Bank Of Models: Sensor Attack Detection and Isolation in Industrial Control Systems

Chuadhry Mujeeb Ahmed[1] and Jianying Zhou[2]

[1] Computer and Information Sciences, University of Strathclyde, Glasgow
[2] Singapore University of Technology and Design

Abstract. Attacks on sensor measurements can take the system to an unwanted state. The disadvantage of using a system model-based approach for attack detection is that it could not isolate which sensor was under attack. For example, if one of two sensors that are physically coupled is under attack, the attack would reflect in both. In this work, we propose an attack detection and isolation technique using a multi-model framework named *Bank of Models* (BoM) in which the same process will be represented by multiple system models. This technique can achieve higher accuracy for attack detection with low false alarm rates. We make extensive empirical performance evaluation on a realistic ICS testbed to demonstrate the viability of this technique.

Keywords: Sensor Security· Attack Detection · Industrial Control Systems.

1 Introduction

An Industrial Control System (ICS) is a combination of computing elements and physical phenomenon [35]. In particular, we will consider examples of a water treatment plant in this paper. 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 advances in communication technologies resulted in the widespread of such systems to better monitor and operate ICS, but this connectivity also exposes physical processes to malicious entities on the cyber domain [6, 28]. Recent incidents of sabotage on these systems [7, 13], have raised concerns on the security of ICS.

Challenges in ICS security are different as compared with conventional IT systems, especially in terms of consequences in case of a security lapse. Attacks on ICS might result in damage to the physical property [10, 46] or severely affecting people who depend on critical infrastructure as was the case of the recent power cutoff in Ukraine [7]. Data integrity is an important security requirement for ICS [20] therefore, the integrity of sensor data should be ensured. Sensor data can either be spoofed in cyber (digital) domain [42] or in physical (analog) domain [40]. Sensors are a bridge between the physical and cyber domains

This is a peer-reviewed, accepted author manuscript of the following article: Ahmed, C. M., & Zhou, J. (Accepted/In press). Bank of models: sensor attack detection and isolation in industrial control systems. Paper presented at The 16th International Conference on Critical Information Infrastructures Security, Lausanne, Switzerland.

in an ICS. 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 the physical/analog domain may go undetected by IDS based only on network traffic [40]. Recently, a live-fire cyber attack-defense exercise on ICS, evaluated commercially available network layer attack detection products with the process-aware research prototypes and concluded that the network-only products do not succeed in detecting process layer attacks [23].

Data integrity attacks on sensor measurement and the impact of such attacks have been studied in theory, including false data injection [33], replay attacks [32], DoS attacks [26] and stealthy attacks [11]. These previous studies proposed attack detection methods based on the system model and statistical fault detectors [3, 1] and also point out the limitations of such fault detectors against an adversarial manipulation of the sensor data. A major limitation of these model based attack detection methods, is that it is difficult to isolate the attacks.

The Attack Isolation Problem: The attack isolation problem also known as determining the source of an attack is important in the context of ICS [47]. Anomaly detection research suffers from this issue, especially methods rooted in machine learning [41, 2]. Using machine learning methods with the available data might be able to raise an alarm but are not able to find the source of the anomaly. In the context of ICS, if a model is created for the whole process it is not clear where does an anomaly is coming from?

Motivating Example: To understand the idea, we need to consider an example from the SWaT testbed [31]. SWaT is a six-stage water treatment plant. Figure 1 shows stage 1 of the SWaT testbed. This stage holds the raw water that is to be processed. The central entity is a water storage tank with a level sensor (LIT-101). For this example, we shall focus on another sensor that is the flow sensor (FIT-201) at the outlet of the tank to measure the outflow of the water from the tank. LIT-101 and FIT-101 are coupled physically, meaning when FIT shows outflow level shall go down in LIT-101. In the following, this explanation will help us understand the problem.

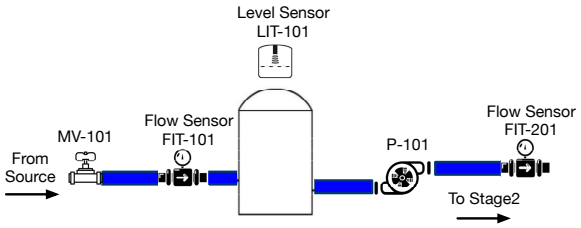


Fig. 1: Stage 1 of the SWaT Testbed.

Figures 2 and 3 show an example of such a problem in a real water treatment process in which both sensor measurements and estimates are obtained through process models. Figure 2 depicts a flow meter at the outlet of the raw water storage tank labeled as FIT-201. A joint physical system model for the stage 1 is created using a Kalman filter (more details on this in Section 2). Such a system model captures the dynamics of the physical process. In our case, the physical

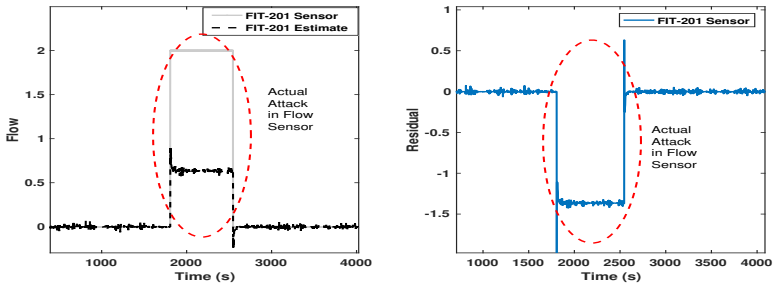


Fig. 2: Flow sensor, FIT-201 is under attack. Flow is simulated to be $2m^3/hr$ while in reality it is zero. From the residual signal it could be detected. However, we will see this attack affecting system model for LIT-101 as well in Figure 3.

process is an example of a water storage tank, which collects a limited amount of water to be used by the subsequent stages of the water treatment testbed. It is intuitive to understand that there is a physical relationship between the physical quantities, for example, consider that when water flows out of the tank through the outlet pipe then the level of the water should fall in the tank. Hence, water level sensor LIT-101 and outlet flow sensor FIT-201 are physically coupled with each other. In the example attack, an attacker spoofs the flow sensor FIT-201 by spoofing the real sensor measurements of zero flow to $2m^3/hr$ volumetric flow level. In the left-hand part of the Figure 2, sensor measurements and estimates are shown before, during and after the attack. The difference between the sensor measurements and estimates is given as residual on the right-hand plot. It can be seen that the attack would be detected using a model-based detector [3] on FIT-201 residual. However, from Figure 3 it can be seen that the same attack is detected using the detector for the LIT-101 sensor. For the Figure 3 it could be seen that using the system model the estimate for the level tends to decrease, for the reason that if there is outflow the level should be decreased, but since there is an attack going on, it could be seen that the estimate deviates from the real sensor measurements. The model-based detectors defined for both level sensor and flow sensor would raise an alarm. It is not possible to figure out where is the actual attack unless manually checked. The problem of attack isolation is important considering the scale and complexity of an ICS. Attack detection and isolating the devices that are under attack is critical for response and recovery.

Proposed Solution: We propose a multi-model framework named Bank of Models (BoM) to detect and isolate attacks on the sensors in an ICS. The proposed attack detection framework improves on the limitations of model-based attack detection schemes[42, 4]. BoM uses the estimates for each sensor obtained from the multiple system models. It then creates a profile for each sensor based on a set of time domain and frequency domain features that are extracted from the residual vector (difference between sensor measurement and sensor estimate). A one-class Support Vector Machine (SVM) is used to detect attacks for a multitude of industrial sensors. Experiments are performed on an operational water treatment facility accessible for research [31]. A class of attacks as explained in

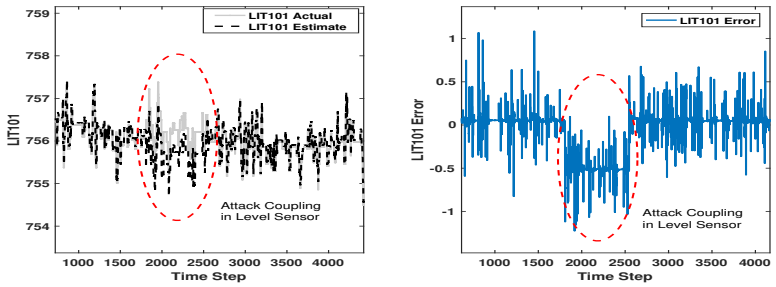


Fig. 3: The attack in flow sensor seen in Figure 2, can be observed in LIT-101 as well. This is precisely the attack isolation problem, meaning an attack originating in one component can appear in multiple devices.

the threat model are launched on a real water treatment testbed. The major contributions of this work are thus:

- A novel BoM framework to detect and isolate sensor attacks.
- A detailed evaluation of the proposed technique as an attack detection method, for a class of sensor spoofing attacks.
- Extensive empirical performance evaluation on a realistic ICS testbed.
- An ensemble of models based algorithm to increase the attack detection rate and reduction in the false alarm rate.

2 System and Threat Model

2.1 System Dynamics

A system model represents the system dynamics in a mathematical form. 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. We used data collected under regular operation (no attacks) and subspace system identification techniques [36] to obtain a system model. For SWaT testbed, resulting system model is a 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 $k \in \mathbb{N}$ 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 $u \in \mathbb{R}^p$ denote the control actions. A, B, C are the state space matrices, capturing the system dynamics. η_k is the sensor measurement noise and v_k is the process noise.

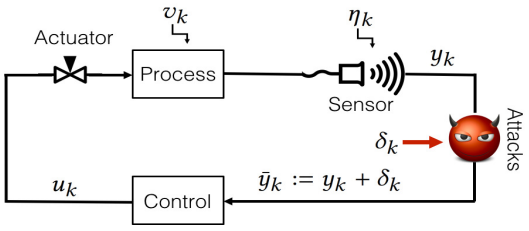


Fig. 4: A general CPS under sensor attacks.

2.2 Threat Model

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 4. 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 compromise the system. For instance, through the *Man-in-The-Middle (MiTM)* attack 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 [42] or a replacement of on board PLC software [17]. 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.
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. All the attacks taken from reference work [18] are executed by compromising the Supervisory Control and Data Acquisition (SCADA) system. An attack toolbox was used to inject an arbitrary value for real sensor measurement.

3 Attack Detection and Isolation

The Problem of State Estimation: State estimation is to estimate the physical state variable of a system given the previous state measurement. A general state estimation problem can be formulated as,

$$\hat{X}_{k+1} = A\hat{X}_k + BU_k + L\big(\bar{Y}_k - \hat{Y}_k\big), \tag{3}$$

Equation (3) presents a general estimator design, where L is a gain matrix calculated to minimize the estimation error. $\hat{Y}$ and $\hat{X}$ are estimated system

output and system state, respectively. Let's consider an example of a system model with two outputs and one control input and equation (3) becomes,

$$\begin{bmatrix} \hat{x}^1_{k+1} \\ \hat{x}^2_{k+1} \end{bmatrix} = \begin{bmatrix} a_{11}\ a_{12} \\ a_{21}\ a_{22} \end{bmatrix} \begin{bmatrix} \hat{x}^1_k \\ \hat{x}^2_k \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} U_k + \begin{bmatrix} l_{11}\ l_{12} \\ l_{21}\ l_{22} \end{bmatrix} \begin{bmatrix} e(y^1_k) \\ e(y^2_k) \end{bmatrix} \tag{4}$$

$$\begin{matrix} \hat{x}^1_{k+1} \\ \hat{x}^2_{k+1} \end{matrix} = \begin{matrix} a_{11}\hat{x}^1_k + a_{12}\hat{x}^2_k + b_{11}u_k + l_{11}e(y^1_k) + l_{12}e(y^2_k) \\ a_{21}\hat{x}^1_k + a_{22}\hat{x}^2_k + b_{21}u_k + l_{21}e(y^1_k) + l_{22}e(y^2_k) \end{matrix} \tag{5}$$

The two system state estimates are labeled as $\hat{x}^1_k$ and $\hat{x}^2_k$. It can be observed from (5) that the state estimate $\hat{x}^1_{k+1}$ at $k+1^{th}$ time instance depends on error from both the outputs, i.e., $e(y^1_k)$ and $e(y^2_k)$ since the estimator is designed for both the sensors as a joint model. In this study, we have used Kalman filter to estimate the state of the system based on the available output y_k,

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L_k\big(\bar{y}_k - C\hat{x}_k\big), \tag{6}$$

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) and the estimator (6), the estimation error is governed by the following difference equation

$$e_{k+1} = \big(A - L_k C\big)e_k - L_k\eta_k - L_k\delta_k + v_k. \tag{7}$$

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 [5]. 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 (6) 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

$$L_k = L := \big(APC^T\big)\big(R_2 + CPC^T\big)^{-1}, \tag{8}$$

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

$$APA^T - P + R_1 = APC^T(R_2 + CPC^T)^{-1}CPA^T. \tag{9}$$

The reconstruction method given by (6)-(9) is referred to as the steady state Kalman Filter, cf. [5].

3.1 Attack Detection Framework

In this section, we explain the details of the proposed attack detection scheme. First, we use the Kalman filter based state estimation to generate residual (difference between sensor measurement and estimate). Then, we present the design of our residual-based attack detection method.

Proposition 1. In steady state [5], residual vector is a function of sensor and process noise. Consider the process (1), the Kalman filter (6)-(9). The residual vector is given as, $r_k = Ce_k + \eta_k$ and $e_k = \sum_{i=0}^{k-2} (A - LC)^i (v_{k-i-1} - L\eta_{k-i-1})$, where $v_k \in \mathbb{R}^n$ is the process noise and $\eta_k \in \mathbb{R}^m$ is the sensor noise.

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

$$e_{k+1} = x_{k+1} - \hat{x}_{k+1} \tag{10}$$

From system state equation (1) and state estimation equation (6), by substituting the equations for x_{k+1} and $\hat{x}_{k+1}$ we get,

$$e_{k+1} = Ax_k + Bu_k + v_k - A\hat{x}_k - Bu_k - L(y_k - \hat{y}_k) \tag{11}$$

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

$$e_{k+1} = A(x_k - \hat{x}_k) + v_k - L(Cx_k + \eta_k - C\hat{x}_k) \tag{12}$$

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

$$e_{k+1} = Ae_k + v_k - LCe_k - L\eta_k \tag{13}$$

$$e_{k+1} = (A - LC)e_k + v_k - L\eta_k \tag{14}$$

■

Using system model and system state estimates it is possible to extract the residual as defined above. Once we have obtained these residual vectors capturing the modeled behaviour of the given ICS, we can proceed with pattern recognition techniques (e.g. machine learning) to detect anomalies.

Design of the Proposed Framework 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 under normal operation.

***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 testbed. The objective of residual collection step is to extract a set of features 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. Each sensor is profiled using variance and other statistical features in the residual vector as shown in the Table 1. A machine learning algorithm is used to profile sensors from fresh readings (test-data).

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	$y_m(0)$

***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 [45] 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. We have performed experiments on a dataset collected over 7 days in SWaT 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 $< F(C_i) >_{i \epsilon [1,m]}$. We have used a one-class SVM algorithm for attack detection. It is found out empirically that a sample size of 120 readings, i.e., m = 120 gave the best results. Most of the machine learning algorithms need a chunk of data to operate on and it is common to find an appropriate chunk size through experimentation [29, 9].

***Size of Training and Testing Dataset*:** For a total of FS feature sets for each sensor, at first we used half ($\frac{FS}{2}$) for training and half ($\frac{FS}{2}$) 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 $\frac{FS}{2}$. Classification is then carried out for the following corresponding range of feature sets for Training : $\{\frac{FS}{2}, \frac{FS}{3}, \frac{FS}{4}, \frac{FS}{5}, \frac{FS}{10}\}$, and for Testing : $\{\frac{FS}{2}, \frac{2FS}{3}, \frac{3FS}{4}, \frac{4FS}{5}, \frac{9FS}{10}\}$, respectively. For the classifier we have used a one-class SVM library [8] and it turns out that the amount of data does not affect the performance. Moreover since we are not using supervised learning for attack detection, therefore, training is only done on the normal data obtained from a particular sensor.

3.2 Attack Isolation

A well known idea in fault isolation literature is to use multiple observers [44, 12, 43]. Consider the dynamic system as expressed by (1) with p outputs,

$$y_k = [y_k^1, y_k^2, ..., y_k^p]^T = Cx_k \tag{15}$$

For the case of an attack on one sensor i, attack vector $\delta_k^i \neq 0$ and $y_k^i = C_i\hat{x_k} + \delta_k^i$. Again consider the example of two sensors in the water tank example we have considered earlier. To use the idea of bank of observers we would drop one sensor at first and design an observer just using the first sensor, i.e., the flow sensor FIT-101 and then we will design another observer by using the second sensor, i.e., the level sensor LIT-101. Let's consider both the cases one by one:

Case 1:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L_i(y_k^i - C_i\hat{x}_k), \tag{16}$$

$$r_k = C\hat{x}_k - y_k \tag{17}$$

Using the first observer designed for FIT-101 gives the output as,

$$\begin{bmatrix}\hat{y}_{k+1}^1\\ \hat{y}_{k+1}^2\end{bmatrix} = C\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}\hat{x}_k^1\\ \hat{x}_k^2\end{bmatrix} + \begin{bmatrix}b_{11}\\ b_{21}\end{bmatrix}U + \begin{bmatrix}l_{11}\\ l_{21}\end{bmatrix}\begin{bmatrix}e(y_k^1) + \delta_k^1\\ e(y_k^1) + \delta_k^1\end{bmatrix}\right) \tag{18}$$

Case 2: Using the second observer designed for LIT-101 gives the output as,

$$\begin{bmatrix}\hat{y}_{k+1}^1\\ \hat{y}_{k+1}^2\end{bmatrix} = C\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}\hat{x}_k^1\\ \hat{x}_k^2\end{bmatrix} + \begin{bmatrix}b_{11}\\ b_{21}\end{bmatrix}U + \begin{bmatrix}l_{21}\\ l_{22}\end{bmatrix}\begin{bmatrix}e(y_k^2) + \delta_k^2\\ e(y_k^2) + \delta_k^2\end{bmatrix}\right) \tag{19}$$

Where δ_k^1 and δ_k^2 are the attack vectors in sensor 1 and sensor 2 respectively. To isolate the attack using a bank of observers, following conditions are considered for p sensors,

> *Condition 1:* if $r_k^j \neq 0$ for one $j \in \{1, 2, ..., i-1, i+1, ..., p\}$, then sensor j is under attack, while sensor i is the one used to design an observer.
>
> *Condition 2:* if $r_k^j \neq 0$ for all $j \in \{1, 2, ..., i-1, i+1, ..., p\}$ then sensor i is under attack while sensor i is used to design the observer.

For a simple example, let's consider two observers as designed in (18) and (19). In the first case we had used FIT-101 sensor measurements to design an observer and also keep in mind that FIT-101 was free of any attacks. This means according to the condition 1 above FIT-101 residual mean should go to zero but for LIT-101, it does not. Figure 5 shows the results for the case 1. It can be seen that the sensor 1 (FIT-101) residual does not deviate form the normal residual, while the sensor 2 (LIT-101) residual deviates from the normal operation, hence detecting and isolating the source of attack. For the case 2, the observer is designed using the sensor 2 (LIT-101) and also remember that the attack is also

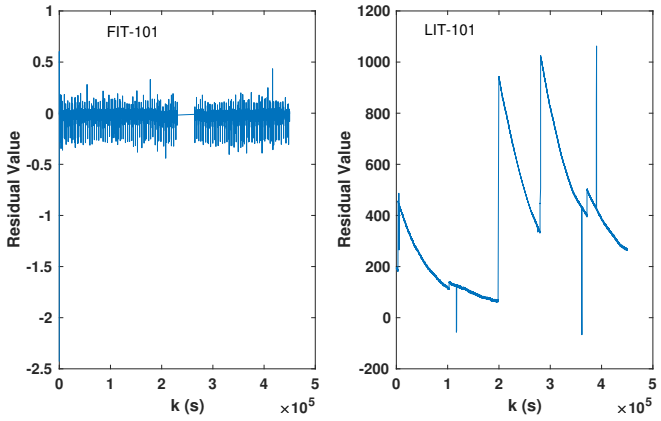


Fig. 5: Sensor 1 FIT-101 is used for observer design but the attack was in sensor 2 LIT-101. Therefore attack can be isolated in residual of LIT-101.

present in the LIT-101. Figure 6 shows the results for this case. This case satisfies the condition 2 as stated above and then we see that the attack is present in both the sensors as the observer used is the one which has the attack. This means δ_k^1 was 0 and δ_k^2 was not zero in (18) and (19) respectively.

However from the results above it could be noticed that the sensor attacks could be isolated using the idea of bank of observers but it would not detect the case when the attack is in multiple sensors at the same time, e.g., multi-point single-stage attacks in an ICS [18]. Towards this end we are proposing the idea of using a Bank of Models (BoM) to isolate and detect attacks on multiple sensors at the same time in an ICS.

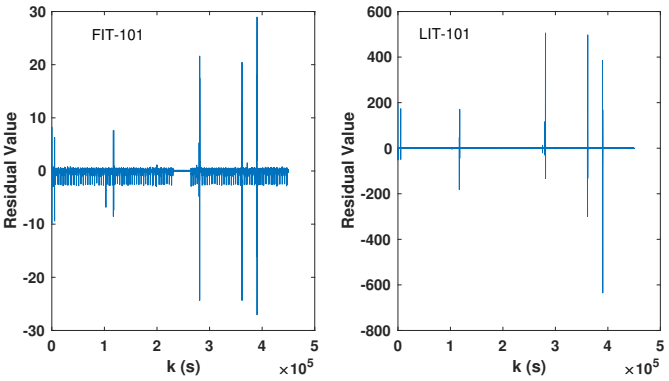


Fig. 6: Sensor 2 LIT-101 used for observer design and the attack was also in sensor 2 LIT-101. Therefore, both the sensor residuals deviate from the normal pattern.

Bank of Models (BoM): The idea is to create multiple models of the physical process rather than the multiple observers. For example if you have two sensors which are physically coupled as in the case of FIT-101 and LIT-101, then we

Algorithm 1: Attack Isolation Method

Result: Output the sensor ID under attack
initialization;
θ_s: {Set of Sensors} ;
$r^i_{joint} = 0$, r^i_{BoM}=0, i $\in \theta_s$;
$sensor^i_m.Attack = False$ #Flag i^{th} sensor Attack;
while *Sensor Signal* **do**
  for i *in* θ_s **do**
    $r^i_{joint} = y^i_{joint} - \hat{y}^i_{joint}$, r^i_{BoM}=$y^i_{BoM} - \hat{y}^i_{BoM}$;
    if $r^i_{BoM}.Attack == True$ && $r^i_{Joint}.Attack == True$ **then**
      $Sensor^i.Attack = True$;
    else
      $Sensor^i.Attack = False$;
    end
  end
end

will create three models, 1) with both the sensors as output, 2) with FIT-101 only as the output and 3) with LIT-101 only as the output. We call the first method as *Joint* model and the rest two models as *BoM*. We can use these models in conjunction with each other to isolate the attacks and call that model as *Ensemble* of models. By having a separate model the sensors are no longer coupled to each other. These separate models could be used to detect attacks but accuracy of detection might be low as we will see in the results. Therefore, we propose a method called Ensemble of models combining the joint and separate models to make an attack detection decision as well as isolate the attack.

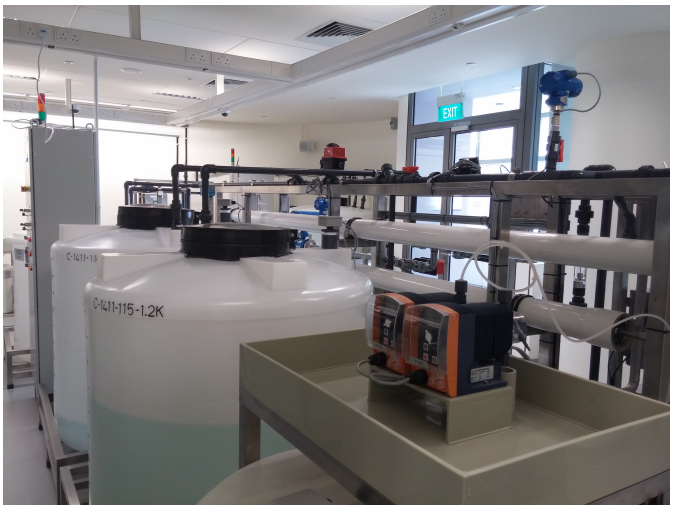


Fig. 7: Experiments conducted at SWaT Testbed.

4 Evaluation

4.1 Experimentation Setup

Industrial Control Systems have a broad domain. In this work, a Secure Water Treatment testbed (SWaT) [31] shown in Figure 7 at the Singapore University of Technology and Design is used as a case study. SWaT is a fully functional testbed and is open for researchers to use. A brief introduction is provided in the following to understand the context of the problem.

The SWaT testbed produces the purified water and it is a scaled-down version of a real water treatment process. There are six stages in the SWaT testbed. Each stage is equipped with a set of sensors and actuators. Sensors include water quantity measures such as level, flow, and pressure and water quality measures such as pH, ORP and conductivity. Actuators include motorized valves and electric pumps. Stage 1 is the raw water stage to hold the raw water for the treatment and stage 2 is the chemical dosing stage to treat the water depending on the measurements from the water quality sensors. Stage 3 is the ultra-filtration stage. Stage 4 is composed of de-chlorinator and stage 5 is equipped with reverse osmosis filters. Stage 6 holds the treated water for distribution. Multiple stages from SWaT are used in this study. Actuator signals are input to the system model and sensor measurements are outputs. Level sensors labelled as LIT-s0q, where LIT stands for level transmitter, s for the stage and q for the specific number, e.g., LIT-101 means level sensor in stage1 and sensor 1. FIT-301 is the flow sensor in stage3 and sensor number 1. The performance is evaluated in three areas, namely, attack detection, attack isolation and the improvement in attack detection rate.

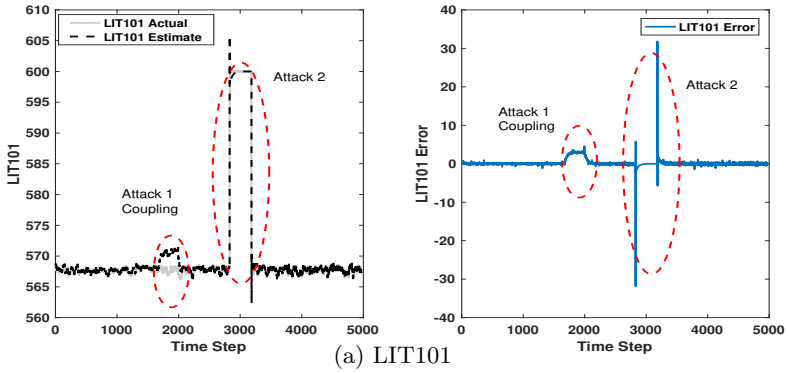
4.2 Attack detection

To show the performance of attack detector we use True Positive Rate (TPR: meaning attack data declared as attack), True Negative Rate (TNR: normal data declared as normal). Attack detection results are shown in Table 2. For each sensor in SWaT testbed attack sequences are shown. These attack sequences and attacked dataset is obtained from already published benchmark attacks [18, 9]. We can see a high TPR and TNR indicating the effectiveness of our proposed scheme. There is an interesting observation to make here, as discussed earlier the proposed technique is based on the system model, it exhibits a strong coupling between inputs and outputs of a system. If attacks are executed on level sensors we could see the effect on associated flow meter and vice versa. This indicates the coupling due to the laws of Physics even though the sensors were of different types. Column 3 and 4 indicates this result in form of TNR-Joint and TPR-Joint respectively. For LIT-101 it could be seen that the TPR is 100%, however, we observe attack detection TPR for FIT-101 to be 88.88% while there were no attacks carried on FIT-101. Column 5 and 6 depict results for the case when we have a separate system model for each sensor labeled as TNR-BoM and TPR-BoM respectively. These two single models can help in detecting attacks just in LIT-101 and none in FIT-101 as expected.

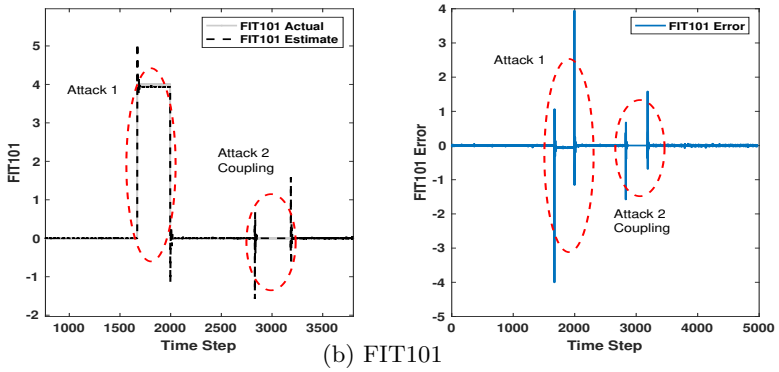
Table 2: Attack detection performance. TPR: Attack detected successfully, TNR: Normal data classified successfully. LIT: level sensor, FIT: flow sensor,DPIT: differential pressure sensor.

Sensor	Atk. seq. [a]	TNR-Joint	TPR-Joint	TNR-BoM	TPR-BoM	TNR-Ensemble	TPR-Ensemble
DPIT-301	8	84.66%	100%	83.14%	100%	88.66%	100%
LIT-101	3,21,30,33,36	83.37%	100%	94.55%	85.18%	96.50%	85.18%
FIT-101	None	91.96%	88.88	71.00%	No Attacks	96.07%	No Attacks
LIT-301	7,16,26,32,41	86.28%	78.37%	92.31%	100%	96.82%	78.37%
FIT-301	None	86.41%	83.78%	86.23%	No Attacks	89.89%	No Attacks
LIT-401	25,27,31	87.12%	74.28%	86.66%	65.21%	90.18%	60.86%
FIT-401	10,11,39,40	87.50%	51.42%	87.40%	100%	91.70%	100%

[a] Attack Sequences [18]



(a) LIT101



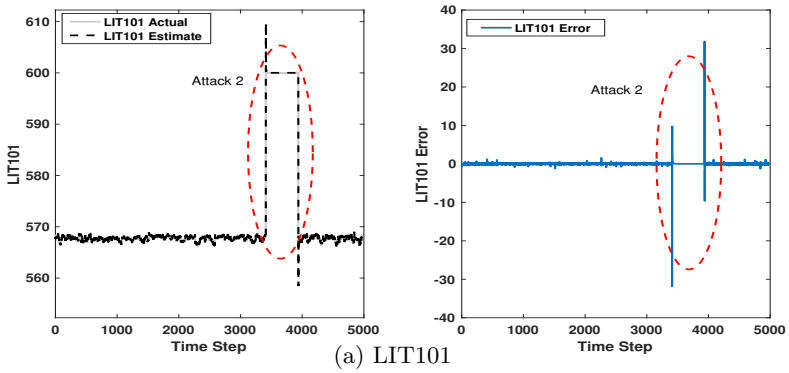
(b) FIT101

Fig. 8: This shows how two different attacks on two different sensors are reflected in residuals of both the sensors due to the physical coupling.

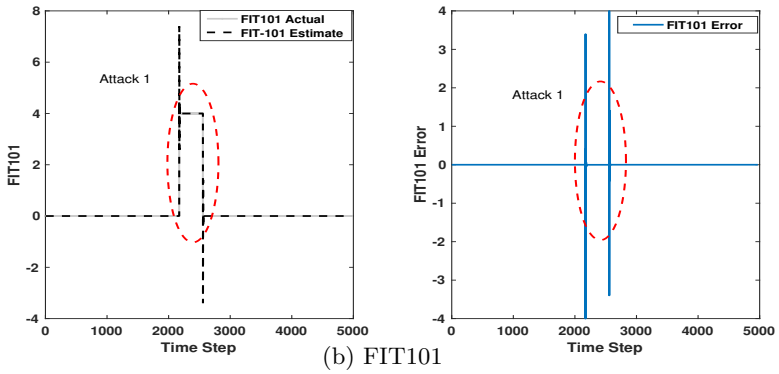
4.3 Attack Isolation

We have seen the attack isolation performance in Table 2 using the separate model for each sensor. To visually present the idea Figure 8 shows two example attacks and the coupling effects. Attack 1 is carried out on the flow meter FIT-101 by spoofing the flow value to $4m^3/hr$ as shown in Figure(b) and this attack can be observed in the residual value on the right-hand side. However, attack 1 could be seen in Figure(a) in the level sensor LIT-101 as well. The Attack 2 is carried out on the level sensor by spoofing the water level value as shown in Figure 8(a). This attack could be seen in the residual of the level sensor LIT-101 and also on the right-hand side in the flow sensor FIT-101 residual.

In Figure 9 it can be seen that separate system models for both the sensors were able to isolate both the attacks. Attack 1 only appears in the residual of FIT-101 and Attack 2 is detected only by LIT-101.



(a) LIT101



(b) FIT101

Fig. 9: In this Figure both the attacks as shown in Figure 8 are shown but for the case when we have two separate models for each sensor. It can be seen that the attacks are isolated to the particular sensor under attack.

4.4 Reduction in False Alarm Rate via Ensemble

From the results above we have seen that the bank of models (BoM) can detect as well as isolate the attacks on the sensors. However, we thought to make use of more information by combining the BoM and Joint models termed as Ensemble model here as outlined in Algorithm 1. Last two columns in Table 2 shows the results for the Ensemble of both the models. It is seen that using the information about residual vectors from two different models and obtaining an ensemble increase the information at hand and can led in reduced false alarms. Observing column TNR-Ensemble and comparing with other TNR columns reveal that the false alarm rate has significantly reduced as compared to prior results.

5 Discussion

TPR and TNR Accuracy: The reason for low TPR and TNR in some cases is that as soon as an attack is ended, we start considering the behavior/ground truth to be a normal operation. But our detection system still raises alarms and these alarms are treated as False positives. In reality, this is the time required by the system to come back to a normal operating range. More so, since we do not record that as a rightful attack detection then it is also counted as wrongful TNR reducing the TNR. For example, as shown in Figure 10 we can observe that as soon as an attack is removed, we observe post-attack effects which persist for some time. In this region, we assume the attack is over but due to attack effects, our detector keeps raising an alarm thus reducing the TNR value. From a defender's perspective, it might be acceptable since operators would be involved even from the first alarm raised. However, from an attacker's perspective this observation highlights, how important it is to terminate attacks in a way to reduce the number of alarms. This is in line with earlier works that have highlighted how important it is to time an attack [25], similarly, it is important to terminate attack as such to avoid abrupt changes.

Scalability: An important consideration is the practicality of the proposed technique for real-world plants. Since the proposed method does not use any extra hardware, scalability is not an issue. Previously in the literature bank of observers has been used for hundreds of sensors, similarly, the proposed idea of a bank of models can be used as it is grounded in the mathematical formulation and software. Practical demonstration in the real-world water treatment testbed also shows the applicability of the proposed technique to real systems without any overhead.

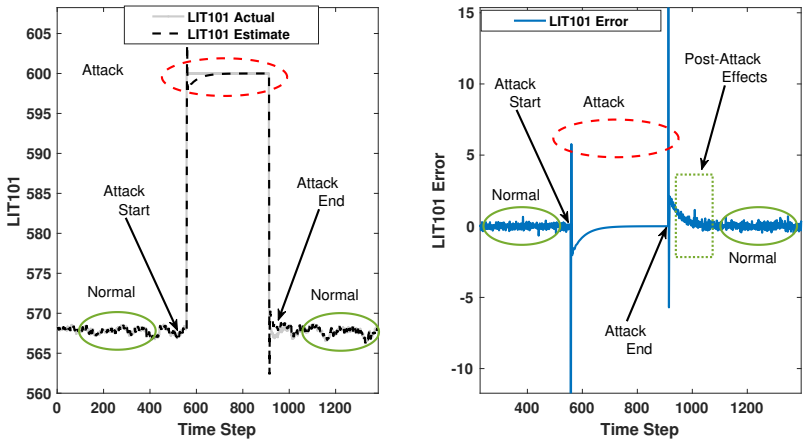


Fig. 10: Explanation for low TPR/TNR dues to post-attack effects.

6 Related Work

Machine Learning-based Approaches: There have been a lot of efforts towards attack detection in the context of ICS. Most of the works took a machine learning-based approach. Most techniques took classification approach and either used unsupervised or semi-supervised machine learning algorithms to detect attacks in an ICS [24, 19, 22, 21, 15, 16, 38]. In particular, some of them have used data from Secure Water Treatment (SWaT) testbed [31]. The design of an anomaly detector for ICS is treated as a "one-class classification problem" and several unsupervised learning methods are effectively employed [22]. Unsupervised learning approaches construct a baseline for normal behavior through feature learning and monitor whether the current behavior is within the specified range or not. Although these techniques can detect zero-day vulnerabilities, they generate high false alarms due to the existence of several hyperparameters and the multivariate nature of ICS data. Similarly, for one class SVM, authors in [22] have fine-tuned the parameters, namely c and γ for better performance on the SWaT dataset. Although there exist several automated approaches, such as grid search, randomized search, and metaheuristic optimization techniques for fine-tuning, a significant challenge faced by these techniques is overfitting. Generally, the error rate during the validation process should be less for the trained model; a higher validation error for the model trained with a large volume of data implies that the model is over-fitted. A context-aware robust intrusion detection system is proposed by [39]. Given the amount of work done in this domain this related worklist is by no means exhaustive but tried to cover the related work tested on the SWaT testbed. These techniques do not include a feature of attack isolation that is the core of this work.

System and Process Model: There have been efforts from the system and the process model perspective. Krotofil et al. [27] detect the spoofed measurements using the correlation entropy in a cluster of related sensors. Most of the work focused on system model-based approaches, the literature volume is huge and it is not possible to cover all the studies here but a few representative [14, 37, 30]. These works capture the process dynamics in the form of system models and use the point change detection methods to detect attacks in the process data. A similar recent approach [47] does isolate the attack but only the attacks on the actuators. Sensor fusion [34] and multi observer techniques[43] are recent efforts on attack isolation problem in simulated environments. However from the results in Section 3.2 it is noticed that the sensor attacks could be isolated using the idea of a bank of observers but it would not detect the case when the attack is in multiple sensors at the same time, e.g., multi-point single-stage attacks in an ICS. Our work is the first effort demonstrating a bank of models on a live water treatment plant.

7 Conclusions and Future Work

The problem of attack isolation is critical in terms of system response and recovery in an event of attack detection. We demonstrated that using the bank of models (BoM) the attack isolation problem on multiple sensors at a time could be solved. This work strengthens the previous studies and provides a novel solution to the problem of determining the source of the attack in a complex industrial control system. In future, we plan to extend this work using bigger city scale process plant. Moreover, we propose to automate the process of data-based system modelling.

References

1. Ahmed, C.M., A.Sridhar, M., A.: Limitations of state estimation based cyber attack detection schemes in industrial control systems. In: IEEE Smart City Security and Privacy Workshop, CPSWeek (2016)
2. Ahmed, C.M., M R, G.R., Mathur, A.P.: Challenges in machine learning based approaches for real-time anomaly detection in industrial control systems. In: Proceedings of the 6th ACM on Cyber-Physical System Security Workshop. p. 23–29. CPSS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3384941.3409588, https://doi.org/10.1145/3384941.3409588
3. Ahmed, C.M., Murguia, C., Ruths, J.: Model-based attack detection scheme for smart water distribution networks. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. pp. 101–113. ASIA CCS '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3052973.3053011, http://doi.acm.org/10.1145/3052973.3053011
4. Ahmed, C.M., Ochoa, M., Zhou, J., Mathur, A.P., Qadeer, R., Murguia, C., Ruths, J.: 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. pp. 483–497. ASIACCS '18,

ACM, New York, NY, USA (2018). https://doi.org/10.1145/3196494.3196532, http://doi.acm.org/10.1145/3196494.3196532
5. Aström, K.J., Wittenmark, B.: Computer-controlled Systems (3rd Ed.). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1997)
6. Cardenas, A., Amin, S., Lin, Z., Huang, Y., Huang, C., Sastry, S.: Attacks against process control systems: Risk assessment, detection, and response. In: 6th ACM Symposium on Information, Computer and Communications Security. pp. 355–366 (2011)
7. Case, D.U.: Analysis of the cyber attack on the ukrainian power grid (2016)
8. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011), software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
9. Chen, Y., Poskitt, C.M., Sun, J.: 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), http://arxiv.org/abs/1801.00903
10. CNN: Staged cyber attack reveals vulnerability in power grid. http://edition.cnn.com/2007/US/09/26/power.at.risk /index.html, year = 2007
11. Dan, G., Sandberg, H.: Stealth attacks and protection schemes for state estimators in power systems. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on. pp. 214–219. IEEE (2010)
12. Esfahani, P.M., Vrakopoulou, M., Andersson, G., Lygeros, J.: A tractable nonlinear fault detection and isolation technique with application to the cyber-physical security of power systems. In: Proceedings of the 51st IEEE Conference on Decision and Control. pp. 3433–3438 (2012)
13. Falliere, N., Murchu, L., Chien, E.: 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)
14. Fawzi, H., Tabuada, P., Diggavi, S.: Secure estimation and control for cyber-physical systems under adversarial attacks. IEEE Trans. Autom.Control **59**(6), 1454–1467 (2014)
15. Filonov, P., Kitashov, F., Lavrentyev, A.: Rnn-based early cyber-attack detection for the tennessee eastman process. arXiv preprint arXiv:1709.02232 (2017)
16. Filonov, P., Lavrentyev, A., Vorontsov, A.: Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. arXiv preprint arXiv:1612.06676 (2016)
17. Garcia, L., Brasser, F., Cintuglu, M.H., Sadeghi, A.R., Mohammed, O., Zonouz, S.A.: Hey, my malware knows physics! attacking plcs with physical model aware rootkit. In: 24th Annual Network & Distributed System Security Symposium (NDSS) (Feb 2017)
18. Goh, J., Adepu, S., Junejo, K.N., Mathur, A.: A dataset to support research in the design of secure water treatment systems. In: Havarneanu, G., Setola, R., Nassopoulos, H., Wolthusen, S. (eds.) Critical Information Infrastructures Security. pp. 88–99. Springer International Publishing, Cham (2017)
19. Goh, J., Adepu, S., Tan, M., Lee, Z.S.: Anomaly detection in cyber physical systems using recurrent neural networks. In: 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). pp. 140–145. IEEE (2017)
20. Gollmann, D., Krotofil, M.: Cyber-Physical Systems Security, pp. 195–204. Springer Berlin Heidelberg, Berlin, Heidelberg (2016).

https://doi.org/10.1007/978-3-662-49301-4_14, https://doi.org/10.1007/978-3-662-49301-4_14
21. Huda, S., Yearwood, J., Hassan, M.M., Almogren, A.: Securing the operations in scada-iot platform based industrial control system using ensemble of deep belief networks. Applied soft computing **71**, 66–77 (2018)
22. Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C.M., Sun, J.: Anomaly detection for a water treatment system using unsupervised machine learning. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 1058–1065. IEEE (2017)
23. iTrust: Sutd security showdown: Live fire cyber exercise. https://itrust.sutd.edu.sg/scy-phy-systems-week/2017-2/s317-event/, year = 2017
24. Kravchik, M., Shabtai, A.: Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. pp. 72–83. ACM (2018)
25. Krotofil, M., Cárdenas, A.A.: Is this a good time? deciding when to launch attacks on process control systems. In: Proceedings of the 3rd International Conference on High Confidence Networked Systems. p. 65–66. HiCoNS '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2566468.2576852, https://doi.org/10.1145/2566468.2576852
26. Krotofil, M., Cárdenas, A.A., Manning, B., Larsen, J.: 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. p. 146–155. ACSAC '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2664243.2664290, https://doi.org/10.1145/2664243.2664290
27. Krotofil, M., Larsen, J., Gollmann, D.: The process matters: Ensuring data veracity in cyber-physical systems. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. pp. 133–144. ASIA CCS '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2714576.2714599, http://doi.acm.org/10.1145/2714576.2714599
28. Krotofil, M., Gollmann, D.: Industrial control systems security: What is happening? In: 2013 11th IEEE International Conference on Industrial Informatics (INDIN). pp. 670–675 (2013). https://doi.org/10.1109/INDIN.2013.6622964
29. Li, X., Ye, N.: Decision tree classifiers for computer intrusion detection. Journal of Parallel and Distributed Computing Practices **4**(2), 179–190 (2001)
30. Liu, Y., Ning, P., Reiter, M.: False data injection attacks against state estimation in electric power grids. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. pp. 21–32 (2009)
31. Mathur, A.P., Tippenhauer, N.O.: 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). pp. 31–36 (April 2016). https://doi.org/10.1109/CySWater.2016.7469060
32. Mo, Y., Sinopoli, B.: Secure control against replay attacks. In: 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton). pp. 911–918 (Sept 2009). https://doi.org/10.1109/ALLERTON.2009.5394956
33. Mo, Y., Sinopoli, B.: Integrity attacks on cyber-physical systems. In: Proceedings of the 1st International Conference on High Confidence Networked Systems. pp. 47–54. HiCoNS '12, ACM, New

York, NY, USA (2012). https://doi.org/10.1145/2185505.2185514, http://doi.acm.org/10.1145/2185505.2185514
34. Mohammadi, A., Yang, C., Chen, Q.w.: Attack detection/isolation via a secure multisensor fusion framework for cyberphysical systems. Complexity **2018** (2018)
35. NIST: Cyber-physical systems. https://www.nist.gov/el/cyber-physical-systems (2014)
36. Overschee, P.V., Moor, B.D.: Subspace identification for linear systems: theory, implementation, applications. Boston: Kluwer Academic Publications (1996)
37. Pasqualetti, F., Dorfler, F., Bullo, F.: Attack detection and identification in Cyber-Physical Systems, models and fundamental limitations. IEEE Transactions on Automatic Control **58**(11), 2715–2729 (2013)
38. Rubio, J.E., Alcaraz, C., Roman, R., Lopez, J.: Analysis of intrusion detection systems in industrial ecosystems. In: SECRYPT. pp. 116–128 (2017)
39. Sethi, K., Rupesh, E.S., Kumar, R., Bera, P., Madhav, Y.V.: A context-aware robust intrusion detection system: a reinforcement learning-based approach. International Journal of Information Security **19**(6), 657–678 (2020)
40. Shoukry, Y., Martin, P., Yona, Y., Diggavi, S., Srivastava, M.: Pycra: Physical challenge-response authentication for active sensors under spoofing attacks. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1004–1015. CCS '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2810103.2813679, http://doi.acm.org/10.1145/2810103.2813679
41. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy. pp. 305–316. IEEE (2010)
42. Urbina, D.I., Giraldo, J.A., Cardenas, A.A., Tippenhauer, N.O., Valente, J., Faisal, M., Ruths, J., Candell, R., Sandberg, H.: Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1092–1105. ACM (2016)
43. Wang, X., Luo, X., Zhang, M., Jiang, Z., Guan, X.: Detection and isolation of false data injection attacks in smart grid via unknown input interval observer. IEEE Internet of Things Journal **7**(4), 3214–3229 (2020). https://doi.org/10.1109/JIOT.2020.2966221
44. Wei, X., Verhaegen, M., van Engelen, T.: 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), 687–707 (2010). https://doi.org/10.1002/acs.1162, http://dx.doi.org/10.1002/acs.1162
45. Welch, P.: 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), 70–73 (1967)
46. Wired: A cyberattack has caused confirmed physical damage for the second time ever. https://www.wired.com/2015/01/german-steel-mill-hack-destruction/ (2015)
47. Yang, T., Murguia, C., Kuijper, M., Nešić, D.: An unknown input multi-observer approach for estimation, attack isolation, and control of lti systems under actuator attacks. In: 2019 18th European Control Conference (ECC). pp. 4350–4355 (2019). https://doi.org/10.23919/ECC.2019.8796178