Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate sensors in CPS

Chuadhry Mujeeb Ahmed
Singapore University of Technology and Design
chuadhry@mymail.sutd.edu.sg

Jianying Zhou
Singapore University of Technology and Design
jianying_zhou@sutd.edu.sg

Aditya P. Mathur
Singapore University of Technology and Design
aditya_mathur@sutd.edu.sg

ABSTRACT

A novel scheme is proposed to authenticate sensors and detect data integrity attacks in a Cyber Physical System (CPS). The proposed technique uses the hardware characteristics of a sensor and physics of a process to create unique patterns (herein termed as fingerprints) for each sensor. The sensor fingerprint is a function of sensor and process noise embedded in sensor measurements. Uniqueness in the noise appears due to manufacturing imperfections of a sensor and due to unique features of a physical process. To create a sensor's fingerprint a system-model based approach is used. A noise-based fingerprint is created during the normal operation of the system. It is shown that under data injection attacks on sensors, noise pattern deviations from the fingerprinted pattern enable the proposed scheme to detect attacks. Experiments are performed on a dataset from a real-world water treatment (SWaT) facility. A class of *stealthy* attacks is designed against the proposed scheme and extensive security analysis is carried out. Results show that a range of sensors can be uniquely identified with an accuracy as high as 98%. Extensive sensor identification experiments are carried out on a set of sensors in SWaT testbed. The proposed scheme is tested on a variety of attack scenarios from the reference literature which are detected with high accuracy.

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, Attacks on Sensors, Sensor Fingerprinting, Authentication.

ACM Reference Format:
Chuadhry Mujeeb Ahmed, Jianying Zhou, and Aditya P. Mathur. 2018. Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate sensors in CPS . In *2018 Annual Computer Security Applications Conference (ACSAC '18), December 3–7, 2018, San Juan, PR, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3274694.3274748

ACSAC '18, December 3–7, 2018, San Juan, PR, USA

ACM ISBN 978-1-4503-6569-7/18/12…$15.00
https://doi.org/10.1145/3274694.3274748

1 INTRODUCTION

A Cyber Physical System (CPS) is composed of a set of actuators, sensors, controllers and communication networks [24]. Examples of common CPS are the smart grid, water treatment plant, autonomous vehicles, and implantable medical devices. Connectivity in a CPS provides improved monitoring and operation of a physical process. Such advancements are helpful but also bring up the challenge of secure operation of the connected devices. Ensuring secure operation in a CPS is an important challenge [7].

Recent research efforts stem from legacy IT infrastructure perspective [22]. Network security measures are suggested for the securing the links between different devices. Network-based intrusion detection based traffic pattern is most widely proposed solution [20, 27]. These methods might work well for legacy IT networks but there is a physical part to CPS which also plays an important role in ensuring secure operation. Previously it has been shown that digital intrusion detection methods fail when attack originates in the physical domain, as there would be no change in network traffic patterns [39]. A lot of information is generated at a sensor in a CPS, which could be attacked in physical domain [39] or cyber domain [44].

Sensor data is transmitted to a programmable logic controller (PLC) to take an appropriate action based on the sensor measurement. If an adversary is able to spoof sensor data in the digital or physical domain, it can take the system to an unsafe state. The focus here is not on the confidentiality of the data as in legacy computer security but on the integrity and trustworthiness of the data [17, 22]. Attacks on sensor measurements have been designed and detection methods have been proposed in recent studies [2, 33, 36, 38–40, 43, 46]. The physical domain poses a security threat for a CPS because an attack executed in the cyber domain can result in catastrophic outcomes in physical space and on the lives of people [8, 11, 14]. Physical domain brings challenges on one hand and it can prove to be useful for security if the physics of the process is utilized. An attacker who tries to defy the rules of physics should expose itself. An understanding of the physics of the process can help secure a CPS.

1.1 Our Solution

A novel technique is proposed to identify sensors and detect data integrity attacks in a Cyber Physical System. The proposed technique uses the hardware characteristics of a sensor to create unique patterns (herein termed as fingerprints) for each sensor. The sensor

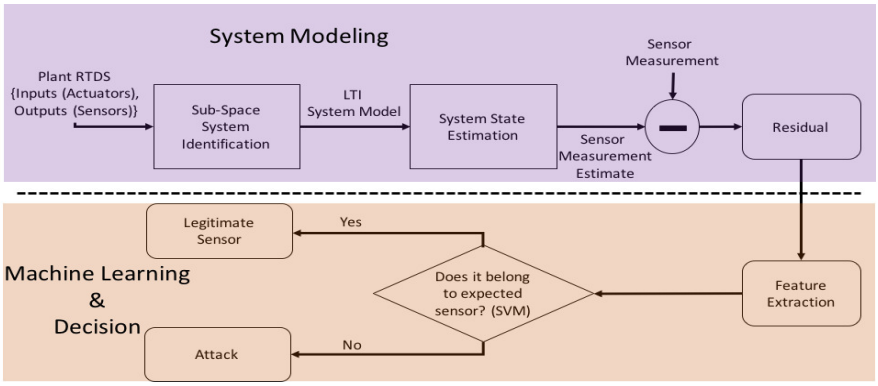

Figure 1: Overview of the proposed technique.

fingerprint is a function of noise in sensor measurements. Uniqueness in the noise appears due to manufacturing imperfections. To create a sensor's fingerprint, a unique challenge is to answer the question *how can noise be extracted from a sensor's measurement*? This is challenging as we need to know the real quantity to extract a noise vector from the sensor data. In process plants, this might not be straightforward as the amount of quantity keeps changing. For example, a level sensor in a process plant is used to measure the level of fluid in a tank. If the level is supposed to stay at a constant amount, then a set of readings measured by the level sensor can be considered as the noise of the sensor because we know the ground truth, that it should return a constant level but because of measurement noise, the readings from the sensor fluctuate. To solve this issue, we came up with the idea of obtaining a system model for the process plant which captures the dynamics of the physical process. By using the system model we can employ an estimator, for example the Kalman filter, to estimate the future measurement of a sensor. An estimate of sensor measurement at time k predicts the next sensor measurement given $k-1$ measurements and a model for system dynamics. We get sensor measurement at time k and we can calculate the estimated measurement for the time k. The difference between the two quantities is actually the noise from the measurements. The difference between sensor measurement (y_k) and an estimate of the sensor measurement ($\hat{y}_k$) is termed as residual (r_k) [4, 28, 31]. A machine learning algorithm is used to create a profile from the noise pattern in the residual.

1.2 How Does the Proposed Technique Work?

A noise pattern based fingerprinting technique is presented. The proposed technique attributes the received data from sensors to its associated sensor using the unique fingerprint of each sensor. Uniqueness in the fingerprint is due to manufacturing imperfections of a device [13] and a random pattern due to the physical process. For example, two water level sensors deployed on top of a tank would exhibit different noise patterns due to manufacturing inconsistencies (sensor noise component), the rate of water flowing in and out, and the structure of a tank (process noise component). The proposed technique can be used as a sensor identification technique and also as an attack detection technique. If an adversary tries to send malicious measurement either by using an external device inside of the system as man-in-the-middle [44] or outside of the system [39] or changing the sensor [3], it can be detected, as the noise profile from the injected data would not match with the reference pattern. In general, it is shown that any attack on sensor measurements could be detected if it changes the statistics of the noise pattern. We can identify a sensor using one-to-one matching, i.e. matching the sensor data with its reference profile created beforehand. A feature vector consisting of eight time domain and frequency domain features captures the uniqueness of sensor and process noise-based patterns. A support vector machine (SVM) classifier is trained and tested for the proposed technique. Experimental results on a real-world water treatment (SWaT) testbed [26] support the idea of fingerprinting noise pattern for device identification and attack detection. Experiments were performed on a total of 18 sensors available in the SWaT testbed and these sensors are industry grade representatives of a general industrial system. Results demonstrated a minimum sensor identification accuracy of 94.5%.

In [5] we presented preliminary results related to this paper but there are substantial differences in this work. In [5] we focused on the idea of noise-based fingerprint and the basis for such fingerprints. One significant difference is the attacker model. In [5], only one type of attack was considered while in this work a range of cyber attacks are considered from a set of benchmark attacks [10, 16] for a real testbed. In this work, a one-class SVM (OC-SVM) classification is used for attack detection in contrast to preliminary analysis where a multi-class classifier was used. For the case of multi-class classification, one has to train a classifier for attack data too but OC-SVM frees us from this restriction, making it more usable in case one does not have the attack dataset a priori. Another major contribution is the analysis of an advanced attacker that tries to compromise the proposed technique. We provide theoretical bounds for state deviation under such attacks. A major contribution of this work is to improve on a major limitation of [5] whereby that did not detect any advanced attacker with the knowledge of noise profile and arbitrary injection of spoofed data to sensors. Here, we provide security proof for the proposed technique. We also addressed research questions regarding delays in attack detection, the effects of data size on the accuracy of the classifier and performance comparisons between different classifier examples. We believe this is the first work which is based on our novel idea of sensor and

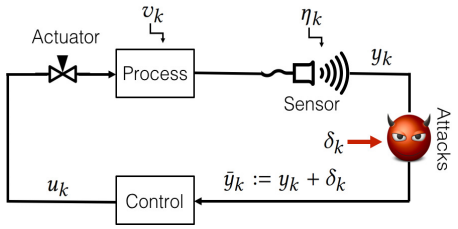

Figure 2: Attack on sensor measurements in a CPS.

process noise fingerprinting and comprehensively carried out an experimental study on a real water treatment testbed.

Our Contributions: This paper proposes novel stealthy attacks against a noise fingerprint based attack detection technique. Security analysis of the proposed fingerprinting technique is also proposed. The main contributions of this work are,

- To obtain and validate a system model, for a real water treatment testbed for the purpose of capturing system dynamics and creating device fingerprints.
- To analytically show the effectiveness of the proposed technique against a powerful stealthy attacker.
- To provide analytical bounds for state deviation for the case of stealthy attacks.
- *Sensor Identification*: To carry out sensor identification for a multitude of sensors in a realistic water treatment testbed.
- *Attack Detection*: To detect sensor attacks under a multitude of adversarial scenarios.

2 SYSTEM AND THREAT MODELS

In this section, we will explain the overview of the proposed technique. Figure 1 shows the block diagram of the proposed technique. The whole method could be divided into two main functional blocks i.e. *system modeling and machine learning*. In the following sections, the details of each implementation are given.

2.1 System Modeling

The challenge in applying noise-based fingerprinting in a process plant is that the system states are dynamic. For example, for a level sensor, if the level of water stays constant, it is simple to extract the noise fingerprint and construct a noise pattern profile for that sensor but in real processes, system states keep changing i.e., fluid level, in a tank keep changing based on actuator actions. It is thus important to capture these variations as a function of control actions so that dynamic sensor measurements can be estimated. To achieve this objective there is a need to obtain an analytical model for the control system. The state model actually captures the system dynamics and can predict future sensor measurement.

System Dynamics: In Figure 1, the first block represents data collection from a cyber physical system. In this work, a dataset from a real water treatment testbed called SWaT [26] is used. SWaT is open to CPS security researchers for experiments. More details on experimentation setup and the SWaT testbed are provided in Section 4. Data was collected over a period of seven days during which the plant ran as per normal. This real-time dataset (RTDS) is composed of data from all sensors and actuators. To obtain a system model subspace system identification technique [32] is used. The resulting system model is the Linear Time Invariant (LTI) discrete time state space model of the form,

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

Where $x_k \in \mathbb{R}^n$ represents the system state, $u_k \in \mathbb{R}^p$ is the control input and $v_k \in \mathbb{R}^n$ is the process noise at time k. $y_k \in \mathbb{R}^m$ and $\eta_k \in \mathbb{R}^m$ are the sensor measurements and measurement noise respectively. A, B, C are the state space matrices of appropriate dimensions, encompassing the system dynamics. 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 $\bar{y}_k$ ($\bar{y}_k$ is the received sensor measurement at a controller which may or may not have been attacked). 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. The focus of this paper is on sensor spoofing attacks, which could be accomplished through a *Man-in-The-Middle (MiTM)* scheme [44] or through hacking into SCADA systems [1]. 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 $k \in \mathbb{N}$ as the discrete-time index of various sequences. Then, we construct a Kalman filter which is used to obtain estimates for the system states and to find the residual vector.

Kalman Filter and Residual: We used the Kalman filter to estimate the state of the system based on the available output $\bar{y_k}$,

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

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 $e_k := x_k - \hat{x}_k$. For 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

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

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 [6]. 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. [6].
The difference between the real-time sensor measurement and sensor measurement estimate is the residual vector ($r_k := \bar{y}_k - \hat{y}_k$). The residual vector is a function of sensor and process noise and can be given as,

$$r_k = C\left\{\sum_{i=0}^{k-2}(A-LC)^i(v_{k-i-1} - L\eta_{k-i-1})\right\} + \eta_k \quad (7)$$

Where r_k is residual at each time-instant $k \in \mathbb{N}$. $v_k \in \mathbb{R}^n$ is the process noise and $\eta_k \in \mathbb{R}^m$ is the sensor noise. A and C are state space matrices and L is steady state Kalman filter gain.
Expression in Eq. (7), is an important intuition behind the idea of a noise-based fingerprint 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 CPS, we can proceed with pattern recognition techniques (e.g. machine learning) to fingerprint the given sensor and process.

2.2 Model Validation

After getting a system model, the next step is to validate the model. The procedure for the system model validation is that 1) First, legitimate control actions are chosen from the plant dataset, 2) The state space matrices (A, B and C) are used to estimate the output of the system. We use the difference equation in (3) to estimate the system state and ultimately estimate the sensor measurements. The estimate of the sensor measurement is compared to the real-time sensor measurement data. An example comparison is shown in Figure 3. The top pane shows the real sensor measurements and estimate of those sensor measurements obtained using the system model. The middle pane plots the difference between the real-time sensor measurements and estimate of the sensor measurements. Distribution (PDF) for the residual vector is plotted on the bottom pane. It can be observed that the estimate for the sensor measurement is very close to the real sensor measurements and the PDF for the residual vector is tightly bounded with a small variance. To quantify the goodness of a system model, mean square error (MSE) is used as a metric. In particular, one minus the root mean square error (RMSE) defines the estimation accuracy or best fit of a model,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (8)$$

MSE is the difference between sensor measurement and sensor measurement estimate squared and essentially gives the distance between measured and estimated value or in other words, how far the estimated value from the measured value is. The model accuracies for 18 sensors used in this study (from SWaT testbed) are shown in Table 1. It can be seen that the obtained system model is very accurate, with most of the sensors achieving scores of 99% and only a few sensors scoring marginally less. In control theory literature models with accuracies as high as 70%, are considered accurate approximations of real system dynamics [45].

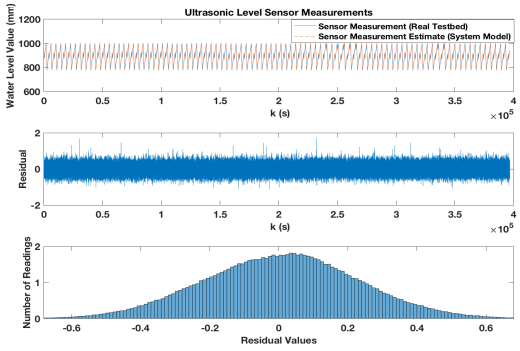

Figure 3: Validating system model obtained using sub-space system identification method.

2.3 Machine Learning and Decision

In the previous section, it has been shown that the residual vector is actually the noise pattern to be fingerprinted. A set of features are extracted from the data of residuals. A list of extracted time domain and frequency domain features is shown in Table 2. Spectral features are obtained by taking the Fourier transform of the time series data of residuals. We used a library for the support vector machine (LibSVM [9]) to train a model on extracted features. Training is performed using seven days of real-time dataset from SWaT testbed under normal operation. Residuals are labeled with a sensor ID. The trained machine learning model is tested on fresh residual vectors to either give a correct sensor ID or raise an alarm for sensor data integrity attacks. More details on data chunking and classification are provided in Section 4.

2.4 Threat Model

In a CPS, sensors play an important role by sending physical measurements to a controller to take a proper control action. An adversary can render a system vulnerable by compromising sensors. Sensors could be compromised in the physical domain (by analog sensor spoofing or physical tampering/replacement of a sensor) and cyber domain (by injecting/modifying sensor data at software layers). It is important to validate the sensor data to authenticate it, whether it is being sent by a legitimate sensor or from an adversary. Due to the limited computational power of sensors in a typical CPS, advanced cryptographic solutions are not feasible. Therefore, we came up with the proposed novel idea of noise-based authentication of sensors. The goal is to identify a sensor based on its physical characteristics. Specific cyber attacks are also considered on sensor measurements in a water treatment plant. In Figure 2, it can be seen that an attacker can modify a rightful sensor measurement by an attack value δ_k. In this section, we introduce the types of attacks launched on the secure water treatment testbed (SWaT). Essentially, the attacker model encompasses the attacker's intentions and capabilities. The attacker may choose its goals from a set of intentions [41], including performance degradation, disturbing a physical property of the system, or damaging a component. In our experiments, a range of attacks are considered from already published attack scenarios in the literature [1, 10, 16].

Table 1: Validating SWAT system model obtained from sub-space system identification. FITs are electromagnetic flow meters, AITs are chemical sensors, LITs are ultrasonic level sensors and PITs are pressure sensors. S1:FIT101, S2:LIT101, S3:AIT201, S4:AIT202, S5:AIT203, S6:FIT201, S7:LIT301, S8:FIT301, S9:DPIT301, S10:LIT401, S11:FIT401, S12:FIT501, S13:PIT501, S14:FIT502, S15:PIT502, S16:FIT503, S17:PIT503, S18:FIT601

Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
RMSE	0.0363	0.2867	0.0346	0.0113	0.0520	0.0313	0.2561	0.0200	0.0612	0.2267	0.0014	0.0096	0.0670	0.0082	0.0267	0.0037	0.0595	0.0035
(1-RMSE)*100%	96.3670	71.3273	96.5409	98.8675	94.8009	96.8656	74.3869	98.0032	93.8757	77.3296	99.8593	99.0377	93.3031	99.1821	97.3313	99.6251	94.0537	99.6501

Table 2: 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	$y_m(0)$

2.4.1 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's measurements). Also, the attacker knows the system dynamics, the state space matrices, the control inputs and outputs, and the implemented detection procedure. A powerful attacker can arbitrarily change sensor measurements to the desired sensor value, by learning and adding the sensor and process noise. We do not consider replay attack in this article because noise profile for process and sensor would be preserved during a replay attack. 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 [15, 42]. An *outsider*, e.g., end user, can also carry out a physical attack on sensors such as smart energy monitors.

2.4.2 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 generic attack to show the performance of the proposed technique. However *stealthy* attack is a worst case scenario for a detection technique where an adversary tries to deceive the detection mechanism by creating attack vectors δ_k based on working principle of the attack detection technique. In this study, the following types of data injection attack scenarios are considered,

- *Generic Sensor Spoofing Attack:* We evaluate the proposed technique for a range of network attack scenarios from benchmark attacks on SWaT testbed [16]. These attacks cover a wide range of 41 attacks on both sensors and actuators. Since the proposed technique extracts the sensor and process noise from the sensor measurements and residuals, a set of attacks on sensors are considered as shown in Table 6 in the Appendix. In general, an attack vector can be defined as,

$$\bar{y}_k = y_k + \delta_k, \tag{9}$$

  Where δ_k is the data injected by an attacker. The detail about each δ_k is described in Table 6 where it can be seen that it ranges from an abrupt injection of data to more slow change in sensor measurements.
- *Stealthy Attacks for the Proposed Technique:* These attacks are designed to be stealthy by changing sensor measurements as such that the proposed detection mechanism might fail. Since the proposed idea uses sensor and process noise fingerprint, an attacker who wants to stay stealthy might try to choose injected readings from the noise distribution of the noise pattern. To do this, an attacker needs to learn the noise pattern for each sensor and process. We assume that an attacker has the ability to do that. We have carried out extensive analysis for such an attack by deriving the bounds on the deviation of system state under such attack and also provided a security proof for attack detection under a stealthy attack scenario in the Section 3.

2.4.3 Attack Execution. All the attacks which are taken from reference work [16], are executed by compromising the Supervisory Control and Data Acquisition (SCADA) system. An attack toolbox is used to inject an arbitrary value for real sensor measurement.

3 STATE ESTIMATION AND SECURITY PROOF UNDER STEALTHY ATTACKS

In this section, the performance of the proposed technique is assessed by quantifying the effect of the attack sequence δ_k on the state of the process. We estimate the state deviation under a stealthy attack, in particular, an upper bound is derived for state estimate. A security proof is given for the performance of the proposed technique under a stealthy attack.

3.1 System State Under Normal Operation

During the normal operation of the plant, state of the system could be estimated using the system model (1) and state estimation (3). In

the following, an analytical model for state estimate and residual is derived to compare and differentiate it from attack scenario.

Proposition 1. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the state estimation error is $e_{k+1} = (A - LC)e_k + v_k - L\eta_k$.

Proposition 2. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the residual vector is $r_k = Ce_k + \eta_k$.

Proposition 3. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the state estimation is $\hat{x}_k = \sum_{j=0}^{k-1}[A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0$.

Proof: Due to space limitations the proof for Proposition (1)-(3) is provided in Appendix A. ■

From the Proposition (1-2), it can be seen that the residual vector is a function of sensor and process noise under normal operation. Proposition (3) derives a state estimate at k^{th} time instance based on system dynamics, sensor measurement and steady state Kalman filter.

3.2 System State under Stealthy Attack

After deriving the system state estimate under normal operation, this section will quantify the damage that the attacker can induce to the system in the worst case scenario of a stealthy attack for the proposed detection technique. It is assumed that the attacker has perfect knowledge of the system dynamics, the Kalman filter, control inputs, sensor measurements and detection procedure. In particular, we are interested in attack sequences δ_k that can induce a change in the system dynamics while trying to be hidden by the detection technique. This class of attacks is known as stealthy attacks. First a general attack vector δ_k is considered to spoof sensor measurements.

$$\bar{y}_k = y_k + \delta_k, \tag{10}$$

Where $\bar{y}_k$ is the sensor measurement under attack and can be represented in terms of attack vector δ_k. Next, state estimation error, residual vector and state estimate are derived for a scenario of an attack on sensor measurements. A generic stealthy attack vector is considered with the following properties,

- An attacker can choose an attack value (δ_k) from the noise distribution of the residual vector.
- An attacker can choose an arbitrary value to inject in the sensor output.
- An attacker can choose a combination or either of the above two attack vectors.

A generic stealthy attack vector δ_k can be expressed as,

$$\delta_k = \beta[Ce_k + \eta_k] + \alpha \tag{11}$$

Where $Ce_k + \eta_k$ is residual vector under normal operation as defined by proposition (2). $\beta \in \mathbb{R}$ is a scalar value to choose multiples of residual vector distribution (noise pattern). $\alpha \in \mathbb{R}$ is to add an arbitrary reading in the sensor measurement. The intention behind this attack vector in Eq. (9) is that if an attacker chooses δ_k from the noise pattern distribution of a sensor, it might evade detection. A powerful attacker can intelligently choose α while keeping $\beta = 0$ to add an arbitrary value and keep original noise pattern from sensor measurements. Since the proposed technique depends on noise pattern in the residual vector under normal operation of the plant $r_k = Ce_k + \eta_k$, choice of expression in Eq.(11) is realistic. In the following, system state deviation is derived under a stealthy attack scenario.

Proposition 4. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the state estimation error is $e_{k+1} = Ae_k + v_k - (\beta + 1)LCe_k - (\beta + 1)L\eta_k - L\alpha$.

Proposition 5. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the residual vector is $r_k = (\beta + 1)Ce_k + (\beta + 1)\eta_k + \alpha$.

Proposition 6. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the state estimation is $\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\alpha + (\beta + 1)A^jL\eta_{k-1-j} + (\beta + 1)A^jLCe_{k-1-j} + A^jBu_{k-1-j}] + A^kx_0$.

Proof: Due to space limitations the proof for Proposition (4)-(6) is provided in Appendix B. ■

One can compare expressions in propositions (1-3) and propositions (4-6). It can be seen that under an attack state estimate and residual vector not only depends on system dynamics but also on attacker's choice of α and β i.e., the attack vector. This intuition is base for the proposed technique where one can differentiate between the distribution of residuals in attack and attack-free scenarios.

3.3 State Degradation under Stealthy Attack

The upper bounds on the deviation of state estimate are derived a stealthy attack. An upper bound on estimation error is also calculated under a powerful stealthy attack.

Proposition 7. Consider the process (1), the Kalman filter (3)-(5). Let the sensors be attacked by the stealthy attack sequence (11). Then, if $\rho([A-(\beta+1)LC]) < 1$, it is satisfied that $\lim_{k\to\infty} ||E[e_k]|| = \gamma$, where $\gamma := ||[I - (A - (\beta + 1)LC)]^{-1}[\bar{v} - (\beta + 1)L\bar{\eta} - L\alpha]||$.

Proof: From proposition 4 we have,

$$e_{k+1} = [A - (\beta + 1)LC]e_k + v_k - (\beta + 1)L\eta_k - L\alpha. \tag{12}$$

Assuming $\rho([A - (\beta + 1)LC]) < 1$,implies that $[I - (A - (\beta + 1)LC)]$ is invertible. Calculating the expectation on eq. (12),

$$E[e_{k+1}] = [A-(\beta+1)LC]E[e_k]+E[v_k]-(\beta+1)LE[\eta_k]-L\alpha. \tag{13}$$

$$E[e_{k+1}] = [A - (\beta + 1)LC]E[e_k] + \bar{v} - (\beta + 1)L\bar{\eta} - L\alpha. \tag{14}$$

Therefore, $\rho([A - (\beta + 1)LC]) < 1$ imply that the equilibrium $\bar{e}$ is exponentially stable [6], i.e., $\lim_{k\to\infty} E[e_k] = \bar{e}$. The Euclidean norm on $\mathbb{R}^n$ is a continuous function from $\mathbb{R}^n$ to $\mathbb{R}_{\geq 0}$ [19]. It follows

that $\lim_{k\to\infty}||E[e_k]|| = ||\lim_{k\to\infty}E[e_k]|| = ||\bar{e}||$. For the case of $\lim_{k\to\infty}E[e_k] \approx \lim_{k\to\infty}E[e_{k+1}]$ converges to $\bar{e}$ from (14),

$$\bar{e} = [A-(\beta+1)LC]\bar{e} + \bar{v} - (\beta+1)L\bar{\eta} - L\alpha, \tag{15}$$

$$[I-(A-(\beta+1)LC)]\bar{e} = \bar{v} - (\beta+1)L\bar{\eta} - L\alpha, \tag{16}$$

$$\bar{e} = [I-(A-(\beta+1)LC)]^{-1}[\bar{v} - (\beta+1)L\bar{\eta} - L\alpha] \tag{17}$$

This completes the proof. ∎

From Eq. (17), it is observed that the error estimate depends on attack components α and β and bounded by attacker's choice of these components and system dynamics.

State estimate under stealthy attack: From Proposition (6) we have, $\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\alpha + (\beta+1)A^jL\eta_{k-1-j} + (\beta+1)A^jLCe_{k-1-j} + A^jBu_{k-1-j}] + A^kx_0$.

For the case of stealthy attack an upper bound is derived on $\hat{x}_k$ and indirectly on $\hat{y}_k$ as $\hat{y}_k := C\hat{x}_k$. The norm of $\hat{x}_k$ is,

$$||\hat{x}_k|| = ||\sum_{j=0}^{k-1}[A^jL\alpha + (\beta+1)A^jL\eta_{k-1-j} + (\beta+1)A^jLCe_{k-1-j} + A^jBu_{k-1-j}] + A^kx_0||, \tag{18}$$

Using triangular inequality for norm [6], following is obtained,

$$||\hat{x}(k)|| \le \sum_{j=0}^{k-1}||A^jL\alpha|| + \sum_{j=0}^{k-1}||(\beta+1)A^jL\eta_{k-1-j}|| + ||A^kx_0|| \sum_{j=0}^{k-1}||(\beta+1)A^jLCe_{k-1-j}|| + \sum_{j=0}^{k-1}||A^jBu_{k-1-j}|| \tag{19}$$

$$||\hat{x}(k)|| \le \sum_{j=0}^{k-1}||A^j||||L||\alpha|| + \sum_{j=0}^{k-1}||(\beta+1)A^j||||L||||\eta_{k-1-j}|| + ||A^kx_0|| + \sum_{j=0}^{k-1}||(\beta+1)A^j||||L||||C||||e_{k-1-j}|| + \sum_{j=0}^{k-1}||A^j||||B||||u_{k-1-j}|| \tag{20}$$

$$||\hat{x}(k)|| \le ||L||||\alpha||\sum_{j=0}^{k-1}||A^j|| + ||L||\sup_{1\le N\le k}||\eta_N||\sum_{j=0}^{k-1}||(\beta+1)A^j|| + ||L||||C||\sup_{1\le N\le k}||e_N||\sum_{j=0}^{k-1}||(\beta+1)A^j|| + ||B||\sup_{1\le N\le k}||u_N||\sum_{j=0}^{k-1}||A^j|| + ||A^k||x_0. \tag{21}$$

The expression in (21), is a geometric series in norm of matrix A, and for geometric series,

$$\sum_{k=0}^{n}ar^k = a\left(\frac{1-r^{n+1}}{1-r}\right), \tag{22}$$

For a system where $\rho(A) < 1$, there exists a matrix norm $||.||_*$ such that $||A||_* < 1$ and series in eq. (21) is convergent and $\hat{x}(k) \le \gamma_k$:

$$\gamma_k = \frac{||L||||\alpha||}{1-||A||_*} + \frac{||U||_\infty||B||}{1-||A||_*} + (\beta+1)\frac{||L||||\eta||_\infty}{1-||A||_*} + (\beta+1)\frac{||L||||C||||e||_\infty}{1-||A||_*} + ||A||^k||x_0||, \tag{23}$$

Where $||J||$ is the induced norm of matrix $J \in \mathbb{R}^{pxp}$ defined as $||J|| := \max_{x\in\mathbb{R}^p, ||x||_2=1}||Jx||$. The notation $||J||_*$ stands for some matrix norm such that $\rho(J) < 1$ implies $||J||_* < 1$, where $\rho(.)$ denotes spectral radius. Spectral radius of a matrix $J \in \mathbb{R}^{pxp}$ is defined as $\rho(J) = \max(\lambda_p)$, where λ_p are the eigen values for J. For $\rho(J) < 1$, such a norm always exist where $||J||_* < 1$ as shown by Lemma 5.6.10 in [19]. If $\rho(J) \ge 1$, we simply take $||J||_* = ||J||$. Expression (23) provides upper bounds for the effects of the stealthy attack on the state estimation given $\rho(A)$ and the design of the proposed detection technique.

3.4 Security Argument

In the previous section, a stealthy attacker is assumed, and it was shown that how system states would deviate from normal behavior under such an attack. In the following, a security proof is provided for such worst case stealthy attack.

Definition 1. *Linear Transformation on a random variable.* A linear transformation takes the form of creating a new variable from the old variable using the equation for a straight line: new variable = a + b* (old variable) where a and b are mathematical constants. What is the mean and the variance of the new variable? To solve this let X denote the old variable and assume that it has a mean of $\bar{X}$ and a variance of S_X^2. Let X* denote the new variable. Then $X^* = a + bX$ with new mean: $\bar{X}^* = a + b\bar{X}$ and new Variance: $S_{X^*} = b^2S_X^2$.

Definition 2. *Sum of two normal distributions.* The sum of two independent normally distributed random variables is normal, with its mean being the sum of the two means, and its variance being the sum of the two variances (i.e., the square of the standard deviation is the sum of the squares of the standard deviations) [18].

Residual vector under normal operation i.e. no attack can be given by $r_k = Ce_k + \eta_k$, as $r_k := y_k - \hat{y}_k$. For an attack δ_k it's $r_k := y_k + \delta_k - \hat{y}_k$ resulting in $r_k^* = Ce_k + \eta_k + \beta(Ce_k + \eta_k) + \alpha$ which is $r_k^* = r_k + \beta(r_k) + \alpha$,

$$r_k^* = (\beta+1)r_k + \alpha \tag{24}$$

Theorem 1. Under a powerful attacker defined as $\delta_k = \beta(Ce_k + \eta_k) + \alpha$ which is a function of probability distribution of residual vector r_k. The original random variable r_k goes to r_k^* as in (24). The statistics of this new random variable deviates from the original probability distribution of r_k. This property enables data injection attack detection on sensors using residual vector r_k.

Proof: We prove the above assertion using following cases for $\delta_k = \beta(Ce_k + \eta_k) + \alpha$.

- Case 1: An attacker choose to inject data from the noise distribution of r_k without any arbitrary data α i.e. $\alpha = 0$. Resultant attack vector is $\delta = \beta(Ce_k + \eta_k)$ with $r_k^* = (Ce_k + \eta_k) + \beta(Ce_k + \eta_k)$. As we have seen that noise pattern can be approximated as a normal distribution with a mean of 0 and

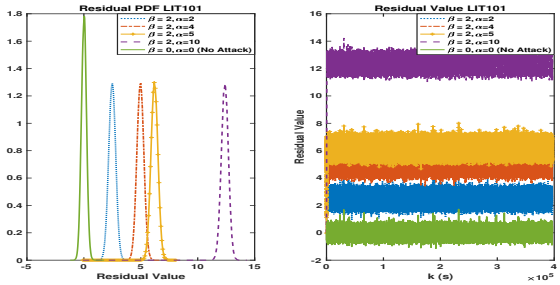

Figure 4: For the case of $\delta_k = (\beta+1)(Ce_k+\eta_k)+\alpha$. An attacker makes an arbitrary choice of β and α. We can see that the residual deviates from the normal pattern under an attack facilitating the detection using noise-based fingerprints.

a variance of S_r^2. From definition 2, we can see that mean for the resultant distribution is still 0 but $S_{r^*}^2 = (\beta+1)^2 S_r^2$. Hence the resultant statistics changed even if an attacker choose to inject data from the noise distribution of the sensor and process.

- Case 2: An attacker choose to inject data arbitrarily and do not add anything from the noise pattern of the residual vector. This means for the case of $\delta_k = \beta(Ce_k + \eta_k) + \alpha$, $\beta = 0$ with resultant attack vector as $\delta_k = \alpha$, where α is an arbitrary scalar value to be added in sensor measurement. This can be considered a very intelligent attack for which residual vector becomes $r_k^* = (Ce_k + \eta_k) + \alpha$ i.e. $r_k^* = r_k + \alpha$. Intuitively it means that noise pattern in r_k is offset with a constant value α, an obvious consequence of which is change of mean of the random variable from 0 to α. Using definition 1, we can see that the resultant residual vector ($r_k^* = r_k + \alpha$) is a linearly transformed version of r_k. Variance would stay the same but mean value goes to $\bar{r}_k^* = \alpha + \bar{r_k}$.
- Case 3: This is a general case for an attack vector δ_k. An attacker uses a linear combination of noise pattern in residual and an arbitrary value α and constructs an arbitrary attack with $\delta_k = \beta(Ce_k + \eta_k)$. The resultant residual vector becomes $r_k^* = (Ce_k + \eta_k) + \beta(Ce_k + \eta_k) + \alpha$, which can also be represented as $r_k^* = (\beta + 1)(r_k) + \alpha$. Using definition 1, we can see that mean for such a residual vector is changed to $\bar{r}_k^* = \alpha + (\beta + 1)\bar{r}_k$ and variance $S_{r^*}^2 = (\beta + 1)^2 S_r^2$. It is proved that if we use noise pattern in residual vector as a fingerprint, any kind of sensor data injection attack can be detected. This completes the proof. ■

4 EXPERIMENTATION SETUP AND EVALUATION

4.1 Secure Water Treatment Testbed (SWaT)

The experiments are carried out in a state-of-the-art water treatment testbed called SWaT [26]. A pictorial abstraction of the SWaT testbed is shown in Figure 5. SWaT imitates the complete process of a real water treatment plant and it produces 5 gallons/minute filtered water.

Water Treatment Process: There are six stages in SWaT and each stage is controlled by a dedicated PLC. Sensor measurements are used by PLCs to control the process plant. For each stage a set of sensors $S = \{S1, S2, ...\}$ and actuators $A = \{A1, A2, ...\}$ are listed in Figure 5. An interested reader is referred to [26], for details on the architecture of SWaT testbed. The proposed technique is tested on different sensors to demonstrate its efficacy on different sensing systems. A system model for complete SWaT testbed is obtained using sub-space system identification technique [32]. Data is collected for seven days of normal operation of the plant. For the next four days, a range of attacks was executed. For sensor identification based on the system model, we used the normal operation data from SWaT testbed and used the proposed technique to identify a sensor against adversarial manipulations of a sensor measurement.

4.2 Feature Extraction and Learning Phase on Normal Data

After obtaining the system model for SWaT testbed and calculating the residual vector for each sensor, the next step is to train a machine learning algorithm to find out anomalies. We used LibSVM [9] to classify the dataset into normal or anomalous behavior. To prepare data for classification, we extract a set of eight features as shown in Table 2 from the residual vector. These extracted features are labeled with a sensor ID (ground truth). For each sensor, SVM is used as a 2-class classifier by labeling the data from the rightful sensor (ground truth) as 1 and 0 for rest of sensors. This way attacks can be treated as kind of data from *other* class. A machine learning model is trained on a normal dataset collected over a period of seven days and validated the obtained machine learning model using cross validation approach. Since attacked data is not available beforehand, supervised learning could miss some of the attacks. To deal with this problem, a one-class SVM classifier is used to train the machine learning model on one class of normal data and in the testing phase, anything else would be declared as an attack. Results for these experiments are presented in the following section.

4.3 Results

Following research questions are formulated and answered in this work.

- RQ1: Proof of Fingerprint. Does sensor and process noise-based sensor fingerprint exist?
- RQ2: Attack Detection Delay. What is the right amount of data to detect correct sensor ID with the highest accuracy?
- RQ3: How the amount of training and testing data affects sensor identification performance?
- RQ4: How well the proposed technique can detect network attacks on sensor measurements?

RQ1: Proof of fingerprint. Does sensor and process noise-based sensor fingerprint exist? In Figure 6 we can see statistical features of the residual vector for two sensors in stage 1 of SWaT testbed. We observe that both the sensors can be uniquely identified based on the noise profile contained in the residual vector. Figure 6 shows that using the three features namely mean, variance and mean average deviation can help us distinguish between two sensors. This visual representation is proof of the existence of a noise-based fingerprint.

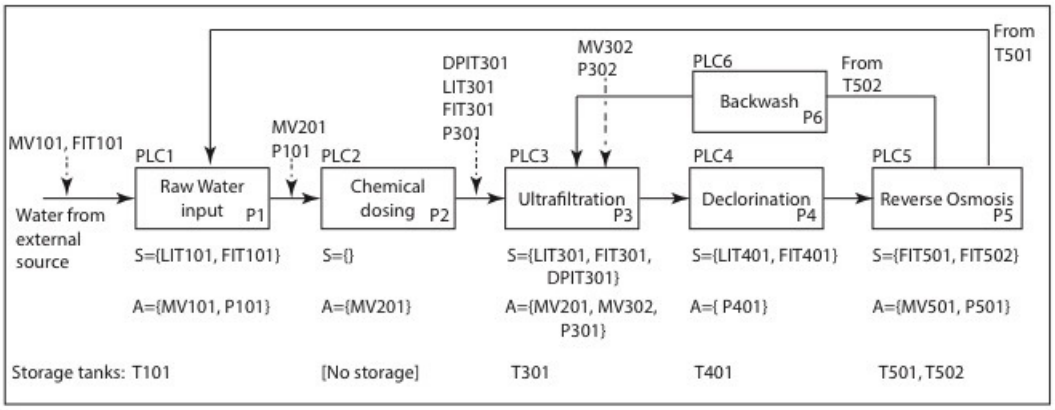

Figure 5: Secure Water treatment testbed (SWaT): P1 though P6 indicate the six stages in the treatment process. Arrows denote the flow of water and of chemicals at the dosing station.

Table 3: Different data chunk size and accuracy of the classifier. This experiment is to establish a trade off between classifier accuracy and amount of data required to make a classification decision.

↓ Sample Size / Sensor →	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
60	95.1097%	94.6%	96.5839%	94.6427%	93.423%	95.9593%	94.5721%	94.6205%	94.5721%	100%	94.5721%	99.1053%	96.3729%	94.9045%	94.5721%	94.5721%	94.6189%	94.5721%
120	95.2125%	95.0253%	96.4684%	94.6821%	94.9547%	96.6162%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.8507%	97.2138%	95.5884%	94.5721%	94.5721%	95.8626%	94.5721%
250	95.2391%	95.4374%	96.6003%	94.8492%	96.4977%	96.8431%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.5122%	97.5922%	96.7%	94.5721%	94.5721%	97.0962%	94.7944%
500	95.0856%	96.0438%	96.8309%	95.7221%	97.2827%	96.7693%	94.5722%	94.5722%	94.5722%	100%	94.5722%	97.666%	97.8029%	97.5565%	94.5722%	94.8665%	97.9261%	95.64%
2000	96.3816%	96.2719%	95.9978%	95.0384%	96.1897%	95.7237%	94.545%	97.5877%	95.6689%	100%	94.5724%	96.1897%	97.8618%	96.3542%	94.4901%	96.6009%	98.1908%	96.9846%

Table 4: Different cross validation k. It does not matter how much data we use for training and testing, accuracy stays the same.

↓ Sample Size / Sensor →	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
2	95.201%	94.9218%	96.3108%	94.6378%	94.8907%	96.3781%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.8524%	96.9856%	95.5589%	94.5721%	94.5721%	95.8232%	94.5721%
3	95.1977%	94.9892%	96.365%	94.6657%	94.9202%	96.521%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.8491%	97.1514%	95.5819%	94.5721%	94.5721%	95.8134%	94.5721%
5	95.2125%	95.0253%	96.4684%	94.6821%	94.9547%	96.6162%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.8507%	97.2138%	95.5884%	94.5721%	94.5721%	95.8626%	94.5721%
10	95.2272%	95.063%	96.4816%	94.692%	94.9777%	96.6819%	94.5721%	94.5721%	94.5721%	100%	94.5721%	98.854%	97.2532%	95.5934%	94.5721%	94.5721%	95.8478%	94.5721%
15	95.2289%	95.0729%	96.4619%	94.6904%	94.9875%	96.7016%	94.5721%	94.5738%	94.5721%	100%	94.5721%	98.8491%	97.2664%	95.5966%	94.5721%	94.5721%	95.8413%	94.5721%

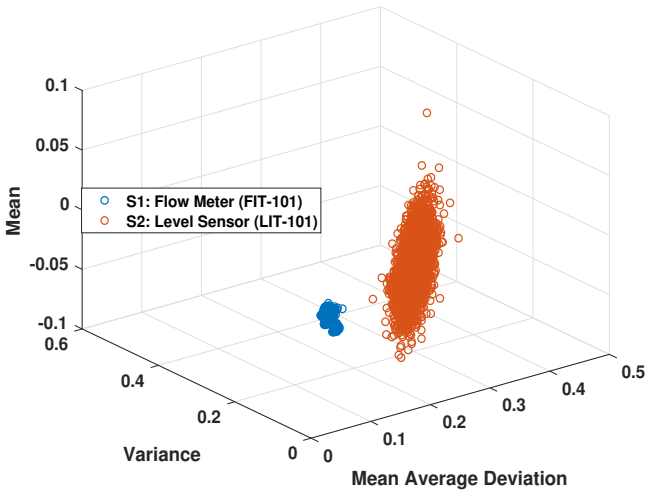

Figure 6: Existence of Fingerprint.

However, to see the performance of the proposed technique, a systematic analysis is carried out by using machine learning and a complete feature set as shown in Table 3. For the classification, a binary classifier is used by labeling data from the sensor of interest as legitimate and data from the rest of all sensors as illegitimate. This way during the testing phase, each sensor is tested against data from all other sensors. Higher sensor identification accuracies in Table 3 point to the existence of the noise-based fingerprint.

RQ2: Attack Detection Delay. What is the right amount of data to detect correct sensor ID with the highest accuracy? To answer this question, chunks of different data size are created for the whole dataset. Data from each sensor is sampled at an interval of one second. The idea is to have a chunk of time series big enough to capture the dynamics of the process and small enough to reduce the delay in the attack detection. Chunk size ranging from 60 readings to 2000 readings is used. The empirical results are shown in Table 3. It can be seen that for a chunk size of 60 readings, the classification accuracy is slightly low. However, for higher chunk size accuracy improves a bit but it means to wait more before making a decision on the received data from a sensor. It is observed that 120 samples

are a good trade-off between accuracy and detection time (i.e. 120 seconds).

RQ3: How the amount of training and testing data affects sensor identification performance? An important question is regarding the amount of data needed for training and testing purpose. A cross validation analysis is done to answer this question. For a k fold cross validation on a dataset means, the whole data set would be divided into k chunks of data, then $k-1$ chunks are used for training and one chunk for testing. For example for 5 fold cross validation, the first 4 of 5 chunks would be used to train the classifier and the last chunk to test the trained machine learning model and then other 4 chunks are used for training and rest one for testing. This procedure is repeated for all chunks and average accuracy is reported. This procedure for k fold cross validation shows how does the choice of data range for training and testing make difference. For a $k = 2$, means whole data set is divided into two chunks and each half is used for training and testing. Similarly, a $k = 15$ means the whole dataset is divided into 15 chunks and validation is done for each chunk as explained above. Ideally, a classifier should be robust to the size of the dataset used for training and testing. It should also be independent of the fact that which dataset range in time series is used for training and testing. Results in Table 4, point out that the accuracy of our chosen classifier function does not depend on the choice of size of the dataset. This gives a practical insight into the case when limited data is available to train the machine learning model. The high classification accuracy proves our hypothesis that sensors can be uniquely identified using a combination of sensor and process noise.

RQ4: How well the proposed technique can detect network attacks on sensor measurements? Previous results demonstrate the ability of the proposed technique to identify sensors uniquely based on their noise fingerprints. This part is to evaluate the performance of the noise fingerprinting technique as an attack detection technique. A set of benchmark attacks are tested [10, 16]. Out of 41 attacks in [16], 18 attacks are on sensors and therefore, chosen for this work. A List of attacks and attack descriptions is provided in Table 6 in the Appendix. The last column in Table 6 shows one-class SVM attack detection accuracy for that particular attack. For each individual attack and a chunk size of 250, it can be seen that one-class SVM could detect all but 2 attacks. A pictorial example of an attack is shown in Figure 7. Figure 7 shows an example of a level sensor at stage 1 (of the SWaT testbed) under attack. On the right and left pane is the zoomed-in plot. In Figure 7, the left pane shows a zoomed-in region when the system is not under any attack. The rightmost pane shows the zoomed-in residual vector during an attack execution period. It can be observed that the residual significantly deviates from the normal profiled noise pattern. This observation provides an intuition for attack detection using the proposed technique. Apparently, from the middle pane, visual inspection also means that one should be able to detect all the attack points. However, to formally show the performance of the attack detector, True Positive Rate (TPR: meaning attacked data declared as an attack), and True Negative Rate (TNR: attack-free data declared as normal) are used. The attack detection results are shown in Table 5. For each sensor in the SWaT testbed, attack sequences are shown. These attack sequences and attacked dataset are obtained from already published benchmark

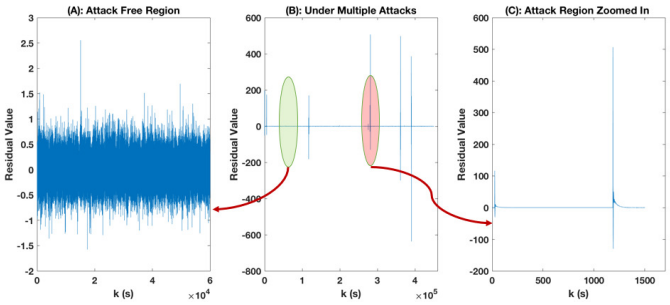

Figure 7: Attack executed on level sensor (LIT-101).

attacks [10, 16]. We can see a high TNR indicating successful attribution of normal data but the TPR is slightly lower using multi-class classification. The reason for the lower TPR is that for multi-class supervised learning, there are no examples of attacked data beforehand to train the classifier. However, the analysis is extended by using a one-class SVM classifier for each sensor. For the case of a one-class classifier, one just needs to have normal data for training and during the testing phase, anything other than that single-class is declared as an anomaly. It can be seen in Table 5 that the TPR has increased significantly making it much easier to detect attacks but it comes at the cost of a slightly lower TNR. It's a trade-off between higher TNR and TPR. To reduce false alarms, one can come up with heuristics e.g. TNR below 85% would raise an alarm for an attack. This kind of heuristic method can reduce the number of false alarms while detecting attacks as the TPR is much higher when using the one-class machine learning method. There is another interesting observation. Since the proposed technique is based on a physical system model, it exhibits a strong coupling between the inputs and outputs of a system. If attacks are executed on level sensors, we could see the effect on associated flow meters and vice versa. This indicates the coupling between the laws of physics even though the sensors are of different types. This observation means even if the TPR rate is less than 100%, it could be aggregated (in a voting manner) from coupled sensors and declare attacks if the aggregate is higher than a learned threshold. We are able to detect most of the sensor attacks from the reference literature [10, 16], indicating the effectiveness of the proposed technique against a range of cyber attack scenarios.

5 DISCUSSION

Scalability: A multitude of sensors is studied from a real water treatment testbed. The high accuracy of sensor identification results in Table 4 points towards the effectiveness of the technique for the case of a number of processes and sensors. A total of 18 sensors were available for experimentation in a *sixstage* water treatment testbed. The problem is formulated as binary classification, by considering the legitimate sensor as (class 1) and rest of the 17 sensors as illegitimate or compromised (class 2). The goal is to authenticate data being received by a particular sensor. In a particular setting of a SCADA system, the data stream is expected to be generated from a particular sensor but the question is, *can the integrity of the sensor data be ensured*? Ideally, a measurement from each sensor should be

Table 5: Attack detection performance. Level sensors in different stages of SWaT are attacked with different attacker goals and strategies as explained in Table 6. Flow meter at stage 1 and 3 are not attacked but from the residual vector of these sensors it is possible to detect anomalies due to attack sequences on level sensors, that's why for those sensors *None* is put in attack sequence column. Detection is possible due to process coupling captured by system model. Average TPR and TNR are shown for all attack sequences. MC-SVM := Multi-class SVM, OC-SVM := One-class SVM.

Sensor	Atk. seq. [a]	Attacked [b]	Detected [c]	MC-SVM TNR	MC-SVM TPR	OC-SVM TNR	OC-SVM TPR
DPIT-301	8	8	5	99.65%	62.5%	86.3%	88.88%
LIT-101	3,21,30,33,36	27	24	97.88%	88.88%	89.4%	93.54%
FIT-101	None	27	22	99.49%	81.48%	94.2%	80.64%
LIT-301	7,16,26,32,41	37	29	91.41%	78.37%	88.7%	80.95%
FIT-301	None	37	22	91.55%	59.45%	88.85%	78.57%
LIT-401	25,27,31	35	20	92.09%	57.14%	89.5%	77.5%
FIT-401	10,11,39,40	12	8	99.86%	66.66%	91.6%	73.3%

[a] Attack Sequences [16]
[b] Total Chunks Attacked
[c] Chunks where Attack is Detected

checked only against its own profile and not with all the rest of the sensors. However since we do not have an attacker's spoofed data beforehand, it is not possible to train a classifier for the illegitimate class. Therefore, data from all the rest of the sensors is considered an "other class". Later we also used one-class SVM to train only for "legitimate class" for each sensor and then tested the performance under various attacks. These observations mean that the proposed technique will scale well even if the number of sensors is huge as it is not necessary to compare a sensor's fingerprint with the whole population. The experiments are performed on a water treatment testbed which has different types and models of sensors. Also, there are 6 stages in the process plant with different process dynamics, which points to the generality of the proposed technique.

Limitation (False Alarm Rate): For any intrusion detection system, false alarms are a limiting factor. In Table 5, it can be seen that using one-class SVM helps to detect attacks with higher accuracy (TPR) but at the cost of slightly lower TNR (i.e. misclassifying normal operation as under attack). Since the lowest TNR is 86%, we can come up with a heuristic threshold of 85% to raise an alarm for an attack. This would significantly lower the false alarm rate. We are also experimenting with a moving average window filter to lower the false alarm rate by tuning the detector parameters on live water treatment testbed.

Performance Comparison with Reference Techniques: Attacks studied in this article are a set of benchmark attacks used by others too [1, 10]. The list in Table 6 shows executed attacks performed on sensors or on a sensor-actuator pair. Accuracy results in Table 6 can be directly compared (for each exact attack sequence) with results in [10]. We observe that our proposed technique performs better but it would not be a fair comparison without pointing out the downside of our proposed technique i.e. false alarms. However, authors in [10] do not provide false alarms in case of an attacked dataset (meaning even for attack dataset, attacks were executed from time to time and most of the readings are normal, therefore one should also provide accuracy for normal data classification). In [1] the detection metrics are not attack detection accuracy but an alarm for the case of an attack. Therefore, from Table 6 it can be seen that the proposed technique performs similarly by successfully detecting attacks on sensors, and also does not require design information (which is required by the method in [1]) to come up with physical invariants.

Implementation and Practical Considerations:

Sensor Replacement/Retraining: Since fingerprints are specific to each sensor, a question arises about what would happen to a fingerprint if a sensor is replaced? In an occasion when a sensor is replaced for whatever reason, the proposed technique does not need to create a new system model because the system dynamics are still the same. We only need to figure out the noise component being contributed by sensor's hardware. To accomplish this objective we just need to run the system and collect data and update profile only for the newly added sensor. It does not require to generate a new system model but just training data for machine learning methods.

Training Phase (Capturing System Dynamics): To be a representative system model, one should capture the whole process dynamics. For example, in the case of a water treatment testbed, a complete cycle of process is involved starting from raw water to filtration stages until we obtain clean drinking water.

6 RELATED WORK

Device fingerprinting is not a new idea on its own but creating new fingerprints for devices in a CPS is less explored. Device fingerprinting for CPS-devices poses unique challenges due to different technologies as compared to IT infrastructures and also encounters different threat models. Previous research efforts focused on threats including privacy compromise or tracking of a certain device. In this work, it is proposed to authenticate the sensors and also to detect attacks in a CPS setting. This work is a continuation of our earlier proposal [5] to create unique fingerprints for sensing devices in a CPS and demonstrating their effectiveness against a range of strong attack scenarios. A summary of the related work is as follows.

Device Fingerprinting: Camera identification based on a CMOS sensor is presented in [25] and it became an inspiration for our work. In [25], a reference noise pattern for each camera is extracted and

later used for camera attribution. A remote device identification based on microscopic deviations in the device's clock [29, 34] is presented in [21]. In [37] network traffic analysis is carried out to fingerprint the device and device-type based on wireless network traffic. In [12], fingerprints for RFID smart cards are created by analyzing the modulation scheme and features derived from that analysis.

CPS Device Fingerprinting and Attack Detection: Among attack detection techniques in the context of CPS, a few of the related works have used the same testbed (SWaT) for experiments. We have used the same testbed and the same dataset as presented in [1, 10]. Both of those techniques use physical invariants to detect attacks. The proposed technique is different from those as we only consider device/sensor characteristics and do not necessarily care about the whole system state. Also, our proposed technique does not need source code or the control system design of a process plant and it does not need to come up with invariants which is a tedious procedure on its own. Another related work is [23] in which authors monitor the variations in process variables to detect attacks. It is related to our work considering that both study process dynamics but a direct comparison of the performance of detection cannot be made because of the principle of detection techniques as well as executed attacks. Authors in [23] used an information theoretic approach (Shannon entropy on sensor data) as a detection method and it is not clear if the proposed method would be effective for the stealthy attacks considered in our work. There are a few research works on the fingerprinting of CPS devices. One approach [15] uses network traffic analysis and physical operation timings of a device. Experiments are performed on 2 latching relays. However, that approach cannot be applied to sensing devices studied in our work because there is no mechanical motion of the components as was the case for electric relays in [15]. Another study in the context of CPS device fingerprinting is carried out on SWaT testbed based on Received Signal Strength (RSS) fingerprints of wireless access points connected to PLCs [35]. A related work proposed [30] to fingerprint sensors based on their noise. However, the method works only in specific states, for example, if the water in the tank has a constant level. To extract sensor noise for certain sensors (e.g. level sensors), one needs to wait for the process to be static. However, the process is not static most of the times and thus introduces another source of noise termed the process noise. The novelty in our proposed approach is that it does not depend on the specific state of a system and that it uses dynamics of the process to create a system model. It combines sensor and process noise to create a fingerprint which is a novel idea.

7 CONCLUSIONS

A technique to fingerprint the sensor and process noise is presented. It is shown that such a fingerprint can uniquely identify the sensor by looking at the sensor measurements passively. The upper bounds for state deviation under a stealthy attack are derived. Results have shown that sensors can be identified with as high an accuracy of 98% by using the noise fingerprint. A multitude of attacks on sensor measurements are detected with a high true positive and true negative rate. A security argument against *stealthy* attacks is provided. It is shown that the proposed technique is able to detect a strong adversary. *Future Work:* The goal is to achieve higher accuracy for attack detection and a very low false alarm rate. An idea is to come up with heuristics such as a voting system among physically coupled sensors and to improve the system model by using a bank of observers scheme.

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.

REFERENCES

[1] 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 (ASIACCS '16)*. ACM, New York, NY, USA, 449–460. https://doi.org/10.1145/2897845.2897855

[2] 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*.

[3] Chuadhry Mujeeb Ahmed and Aditya 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

[4] 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 (ASIACCS '17)*. ACM, New York, NY, USA, 101–113. https://doi.org/10.1145/3052973.3053011

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

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

[7] 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.

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

[9] 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.

[10] Yuqi Chen, Christopher M. Poskitt, and Jun Sun. 2018. Learning from Mutants: Using Code Mutation to Learn and Monitor Invariants of a Cyber-Physical System. *IEEE Security and Privacy 2018* abs/1801.00903 (2018). arXiv:1801.00903 http://arxiv.org/abs/1801.00903

[11] 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.

[12] 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

[13] Sanorita. Dey, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury, and Srihari Nelakuditi. 2014. Accelprint: Imperfections of accelerometers make smartphones trackable. In *Network and Distributed System Security Symposium (NDSS)*.

[14] 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.

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

[16] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2017. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *Critical Information Infrastructures Security*, Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos, and Stephen Wolthusen (Eds.). Springer International Publishing, Cham, 88–99.

[17] 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
[18] Charles M. Grinstead. [n. d.]. *Introduction to Probability*. Swarthmore College J. Laurie Snell Dartmouth College. http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/amsbook.mac.pdf
[19] Roger A. Horn and Charles R. Johnson. 2012. *Matrix Analysis* (2nd ed.). Cambridge University Press, New York, NY, USA.
[20] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. 2017. Cyber-Physical Systems Security - A Survey. *CoRR* abs/1701.04525 (2017). arXiv:1701.04525 http://arxiv.org/abs/1701.04525
[21] Tadayoshi Kohno, Andre Broido, and KC 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
[22] Marina Krotofil, Alvaro A. Cárdenas, Bradley Manning, and Jason Larsen. 2014. CPS: Driving Cyber-physical Systems to Unsafe Operating Conditions by Timing DoS Attacks on Sensor Signals. In *Proceedings of the 30th Annual Computer Security Applications Conference (ACSAC '14)*. ACM, New York, NY, USA, 146–155. https://doi.org/10.1145/2664243.2664290
[23] Marina Krotofil, Jason Larsen, and Dieter Gollmann. 2015. The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '15)*. ACM, New York, NY, USA, 133–144. https://doi.org/10.1145/2714576.2714599
[24] Edward 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
[25] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. 2006. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security* 1, 2 (2006).
[26] Aditya P. Mathur and Nils 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
[27] Robert Mitchell and Ing-Ray Chen. 2014. A Survey of Intrusion Detection Techniques for Cyber-physical Systems. *ACM Comput. Surv.* 46, 4, Article 55 (March 2014), 29 pages. https://doi.org/10.1145/2542049
[28] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. 2015. Physical Authentication of Control Systems: Designing Watermarked Control Inputs to Detect Counterfeit Sensor Outputs. *IEEE Control Systems* 35, 1 (Feb 2015), 93–109. https://doi.org/10.1109/MCS.2014.2364724
[29] Sue B. Moon, Paul Skelly, and Don 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
[30] Mujeeb Ahmed, Aditya Mathur, and Martin Ochoa. 2017. NoiSense: Detecting Data Integrity Attacks on Sensor Measurements using Hardware based Fingerprints. *ArXiv e-prints* (Dec. 2017). arXiv:cs.CR/1712.01598
[31] Carlos Murguia and Justin 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
[32] P. Van Overschee and B. De Moor. 1996. Subspace Identification for Linear Systems: theory, implementation, applications. *Boston: Kluwer Academic Publications* (1996).
[33] Youngseok Park, Yunmok Son, Hocheol Shin, Dohyun Kim, and Yongdae Kim. 2016. This Ain't Your Dose: Sensor Spoofing Attack on Medical Infusion Pump. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*. USENIX Association, Austin, TX. https://www.usenix.org/conference/woot16/workshop-program/presentation/park
[34] 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
[35] Jay Prakash and Mujeeb 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
[36] 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*.
[37] Sakthi V. Radhakrishnan, Selcuk Uluagac, and Raheem 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
[38] Hocheol Shin, Yunmok Son, Youngseok Park, Yujin Kwon, and Yongdae Kim. 2016. Sampling Race: Bypassing Timing-based Analog Active Sensor Spoofing Detection on Analog-digital Systems. In *Proceedings of the 10th USENIX Conference on Offensive Technologies (WOOT'16)*. USENIX Association, Berkeley, CA, USA, 200–210. http://dl.acm.org/citation.cfm?id=3027019.3027037
[39] 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
[40] 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
[41] Adepu Sridhar and Mathur Aditya. 2016. Generalized Attacker and Attack Models for Cyber Physical Systems. In *40th IEEE COMPSAC*.
[42] Siddharth Sridhar, Adam Hahn, and Manimaran 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
[43] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*. 3–18. https://doi.org/10.1109/EuroSP.2017.42
[44] 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.
[45] 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
[46] Shoukry Yasser, Martin Paul, Tabuada Paulo, and Srivastava Mani. 2013. Non-invasive Spoofing Attacks for Anti-lock Braking Systems. In *CHES, Springer Link*, Vol. 8086. 55–72.

A SYSTEM STATE UNDER NORMAL OPERATION

During the normal operation of the plant, we can estimate the state of the system based on system model equations (1) and state estimation equations (3).

Proposition 1. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the state estimation error is $e_{k+1} = (A - LC)e_k + v_k - L\eta_k$.

Proof: The state estimation error is the difference between real system state and estimated system state and can be presented as,

$$e_{k+1} = x_{k+1} - \hat{x}_{k+1} \tag{25}$$

From system state equation (1) and state estimation equation (3), by substituting the equations for $x_{k+1} and \hat{x}_{k+1}$ we get,

$$e_{k+1} = Ax_k + Bu_k + v_k - A\hat{x}_k - Bu_k - L(y_k - \hat{y}_k) \tag{26}$$

For $y_k = Cx_k + \eta_k$ and $\hat{y}_k = C\hat{x}_k$ we get,

$$e_{k+1} = A(x_k - \hat{x}_k) + v_k - L(Cx_k + \eta_k - C\hat{x}_k) \tag{27}$$

As $e_k = x_k - \hat{x}_k$ we get,

$$e_{k+1} = Ae_k + v_k - LCe_k - L\eta_k \tag{28}$$

$$e_{k+1} = (A - LC)e_k + v_k - L\eta_k \tag{29}$$

This completes the proof. ■

Proposition 2. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the residual vector is $r_k = Ce_k + \eta_k$.

Proof: The residual vector is the difference between real sensor measurement and estimated sensor reading and can be presented as,

$$r_k = y_k - \hat{y}_k \tag{30}$$

From equation for system model (1) we know $y_k = Cx_k + \eta_k and \hat{y}_k = C\hat{x}_k$ then,

$$r_k = Cx_k + \eta_k - C\hat{x}_k \tag{31}$$

$$r_k = C(x_k - \hat{x}_k) + \eta_k \tag{32}$$

where we have $e_k = x_k - \hat{x}_k$,

$$r_k = Ce_k + \eta_k \tag{33}$$

This completes the proof. ■

Proposition 3. Consider the process (1), the Kalman filter (3)-(5). Under the normal operation of the plant, it can be shown that the state estimation is $\hat{x}_k = \sum_{j=0}^{k-1}[A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0$.

Proof: As shown previously state estimation equation using Kalman filter gain L is given as,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(\bar{y}_k - \hat{y}_k), \tag{34}$$

where $\bar{y}(k)$ is the sensor measurement which might be under attack and can be represented in terms of attack vector δ_k,

$$\bar{y}_k = y_k + \delta_k, \tag{35}$$

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k + \delta_k - \hat{y}_k), \tag{36}$$

For the normal operation, there is no attack and $\delta_k = 0$,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k), \tag{37}$$

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(Cx_k + \eta_k - C\hat{x}_k), \tag{38}$$

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + LCe_k + L\eta_k, \tag{39}$$

Let $\hat{x}_0 = x_0$, and by iterative solution of above equation with $k = 0$, we get, $\hat{x}_1 = A\hat{x}_0 + Bu_0 + LCe_0 + L\eta_0$ and as $\hat{x}_0 = x_0$, this gives,

$$\hat{x}_1 = Ax_0 + Bu_0 + LCe_0 + L\eta_0, \tag{40}$$

For $\hat{x}_2$ we have, $\hat{x}_2 = A\hat{x}_1 + Bu_1 + LCe_1 + L\eta_1$. Substituting $\hat{x}_1$ from previous iteration, we get,

$$\hat{x}_2 = A^2x_0 + ABu_0 + ALCe_0 + AL\eta_0 + Bu_1 + LCe_1 + L\eta_1, \tag{41}$$

For $\hat{x}_3$ we have, $\hat{x}_3 = A\hat{x}_2 + Bu_2 + LCe_2 + L\eta_2$. Substituting $\hat{x}_2$ from previous iteration, we get,

$$\hat{x}_3 = A^3x_0 + A^2Bu_0 + A^2LCe_0 + A^2L\eta_0 + ABu_1 + ALCe_1 + AL\eta_1 + Bu_2 + LCe_2 + L\eta_2, \tag{42}$$

By induction we can write for the k^{th} entry as,

$$\hat{x}_k = \sum_{j=0}^{k-1}[A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0 \tag{43}$$

This completes the proof. ■

B STATE DEGRADATION PROOFS UNDER ATTACK

B.1 Under a General Attack

Following results are for the case of a general attack δ_k.

Proposition 4A. Consider the process (1), the Kalman filter (3)-(5). Under an attack δ_k on the plant, it can be shown that the state estimation error is $e_{k+1} = Ae_k + v_k - LCe_k - L\eta_k - L\delta_k$.

Proof: The proof follows on the same lines as proposition 1 by modifying $\bar{y}_k = y_k + \delta_k$. ■

Proposition 5A. Consider the process (1), the Kalman filter (3)-(5). Under an attack δ_k on the plant, it can be shown that the residual vector is $r_k = Ce_k + \eta_k + \delta_k$.

Proof: The proof follows on the same lines as proposition 2 by modifying $\bar{y}_k = y_k + \delta_k$. ■

Proposition 6A. Consider the process (1), the Kalman filter (3)-(5). Under an attack δ_k on the plant, it can be shown that the state estimation is $\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\delta_{k-1-j} + A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0$.

Proof: The proof follows on the same lines as proposition 2 by modifying $\bar{y}_k = y_k + \delta_k$. ■

B.2 Under a Stealthy Attack

Following results are for the case of a stealthy attack $\delta_k = \beta[Ce_k + \eta_k] + \alpha$.

Proposition 4. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the state estimation error is $e_{k+1} = Ae_k + v_k - (\beta+1)LCe_k - (\beta+1)L\eta_k - L\alpha$.

Proof: The proof follows from the proposition 4A by using attack vector in equation (11),

$$e_{k+1} = Ae_k + v_k - LCe_k - L\eta_k - L\delta_k \tag{44}$$

Using $\delta_k = \beta[Ce_k + \eta_k] + \alpha$ we get,

$$e_{k+1} = Ae_k + v_k - LCe_k - L\eta_k - L[\delta_k = \beta[Ce_k + \eta_k] + \alpha] \tag{45}$$

$$e_{k+1} = Ae_k + v_k - (\beta+1)LCe_k - (\beta+1)L\eta_k - L\alpha \tag{46}$$

This completes the proof. ■

Proposition 5. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the residual vector is $r_k = (\beta+1)Ce_k + (\beta+1)\eta_k + \alpha$.

Proof: The proof follows from the proposition 5A by using attack vector in equation (11),

$$r_k = Ce_k + \eta_k + \delta_k \tag{47}$$

Using $\delta_k = \beta[Ce_k + \eta_k] + \alpha$ we get,

$$r_k = Ce_k + \eta_k + \beta[Ce_k + \eta_k] + \alpha \tag{48}$$

$$r_k = (\beta+1)Ce_k + (\beta+1)\eta_k + \alpha \tag{49}$$

This completes the proof. ■

Proposition 6. Consider the process (1), the Kalman filter (3)-(5). Under the generalized attack (11) δ_k on the plant, it can be shown that the state estimation is $\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\alpha + (\beta+1)A^jL\eta_{k-1-j} + (\beta+1)A^jLCe_{k-1-j} + A^jBu_{k-1-j}] + A^kx_0$.

Proof: The proof follows from the proposition 6A by using attack vector in equation (11),

$$\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\delta_{k-1-j} + A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0 \tag{50}$$

Using $\delta_k = \beta[Ce_k + \eta_k] + \alpha$ we get,

$$\hat{x}_k = \sum_{j=0}^{k-1}[A^jL[\beta[Ce_k + \eta_k] + \alpha] + A^jLCe_{k-1-j} + A^jBu_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0 \tag{51}$$

$$\hat{x}_k = \sum_{j=0}^{k-1}[\beta A^jLCe_{k-1-j} + A^jBu_{k-1-j} + \beta A^jL\eta_{k-1-j} + A^jL\alpha + A^jLCe_{k-1-j} + A^jL\eta_{k-1-j}] + A^kx_0 \tag{52}$$

$$\hat{x}_k = \sum_{j=0}^{k-1}[A^jL\alpha + (\beta+1)A^jL\eta_{k-1-j} + (\beta+1)A^jLCe_{k-1-j} + A^jBu_{k-1-j}] + A^kx_0 \tag{53}$$

This completes the proof. ■

Table 6: Executed Attacks on SWaT Testbed from reference [16]

Attack Sequence Number	Start Time	End Time	Attack Point	Start State	Attack	Expected Impact or Attacker Intent	OC-SVM Detection Accuracy
3	28/12/2015 11:22:00	11:28:22	LIT-101	Water level between L and H	Increase by 1 mm every second	Tank Underflow; Damage P-101	100%
7	28/12/2015 12:08:25	12:15:33	LIT-301	Water level between L and H	Water level increased above HH	Stop of inflow; Tank underflow; Damage P-301	100%
8	28/12/2015 13:10:10	13:26:13	DPIT-301	Value of DPIT is <40kpa	Set value of DPIT as >40kpa	Backwash process is started again and again; Normal operation stops; Decrease in water level of tank 401. Increase in water level of tank 301	100%
10	28/12/2015 14:16:20	14:19:00	FIT-401	Value of FIT-401 above 1	Set value of FIT-401 as <0.7	UV shutdown; P-501 turns off; UV did not shutdown; P-501 did not turn off	100%
11	28/12/2015 14:19:00	14:28:20	FIT-401	Value of FIT-401 above 1	Set value of FIT-401 as 0	UV shutdown; P-501 turns off	100%
16	29/12/2015 11:57:25	12:02:00	LIT-301	Water level between L and H	Decrease water level by 1mm each second	Tank Overflow	100%
21	29/12/2015 18:30:00	18:42:00	MV-101, LIT-101	MV-101 is open; LIT-101 between L and H	Keep MV-101 on continuously; Value of LIT-101 set as 700 mm	Tank overflow	100%
25	30/12/2015 10:01:50	10:12:01	LIT-401, P-401	Value of LIT-401 <1000; P-402 is on	Set value of LIT-401 as 1000; P402 is kept on	Tank underflow	100%
26	30/12/2015 17:04:56	17:29:00	P-101, LIT-301	P-101 is off; P-102 is on; LIT-301 is between L and H	P-101 is turned on continuously; Set value of LIT-301 as 801 mm	Tank 101 underflow; Tank 301 overflow	66.66%
27	31/12/2015 01:17:08	01:45:18	P-302, LIT-401	P302 is on, LIT401 Is between L and H	Keep P-302 on continuously; Value of LIT401 set as 600 mm till 1:26:01	Tank overflow	37.5%
30	31/12/2015 15:47:40	16:07:10	LIT-101, P-101, MV-201	P-101 is off; MV-101 is off; MV-201 is off; LIT-101 is between L and H; LIT-301 is between L and H	Turn P-101 on continuously; Turn MV-101 on continuously; Set value of LIT-101 as 700 mm; P-102 started itself because LIT301 level became low	Tank 101 underflow; Tank 301 overflow	100%
31	31/12/2015 22:05:34	22:11:40	LIT-401	Water level between L and H	Set LIT-401 to less than L	Tank overflow	100%
32	1/01/2016 10:36:00	10:46:00	LIT-301	Water level between L and H	Set LIT-301 to above HH	Tank underflow; Damage P-302	100%
33	1/01/2016 14:21:12	14:28:35	LIT-101	Water level between L and H	Set LIT-101 to above H	Tank underflow; Damage P-101	100%
36	1/01/2016 22:16:01	22:25:00	LIT-101	Water level between L and H	Set LIT-101 to less than LL	Tank overflow	100%
39	2/01/2015 11:43:48	11:50:28	FIT-401, AIT-502	In Normal Range	Set value of FIT-401 as 0.5; Set value of AIT-502 as 140 mV	UV will shut down and water will go to RO UV did not shutdown	100%
40	2/01/2015 11:51:42	11:56:38	FIT-401	In Normal Range	Set value of FIT-401 as 0	UV will shut down and water will go to RO P-402 did not close, both should be interlinked	100%
41	2/01/2015 13:13:02	13:40:56	LIT-301	Water level between L and H	decrease value by 0.5 mm per second	Tank overflow Rate of decrease in water level reduced after 1:33:25 PM	100%