents the volume of the tank, A is the cross-sectional area of the tank, and h is the height of the water inside the tank, (8) provides a linear equation, the term $[Q_{in} - Q_{out}]$ is the water flow which depends upon the PLC control actions. Let us consider water level in the tank as state of the system. Discretization leaves us with the following system of state space difference equations,

$$\begin{cases} x_{k+1} = x_k + u_k + v_k, \\ y_k = x_k + \eta_k, \end{cases} \quad (9)$$

where u_k is the PLC control action and y_k is the sensor measurement driven by noise η_k. Since we have the system model now, we can use Kalman filter to estimate the state of the system. We designed the estimator so that it predicts the states of the two-tanks simultaneously. The Kalman filter can be expressed as follows,

$$\hat{x}_{k+1} = F\hat{x}_k + Gu_k + L(\bar{y}_k - \hat{x}_k), \quad (10)$$

where $\hat{x}_k$ is the estimate of the system state and $\bar{y}_k$ is the last (possibly attacked) sensor measurement. L is the Kalman gain matrix which is a weighting factor for the computation of the Kalman estimate. The value of the Kalman gain could be between 0 and 1 [10] which makes the estimation to either give more weight to current measurement of the sensor or to the previous estimate of the sate. Since we have a combined system model, a combined estimator is derived which is a matrix of a 2×2 such that,

$$L = \begin{pmatrix} L_1 & L_2 \\ L_3 & L_4 \end{pmatrix}, \quad F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

For two-tank experiment we have vectored values now, so

$$\hat{x}_k = \begin{pmatrix} \hat{x}_k^1 \\ \hat{x}_k^2 \end{pmatrix}, \bar{y}_k = \begin{pmatrix} \bar{y}_k^1 \\ \bar{y}_k^2 \end{pmatrix}.$$

where $\hat{x}_k^1, \hat{x}_k^2, \bar{y}_k^1, \bar{y}_k^2$ are the previous estimate and last measurements of Tank-1 and Tank-2 respectively. By putting all these values in (10), we get combined Kalman estimation equations for Tank-1 and Tank-2.

$$\begin{cases} \hat{x}_{k+1}^1 = \hat{x}_k^1 + u_k^1 + L_1(\bar{y}_k^1 - \hat{x}_k^1) + L_2(\bar{y}_k^2 - \hat{x}_k^2) \\ \hat{x}_{k+1}^2 = \hat{x}_k^2 + u_k^2 + L_3(\bar{y}_k^1 - \hat{x}_k^1) + L_4(\bar{y}_k^2 - \hat{x}_k^2) \end{cases} \quad (11)$$

In (11), $\hat{x}_{k+1}^1$ represents the state estimation of Tank-1 and $\hat{x}_{k+1}^2$ represents the state estimation of Tank-2 in a combined manner. The gain values were computed as of $L_1 = 0.35$, $L_2 = 0.15$, $L_3 = -0.15$, $L_4 = 0.65$. This system model is implemented in real-time at SWaT testbed [31], attack executed and results obtained are discussed in the following sections.

4.2 WADI System Model

Figure 8 in appendix E shows a system level abstraction of the water distribution testbed [6]. It has three major stages: Primary Grid, Secondary Grid and Return Water. Each stage consists of set of sensors and actuators. We consider sensor measurements as outputs and actuation control actions as inputs. There are multitude of sensors, actuating devices and six consumers nodes in WADI, which makes it a complex system to obtain a system model from first principles. To derive a system model, the plant is run for 14 days and data is collected for inputs and outputs. Using sub-space system identification [40] techniques, a model of the following form is obtained.

$$x_{k+1} = Ax_k + Bu_k + v_k \quad (12)$$
$$y_k = Cx_k + \eta_k \quad (13)$$

where $k \in \mathbb{N}$ is the discrete time index, $x_k \in \mathbb{R}^n$ is the state of the approximated model, (its dimension depends on the order of the approximated model), $y \in \mathbb{R}^m$ are the measured outputs, and $u \in \mathbb{R}^p$ denotes the actuator action which depends on the demand patterns. The system identification problem is to determine the system matrices A, B, C from input-output data. The obtained model provides a good fit (as shown in next section) between measurements and modeled outputs (generated using the identified system model) with 10 states, i.e., $n = 10$. We also identified a few higher and lower order models. Ultimately, the model with 10 states has a nice trade-off between prediction error and the dimensions of the model.

4.3 System Model Validation

The identified model is validated by looking at the system state evolution based on the identified state space matrices and initial state x_1. The closeness of the system evolution to the sensor measurements obtained from real testbed indicates that this model is a faithful representation of the water distribution network, as shown in Figure 5. The top pane shows the sensor readings from real-testbed as well as the modeled output for the electromagnetic flow meter using system matrices. We can observe that modeled output is very close to sensor readings, resulting in small error. (Error is shown in the middle pane, while error's probability distribution is shown in the bottom pane.) In Figure 11 shown in appendix D, we can see that real sensor measurement and sensor estimate for Tank-1 in SWaT is the same, thus validating the model and ensuring that it is representative of the real testbed. The middle pane shows the difference between real sensor measurement and the sensor estimate. The bottom pane shows the plot of PDF for the residual vector, and for the level sensor in the SWaT testbed.

Besides visual representation of the model, we also analyzed the statistical metric for the obtained model. Variance Account For (VAF) values [54] are used on a data set from the real testbeds. VAF is defined as,

$$VAF = max\{1 - (var(y_k - \hat{y_k}))/var(y_k), 0\} * 100, \quad (14)$$

where $\hat{y_k}$ denotes the estimated output signal, y_k sensor measurements, and var the variance of a signal. The VAF values are shown in appendix D Table 6 for SWaT testbed and in Table 2 for WADI testbed. We can see that for SWaT both level sensor's VAF values are 100%, because $(y - \hat{y})$ value is very small, i.e. for level sensor on

Table 2: Validating WADI system model obtained from subspace system identification.

Sensor (Output Channel)	VAF value
RADAR Level Sensor (Primary Grid)	99.82%
RADAR Level Sensor (Secondary Grid)	99.94%
RADAR Level Sensor (Secondary Grid)	99.92%
Differential Pressure Transmitter (Secondary Grid)	96.86%
Differential Pressure Transmitter (Secondary Grid)	92.56%
Electromagnetic Flowmeter (Primary Grid)	99.74%
Electromagnetic Flowmeter (Secondary Grid)	99.54%
Electromagnetic Flowmeter (Secondary Grid)	98.70%
Electromagnetic Flowmeter (Secondary Grid)	97.10%

Tank-1 it is $1.87 * 10^{-7}$ and for level sensor on Tank-2 it is $1.85 * 10^{-7}$. For WADI VAF value for each output channel is as high as 99%, with a lowest of 92%. In literature a system model with a VAF value of 80% is considered a good fit model [54].

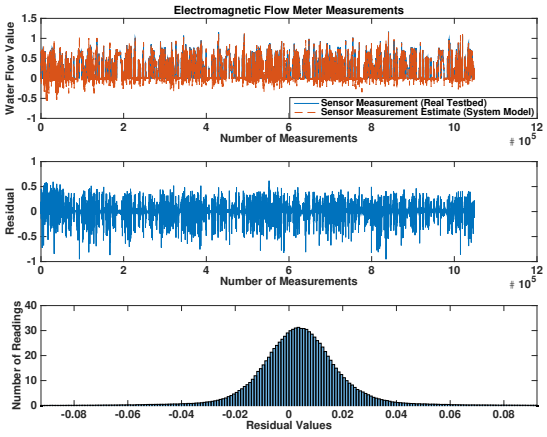

Figure 5: WADI System Model Validation.

5 PERFORMANCE EVALUATION

In this section a brief background on statistical detectors is given, followed by attack design against such detectors and evaluation of the proposed scheme.

5.1 Statistical Detectors: A Primer

Residuals and Hypothesis Testing: For the case of statistical detectors, estimated state values are compared with sensor measurements $\bar{y}_k$ (which may have been attacked). The difference between the two should stay within a certain threshold under normal operation, otherwise an alarm is triggered to point a potential attack. Define the residual random sequence $r_k, k \in \mathbb{N}$ as,

$$r_k := \bar{y}_k - C\hat{x}_k = Ce_k + \eta_k + \delta_k. \tag{15}$$

If there are no attacks, the mean of the residual is,

$$E[r_{k+1}] = CE[e_{k+1}] + E[\eta_{k+1}] = \mathbf{0}_{m\times 1}. \tag{16}$$

where $\mathbf{0}_{m\times 1}$ denotes an $m \times 1$ matrix composed of only zeros, and the covariance is given by,

$$\Sigma := E[r_{k+1}r_{k+1}^T] = CPC^T + R_2. \tag{17}$$

For this residual, we identify two hypotheses to be tested, $\mathcal{H}_0$ the *normal mode* (no attacks) and $\mathcal{H}_1$ the *faulty mode* (with attacks). We can formulate the hypothesis testing in a more formal manner using existing change detection techniques (as explained in the following) based on the statistics of the residuals.

Cumulative Sum (CUSUM) Detector: The CUSUM procedure is driven by the residual sequences. In particular, the input to the CUSUM procedure is a *distance measure*, i.e., a measure of how deviated the estimator is from the actual system, and this measure is a function of the residuals. We propose the absolute value of the entries of the residual sequence as distance measure, i.e.,

$$z_{k,i} := |r_{k,i}| = |C_i e_k + \eta_{k,i} + \delta_{k,i}|. \tag{18}$$

For a given *distance measure* $z_{k,i} \in \mathbb{R}$, the CUSUM of Page [41] is written as follows.

CUSUM: $S_{0,i} = 0, \quad i \in \mathcal{I}$,

$$\begin{cases} S_{k,i} = \max(0, S_{k-1,i} + z_{k,i} - b_i), & \text{if } S_{k-1,i} \le \tau_i, \\ S_{k,i} = 0 \text{ and } \tilde{k}_i = k-1, & \text{if } S_{k-1,i} > \tau_i. \end{cases} \tag{19}$$

Design parameters: bias $b_i > 0$ and threshold $\tau_i > 0$.
Output: alarm time(s) $\tilde{k}_i$.

Bad-Data Detector: For the residual sequence $r_{k,i}$ given by (15), the Bad-Data detector is defined as follows.

Bad-Data Procedure:

$$\text{If } |r_{k,i}| > \alpha_i, \quad \tilde{k}_i = k, \quad i \in \mathcal{I}. \tag{20}$$

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 $|r_{k,i}|$ exceeds the threshold α_i. In Appendix B, more details on these statistical detectors are given for an interested reader.

5.2 Zero-Alarm Attack Design

We executed the two types of *zero-alarm* attacks on SWaT testbed against the introduced statistical detectors.

Zero-Alarm Attack for Bad-Data Detector: This attack is designed to stay undetected by the Bad-Data detectors. The attacker knows the system dynamics, has access to sensor readings, and knows the detector parameters, it is able to inject false data into real-time measurements and stay undetected. Consider the Bad-Data procedure and write (20) in terms of the estimated state $\hat{x}_k$,

$$|r_{k,i}| = |y_{k,i} - C_i\hat{x}_k + \delta_{k,i}| \le \alpha_i, \quad i \in \mathcal{I}. \tag{21}$$

By assumption, the attacker has access to $y_{k,i} = C_i x_k + \eta_{k,i}$. Moreover, given its perfect knowledge of the observer, the opponent can

compute the estimated output $C_i\hat{x}_k$ and then construct $y_{k,i} - C_i\hat{x}_k$. It follows that,

$$\delta_{k,i} = C_i\hat{x}_k - y_{k,i} + \alpha_i - \epsilon_i, (\alpha_i > \epsilon_i) \rightarrow |r_{k,i}| = \alpha_i - \epsilon_i, \quad i \in \mathcal{I}, \tag{22}$$

is a feasible attack sequence given the capabilities of the attacker. The constant $\epsilon_i > 0$ is a small positive constant introduced to account for numerical precision. These attacks maximize the damage to the CPS by immediately saturating and maintaining $|r_{k,i}|$ at the constant $\alpha_i - \epsilon_i$. Therefore, for this attack, the sensor measurements received by the controller take the form,

$$\bar{y}_{k,i} = C_i\hat{x}_k + \alpha_i - \epsilon_i. \tag{23}$$

Zero-Alarm Attack for CUSUM Detector: This attack is designed to stay undetected by the CUSUM detectors. Consider the CUSUM procedure and write (19) in terms of the estimated state $\hat{x}_k$,

$$S_{k,i} = \max(0, S_{k-1,i} + |y_{k,i} - C_i\hat{x}_k + \delta_{k,i}| - b_i), \tag{24}$$

if $S_{k-1,i} \leq \tau_i$ and $S_{k,i} = 0$ if $S_{k-1,i} > \tau_i$. As with the Bad-Data procedure, we look for attack sequences that immediately saturate and then maintain the CUSUM statistic at $S_{k,i} = \tau_i - \epsilon_i$ where ϵ_i ($\min(\tau_i, b_i) > \epsilon_i > 0$) is a small positive constant introduced to account for numerical precision. Assume that the attack starts at some $k = k^* \geq 1$ and $S_{k^*-1,i} \leq \tau_i$, i.e., the attack does not start immediately after a false alarm. Consider the attack,

$$\delta_{k,i} = \begin{cases} \tau_i - \epsilon_i + b_i - y_{k,i} + C_i\hat{x}_k - S_{k-1,i}, & k = k^*, \\ b_i - y_{k,i} + C_i\hat{x}_k, & k > k^*. \end{cases} \tag{25}$$

This attack accomplishes $S_{k,i} = \tau_i - \epsilon_i$ for all $k \geq k^*$ (thus zero alarms). Note that the attacker can only induce this sequence by exactly knowing $S_{k^*-1,i}$, i.e., the value of the CUSUM sequence one step before the attack. This is a strong assumption since it represents a real-time quantity that is not communicated over the communication network. Even if the opponent has access to the parameters of the CUSUM, (b_i, τ_i), given the stochastic nature of the residuals, the attacker would need to know the complete history of observations (from when the CUSUM was started) to be able to reconstruct $S_{k^*-1,i}$ from data. This is an inherent security advantage in favor of the CUSUM over static detectors like the Bad-Data. Nevertheless, for evaluating the worst case scenario, we assume that the attacker has access to $S_{k^*-1,i}$. Therefore, for this attack, the sensor measurements received by the controller take the form,

$$\bar{y}_{k,i} = \begin{cases} C_i\hat{x}_k + \tau_i - \epsilon_i + b_i - S_{k-1,i} - \epsilon_i, & k = k^*, \\ C_i\hat{x}_k + b_i, & k > k^*. \end{cases} \tag{26}$$

5.3 Performance Metrics

In our experiments, each sensor is assigned a unique ID and a two-class classification is applied to identify each sensor. To evaluate the performance, we use identification accuracy as a performance metric. Let c be the total number of classes. We define TP_i as true positive for class c_i when it is rightly classified based on the ground truth. False negative FN_i is defined as the wrongly rejected, and False positive FP_i as wrongly accepted. True negative TN_i is the rightly rejected class. The overall accuracy (acc) for total of c classes is defined as,

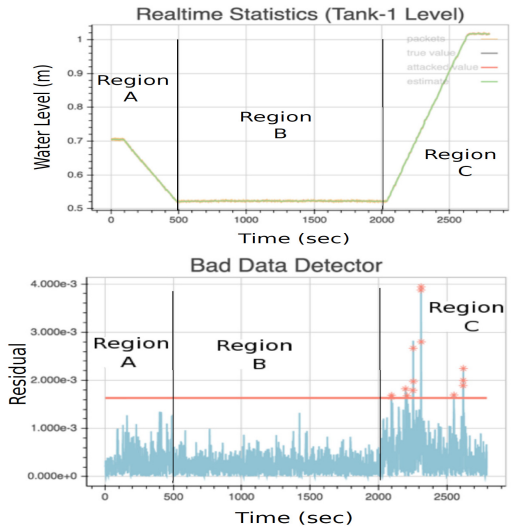

Figure 6: Residual vector for the Tank-1 during the normal operation of the SWaT plant.

$$acc = \frac{\sum_{i=1}^{c} TP_i + \sum_{i=1}^{c} TN_i}{\sum_{i=1}^{c} TP_i + \sum_{i=1}^{c} TN_i + \sum_{i=1}^{c} FP_i + \sum_{i=1}^{c} FN_i}. \tag{27}$$

5.4 Attack Detection Performance

Attack detection performance of the proposed scheme is presented and compared with the statistical detectors from the literature.

Residual Vector for Normal Operation: Figure 6 shows plot for residual vector for the case of normal operation in SWaT testbed. Residue vector is shown for three different states of the system, i.e. region A is the case for water emptying process in Tank-1, region B is the case for static process and region C for water filling process. The randomness in the residue vector is a function of sensor and process noise as given by proposition 1. The intuition for the proposed scheme is based on this noise pattern in the residue vector. Sensor noise part is due to physical structure of the sensor [19] and process noise is property of the process e.g. water sloshing in the tank [4]. The horizontal line is the threshold for the Bad-Data detector.

Threshold Validation for the Statistical Detectors: As explained in previous section that the threshold has to be selected such that it could satisfy the alarm rate, and it should not have too much margin so that the false alarm rate is too high or is too low. Here we tested our combined detector of both Bad-Data and CUSUM on both tanks dynamically and the result is shown in Figure 9. The plot shows that the alarm rate for tanks is between 0.02 to 0.04. Both detectors running at Tank-1 converged to a 0.025 false alarm rate when running for more than 1500 time stamps. Similarly detectors running at Tank-2 converged to 0.045 false alarm rate. To achieve this false alarm rate we used the threshold settings as shown in Table 3.

Residual Vector for Zero-Alarm Attack: Figure 7 shows a plot for the residual vector when system is under *zero-alarm* attack.

Table 3: Threshold Validation for Statistical Detectors.

Parameter	Tank-1	Tank-2
α	0.00046072	0.00045890
τ	0.00015972	0.00014750
bias b	0.0003269	0.0003256

The left most plot shows real-time data for level sensor in Tank-1, while two plots on the right show residual vectors for Bad-Data and CUSUM detectors. From the design of *zero-alarm* attacks in previous section, it was expected that the attacker would spoof the sensor data to stay stealthy for the statistical detectors. In an attempt to be stealthy but still be able to damage the plant [5, 17, 37], an attacker would inevitably modify the noise pattern of the residual vector. A visual comparison of normal operation in Figure 6 and system under attack in Figure 7, reveals the deviation from the normal noise pattern when system is under attack.

Attack Detection: Table 4 shows the results for the performance of the attack detectors. A comparison between statistical detectors and *NoisePrint* reveals that the proposed scheme is able to detect sensor spoofing attacks using the same residual vector as used by Bad-Data and CUSUM. Hence, *NoisePrint* removes the limitations of these detectors and could detect the *zero-alarm* attacks.

- **Constant Bias Attack:** Figure 10 in appendix F shows the water level at the Tank-1 when the system is under a constant bias attack of $\delta_1 = 0.01$m. The PLC received this attacked measurement value. The true value (plotted in gray) of the level at Tank-1 is about 0.5m. This true level remains constant throughout the attack and the inlet pump and valve are switched OFF. The attack is launched at $k = 11$s (time instant in plot) and the Bad-Data detector monitoring Tank-1 detects it immediately. Furthermore this attack was also detected by the CUSUM detector running at Tank-1.
  NoisePrint also detects this attack using the SVM model trained using residual from the normal operation of the plant. The deviation in the residual vector from the normal operation is pictorially seen in Figure 10.
- **Zero-Alarm Attack for Bad-Data and CUSUM Detector:** We launched *zero-alarm* attack for Bad-Data and CUSUM detectors for level sensor installed in two tanks at SWaT testbed. Since this attack is designed to raise no alarms for the Bad-Data or the CUSUM detectors, neither detector on tanks detect the attack. The attacker has the complete knowledge of the detectors, so he can deviate the level of the tank in such a way that Bad-Data and CUSUM detectors would not be able to detect it. Figure 7 shows the Tank-1 level sensor under such an attack. It can be seen that attacker spoofs sensor data in a way that residual vectors stay under the detection threshold.
  NoisePrint is able to detect *zero-alarm* attacks as noise pattern is changed from the fingerprint created under the normal operation.

Table 4: Attack detection performance and comparison between detectors.

Attack Type / Detector	Bad-Data Detector	CUSUM Detector	***NoisePrint***
Zero-Alarm Attack	Not Detected	Not Detected	Detected (100% Accuracy)
Constant Bias Attack	Detected	Detected	Detected (100% Accuracy)

5.5 Sensor Identification Accuracy

In Table 5, sensor identification accuracies are given for nine different sensors in the water distribution testbed. We can see that the lowest identification accuracy is 90% and the highest is 96.41%. The sensors can be identified with a very high accuracy even though few processes are of similar type e.g. flow of water, level of water or pressure at the junctions. Two-class SVM is used for sensor identification. One class is labeled as *legitimate* for the case of right sensor and data from all other sensors, while attackers are labeled as *illegitimate*. Since the residual vector (source of fingerprint) is a function of sensor and process noise, if an adversary physically manipulates the sensor or execute analog sensor spoofing [47], it will modify the sensor noise pattern. In case an adversary swaps level sensors on two different tanks (processes) [4], the process noise would deviate from the reference fingerprint. The proposed method is able to detect such physical/analog domain manipulations. These results highlight the significance of *NoisePrint*.

6 DISCUSSION

Security Argument: Attacks on sensor measurements can be detected using *NoisePrint* for the case of an attacker with the knowledge of either the system model including estimator gain or the noise profile. However, for the case of a strong adversary (possessing knowledge of system model and noise distribution for a sensor) the proposed scheme would fail only when an attacker strictly follows the system model and imitates the noise profile. To stay stealthy against *NoisePrint* an attacker should stay within the bounds of noise distribution of a residual vector and can not deviate from the system model, which means it can not inject arbitrary values. An attacker injecting values from the noise distribution of residual vector would not be able to achieve its objectives as stated in the attacker model. *NoisePrint* raises the bar for such an advanced attacker. For a more advanced attacker, we can complement *NoisePrint* with a *challenge-response protocol*, to detect replay attacks. A challenge is generated from the physical quantity to be measured and the challenger-sensor pair is fingerprinted, which would help us detect replay attacks.

Scalability: In this article we considered a multitude of sensors from two CPS testbeds. For evaluation of the proposed scheme we have used two-class classification (LibSVM) by considering the legitimate sensor (class 1) and rest of the sensors as illegitimate or compromised (class 2). We also considered a variety of processes. Multitude of devices, processes and the classification algorithm indicates that *NoisePrint* is scalable. We studied the feasibility of the proposed scheme on two different testbeds which also points out the generality and scalability of the *NoisePrint*.

Attack Detection Speed: In this article we executed *zero-alarm*

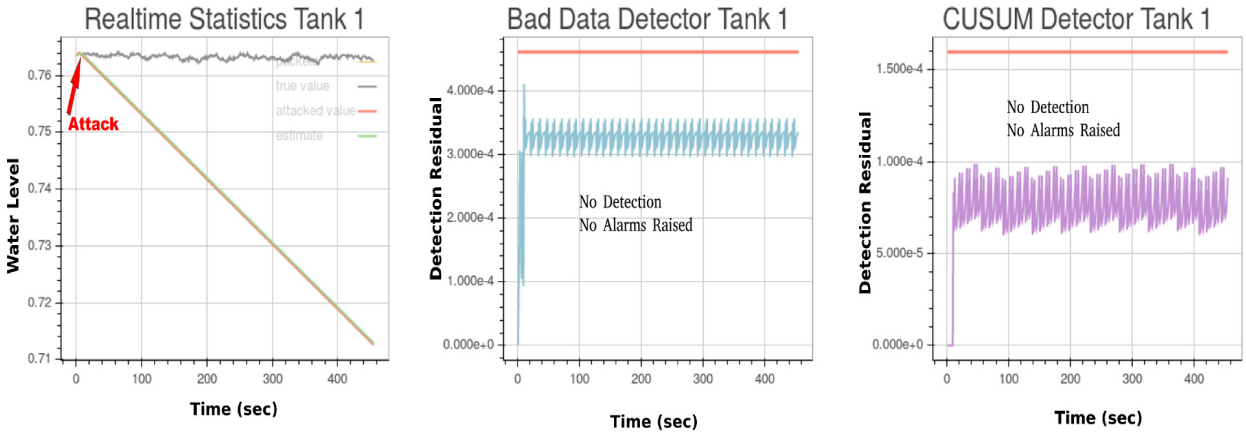

Figure 7: Zero-Alarm Attack for Bad-Data and CUSUM detectors on Tank-1 in SWaT. Horizontal line in right hand plots is the threshold for the particular detector.

Table 5: WADI Sensor Identification Accuracy Result

Sensor	Type and Model	Identification Accuracy
RADAR Level Sensor (Primary Grid)	iSOLV RD700	90.87%
RADAR Level Sensor (Secondary Grid)	iSOLV EFS803/CFT183	96.41%
RADAR Level Sensor (Secondary Grid)	iSOLV EFS803/CFT183	91.52%
Differential Pressure Transmitter (Secondary Grid)	iSOLV SPT 200	92.02%
Differential Pressure Transmitter (Secondary Grid)	iSOLV SPT 200	92.95%
Electromagnetic Flowmeter (Primary Grid)	iSOLV EFS803/CFT183	92.76%
Electromagnetic Flowmeter (Secondary Grid)	iSOLV EFS803/CFT183	90.76%
Electromagnetic Flowmeter (Secondary Grid)	iSOLV EFS803/CFT183	90.0%
Electromagnetic Flowmeter (Secondary Grid)	iSOLV EFS803/CFT183	92.04%

attacks on two stages of the SWaT testbed and compared the performance of *NoisePrint* and legacy statistical methods based on certain thresholds. The proposed scheme can detect these attacks while legacy methods fail. However, there is a trade-off for such a good performance, in terms of detection time. For threshold based schemes an attack detection decision is made at each time instant, by comparing the residual value to a threshold, while for *NoisePrint* we need 120 samples to extract features and then make a detection decision. There is a delay of 120 samples to raise an alarm if any attack is being executed. However, we propose an idea where we only wait for initial 120 readings and then at each time instant use previous readings in a moving window manner plus a set of fresh readings to extract a feature vector. This way, we do not have to wait for 120 readings and an attack can be detected in less time. We have not tested this proposal yet, which is part of our future work.

Application in Real-World CPS: We have tested the proposed method for a data set collected over a period of two weeks from a water distribution testbed. The results are promising for such a time period. However, it is recommended to train the classifiers after every plant maintenance cycle. Moreover, being used in a testbed for few weeks is different from being used in a real-world production system of physical plants with possibly more harsh environment especially for the case of level measurements including rivers, dams etc. Although the testbeds used in the reported experiments imitate real water treatment plants as close as possible but we believe the sensors and actuators wear out with time, rendering them less accurate. There is a possibility that those environmental effects may change the fingerprint but according to our hypothesis each sensor will be affected in a distinct way and, if retrained, will possess a unique fingerprint. As far as the ambient noise or interference is concerned that would affect all the devices in a similar manner, letting us to cancel out those effects from sensor fingerprint.

Implementation and Practical Considerations:

Sensor Replacement: Replacement of a sensor requires the generation of a new fingerprint for the new sensor. Currently we have a system-wide model for a testbed which is an advantage of the proposed method in that it is scalable for a complete realistic plant. Hence, if we are retraining the model, we need to do so for the entire system (plant). If only one sensor is replaced then we need to collect fresh data for that sensor and update the system model.

Training: For training we need at least one complete cycle of a process. For example, if we are modeling a water storage tank, then the dynamics of emptying and filling a tank should be captured. **Results:** There are three main results in the paper: a). Constant Bias

Attack Detection, b). Zero-Alarm Attack Detection and c). Device Identification. We trained the SVM for two-class classification on a labeled data set during normal operation using a legitimate sensor (class 1) and all other sensors and scenarios as (class 2). For testing phase, we run the plant while under attack (attacks are launched multiple times and residual vector is collected). SVM is able to detect the change in noise profile when sensors are under attack with an 100% accuracy. **Performance Comparison:** Attacks studied in this article emulate the system states. For example a *zero-alarm* attack when executed tries to imitate the emptying or a filling process but by adding a small δ value to the sensor measurement which can not be detected by legacy statistical detectors considered in this work. While the real system state would be something different but readings sent to SCADA system would imitate the physical process (emptying or filling) during the attack execution. CUSUM and Bad-Data detectors fail as the attacker knows their parameters (e.g. threshold) but *NoisePrint* is successful because an arbitrary spoofing of sensor reading leads to deviation from the normal noise fingerprint. *NoisePrint* is comparable to these other methods because input to all these detectors is the same, a residual vector and all of these depend on an accurate system model.

7 RELATED WORK

Device Fingerprinting: The approach presented in this article is inspired by the idea of using sensor noise as a fingerprint for camera identification [30]. In [30], images are taken by a camera and filtered to obtain noise components and averaged for all images. This resultant noise vector acts as a reference pattern for test images. An image is tested against reference patterns for all cameras being studied and matched with one having the highest correlation with image's noise vector. The idea of fingerprinting a device remotely based on its hardware is presented in [27]. Small microscopic deviations in device's clock [35, 42] are used as fingerprint for the particular device. In [45] inter arrival time of packets is analyzed to fingerprint devices on a small campus network. In [18], 50 RFID smart cards from the same manufacturer and type are tested for fingerprints. Performance analysis on Received Signal Strength (RSS) based fingerprinting of wireless access point is presented in [43].

CPS Device Fingerprinting: In [21] authors focus on the idea of device fingerprinting in ICS. One approach in [21] is based on traditional network traffic monitoring and observing message response time, while the second approach is based on physical operation time of a device. Analysis is carried out on 2 latching relays based on their operation timings. This approach can not be applied to devices studied in our work because there is no mechanical motion of the components as was the case for electric relays in [21]. A preliminary study, on the idea of sensor fingerprinting is presented as a short paper in [4], using 2 sensors, based on correlation analysis with an accuracy of 86%. Besides lower accuracy, another limitation of [4] is that it requires a complete process cycle (ten's of minutes) to make a detection decision, which is slow considering the critical nature of real time CPS. Another related work in CPS presented a study on the idea of sensor fingerprinting in [36]. However, to the best of our knowledge, this paper presents the first attempt for a rigorous analysis on fingerprinting the combined process and sensor noise. The new approach is also able to detect attacks where an adversary swaps sensors among processes [4]. Sensor swap attack would not be detected by using only sensor noise. Another limitation of [36] is that, to extract sensor noise for certain sensors (e.g. level sensors), one needs to wait for process to be static. However, process is not static most of the time and thus introducing another source of noise i.e. process noise. The proposed scheme takes residual vector as an input which is also an input for the statistical detectors (e.g. CUSUM and Bad-Data). *NoisePrint* removes the limitations of these statistical detectors against a class of well studied *zero-alarm* attacks [5, 17, 37]. To the best of our knowledge, no prior work has applied sensor and process noise fingerprinting scheme to the detection of sensor data integrity attacks on ICS.

8 CONCLUSIONS AND FUTURE WORK

Summary: An idea for fingerprinting the sensor and process noise for the purpose of device identification and attack detection is proposed. We need a representative model for the system under consideration. Towards that end we had access to two real world water treatment (SWaT) and distribution (WADI) testbeds. We used first principles and obtained the system model for a part of SWaT testbed, based on the physics of the system. For WADI testbed, we obtained a system model by using a well known technique of sub-space system identification. Once we have system model for our system, we can design a Kalman filter for the purpose of state estimation. By subtracting the state estimates from the real system estimates, a residual vector is obtained in steady state that residual vector is a function of process and sensor noise.
Conclusions: A novel method to fingerprint these sensor and process noise is presented. Our results have shown that *zero-alarm* attacks cannot be detected by reference statistical methods but can be detected by the proposed scheme. Moreover, we have shown that sensors can be uniquely identified with accuracy higher than 90%.
Future Work: In future, we plan to isolate the sensor noise from the process noise to identify the individual sensors in the plant. Another interesting problem is to increase the accuracy of device identification. Towards that end we are working on generating and using multiple system models by deploying a bank of observers for each sensor and isolate the sensor under attack.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cyber Security R&D Programme (Award No. NRF2014NCR-NCR001-40) and administered by the National Cybersecurity R&D Directorate. The 3^{rd} author's research was supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

REFERENCES

[1] B.M. Adams, W.H. Woodall, and C.A. Lowry. 1992. The use (and misuse) of false alarm probabilities in control chart design. *Frontiers in Statistical Quality Control 4* (1992), 155–168.
[2] Sridhar Adepu and Aditya Mathur. 2016. Distributed Detection of Single-Stage Multipoint Cyber Attacks in a Water Treatment Plant. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. ACM, New York, NY, USA, 449–460. https://doi.org/10.1145/2897845.2897855

[3] C. M. Ahmed, A.Sridhar, and M. Aditya. 2016. Limitations of state estimation based cyber attack detection schemes in industrial control systems. In *IEEE Smart City Security and Privacy Workshop, CPSWeek*.

[4] C. M. Ahmed and A. P. Mathur. 2017. Hardware Identification via Sensor Fingerprinting in a Cyber Physical System. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 517–524. https://doi.org/10.1109/QRS-C.2017.89

[5] Chuadhry Mujeeb Ahmed, Carlos Murguia, and Justin Ruths. 2017. 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 '17)*. ACM, New York, NY, USA, 101–113. https://doi.org/10.1145/3052973.3053011

[6] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWATER '17)*. ACM, New York, NY, USA, 25–28. https://doi.org/10.1145/3055366.3055375

[7] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. 2014. Good Practice in Large-Scale Learning for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (March 2014), 507–520. https://doi.org/10.1109/TPAMI.2013.146

[8] Rajeev Alur. 2015. *Principles of cyber-physical systems.* MIT Press.

[9] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. 2013. Cyber Security of Water SCADA Systems x2014;Part I: Analysis and Experimentation of Stealthy Deception Attacks. *IEEE Transactions on Control Systems Technology* 21, 5 (Sept 2013), 1963–1970. https://doi.org/10.1109/TCST.2012.2211873

[10] Karl J. Aström and Björn Wittenmark. 1997. *Computer-controlled Systems (3rd Ed.).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[11] A. Cardenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry. 2011. Attacks against process control systems: Risk assessment, detection, and response. In *6th ACM Symposium on Information, Computer and Communications Security*. 355–366.

[12] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and Shankar Sastry. 2009. Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*. 5.

[13] Defense Use Case. 2016. Analysis of the Cyber Attack on the Ukrainian Power Grid. (2016).

[14] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2 (2011), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[15] Eyasu Getahun Chekole, John Henry Castellanos, Martín Ochoa, and David K. Y. Yau. 2018. Enforcing Memory Safety in Cyber-Physical Systems. In *Computer Security*, Sokratis K. Katsikas, Frédéric Cuppens, Nora Cuppens, Costas Lambrinoudakis, Christos Kalloniatis, John Mylopoulos, Annie Antón, and Stefanos Gritzalis (Eds.). Springer International Publishing, Cham, 127–144.

[16] CNN. [n. d.]. Staged cyber attack reveals vulnerability in power grid. http://edition.cnn.com/2007/US/09/26/power.at.risk /index.html, year = 2007. ([n. d.]).

[17] Gyorgy Dan and Henrik Sandberg. 2010. Stealth attacks and protection schemes for state estimators in power systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 214–219.

[18] Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. 2009. Physical-layer Identification of RFID Devices. In *Proceedings of the 18th Conference on USENIX Security Symposium (SSYM'09)*. USENIX Association, Berkeley, CA, USA, 199–214. http://dl.acm.org/citation.cfm?id=1855768.1855781

[19] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. 2014. Accelprint: Imperfections of accelerometers make smartphones trackable. In *Network and Distributed System Security Symposium (NDSS)*.

[20] N. Falliere, L.O. Murchu, and E. Chien. 2011. W32 Stuxnet Dossier. Symantec, version 1.4. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. (Feb. 2011).

[21] David Formby, Preethi Srinivasan, Andrew Leonard, Jonathan Rogers, and Raheem Beyah. 2016. WhoâĂŹs in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. In *NDSS*.

[22] 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 *24th Annual Network & Distributed System Security Symposium (NDSS)*.

[23] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve F. Russell. 2006. Device Identification via Analog Signal Fingerprinting: A Matched Filter Approach. In *NDSS*.

[24] Dieter Gollmann and Marina Krotofil. 2016. *Cyber-Physical Systems Security*. Springer Berlin Heidelberg, Berlin, Heidelberg, 195–204. https://doi.org/10.1007/978-3-662-49301-4_14

[25] Naman Govil, Anand Agrawal, and Nils Ole Tippenhauer. 2017. On Ladder Logic Bombs in Industrial Control Systems. *CoRR* abs/1702.05241 (2017). http://arxiv.org/abs/1702.05241

[26] Y. Gu, T. Liu, D. Wang, X. Guan, and Z. Xu. 2013. Bad Data Detection Method for Smart Grids based on Distributed Estimation. In *IEEE ICC*.

[27] T. Kohno, A. Broido, and K. C. Claffy. 2005. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing* 2, 2 (April 2005), 93–108. https://doi.org/10.1109/TDSC.2005.26

[28] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu. 2013. Ghost Talk: Mitigating EMI Signal Injection Attacks against Analog Sensors. In *2013 IEEE Symposium on Security and Privacy*. 145–159. https://doi.org/10.1109/SP.2013.20

[29] E. A. Lee. 2008. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. 363–369. https://doi.org/10.1109/ISORC.2008.25

[30] J. Lukas, J. Fridrich, and M. Goljan. 2006. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security* 1, 2 (2006).

[31] A. P. Mathur and N. O. Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. 31–36. https://doi.org/10.1109/CySWater.2016.7469060

[32] L. Mili, T.V. Cutsen, and M.R.-Pavella. 1985. Bad Data Identification Methods in Power System State Estimation - A Comparative Study. *IEEE Trans. on Power Apparatus and Systems* (1985).

[33] Y. Mo and B. Sinopoli. 2009. Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 911–918. https://doi.org/10.1109/ALLERTON.2009.5394956

[34] Yilin Mo and Bruno Sinopoli. 2012. Integrity Attacks on Cyber-physical Systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems (HiCoNS '12)*. ACM, New York, NY, USA, 47–54. https://doi.org/10.1145/2185505.2185514

[35] S. B. Moon, P. Skelly, and D. Towsley. 1999. Estimation and removal of clock skew from network delay measurements. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 1. 227–234 vol.1. https://doi.org/10.1109/INFCOM.1999.749287

[36] C. Mujeeb Ahmed, A. Mathur, and M. Ochoa. 2017. NoiSense: Detecting Data Integrity Attacks on Sensor Measurements using Hardware based Fingerprints. *ArXiv e-prints* (Dec. 2017). arXiv:cs.CR/1712.01598

[37] C. Murguia and J. Ruths. 2016. Characterization of a CUSUM model-based sensor attack detector. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. 1303–1309. https://doi.org/10.1109/CDC.2016.7798446

[38] C. Murguia and J. Ruths. 2016. CUSUM and chi-squared attack detection of compromised sensors. In *2016 IEEE Conference on Control Applications (CCA)*. 474–480. https://doi.org/10.1109/CCA.2016.7587875

[39] NIST. 2014. Cyber-Physical Systems. https://www.nist.gov/el/cyber-physical-systems. (2014).

[40] P. Van Overschee and B. De Moor. 1996. Subspace Identification for Linear Systems: theory, implementation, applications. *Boston: Kluwer Academic Publications* (1996).

[41] E. Page. 1954. Continuous Inspection Schemes. *Biometrika* 41 (1954), 100–115.

[42] Vern Paxson. 1998. On Calibrating Measurements of Packet Transit Times. In *Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '98/PERFORMANCE '98)*. ACM, New York, NY, USA, 11–21. https://doi.org/10.1145/277851.277865

[43] J. Prakash and C. M. Ahmed. 2017. Can You See Me On Performance of Wireless Fingerprinting in a Cyber Physical System. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. 163–170. https://doi.org/10.1109/HASE.2017.40

[44] Qadeer R., Murguia C.and Ahmed C.M., and Ruths J. 2017. Multistage Downstream Attack Detection in a Cyber Physical System. In *CyberICPS Workshop 2017, in conjunction with ESORICS 2017*.

[45] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah. 2015. GTID: A Technique for Physical Device and Device Type Fingerprinting. *IEEE Transactions on Dependable and Secure Computing* 12, 5 (Sept 2015), 519–532. https://doi.org/10.1109/TDSC.2014.2369033

[46] M. Ross. 2006. *Introduction to Probability Models, Ninth Edition.* Academic Press, Inc., Orlando, FL, USA.

[47] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. 2015. PyCRA: Physical Challenge-Response Authentication For Active Sensors Under Spoofing Attacks. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 1004–1015. https://doi.org/10.1145/2810103.2813679

[48] J. Slay and M. Miller. 2008. Lessons Learned from the Maroochy Water Breach. *Springer 620 US, Boston, MA* (2008), 73–82.

[49] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors. In *Proceedings of the 24th USENIX Conference on Security Symposium (SEC'15)*. USENIX Association, Berkeley, CA, USA, 881–896. http://dl.acm.org/citation.cfm?id=2831143.2831199

[50] A. Sridhar and M. Aditya. 2016. Generalized Attacker and Attack Models for Cyber Physical Systems. In *40th IEEE COMPSAC*.
[51] S. Sridhar, A. Hahn, and M. Govindarasu. 2012. Cyber Physical System Security for the Electric Power Grid. *Proc. IEEE* 100, 1 (Jan 2012), 210–224. https://doi.org/10.1109/JPROC.2011.2165269
[52] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. 2016. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1092–1105.
[53] C.S. van Dobben de Bruyn. 1968. *Cumulative sum tests : theory and practice*. London : Griffin.
[54] Xiukun Wei, Michel Verhaegen, and Tim van Engelen. 2010. Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques. *International Journal of Adaptive Control and Signal Processing* 24, 8 (2010), 687–707. https://doi.org/10.1002/acs.1162
[55] Peter Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics* 15, 2 (1967), 70–73.
[56] Wired. 2015. A Cyberattack Has Caused Confirmed Physical Damage for the Second Time Ever. https://www.wired.com/2015/01/german-steel-mill-hack-destruction/. (2015).
[57] S. Yasser, M. Paul, T. Paulo, and S. Mani. 2013. Non-invasive Spoofing Attacks for Anti-lock Braking Systems. In *CHES, Springer Link*, Vol. 8086. 55–72.

A PROOF PROPOSITION 1.

For the process (1), the Kalman filter (3)-(6), we can represent estimation error as, $e_{k+1} = x_{k+1} - \hat{x}_{k+1}$, which in turn gives, $e_{k+1} = A(x_k - \hat{x}_k) - L\eta_k + v_k$.

$$e_{k+1} = (A - LC)e_k + v_k - L\eta_k \tag{28}$$

Considering estimation error as in (28), we start with estimator's initial state same as real system state, then $e_1 = 0$. For second iteration of estimation, from (28) we have $e_2 = (A-LC)e_1 + v_1 + L\eta_1$, for $e_1 = 0$, it gives $e_2 = v_1 + L\eta_1$. Similarly we get $e_3 = (A-LC)(v_1 - L\eta_1) + v_2 - L\eta_2$ and $e_4 = (A-LC)^2(v_1 - L\eta_1) + (A-LC)(v_2 - L\eta_2) + (v_3 - L\eta_3)$. By induction, we can generalize expression for k iterations of an estimator and estimation error can be represented as,

$$e_k = \sum_{i=0}^{k-2} (A - LC)^i (v_{k-i-1} - L\eta_{k-i-1}) \tag{29}$$

For residual we have $r_k = y_k - \hat{y}_k$, with $\hat{y}_k = C\hat{x}_k$, it becomes $r_k = C(x_k - \hat{x}_k) + \eta_k$,

$$r_k = Ce_k + \eta_k \tag{30}$$

By replacing e_k in (30) we get an expression for residual in steady state that is a function of process and sensor noise as given by following expression,

$$r_k = C \left\{ \sum_{i=0}^{k-2} (A - LC)^i (v_{k-i-1} - L\eta_{k-i-1}) \right\} + \eta_k \tag{31}$$

■

B STATISTICAL DETECTORS: A PRIMER

Residuals and Hypothesis Testing: In this work, we assess the performance of two model-based fault detection procedures (the Bad-Data and the CUSUM detectors) for a variety of attacks. These procedures rely on a state estimator (e.g., Kalman filter) to predict the evolution of the system. The estimated values are compared with sensor measurements $\bar{y}_k$ (which may have been attacked). The difference between the two should stay within a certain threshold under normal operation, otherwise an alarm is triggered to point a potential attack. Define the residual random sequence $r_k, k \in \mathbb{N}$ as

$$r_k := \bar{y}_k - C\hat{x}_k = Ce_k + \eta_k + \delta_k. \tag{32}$$

If there are no attacks, the mean of the residual is

$$E[r_{k+1}] = CE[e_{k+1}] + E[\eta_{k+1}] = \mathbf{0}_{m\times 1}. \tag{33}$$

where $\mathbf{0}_{m\times 1}$ denotes an $m \times 1$ matrix composed of only zeros, and the covariance is given by

$$\Sigma := E[r_{k+1} r_{k+1}^T] = CPC^T + R_2. \tag{34}$$

For this residual, we identify two hypotheses to be tested, $\mathcal{H}_0$ the *normal mode* (no attacks) and $\mathcal{H}_1$ the *faulty mode* (with attacks). For our particular case of study, the pressure at the nodes and the water level in the tank are the outputs of the system. Using this data along with the state estimates, we construct our residuals. Then, we have:

$$\mathcal{H}_0 : \begin{cases} E[r_k] = \mathbf{0}_{m\times 1}, \\ E[r_k r_k^T] = \Sigma, \end{cases} \text{ or } \quad \mathcal{H}_1 : \begin{cases} E[r_k] \neq \mathbf{0}_{m\times 1}, \\ E[r_k r_k^T] \neq \Sigma. \end{cases}$$

We can formulate the hypothesis testing in a more formal manner using existing change detection techniques (as explained in the following) based on the statistics of the residuals.

Cumulative Sum (CUSUM) Detector: The CUSUM procedure is driven by the residual sequences. In particular, the input to the CUSUM procedure is a *distance measure*, i.e., a measure of how deviated the estimator is from the actual system, and this measure is a function of the residuals. In this work, we assume there is a dedicated detector on each sensor (or on any sensor we want to include in the detection scheme). Throughout the rest of this paper we will reserve the index i to denote the sensor/detector, $i \in \mathcal{I} := \{1, 2, \ldots, m\}$. Thus, we can partition the attacked output vector as $\bar{y}_k = \text{col}(\bar{y}_{k,1}, \ldots, \bar{y}_{k,m})$ where $\bar{y}_{k,i} \in \mathbb{R}$ denotes the i-th entry of $\bar{y}_k \in \mathbb{R}^m$; then

$$\bar{y}_{k,i} = C_i x_k + \eta_{k,i} + \delta_{k,i}, \tag{35}$$

with C_i being the i-th row of C and $\eta_{k,i}$ and $\delta_{k,i}$ denoting the i-th entries of η_k and δ_k, respectively. Inspired by the empirical work in [11], we propose the absolute value of the entries of the residual sequence as distance measure, i.e.,

$$z_{k,i} := |r_{k,i}| = |C_i e_k + \eta_{k,i} + \delta_{k,i}|. \tag{36}$$

Note that, if there are no attacks, $r_{k,i} \sim \mathcal{N}(0, \sigma_i^2)$, where σ_i^2 denotes the i-th entry of the diagonal of the covariance matrix Σ. Hence, $\delta_k = \mathbf{0}$ implies that $|r_{k,i}|$ follows a *half-normal distribution* [46] with

$$E\big[\,|r_{k,i}|\,\big] = \frac{\sqrt{2}}{\sqrt{\pi}}\sigma_i \text{ and } \text{var}\big[\,|r_{k,i}|\,\big] = \sigma_i^2\Big(1 - \frac{2}{\pi}\Big). \tag{37}$$

Next, having presented the notion of distance measure, we introduce the CUSUM procedure. For a given *distance measure* $z_{k,i} \in \mathbb{R}$, the CUSUM of Page [41] is presented in (19).

From (19), it can be seen that $S_{k,i}$ accumulates the distance measure $z_{k,i}$ over time. The thresholds τ_i and bias b_i are selected based on a certain false alarm rate [1, 38, 53].

Bad-Data Detector: We have also implemented the Bad-Data detector for this case study because it is widely used in the CPS security literature [26, 32]. For the residual sequence $r_{k,i}$ given by

(15), the Bad-Data detector is defined in (20). Using the Bad-Data detector an alarm is triggered if distance measure $|r_{k,i}|$ exceeds the threshold α_i. Similar to the CUSUM procedure, the parameter α_i is selected to satisfy a required false alarm rate $\mathcal{A}_i^*$. An interested reader is referred to [37].

C SUPPORT VECTOR MACHINE CLASSIFIER

SVM is a data classification technique used in many areas such as speech recognition, image recognition and so on [7]. The aim of SVM is to produce a model based on the training data and give classification results for testing data. For a training set of instance-label pairs $(x_i, y_i), i = 1, ..., k$ where $x_i \in \mathbb{R}^n$ and $y \in \{1, -1\}^k$, SVM require the solution of the following optimization problem:

$$\begin{aligned} \underset{w,b,\zeta}{\text{minimize}} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{k} \zeta_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ \text{where } \zeta_i \geq 0. & \end{aligned} \tag{38}$$

The function ζ maps the training vectors into a higher dimensional space. In this higher dimensional space a linear separating hyperplane is found by SVM, where $C > 0$ is the penalty parameter of the error term. For the kernel function in this work we use the radial basis function:

$$K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0. \tag{39}$$

In our work, we have multiple sensors to classify. Therefore, multi-class SVM library LIBSVM [14] is used.

D WATER TREATMENT TESTBED

It is a fully operational (research facility) scaled down water treatment plant producing 5 gallons/minute of doubly filtered water, this testbed mimics large modern plants for water treatment. Following is the brief overview of the testbed, for further details, please refer to [31].

Water Treatment Process: The treatment process consists of six distinct stages each controlled by an independent Programmable Logic Controller (PLC). Control actions are taken by the PLCs using data from sensors. Stage P1 controls the inflow of water to be treated by opening or closing a motorized valve MV-101. Water from the raw water tank is pumped via a chemical dosing station (stage P2, chlorination) to another UF (Ultra Filtration) feed water tank in stage P3. A UF feed pump in P3 sends water via UF unit to RO (Reverse Osmosis) feed water tank in stage P4. Here an RO feed pump sends water through an ultraviolet dechlorination unit controlled by a PLC in stage P4. This step is necessary to remove any free chlorine from the water prior to passing it through the reverse osmosis unit in stage P5. Sodium bisulphate (NaHSO3) can be added in stage P4 to control the ORP (Oxidation Reduction Potential). In stage P5, the dechlorinated water is passed through a 2-stage RO filtration unit. The filtered water from the RO unit is stored in the permeate tank and the reject in the UF backwash tank. Stage P6 controls the cleaning of the membranes in the UF unit by turning on or off the UF backwash pump.

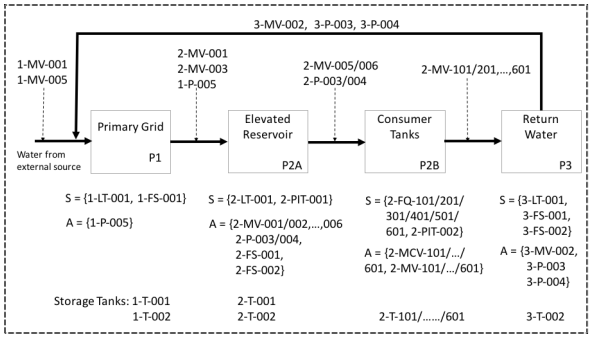

Figure 8: Overview of WADI testbed [6]. Solid arrows indicate flow of water and sequence of processes. S and A represent, respectively, sets of sensors and actuators.

Table 6: Validating SWaT system model obtained from first principles.

Sensor (Output Channel)	VAF value
Ultrasonic Level Sensor (Tank-1)	100.0%
Ultrasonic Level Sensor (Tank-2)	100.0%

E WATER DISTRIBUTION TESTBED

It is an operational testbed supplying 10 US gallons/min of filtered water. It represents a scaled-down version of a large water distribution network in a city. It contains three distinct control processes labeled P1 through P3, each controlled by its own set of PLCs as shown in Figure 8. An interested reader might look at [6] to understand the functionality of the testbed. Following is a brief overview of the WADI.

Stages in WADI: Water distribution process is segmented into the following sub-processes: P1: Primary grid, P2: Secondary grid, P3: Return water grid.

Primary grid: The primary grid contains two raw water tanks of 2500 liters each, and a level sensor (1-LIT-001) to monitor the water level in the tanks. Water intake into these two tanks can be from the water treatment plant, from Public Utility Board inlet, or from the return water grid. A chemical dosing system is installed to maintain adequate water quality. Sensors are installed to measure the water quality parameters of the water flowing into and out of the primary grid.

Secondary grid: This grid has two elevated reservoir tanks and six consumer tanks. Raw water tanks supply water to the elevated reservoir tanks and, in turn, these tanks supply water to the consumer tanks based on a pre-set demand pattern. Once consumer tanks meet their demands, water drains to the return water grid. Return water grid is equipped with a tank.

F SUPPORTING FIGURES

In the following, supporting figures auxiliary results are shown. Figure 9 shows threshold validation for CUSUM and Bad-Data detectors on SWaT testbed. Figure 10 shows detection using Bad-Data detector. Figure 11 shows system model validation of SWaT.

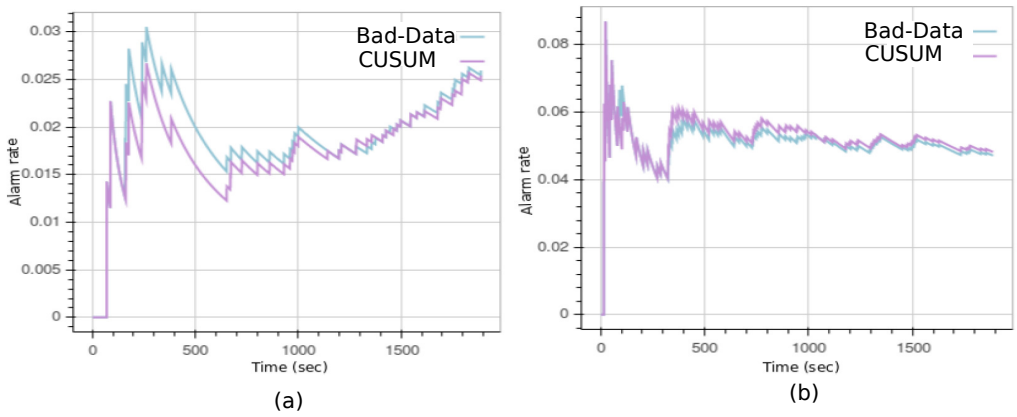

Figure 9: (a): False alarm rate of Bad-Data and CUSUM detector at Tank-1. (b): False alarm rate of Bad-Data and CUSUM detector at Tank-2.

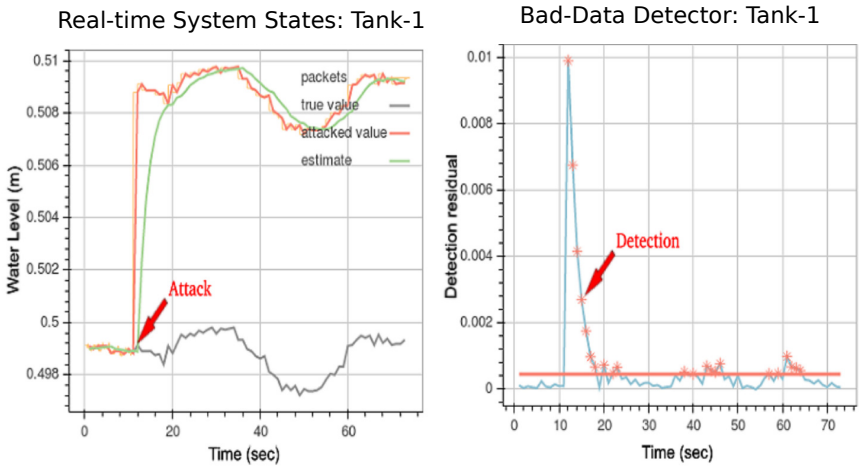

Figure 10: Constant bias attack detection by Bad-Data detector. It can be observed that as attack starts at 11s, it's detected.

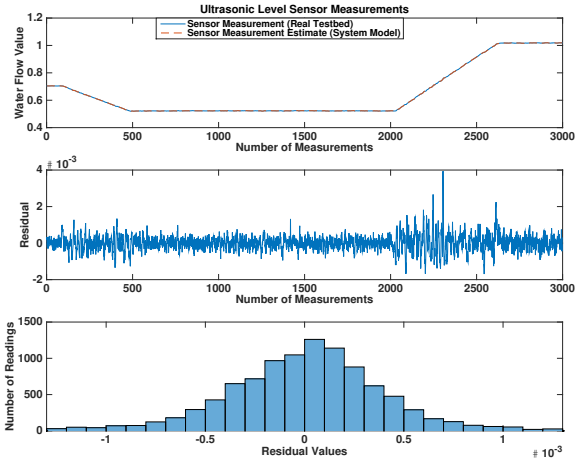

Figure 11: SWaT System Model Validation for Tank-1.