Zero Residual Attacks on Industrial Control Systems and Stateful Countermeasures

Hamid Reza Ghaeini
Singapore University of Technology and Design
Singapore
ghaeini@acm.org

Nils Ole Tippenhauer
CISPA-Helmholtz Center for Information Security
Saarbrucken, Germany
tippenhauer@cispa.saarland

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

ABSTRACT

In this paper, we discuss the practical implementation of stealthy attacks on industrial control systems. We start by reviewing the attacks proposed in prior works. Then, we offer Zero-Residual Attacks (ZeRA), which allow the attacker to launch stealthy attacks leveraging estimation of the stateful anomaly detector and matching of residuals as a fraction of actual estimation residual. To perform the zero residual attack, the attacker will require the use of two state estimators each for the physical system state and the detector system state, adding complexity that was so far not discussed. We implement ZeRA and demonstrate its efficacy. Then, we propose to use a Stateful Detector (SD) to precisely detect such stealthy attacks. We design and implement the SD detector. The obtained results from the performance evaluation demonstrate that we can detect stealthy attacks such as the ZeRA, with precision above 99%, sensitivity above 99%, and Matthews correlation coefficient above 0.98.

CCS CONCEPTS

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; **Systems security**; • **Computing methodologies** → *Machine learning*; Modeling and simulation; • **Computer systems organization** → *Embedded and cyber-physical systems*; *Sensors and actuators*;

KEYWORDS

Industrial Control System, Stealthy Attack, Stateful Anomaly Detection

ACM Reference Format:
Hamid Reza Ghaeini, Nils Ole Tippenhauer, and Jianying Zhou. 2019. Zero Residual Attacks on Industrial Control Systems and Stateful Countermeasures. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019) (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3339252.3340331

1 INTRODUCTION

Modern Industrial Control Systems (ICS) can be connected to the Internet for remote supervision and maintenance, and they are using industrial protocols on top of IP and TCP protocols. Such connectivity raises security concerns of the ICS in both cyber and physical levels. A modern advanced persistent threat (APT) has compromised even systems that are "air-gapped" (i.e., in an isolated network) designed to control the ICS systems. For example, Stuxnet [10] compromised programmable logic controllers (PLC) over the industrial network, and the attacker eventually was able to manipulate the industrial process state. Modern cryptographic solutions often cannot be implemented in existing ICS for legacy compliance and performance reasons.

In recent years, different approaches to detect such attacks were proposed, among them (i) network-based attack detection (specialized on industrial protocols) similar to traditional IDS [1, 6, 19], and (ii) stateful detection schemes that verify correct physical process behavior and controls (focused on process models and control theory) [3, 20–22]. The latter approach often uses stateful anomaly detection techniques like CUmulative SUM (CUSUM), in the context of water treatment systems, water distribution networks [20, 21], and smart grids [7]. The CUSUM aggregates the residual of observed system and estimated system state. The alarm will be raised once the CUSUM crosses a threshold. In [5], the authors proposed a state-aware detection scheme that considers process states in CUSUM computation to provide a tighter bound for stealthy attackers.

To motivate our work, we discuss practical implementations of stealthy attacks proposed in prior works. We demonstrate the feasibility of performing attacks in a real ICS, forcing the process to enter an unsafe state with zero residual of the stateful system state estimation, and, without passing the detection threshold. This attack is designed to perform the stealthy attack without any prior knowledge of the stateful detection parameter settings, and of course, without raising alarms from prior stealthy attack detection techniques.

We propose Zero-Residual Attacks (ZeRA), which allow the attacker to launch stealthy attacks even if such thresholds are unknown by the attacker, leveraging estimation of the stateful anomaly detector and matching of residuals to noise levels. To achieve that, the attacker leverages two system state estimators, one estimator for the system state estimator at the detector side to prevent the detection and a second system state estimator based on the actual physical state of the ICS to estimate how close she is to her physical goal of the attack. We implement the ZeRA attack, and we demonstrate its efficacy. We consider additional noises to measure the effect of noise on both attacker performance and detector performance.

As a second main contribution, we propose a defense-in-depth countermeasure against such *strong* attackers who perform stealthy attacks. We propose the Stateful Detector (SD) which uses stateful features of the industrial control system in our detection scheme,

ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340331

which is a novel stealthy attack detection scheme with a machine-learning based classifier that uses both stateless and stateful physical process features.

We evaluate the performance of the ZeRA attack, and its countermeasure SD detector, by benchmarking the attack in a real water treatment ICS. The implemented stealthy attack will cause an overflow of the water tank in a realistic water treatment system. Then, we simulate additional noise to evaluate our proposal in a non-deterministic case comprehensively.

We summarize the main contributions of this work as follows:

- We design and implement the ZeRA attack, which is a strong stealthy attack and it will not trigger state-of-the-art stateful detection techniques. We note that prior attacks require the attacker to know the detection threshold precisely to avoid detection. The ZeRA will keep the attacked residual at a fraction of the actual residual.
- As a countermeasure to our new attack, we propose SD, which is a stateful anomaly detection framework that extracts the required detection features directly from industrial network packets.
- We evaluate the SD detector's performance, leveraging the ZeRA attack by a comprehensive set of realistic implementation and a simulation-based case study of the additional noise.

Organization. The rest of this paper is organized as follows. Section 2 provides the background of this paper. Section 3 discusses practical implementation of stealthy attacks. In Section 4, we present our proposed ZeRA attack. We propose the SD detector in Section 5, and we present the evaluation and discussion in Section 6. We explore the related state-of-the-art in Section 8. Finally, the paper concludes in Section 9.

2 BACKGROUND

In this section, we will present the industrial control system, the water treatment system used in our performance evaluation, and the CUSUM change point detector.

2.1 Industrial Control System

The term "Industry 4.0" or "Smart Factory" refers to the connected industrial control systems to support aspects including Cyber-Physical Systems, Cloud Computing, and the Internet of Things [8]. The modern industrial control systems consist of three major levels:

- Supervisory Control And Data Acquisition (SCADA): this level of the ICS mainly used for the control and monitoring of the industrial process that may consist of large-scale geographical distributed computers. Five major components of the SCADA are the human-machine interface (HMI), data acquisition server, historian, engineer workstations and remote workstation.
- Programmable Logic Controllers (PLC): The local control component that is mostly designed for managing a single process in ICS. PLCs are industrial computers that developed for handling the process level devices like sensors and actuators.
- Fieldbus: The physical elements like sensors and actuators are connected to the PLC at this level. Most of the recent Fieldbus implementations use the Device Level Ring (DLR) with two redundant PLCs and a ring topology between those PLCs and physical elements.

This layered design of the industrial control systems provides a better implementation and maintenance of the whole process.

2.2 The Water Treatment System

We used a real water treatment ICS to perform the proposed attack and evaluate the performance of the proposed detection technique. The SWaT water treatment system is a six-stage testbed designed for security and safety analysis of water treatment industry [12]. The first stage is intended to control inflow water to the water tank by opening and closing a valve. Stage P2 is responsible for chemical dosing, and it pumps the water reserved in tank 1 to the ultrafiltration feed water tank of stage 3. Stage 3 is responsible for pumping the water from its tank to reverse osmosis feed water tank of stage 4.

Stage 4 controlling the water pumping through the ultraviolet dechlorination. Stage 5 is responsible for passing the water through a reverse osmosis unit, and it will store in a permeate tank. The backwash process will be done in stage 6, and the water will be rejected to ultrafiltration. In stage 6, the ultrafiltration pump will open and close to clean the membranes from the water.

2.3 CUSUM

The residual is the absolute difference between the system reading and its estimation. The residual is defined as:

$$r_k = |y_k - \hat{y_k}| \tag{1}$$

where y_k is the sensor measurement and $\hat{y_k}$ is the estimated sensor measurement. CUmulative SUM (CUSUM) is one the most promising proposals for possible change detection in the ICS at an unknown change point [3]. The non-parametric CUSUM statistic is recursively computed as follows:

$$S_k = \begin{cases} 0 & \text{where } k = 0 \\ (S_{k-1} + r_k - \alpha)^+ & \text{where } k \neq 0 \end{cases} \tag{2}$$

where $(x)^+$ means $max(0, x)$ and α is the tuning value that selected to keep $|r_k| - \alpha < 0$ under a normal operation. Then, the CUSUM test will restart after crossing the threshold at time k, i.e., $S_{k+1} = 0$. The state-aware anomaly detector considers the physical state in CUSUM computation and it will raise the alarm when the computed CUSUM passes the threshold [5].

3 PRACTICAL IMPLEMENTATIONS OF STEALTHY ATTACKS

In this section, we discuss the system and attacker model, the system state, greedy stealthy attack, and practical implementations of stealthy attacks. We demonstrate that an attacker needs two state estimators to perform stealthy attacks without prior knowledge of detectors configuration.

3.1 System and Attacker Model

We assume that the physical processes under attack can be modeled with sufficient precision through a linear model, which is available to the defender. The defender is monitoring the reported sensor and actuator data (i.e., by monitoring Fieldbus or SCADA network traffic), and uses that data and process-aware detection mechanisms

such as the ones proposed in [21, 22] to detect ongoing attacks. Attacks on the monitoring system itself (i.e., remote compromise) are out of the scope of this work. The attacker has either compromised a device in the Fieldbus or obtained access to the plant network through other means, and is able to perform the stealthy attack by manipulating the flowing traffic. The attacker's goal is to manipulate the physical process state, e.g., to damage the system. To achieve that goal, the attacker can either manipulate data contained in the network traffic, or compromise sensors or actuators to directly manipulate the sensing and actuation of the physical process while she will remain undetected by other conventional network security solutions, cyber anomaly detectors, or physical anomaly detection systems. The attacker might remain undetected in the system and be present in isolated networks such as the Fieldbus network. The attacker knows the system and the cyber detection strategies.

Figure 1 shows the attacker setup inside the control process to perform the stealthy attack. The controller sends the controller commands (u_k) via the Fieldbus to the actuators. The actuators would perform the actuation commands (v_k) and send the controller commands (u_k) via the Fieldbus to the sensors. The sensors measure the physical parameters from the physical process (z_k) and send the controller commands (u_k), and sensor readings (y_k) via the Fieldbus to the attacker. The attacker sends the controller commands (u_k), and manipulated sensor readings (y_k^a) via the Fieldbus to the controller. The detector is wiretapping the Fieldbus network traffic and reads the controller commands (u_k), and manipulated sensor readings (y_k^a). The controller will receive the controller commands (u_k), and manipulated sensor readings (y_k^a) and perform the next actuation commands (u_{k+1}) and this process will continue.

3.2 The System State

The system state will be determined by a set of variables modeling the state of the system. We leverage approaches from control theory where the next state of the system will be predicted by considering the current state of the system. The process states are the specific states determined by the discretization of the process over time. We used the Linear Dynamical State-space (LDS) model to perform the physical modeling of the processes:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \epsilon_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned} \tag{3}$$

where A, B, C, and D are the system matrices that determined by system identification, k is the current state of the system and $k+1$ is the next state of the system, u_k is control commands, x_k is the state of the estimated model, x_{k+1} is next state of the system, and y_k is sensor measurements.

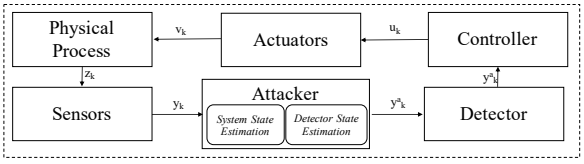


Figure 1: The stealthy attacker setup inside the Fieldbus network with the detector.

3.3 Greedy Stealthy Attack

The stealthy attacker that knows the detection threshold will maximize the possible difference between reported sensor value and actual sensor value while remaining undetected by stateful anomaly detection techniques. We consider such an attacker that knows our parameter settings and tries to avoid detection [21, 22]. An optimal greedy attacker (y^{a*}) at time t will try to maximize the residual while remaining below the threshold.

$$y_{k+1}^{a*} = \begin{cases} \arg\max^a_{y_{k+1}} |y_{k+1} - y^a_{k+1}| \\ \arg\min^a_{y_{k+1}} |y_{k+1} - y^a_{k+1}| \end{cases} \tag{4}$$

The attacker goal in stateless detection techniques will be:

$$y_{k+1}^{a*} = y_{\hat{k+1}} \pm \tau \tag{5}$$

As such, the attacker's goal to avoid stateful detection techniques will be:

$$y_{k+1}^{a*} = \max\{y_{k+1} : S_{k+1} \leq \tau\} \tag{6}$$

The CUSUM is computed by Equation 2. A greedy optimal attacker tries not to pass the threshold of CUSUM, i.e. $S_k = \tau$. Hence, the attacker goal will be:

$$y_{k+1}^{a*} = y_{\hat{k+1}} \pm (\tau + \alpha - S_k) \tag{7}$$

3.4 Implementation of Stealthy Attacks

We note that to solve Equation 7 (i.e., to find values for y that will allow a stealthy attack), the attacker requires knowledge of τ, a precise estimate of the defender's current CUSUM value S_K, and the residual that will be caused by the attacker signal $y_k^{a*} - \hat{y_k}$ (which requires an estimate of the defender's estimate $\hat{y_k}$). In other words, the attacker will have to run a *estimation of the defender's process estimation*, while the defender is trying to estimate the state of a system based on the attacker's signals. In particular, any noise on the sensor signals that is unknown (and unpredictable) to the attacker will diverge the attacker's estimation of the defender's detection system state. We argue that this important requirement for stealthy attacks was not sufficiently discussed so far in prior work, and calls for a solution in which the attacker does not need to precisely estimate S_K, and ideally does not require knowledge of τ at all.

4 PROPOSED ZERA ATTACK

In this section, we present the design of the Zero-Residual Attack (ZeRA), a novel stealthy attack which will not trigger state-of-the-art stateful detection techniques and it will keep the attacked residual at a fraction of the actual residual. This new attack will generate zero residual in control theoretical techniques such as stateless and stateful detection techniques. In addition to zero residual characteristics of the attack, the ZeRA will generate a residual as a fraction of the actual residual, which will harden the detection of the stealthy attack.

4.1 Zero Residual Attack

The non-parametric CUSUM test will raise an alarm where the computed S_k passes the detection threshold:

$$S_k > \tau \tag{8}$$

Considering an arbitrary threshold of τ, the primary goal of the ZeRA attack is to keep the computed CUSUM to be zero while she is performing the attack. In this way, regardless of the value of the τ, the attacker will execute the attack while remaining undetected. We could rewrite the primary goal of the ZeRA attack as:

$$(r_k - \alpha)^+ = 0 \tag{9}$$

In other words, the attacker will maximize the residual in a way that it will not pass the α in the residual computation at the detector. The ZeRA attacker will perform the attack as follows:

$$y^a_{k+1} = y^{\hat{a}}_{k+1} \pm \alpha \tag{10}$$

where the y^a_{k+1} is the signal value reported to the detector by the attacker, and $y^{\hat{a}}_{k+1}$ is the detectors estimation of the system state at state $k+1$. Hence, the detector would compute the detection residual as:

$$|r_k| = |y^a_{k+1} - y^{\hat{a}}_{k+1}| = \alpha \tag{11}$$

By performing the attack as proposed in Equation 10, the ZeRA attacker will bypass the stateless and stateful detection techniques introduced in [6, 22].

4.2 ZeRA with a Residual as a Fraction of Actual Residual

The residual might be used as a fingerprinting countermeasure to detect the stealthy attack. Hence, the ZeRA attacker wants to keep the shape of the noises computed by LDS techniques at the detector as a fraction of the actual estimation residual. The attacker would report the y^a_{k+1} as:

$$y^a_{k+1} = \begin{cases} y^{\hat{a}}_{k+1} - \beta \times r_k & \text{if the signal is } \textit{increasing} \\ y^{\hat{a}}_{k+1} + \beta \times r_k & \text{if the signal is } \textit{decreasing} \end{cases} \tag{12}$$

where the β is the tuning value of the ZeRA attack and the attacker computes the residual r_k by the following equation:

$$r_k = y_k - \hat{y_k} \tag{13}$$

Hence, the detector would compute the detection residual as:

$$|r_k| = |y^a_{k+1} - y^{\hat{a}}_{k+1}| = \beta \times r_k \tag{14}$$

As discussed before, the attacker needs two state estimators:

- System State Estimation: the attacker would estimate the current system state to find out how close she is to her physical goal and the attacker uses this estimation to compute $\hat{y_k}$ and to perform the ZeRA attack as a β fraction of the actual residual (r_k).
- Detector State Estimation: the attacker needs to estimate the detector's system state to avoid being detected.

Figure 2 shows the two-state estimators that we have used in the implementation of the ZeRA attack. The first system state estimator estimates the current state of the physical process based on the actual sensor readings (y_k). The second system state estimator is used to prevent the detection, and the attacker estimates the detectors system state estimation and reports the manipulated sensor readings (y^a_k). The ZeRA attacker will use the fraction of the noise of the system to generate the attacked value (see Equation 12). Then, the ZeRA attacker estimates the detector's state to avoid the detection while maintaining the noise (residual) of the performed ZeRA attack as a fraction of actual noise (residual) of the system. Finally, the ZeRA will examine whether the reported attacked value will pass the α or not. If the attacked value does not pass the α, the ZeRA attacker will report the attacked value to the controller. Otherwise, the attacker performs some tuning over the attacked value and examines the tuned value again. As discussed before, by performing such an attack, the conventional stateful detectors will not be able to detect the ZeRA, cause the ZeRA will not pass the α.

5 SD DETECTOR

In the previous section, we introduced the ZeRA attack. The ZeRA was designed to bypass the conventional detectors that are based on stateful detection, while ZeRA can reach its physical goal. We now propose the stateful (SD) that would be able to detect stealthy attacks such including the ZeRA, by a novel feature sets that will includes both stateless and stateful physical state estimation features of the system to detect the ZeRA attack, in addition to other previously proposed attacks.

5.1 Design Overview

Figure 3 shows the SD structure and its data processing module. The data processing module consists of three phases:

- Training: The training phase will be done with the historical record of the network packets that are labeled with normal or attack label.
- Classification: The classification phase will be done during the operational process of the ICS and reads the real-time record of ICS network packets.
- Detection: At the detection phase, if the classifier reports an anomaly.

The training phase will create the machine learning model, and the historical record of the network packets will be pre-processed to generate the desired cyber and physical records for the next component which is the feature extraction. The features will be extracted from the cyber and physical records and will be passed to the training classifiers to generate the machine learning model (ML model). We will store the ML model for real-time processing of the real-time record of ICS network packets. During the classification phase the same pre-processing and feature extraction components will generate the stateless and stateful features and by using the stored ML model the corresponding label to the real-time record of ICS network packets will be generated.

5.2 Feature Sets

The feature sets of the SD detector will generate a machine learning model that will be used during real-time anomaly detection. By providing the right features to the SD detector, the SD will offer an online detection framework for detection of strong stealthy attacks such as the ZeRA attack. Table 1 provides the features used in the proposed detection scheme. As we will see in Table 1, the SD detector considers the ICS features including the actuator states, the sensors reading, the estimation of the system state, and the residual of that estimation.

The stateless features includes the actuation commands (process states), the current sensor reading (or sensor signal), the estimation of the sensor reading, and the residual of the estimation. We also used stateful features over a window (κ) to have a windowed

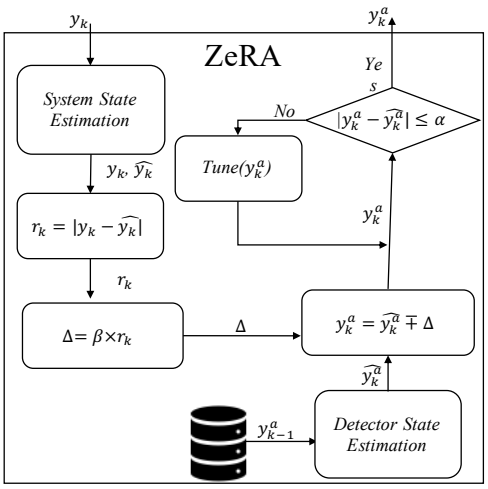


Figure 2: The structure of ZeRA attacker inside the Fieldbus network.

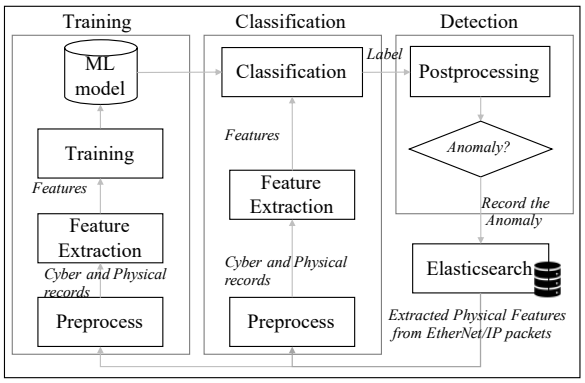


Figure 3: The structure of the SD detector inside the Fieldbus network.

CUSUM of the actuation commands, sensor reading, estimation of the sensor reading, and the residual of the estimation. During the training process, a machine learning hyper-plane will be generated that includes the behavior of the system in normal operation by provided features.

To generate those features, we will perform pre-processing over the cyber and physical records. Here, our focus is on the CUSUM that used in several stateful state estimation applications. We used a window (κ) of the pre-processing to provide an upper bound of κ for the time-to-detect. The κ value is a system specific value and will be computed in a way that the detector will detect the attack before the attacker would enter the system to an unsafe state. The window of actuation commands provides a window of the actuation commands and the classifiers would classify the actuation commands based on the history of actuation commands generated during the training phase. The window of signal difference provides the CUSUM of differences of the signal over the window. The window of estimation difference provides the CUSUM of differences of the estimation over the window.

5.3 Implementation

We used Raspberry PI devices to perform deep packet inspection in the Fieldbus and SCADA network by an ICS extension of the Bro [16] intrusion detection system (IDS) and a central server to records logs of those IDS components. These computing devices provide detailed information about the process inside the PLC network. We gather that information from all six stages in the central server by an SSH channel. The central server receives the network traffic from the network switch at the SCADA level. As a result, the central server has complete information about the ICS traffic both at Fieldbus and SCADA network level.

The central server consists of five software components. The packet parser is an ICS extension of the Bro IDS that can parse industrial control network packets, in particular, Ethernet/IP packets and DLR packets. The packet parser will generate detailed log files for network-based intrusion detection, cyber, and physical features. The cyber features are the type of packets, timing of the packet, and information about the payload of packets. The physical features are extracted from the payload of the packets, including actuator states, sensor reading, and control commands. The Logstash will read the generated logs and interpret them to store them in our triple-store database which is Elasticsearch. We compute the residual at the Logstash. The processed logs will be stored in the Elasticsearch, and we use its computation capabilities on upper layers of the framework. Kibana is an interface for visualization of the data and running the commands on Elasticsearch. We used Kibana for real-time visualization of the traffic. Besides, we have the data processing module, which will process the Elasticsearch stored data and return the results of the SD detector to be stored in Elasticsearch. The scripts of the anomaly detection are based on the extracted machine learning model from the training phase. We used the WEKA libraries to generate the machine learning model from our stored data.

6 EVALUATION AND DISCUSSION

In this section, we will present the evaluation, and discussion of the detection performance.

6.1 ICS Use-case

We used a water tank filling process in a water treatment system to benchmark the ZeRA attacker, and the SD detector. The Linear Dynamical State-space model of the water tank level based on the input Q^{in} and output Q^{out} volume of water tank is:

$$Area\frac{dh}{dt} = Q^{in} - Q^{out} \tag{15}$$

With time discretization over one second, the water level model will be:

$$h_{k+1} = h_k + \frac{Q_k^{in} - Q_k^{out}}{Area} \tag{16}$$

Hence, the ZeRA attacker will report the sensor reading by:

$$h_{k+1}^a = \begin{cases} h_k^a + \beta(h_k - \hat{h_k}) & \text{if the signal is } increasing \\ h_k^a - \beta(h_k - \hat{h_k}) & \text{if the signal is } decreasing \end{cases} \tag{17}$$

where $\beta(h_k - \hat{h_k}) \leq \alpha$. In this way, the ZeRA attacker will remain undetected without knowing the detection scheme of the stateful

Table 1: The features classes in the proposed scheme to detect stealthy attacks.

	Formula	Stateless	Stateful
Actuation Commands	u_k	●	○
Sensor Signal	y_k	●	○
Sensor Estimation	$\hat{y_k}$	●	○
Residual	$y_k - \hat{y_k}$	●	○
Window of Actuation Commands	$\sum_{i=k-\kappa}^{k} u_i$	○	●
Window of Signal Difference	$\sum_{i=k-\kappa}^{k-1} \lvert y_{i+1} - y_i \rvert$	○	●
Window of Estimation Difference	$\sum_{i=k-\kappa}^{k-1} \lvert \hat{y_{i+1}} - \hat{y_i} \rvert$	○	●
Window of Residuals	$\sum_{i=k-\kappa}^{k} \lvert y_i - \hat{y_i} \rvert$	○	●

anomaly detector. Our implemented ZeRA attack has a physical goal of causing a water tank overflow in the water treatment system.

To evaluate the performance of the proposed detection framework, we implemented the ZeRA attack with $\beta = 0.7$, and we used this implementation to assess our proposed detection scheme. As a countermeasure, we performed the SD detector with $\kappa = 2$ minutes to detect the ZeRA attack, before causing a water tank overflow. We collect the data of normal operation and system under attack of three days, and the overall size of the data is more than 120 GB of historical data. This data includes both extracted Fieldbus and SCADA network traffic features and physical features. To perform a comprehensive evaluation of those machine learning techniques, we executed ten-fold cross-validation to measure its performance for randomized train and test sets during the training phase.

6.2 Extension of Datasets with Simulated Additional Noise

We consider some noise after the attacker to measure the impact of attacker noise on the detector. We simulated noises of 5%, 25%, and 50% of the sensor reading precision at the reported sensor reading of the attacker to the detector. We used three scenarios of attack where the first scenario is the actual implementation with real data obtained from the ICS, and second and third scenario have a simulation of an attacker that induce the noise to the detector. In the second scenario, the start of attack will cause additional noise, and there is not any extra noise before the start of the physical attack (see Fig. 4c). In the third scenario, the attacker itself has an additional noise, and the extra noise is present before and after the start of the stealthy attack (see Fig. 4e). These attacks are simulated with MATLAB to evaluate the effect of the noise on attacker and detection performance while they will keep the shape of the system noise as a Gaussian distribution. Considering the mean of the additional random noise of the attacker of n_a, we could rewrite Equation 12 as:

$$y^a_{k+1} = \begin{cases} y_k + \beta \times r_k - n_a & \text{if the goal is } \textit{increasing} \\ y_k - \beta \times r_k + n_a & \text{if the goal is } \textit{decreasing} \end{cases} \tag{18}$$

where the β is the tuning value of the ZeRA attack and $\beta \times r_k \leq \alpha$. We consider the following scenarios for the additional noise, and we used these three data sets in our performance evaluation:

- Scenario I (Noiseless): There would be no additional noise of the attacker or attack itself (see Fig. 4a, and Fig. 4b).
- Scenario II (Noisy Attacker): There is additional noise of the attack and after the start of the stealthy attack, the manipulated sensor reading will have 5%, 25%, and 50% of additional noise (see Fig. 4c, and Fig. 4d).
- Scenario III (Noisy Channel): There is additional noise of the attacker and during the operational process the manipulated sensor reading will have 5%, 25%, and 50% of additional noise (see Fig. 4e, and Fig. 4f).

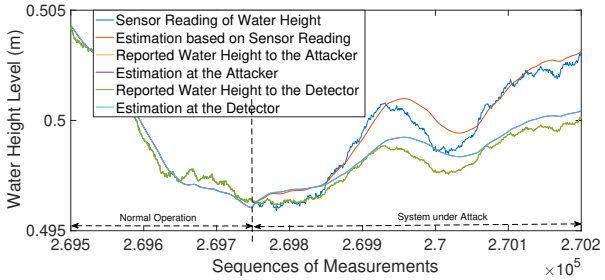
Figures 4a, and 4b show the effect of implemented ZeRA attack on water level reading, and estimation residuals with real data extracted from the water treatment system (Scenario I). As we will see in 4a, the reported water level to the controller and detector is less than what the actual sensor is reporting, and the effect of the ZeRA attack is shown on Fig. 4b where the actual residual of the physical process estimation is a fraction of the attacker estimation of the system state, while the ZeRA attack manipulates the detector residual and it is different from the actual water height estimation of the reported sensor value. We can see the effect of attacker noise (Scenario II) at Fig. 4c, and Fig. 4d after the start of the attack. We can see some distortion of reported sensor reading after the start of the attack. Also, we can observe that the attacker's residual is still close to the residual of the system state estimation based on actual reported sensor value. Figures 4e and 4f show the effect of the additional noise of the channel (Scenario III) to the system state estimation. We can see the distortion of the reported sensor reading after the attacker. Still, the attacker can successfully perform the ZeRA attack, and there is a divergence of the detectors estimated system state and the actual system state.

7 PERFORMANCE METRICS

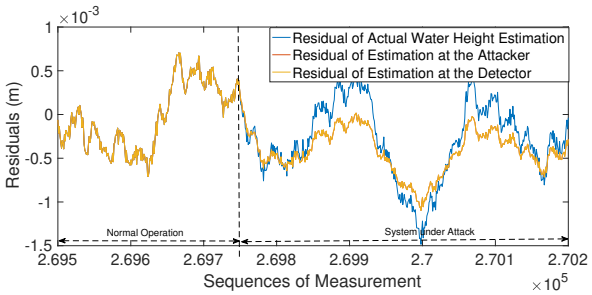
To evaluate the performance of the proposed method we used eight performance metrics. The true positive (TP) is the number of retrieved relevant instances. The false positive (FP) is the number of retrieved non-relevant instances. The true negative (TN) is the number of not retrieved non-relevant instances. The false negative (FN) is the number of not retrieved relevant instances. The Sensitivity rate (Recall, eq. 19) presents the rate of retrieved relevant instances (TP) in overall relevant instances (TP + FN). The Precision rate (specificity, eq. 20) demonstrate the fraction of relevant instances (TP) in overall retrieved instances (TP + FP).

$$\text{Sensitivity rate} = \frac{TP}{TP + FN} \tag{19}$$

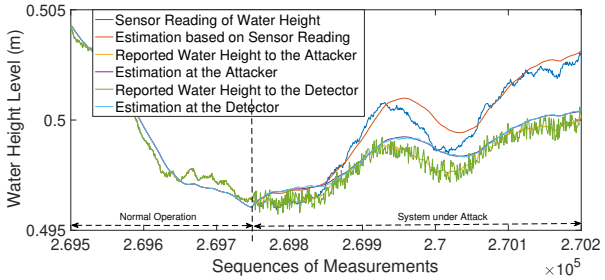
$$\text{Precision rate} = \frac{TP}{TP + FP} \tag{20}$$



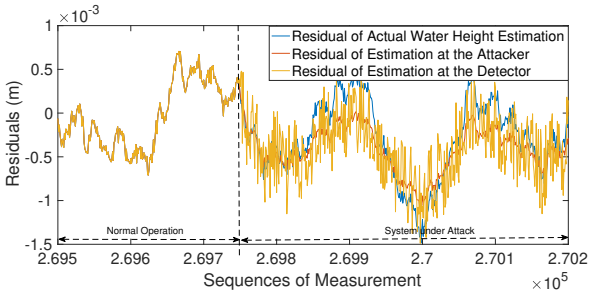
(a) Water tank level reading and estimation (Scenario I).



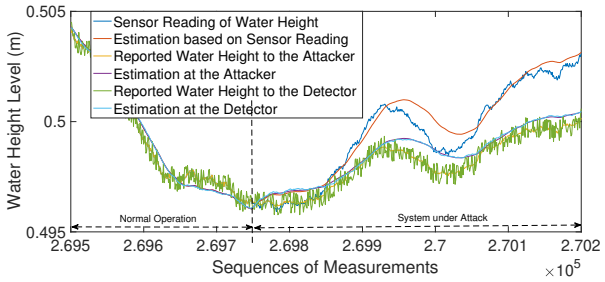
(b) Estimation residual (Scenario I).



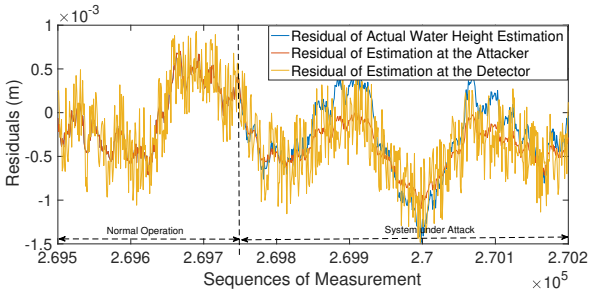
(c) Water tank level reading and estimation (Scenario II).



(d) Estimation residual (Scenario II).



(e) Water tank level reading and estimation (Scenario III).



(f) Estimation residual (Scenario III).

Figure 4: Sensor readings estimation, and residuals of system state estimation.

Table 2: Performance comparison of the classifiers with data set of Scenario I with real data obtained from the control process. The classifiers: Random Forests (RF), Naive-Bayes Tree (NBTree), Logistic Model Tree (LMT), J48, PART, Multilayer Perceptron (MLP), Hoeffding Trees (HTree), Logistic Function (LogF), and Support Vector Machine (SVM).

Algorithms	Precision	Sensitivity	FP	FN	FD	F1-score	MCC
RF	0.9935	0.9956	0.0004	0.0044	0.0065	0.9945	0.9942
NBTree	0.9908	0.9903	0.0006	0.0097	0.0092	0.9905	0.9899
LMT	0.9682	0.9754	0.0020	0.0246	0.0318	0.9718	0.9700
J48	0.9623	0.9680	0.0024	0.0320	0.0377	0.9651	0.9629
PART	0.9375	0.8933	0.0037	0.1067	0.0625	0.9149	0.9100
MLP	0.7556	0.7883	0.0160	0.2117	0.2444	0.7716	0.7571
HTree	0.6950	0.7132	0.0196	0.2868	0.3050	0.7040	0.6852
LogF	0.6886	0.6656	0.0189	0.3344	0.3114	0.6769	0.6571
SVM	0.5270	0.1474	0.0083	0.8526	0.4730	0.2304	0.2573

The false positive rate (eq. 21) is the rate of retrieved non-relevant instances (FP) in overall non-relevant instances (FP + TN). The false negative rate (eq. 22) is the rate of not retrieved relevant instances (FN) in overall relevant instances (FN + TP). The false discovery rate (eq. 23) is the rate of retrieved non-relevant instances (FP) in overall retrieved instances (FP + TP).

$$\text{False positive rate} = \frac{FP}{FP + TN} \tag{21}$$

$$\text{False negative rate} = \frac{FN}{FN + TP} \tag{22}$$

$$\text{False discovery rate} = \frac{FP}{FP + TP} \tag{23}$$

The F1-score (eq. 24) is a metric for the test's accuracy. The F1-score (also F-score or F-measure) is defined as follows:

$$\text{F1-score} = \frac{2 \times \text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} \ + \ \text{Precision}} \tag{24}$$

The Matthews correlation coefficient (MCC, (eq. 25)) is a metric for the quality of two-class classification. The MCC metric is one of the most interesting metrics in anomaly detection where the physical feature will be classified to normal and abnormal classes. The MCC is defined as follows:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{25}$$

7.1 Evaluated Classifiers

We reviewed the most recent related papers, and we choose the most promising classifiers used in the related works like [2, 18]. Then, we evaluated classifiers with ten-fold cross-validation to find the best three classifiers for our comprehensive evaluation. The data-sets was extracted from a database of normal operation and system under attack of three days, and the overall size of the database was more than 120 GB of historical data. We performed 900 ten-fold cross-validation with a random seed in each run. Our results showed that the LMT, Naive Bayesian Tree, and Random Forests are suitable classifiers to be implemented inside of the framework due to their high precision, sensitivity, and MCC. Table 2 shows the performance evaluation of nine classifiers that we used in our experimental evaluation process. The Random Forests (RF) shows the best performance in comparison to other classifiers, and it could detect ZeRA with precision above 99%, sensitivity above 99%, and Matthews correlation coefficient above 0.99. In ICS security, we are interested in classifiers that will provide zero false positives. In our experiments, the Random Forests and NBTree classifiers provided a false positive rate of maximum 0.0006. The performances of LMT and NBTree are close to the Random Forests, and their MCC is higher than 0.97. The other classifiers like J48 and Part are still good candidates for stealthy attack detection in the ICS. The performance of Hoeffding Tree (HTree), Multilayer Perceptron (MLP), Logistic Functions (LogF), and SVM was insufficient to be considered in our evaluation against the additional simulated noise of the attacker.

7.2 Evaluation of Noise Effect on the Detection Performance:

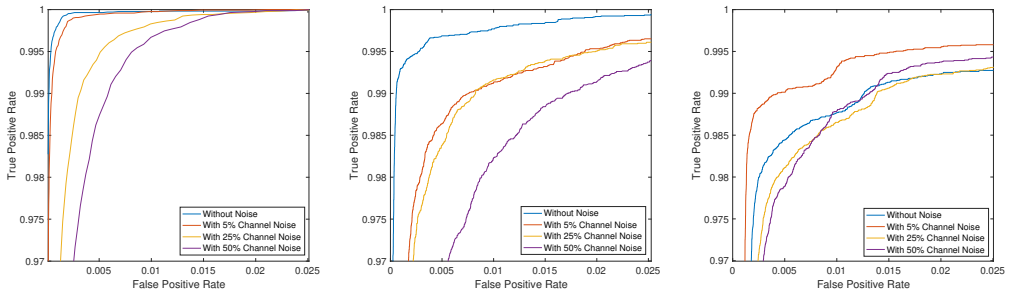
In our next experiment, we evaluated the performance of the three selected classifiers from our previous experiments against some additional simulated noise of the attacker. The selected classifiers are Random Forests, NBTree, and LMT. During the training phase, we used ten runs of ten key-folds cross-validation for each classifier to have a comprehensive evaluation of that classifier, and in total, we had 2100 ten-fold cross-validation with a random seed in each run. We will see in Figures 5a, 5b, and 5c that the Random Forests works better than NBTree, and LMT. Also, the NBTree performs better than the LMT.

Figure 5 shows the comparison between the different noise level of the channel or the attacker. Figure 5a and 5d show the 1-Precision ROC curve of Random Forest classifier in the presence of the noise of the attacker and channel, respectively. By comparing the ROC curve of Random Forest classifier with the different noise level in both Figure 5a and Figure 5d, we would conclude that the noise will decrease the overall performance of the classifier in the detection of the ZeRA attack. Also, we could compare the Figure 5a and Figure 5d directly, and we would conclude that the knowledge of the attacker about the detection mechanism would improve the overall performance of the attack. This is the case in Figure 5a where the additional noise is deterministic and the attacker estimates the channel noise. As we see in 5d, the extra noise of the attacker would not help the attacker to reduce the classifier performance during the detection process. We see the same behavior by comparing Figures 5b and 5e for NBTree, and Figures 5c and 5f for LMT classifier. Table 3 shows the performance evaluation of three selected classifiers that we used in our experimental evaluation of noise effect. The Random Forests classifier detects the ZeRA attack with precision above 99%, sensitivity above 99%, and Matthews correlation coefficient above 0.98 with presence of the noise in Scenario II and it detects the ZeRA attack with precision above 99%, sensitivity above 99%, and Matthews correlation coefficient above 0.96 with presence of the noise in Scenario III. We would conclude that the additional noise in both Scenario II and Scenario III will reduce the detection performance of the SD detector.

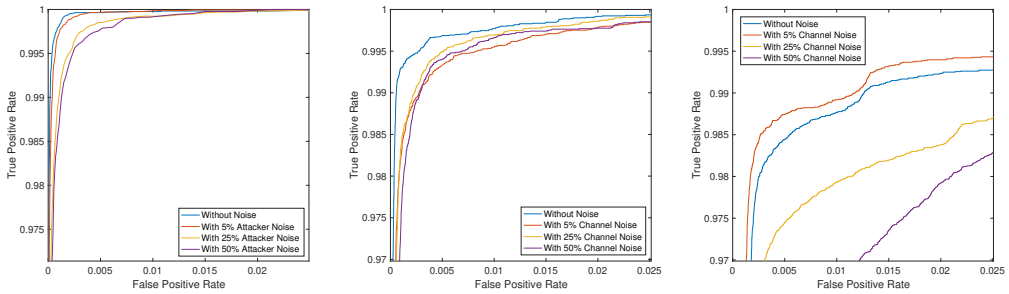
7.3 Discussion

The Random Forests learning techniques are providing the best performance comparing to other classifiers. As discussed in the literature, the random forest is a suitable tree-based learning technique for process modeling in cyber-physical systems. Ths industrial control systems are a subclass of the cyber-physical systems, and as we see in this paper, random forests showed the best performance to be used as a machine learning model inside the SD detector.

Based on the performance evaluation of the classifiers, we used Random Forests together with the stateless, and stateful features. We performed ten-fold cross-validation during the training phase, and we measured several learning algorithms with seven metrics. The results show that we could accurately detect the attacks with the accuracy above 99%. In addition, the random forests learning algorithms have 0.0003 false positive rate, which is a significant metric for ICS. To measure the normal and abnormal classification, we measure the performance by the MCC, and we can reach the 0.99 of MCC. In our experiments, the implementation of our proposed framework had a false negative rate of 0.4%. However, we can reliably detect the attack once it causes enough differentiate from



(a) The Random Forest classifier evaluation with Scenario II. (b) The Naive Bayes Tree classifier evaluation with Scenario II. (c) The Logistic Model Tree classifier evaluation with Scenario II.



(d) The Random Forest classifier evaluation with Scenario III. (e) The Naive Bayes Tree classifier evaluation with Scenario III. (f) The Logistic Model Tree classifier evaluation with Scenario III.

Figure 5: ROC curve of true positive rate (sensitivity) against false positive rate (1-precision).

Table 3: Performance comparison of different tested detection techniques in the three data sets of Scenario I, Scenario II, and Scenario III. The classifiers: Random Forests (RF), Naive-Bayes Tree (NBTree), and Logistic Model Tree (LMT).

Algorithm	Scenario	Noise	Precision	Sensitivity	FP	FN	F1-score	MCC
RF	Scenario I	0%	0.9997	0.9996	0.0003	0.0041	0.9996	0.9943
RF	Scenario II	5%	0.9995	0.9995	0.0004	0.0078	0.9995	0.9921
RF	Scenario II	25%	0.9987	0.9995	0.0004	0.0201	0.9991	0.9852
RF	Scenario II	50%	0.9983	0.9995	0.0004	0.0269	0.9989	0.9818
RF	Scenario III	5%	0.9994	0.9994	0.0005	0.0088	0.9994	0.9905
RF	Scenario III	25%	0.9980	0.9986	0.0013	0.03040	0.9983	0.9725
RF	Scenario III	50%	0.9971	0.9982	0.0017	0.0447	0.9977	0.9608
NBTree	Scenario I	0%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario II	5%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario II	25%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario II	50%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario III	5%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario III	25%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
NBTree	Scenario III	50%	0.9984	0.9979	0.0020	0.0242	0.9945	0.9942
LMT	Scenario I	0%	0.9984	0.9979	0.0020	0.0242	0.9982	0.9699
LMT	Scenario II	5%	0.9986	0.9983	0.0016	0.0211	0.9985	0.9749
LMT	Scenario II	25%	0.9976	0.9983	0.0016	0.0379	0.9979	0.9659
LMT	Scenario II	50%	0.9968	0.9981	0.0018	0.0508	0.9975	0.9573
LMT	Scenario III	5%	0.9988	0.9984	0.0015	0.0184	0.9986	0.9774
LMT	Scenario III	25%	0.9977	0.9977	0.0022	0.0355	0.9977	0.9617
LMT	Scenario III	50%	0.9977	0.9977	0.0022	0.0356	0.9977	0.9616

the trained noise. We generated the machine learning model with the decision tree made from the random forest learning technique, and the SD detector could detect the anomalies by online parsing the Fieldbus network packets.

8 RELATED WORK

The stateful detection techniques and machine learning based techniques were discussed in the literature. However, in this paper, we presented an online anomaly detection based on the machine learning technique that uses stateful computation as detection features. The rest of this section explores the related state-of-the-art works.

Stateful Anomaly Detection. The authors of [20] discussed the impact of the attacking Fieldbus communication. In [22] the authors proposed stateful CUSUM to limit the impact of Fieldbus attacks. In this paper, we used the *strong attack against sensor reading variables* that the attacker tries to change the sensor reading of the tank level with a constant value to remain undetected by changing the sensor measurement slowly.

Process State-aware Anomaly Detection. State-aware anomaly detection techniques designed to model the systems that randomly transit between state over a discretized time. The authors of [5] presented an anomaly detection technique that considers the process-states of the industrial control system over a discretized time. As shown in the [5], the overall performance of stateful anomaly detection techniques will drastically improve by considering the fact of states in ICS processes.

Machine Learning based Anomaly Detection. The authors of [13] discussed the machine learning proposals for anomaly detection in the ICS. Also, the authors of [9] proposed to use convolutional neural networks to detecting the cyber attacks in industrial control systems. Machine learning techniques for anomaly detection in industrial arm applications is discussed in [14]. The authors of [11] used the k-mean clustering to detect traffic phase shifts inside the SCADA automatically. The authors of [15] proposed a hybrid IDS that learns temporal state-based specifications of the power system, and they used data mining techniques to classify the scenarios of disturbances, normal control operations, and cyber-attacks. In [4], the authors discussed the data mining and machine learning techniques for cybersecurity. There are many successful applications of machine learning in cyber-physical system security. The authors of [17] proposed the measurement and verification of transmitted network data. They used telemetry based intrusion detection by machine learning techniques like REPTree, NaiveBayes, and Logistic.

9 CONCLUSIONS

In this paper, we discussed the practical implementations of stealthy attacks on industrial control systems proposed in prior works. We introduced ZeRA attack, which allows the attacker to launch stealthy attacks, leveraging estimation of the stateful anomaly detector and matching of residuals as a fraction of actual estimation residual. We implemented the ZeRA attack in a realistic water treatment ICS, and we demonstrate the effect of additional noise on the executed ZeRA attack. Then we presented the SD detector that leverages the stateless, and stateful features of the ICS during the detection process. We performed and verified our proposal in a realistic water treatment ICS. The obtained results from the performance evaluation showed that we could detect ZeRA with precision above 99%, sensitivity above 99%, and Matthews correlation coefficient above 0.98.

10 ACKNOWLEDGMENT

The authors would like to thank the Singapore University of Technology and Design (SUTD) for supporting this research by providing financial means and access to the test-beds. In particular, Jianying Zhou's work was supported by SUTD start-up research grant SRG-ISTD-2017-124.

REFERENCES

[1] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems; analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, 2013.
[2] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.
[3] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 355–366. ACM, 2011.
[4] S. Dua and X. Du. *Data mining and machine learning in cybersecurity*. CRC press, 2016.
[5] H. R. Ghaeini, D. Antonioli, F. Brasser, A.-R. Sadeghi, and N. O. Tippenhauer. State-aware anomaly detection for industrial control systems. In *The 33rd ACM/SIGAPP Symposium On Applied Computing (SAC)*, Apr. 2018.
[6] H. R. Ghaeini and N. O. Tippenhauer. Hamids: Hierarchical monitoring intrusion detection system for industrial control systems. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pages 103–111. ACM, 2016.
[7] J. Giraldo, A. Cárdenas, and N. Quijano. Integrity attacks on real-time pricing in smart grids: impact and countermeasures. *IEEE Transactions on Smart Grid*, 2017.
[8] M. Hermann, T. Pentek, and B. Otto. Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016.
[9] M. Kravchik and A. Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 72–83. ACM, 2018.
[10] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.
[11] C. Markman, A. Wool, and A. A. Cardenas. Temporal phase shifts in scada networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 84–89. ACM, 2018.
[12] A. Mathur and N. O. Tippenhauer. A water treatment testbed for research and training on ICS security. In *Proceedings of Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*, 2016.
[13] A. Meshram and C. Haas. Anomaly detection in industrial networks using machine learning: a roadmap. In *Machine Learning for Cyber Physical Systems*, pages 65–72. Springer, 2017.
[14] V. Narayanan and R. B. Bobba. Learning based anomaly detection for industrial arm applications. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 13–23. ACM, 2018.
[15] S. Pan, T. Morris, and U. Adhikari. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, 6(6):3104–3113, 2015.
[16] V. Paxson. Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.
[17] S. Ponomarev and T. Atkison. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing*, 13(2):252–260, 2016.
[18] A. K. Sikder, H. Aksu, and A. S. Uluagac. 6thsense: A context-aware sensor-based attack detector for smart devices. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 397–414, Vancouver, BC, 2017. USENIX Association.
[19] R. Udd, M. Asplund, S. Nadjm-Tehrani, M. Kazemtabrizi, and M. Ekstedt. Exploiting bro for intrusion detection in a SCADA system. In *Proceedings of Cyber-Physical System Security Workshop (CPSS)*, 2016.
[20] D. Urbina, J. Giraldo, N. O. Tippenhauer, and A. Cárdenas. Attacking fieldbus communications in ICS: Applications to the SWaT testbed. In *Proceedings of Singapore Cyber Security Conference (SG-CRC)*, Jan. 2016.
[21] D. I. Urbina, J. Giraldo, A. A. Cardenas, J. Valente, M. Faisal, N. O. Tippenhauer, J. Ruths, R. Candell, and H. Sandberg. Survey and new directions for physics-based attack detection in control systems. *National Institute of Standards and Technology*, GCR 16-010, 2016.
[22] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105. ACM, 2016.