Revealing Ongoing Sensor Attacks in Industrial Control System via Setpoint Modification

Zhihao Dai*, Ligang He*, Shuang-Hua Yang†, and Matthew Leeke‡
*Department of Computer Science, University of Warwick, Coventry, UK
†Department of Computer Science, University of Reading, Reading, UK
‡School of Computer Science, University of Birmingham, Birmingham, UK
*{zhihao.dai, ligang.he}@warwick.ac.uk †shuang-hua.yang@reading.ac.uk §m.leeke@bham.ac.uk

***Abstract*—A variety of Intrusion Detection Systems (IDSs) for Industrial Control Systems have been proposed to detect attacks and alert operators. Passive and active detection schemes are characterised by whether or not they interact with the process under control, though both categories of approach have limitations relating to either known correlations in the process data or the use of explicit system modelling. We propose setpoint modification as a strategy to address those limitations. The approach superimposes Gaussian noises on setpoint values, which aids in revealing latent correlations between setpoints and measurements, thereby allowing machine learning-based IDSs to learn them during training and verify during inference. We show that by applying the approach to a linear system with PID control, statistical tests can be configured such that the distortion power of sensor attacks is nullified. Building on this foundation, we further adapt passive IDSs for active discovery of sensor attacks in a process-agnostic fashion. The proposed strategy is evaluated using a nonlinear and simulated industrial benchmark, affirming that the approach enhances intrusion detection performance when the specific sensor under consideration is targeted whilst incurring marginal cost. Finally, we explore changing setpoints concurrently when the attacker could manipulate an arbitrary sensor, which also boosts detection performance and motivates the exploration of setpoint selection.**

***Index Terms*—Industrial Control System, Active Defence, Intrusion Detection, Unsupervised Learning**

I. Introduction

An Industrial Control System, or ICS, can be described as a network of computers and devices responsible for monitoring and controlling an industrial process, such as manufacturing, power generation, or water treatment. Most ICSs are closed-loop control systems. At its core, the system functions by iteratively measuring the output using field devices such as sensors, calculating the difference between the output and the desired values, computing the control signals within controllers, and executing these signals via actuators. The goal is to maintain the output at the desired values, which are known as setpoints.

Compromising the security of an ICS can have significant financial and safety implications. A Kaspersky study [1] in 2022 showed that at least 31.8% of ICSs across the globe have detected cyberattacks, highlighting the urgent need to enhance the security of ICSs.

Intrusion detection represents the first step in countering ongoing attacks. Recent research has proposed many ICS-specific Intrusion Detection Systems (IDSs) [2], [3]. Once an intrusion is identified, manual or automated intervention [4] follows to prevent or, in the worst cases, mitigate process disruptions and physical damage. Based on its interactivity with the industrial process controlled by the ICS, an IDS can be classified as either passive or active. Most IDSs are passive, meaning that they do not alter the control functionality of the ICS while scanning for possible intrusions.

Active defence of ICSs involves altering the system behaviours in order to make the traces of attacks more visible or thwart attempted attacks. Examples of active defence include varying system configurations to obfuscate attacker's understanding of the system [5], [6] and coupling heterogeneous redundant controllers with a data-driven selection scheme to prevent single-point breaches [7].

Active defence is further exemplified by dynamic watermarking [8], which inserts private noises into control signals to expose malicious sensors. Actuators generate small random signals, known as watermarks, superimpose them on control signals, and subsequently validate measurements reported by the sensors using statistical tests based on process modelling. However, dynamic watermarking requires knowledge of the system dynamics, which might not be readily available.

In this paper, we propose a novel setpoint-based active defence strategy to reveal ongoing sensor attacks in ICSs. The approach is built on the observation that, under a closed-loop control policy, measurements controlled by the setpoints should closely follow the setpoints. In most ICSs, absent a shift in control targets or operation modes, such as ramping up production rates, setpoints are maintained at constant values.

Existing IDSs fail to recognise the importance of correlated relationships between setpoints and measurements, and do not consider setpoint values in defending ICS. This consequently limits IDSs to passively rely on remaining correlations among collected data or actively perturb

This work was supported by the Royal Society International Exchanges (IEC\NSFC\211297).

process actuation for detection purposes, i.e., dynamic watermarking. Additionally, many datasets do not record setpoint values, hindering passive IDSs from effectively capitalising on setpoint-measurement correlations.

The proposed approach superimposes a randomised Gaussian signal on the setpoint values via the commands from Human Machine Interface (HMI). This action causes the corresponding changes in the measurements and reveals previously latent correlations between setpoint values and the measurements. As the superimposition remains private and is only known to the HMI and the controller, the attacker, lacking access to both, cannot manipulate the measurements to match the changing setpoints without reporting the actual values. By enabling setpoint modification, passive machine learning-based IDSs can transition into active systems. The correlations between setpoints and measurements can be captured during training and are later verified during inference. To the best of our knowledge, existing research has not explored the utilisation of varying setpoints as an active defence strategy for ICS.

In this paper, we first investigate the case of a Linear Time-Invariant (LTI) system, where a sensor is controlled by a setpoint using a proportional–integral–derivative (PID) controller. By constructing an alternative PID control policy whose difference to the changing setpoint policy is proportional to the inserted Gaussian noise, we prove that statistical tests such as χ^2 can be applied in a dynamic watermarking [8] manner. The proposed approach ensures that an attacker who manipulates the sensor measurements cannot pass these tests without reporting the actual values. Consequently, the distortion power of sensor attacks is reduced to zero.

Building upon the proof, we develop a framework to implement the setpoints modification policy in any closed-loop controlled system, no matter they are linear and non-linear systems. The framework enables existing passive machine learning-based IDSs to actively discover sensor attacks. It comprises several stages, including setpoint identification, variance measurement, noise generation, data pre-processing and unsupervised learning of setpoint-measurement correlations.

To evaluate the framework, we conducted experiments on a non-linear simulated industrial benchmark for chemical production [9]. We evaluate the performance improvements in detection when noises are inserted into one setpoint and the attacker is targeting the corresponding sensor relative to static setpoint control. We then extend the experiments to simulate simultaneous changes in all setpoints, aiming to counter arbitrary sensor attacks. Finally, we examine the overheads of detection and the policy's impact on system output.

A. Contributions

In this paper, we make several contributions to intrusion detection for ICSs. In particular,

1) We propose a novel active defence framework based on the modification of setpoints, eliminating the need for system modelling as a prerequisite.
2) We prove the detection efficacy of the framework in a linear system, showing that changing setpoints under PID control provides perfect intrusion detection.
3) We conduct experiments in a non-linear system to demonstrate significant improvements in detection performance compared to static setpoint control, with only marginal overheads.

II. Related Work

In this section we consider the IDSs that are most relevant to our approach, including dynamic watermarking as an active defence strategy and unsupervised learning models trained on historical data. The proposed approach exists at the intersection of these areas.

A. Dynamic Watermarking

Physical watermarking was developed in [10], [11]. The approach inserts an additive Gaussian sequence into control signals generated by a fixed-gain Linear-Quadratic Gaussian controller. As replay attacks alter the covariance of the residue, that is, the difference between sensor readings and their estimates, and consequentially the new distribution can be derived, the authors design a Neyman–Pearson detector to reject the null hypothesis of ”no attacks” in favour of the alternative of ”under attacks” in the event of such an attack. The term dynamic watermarking was first coined in [8] to differentiate the dynamic technique in its ability to detect arbitrary sensor attacks with two statistical tests, measuring the variance of the residue before and after subtracting the impact of the watermark.

The modelling of system dynamics is a prerequisite of dynamic watermarking, limiting its usage to well-studied industrial processes. Whilst also being an active defence, our proposed strategy circumvents the modelling limitation by perturbing the setpoint values to expose ongoing attacks. It can boost intrusion detection for an ICS under closed-loop control.

B. Unsupervised IDSs

Supervised learning-based anomaly detection learns from labelled data. The latter could be time-consuming and even prohibitively expensive to capture. In comparison, passive IDSs built from unsupervised machine learning models thrives on the wealth of unlabelled data to model normal behaviours of the system and subsequently learn to identify anomalies, such as intrusions or faults. MAD-GAN [12] deploys two Long-Short-Term-Memory (LSTM) Recurrent Neural Networks (RNN) for generating and distinguishing fake samples during training and obtaining combined anomaly score during detection. GDN [13] first learns the adjacency matrix of sensors and then reconstructs the current sensor values from historical time series using a graph neural network directed by the leaned matrix.

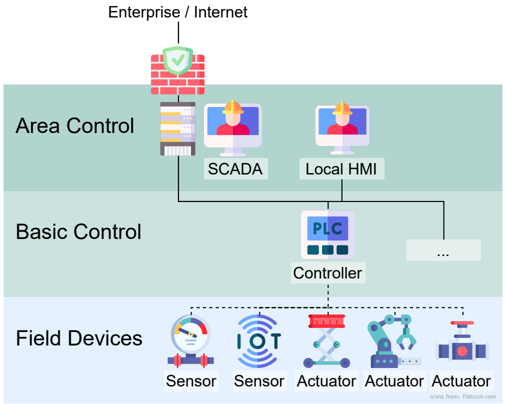


Fig. 1. System architecture.

Anomaly and error detection are widely studied [14], [15]. Anomaly detection algorithms targeted at general domains are also commonly found in IDSs for ICSs and hence we also include them in our evaluation. Isolation Forest [16] is a classic anomaly detection algorithm that randomly splits the feature space using a forest of trees to isolate anomalies. Robust Random Cut Forest (RRCF) [17] improves upon Isolation Forest with proportional dimension selection and a robust anomaly measure named Collusive Displacement to handle high-dimensional data streams. TranAD [18] is a state-of-the-art anomaly detection technique that refactors the transformer architecture [19] to enable adversarial training and self-conditioning to amplify deviations of the anomalies and make the model robust. In Section IV-C, we adapt unsupervised IDSs to leverage the latent setpoint-measurement correlations in our proposed framework and to advance the detection performance.

III. Attack Model

A. System Architecture

The ICS architecture in interest is a stripped-down version of the classical Purdue Reference Model of ICS, as depicted in Figure 1. Controllers in Level 1 interface with sensors and actuators to monitor and manipulate the industrial process. Supervisory Control and Data Acquisition (SCADA) and local Human Machine Interface (HMI) alike supervise the monitoring and control of the process in real time and are located in Level 2. Non-real-time components such as operational management and business logistics systems in Level 3 and above are omitted here for conciseness. While it mainly addresses security in one local plant, our approach can be independently deployed across distributed processes and plants in a large-scale ICS.

B. Problem Formulation

We consider the real-time detection of malicious actions by an actor, a.k.a., the attacker, against sensors in an ICS. The system consists of n sensors $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ and m actuators $U = \{u^{(1)}, u^{(2)}, \dots, u^{(m)}\}$, with each sensor $x^{(j)} \in X$ measuring a process variable and each actuator $u^{(i)} \in U$ acting on a manipulated variable. We assume the system is under closed-loop control and all setpoints $S = \{s^{(1)}, s^{(2)}, \dots, s^{(p)}\}$ have constant values. For a setpoint and its corresponding sensor $(s^{(i)}, x^{(j)})$ where the process variable measured by the sensor is directly controlled by the setpoint, we would expect its mean value equal to the setpoint value, i.e., $\bar{x}^{(j)} = s^{(i)}$, once the system stabilises. We assume that the measurement noise are i.i.d. and Gaussian, i.e., $x^{(j)} \sim \mathcal{N}(\bar{x}^{(j)},\ \sigma^2_{x^{(j)}})$ where $\sigma^2_{x^{(j)}}$ is the measurement variance.

When the system is stable, the attacker chooses their targets of sensors $O \subset X$ and performs malicious actions of blocking, delaying, or falsifying the measurements reported by the sensors. Such actions can be achieved through physical interference, planted backdoors, or communication hijacking. We assume controllers and actuators are honest and will always forward values reported by the sensors and execute the control policy, though the sensors might not be fully honest in themselves.

Some sensor attacks mislead the controllers into outputting erroneous control signals, degrades the performance, and even destabilises the system, e.g., reporting the pressure readings as high when they are low. Others feed deceptive or distorted measurements into data historians and production applications and cause long-term financial and operational inefficiencies. Additionally, sensor attacks can be combined with attacks targeting controllers and actuators to maximise the disruptive impact and to hinder detection, though the detection of which is outside the scope of this paper and left for future study.

C. Sensor Attacks

We consider three common categories of sensor attack. In the following formulation, all attacks start at time step k_a.

1) Denial-of-Service Attack: The attacker floods the communication channel between sensors and the controller with invalid requests. The controller will estimate the current measurement with historical values according to:

$$\hat{x}^{(j)}_k = \bar{x}^{(j)}_{1:k_a-1} \text{ for } k \geq k_a \tag{1}$$

where $x^{(j)} \in O$ and $\bar{x}^{(j)}_{1:k_a-1}$ is the mean value of $x^{(j)}$ from previous steps.

2) Time-Delayed Replay Attack: The attacker delays reporting of the actual measurement values by T_D hours.

$$\hat{x}^{(j)}_k = x^{(j)}_{k-T_D/\Delta t} \text{ for } k \geq k_a \tag{2}$$

where Δt is the sampling period.

3) Injection Attack: The attacker falsifies measurement values, providing values that share statistical characteristics with the actual values.

$$\hat{x}^{(j)}_k \sim \mathcal{N}(\alpha\bar{x}^{(j)}_{1:k-1},\ \beta^2\sigma^2_{x^{(j)}}) \text{ for } k \geq k_a \tag{3}$$

where α and β are amplification factors of mean and variance.

IV. Setpoints and Intrusion Detection

A. Linear Case I

An LTI system under PID control is first studied. We will show that adding Gaussian noises onto the setpoint is equivalent to applying dynamic watermarking to the actuation signals. Two statistical tests can thus be devised to validate the sensor in interest.

The LTI system has a single process variable x that is controlled by a setpoint s via manipulated variable u. System dynamics at any time step $k+1$ are described as:

$$x_{k+1} = ax_k + bu_k + w_{k+1} \tag{4}$$

where $w \sim \mathcal{N}(0,\ \sigma_w^2)$ is the process noise and i.i.d.

Applying the discrete form of PID controller, the manipulated variable in the previous step, $u_k = u_{k-1} + K_p I_k$ and

$$I_k = (1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t})e_k + (-1 - \frac{2T_d}{\Delta t})e_{k-1} + \frac{T_d}{\Delta t}e_{k-2} \tag{5}$$

where K_p, $K_i = K_p/T_i$, and $K_d = K_p T_d$ are proportional, integral, and derivative gains of the PID control. The error term e captures the difference between the process variable and the target value, $e_k = s_k - x_k$.

Instead of having a constant setpoint $s_k = \bar{s}$, we superimposes an i.i.d. and Gaussian process signal $\Delta s \sim \mathcal{N}(0,\ \sigma_{\Delta s}^2)$ on the control target $\bar{s}$ such that $s_k = \bar{s} + \Delta s_k$. An alternative PID control policy g_k dependent on k is $g_k(x_k) = u_{k-1} + K_p I' k$ and

$$I'_k = (1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t})e'_k + (-1 - \frac{2T_d}{\Delta t})e_{k-1} + \frac{T_d}{\Delta t}e_{k-2} \tag{6}$$

where $e'_k = \bar{s} - x_k$. The policy g_k, in comparison to u_k, effectively restores the setpoint value to $\bar{s}$ at time step k, while preserving the previous setpoint values.

Substituting (6) from (5), we get: $u_k = g_k(x_k) + \beta \Delta s_k$ where $\beta = K_p(1 + \Delta t/T_i + T_d/\Delta t)$. This shows that adding the noise Δs on actuation u is equivalent to applying dynamic watermarking over an alternate g_k. Substituting u_k in (4), the process variable x must satisfy $x_{k+1} - ax_k - bg_k(x_k) = b\beta\Delta s_k + w_{k+1}$. Since Δs_k and w_{k+1} are independent at any time step k, we have:

$$x_{k+1} - ax_k - bg_k(x_k) \sim \mathcal{N}(0,\ b^2\beta^2\sigma_{\Delta s}^2 + \sigma_w^2) \tag{7}$$

$$x_{k+1} - ax_k - bg_k(x_k) - b\beta\Delta s_k \sim \mathcal{N}(0,\ \sigma_w^2) \tag{8}$$

Definition 1. Distortion Power P of a sensor [8] is defined as:

$$P = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} v_k^2$$

where $v_k = \hat{x}_k - a\hat{x}_{k-1} - bg_{k-1}(\hat{x}_{k-1}) - b\beta\Delta s_{k-1} - w_k$ is the residual term at time step k and $\hat{x}$ is the measurement reported by the potentially malicious sensor. Assuming that initial reported value $\hat{x}_0 \equiv x_0$, then $P = 0$ if and only if the sensor is honest, i.e., $\hat{x}_k = x_k$ for any k.

We transform (7) and (8) into statistical tests for $\hat{x}$.

Test 1.

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} (\hat{x}_{k+1} - a\hat{x}_k - bg_k(\hat{x}_k))^2 = b^2\beta^2\sigma_{\Delta s}^2 + \sigma_w^2$$

Test 2.

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} (\hat{x}_{k+1} - a\hat{x}_k - bg_k(\hat{x}_k) - b\beta\Delta s_k)^2 = \sigma_w^2$$

Theorem 1. *If $\hat{x}$ satisfies both Tests 1 and 2, then $P = 0$. In other words, the sensor is honest.*

Our approach is inspired by and built upon the dynamic watermarking framework. Specifically, we use a similar method to construct Tests 1 and 2. For a detailed proof of the above theorem, we suggest referring to Section V in [8]. However, instead of directly adopting the proof, it is essential to make an important adjustment: in Theorem 1 of the aforementioned work, substitute $e[k]$ with $\beta\Delta s_k$ for our context. This adjustment, along with the updated v_k, highlights the uniqueness of our contribution, by considering an alternative control policy and showing the linearity of changing setpoint influence to construct statistical tests in our framework.

B. Linear Case II

A LTI system with n process variables and m manipulated variables is considered. We will show that adding independent Gaussian noises onto all setpoints is equivalent to applying dynamic watermarking to the actuation signals. Two statistical tests can thus be devised to validate the sensors in interest.

The LTI system dynamics can be described as:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + \boldsymbol{w}_{k+1} \tag{9}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$, $\boldsymbol{w} \in \mathbb{R}^n$, $\boldsymbol{w} \sim \mathcal{N}(0,\ \Sigma_w)$, $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$. The covariance matrix $\Sigma_w \in \mathbb{R}^{n\times n}$ has $\boldsymbol{\sigma}_w^2 \in \mathbb{R}^n$ on the diagonal and zeros elsewhere.

Under PID control, assuming that each manipulated variable $u^{(i)}$ is linked to a distinct setpoint $s^{(i)}$ to control the process variable $x^{(j)}$, we have $u_k^{(i)} = u_{k-1}^{(i)} + K_p^{(i)} I_k^{(i)}$ and

$$I_k^{(i)} = (1 + \frac{\Delta t}{T_i^{(i)}} + \frac{T_d^{(i)}}{\Delta t})e_k^{(i)} + (-1 - \frac{2T_d^{(i)}}{\Delta t})e_{k-1}^{(i)} + \frac{T_d^{(i)}}{\Delta t}e_{k-2}^{(i)} \tag{10}$$

where the error term $e_k^{(i)} = s_k^{(i)} - x_k^{(j)}$, $K_p^{(i)}$, $K_i^{(i)} = K_p^{(i)}/T_i^{(i)}$, and $K_d^{(i)} = K_p^{(i)} T_d^{(i)}$ are proportional, integral, and derivative gains of the i-th PID controller. Like in the previous section, we superimpose an i.i.d. and Gaussian process signal $\Delta s^{(i)} \sim \mathcal{N}(0,\ \sigma_{\Delta s^{(i)}}^2)$ on each control target such that $s_k^{(i)} = \bar{s}^{(i)} + \Delta s_k^{(i)}$.

Likewise, consider an alternate PID control policy $g_k^{(i)}$: $g_k^{(i)}(x_k^{(j)}) = u_{k-1}^{(i)} + K_p^{(i)} I_k^{(i)'}$ and

$$I_k^{(i)'} = I_k^{(i)} + (1 + \frac{\Delta t}{T_i^{(i)}} + \frac{T_d^{(i)}}{\Delta t})(\bar{s}^{(i)} - s_k^{(i)}) \qquad (11)$$

Substituting (11) from (10), we get $u_k^{(i)} = g_k^{(i)}(x_k^{(j)}) + q_k^{(i)}$ where $q_k^{(i)} = \beta^{(i)} \Delta s_k^{(i)}$, $\beta^{(i)} = K_p^{(i)}(1 + \Delta t / T_i^{(i)} + T_d^{(i)} / \Delta t)$. This shows that adding the noise $\Delta s^{(i)}$ on actuation $u^{(i)}$ is equivalent to applying dynamic watermarking over an alternate $g_k^{(i)}$.

The process variables $\boldsymbol{x}$ must now satisfy $\boldsymbol{x}_{k+1} - A\boldsymbol{x}_k - Bg_k(\boldsymbol{x}_k) = B\boldsymbol{q}_k + \boldsymbol{w}_{k+1}$. Since $\boldsymbol{q}_k$ and $\boldsymbol{w}_{k+1}$ are independent at any k, we have

$$\boldsymbol{x}_{k+1} - A\boldsymbol{x}_k - Bg_k(\boldsymbol{x}_k) \sim \mathcal{N}(0,\ B\Sigma_q B^T + \Sigma_w) \qquad (12)$$

where the covariance matrix Σ_q has $\sigma_q^2 \in \mathbb{R}^m$ on the diagonal and zeros elsewhere.

Definition 2. Distortion Power P of sensors [8] is defined as

$$P = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} ||\boldsymbol{v}_k||_2$$

where $\boldsymbol{v}_k = \hat{\boldsymbol{x}}_k - A\hat{\boldsymbol{x}}_{k-1} - Bg_{k-1}(\hat{\boldsymbol{x}}_{k-1}) - B\boldsymbol{q}_{k-1} - \boldsymbol{w}_k$ and $\hat{\boldsymbol{x}}$ is the measurement reported by sensors.

We transform (12) into a statistical test for $\hat{\boldsymbol{x}}$.

Test 3.

$$\lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} (\hat{\boldsymbol{x}}_{k+1} - A\hat{\boldsymbol{x}}_k - Bg_k(\hat{\boldsymbol{x}}_k))(\hat{\boldsymbol{x}}_{k+1} - A\hat{\boldsymbol{x}}_k - Bg_k(\hat{\boldsymbol{x}}_k))^T = B\Sigma_q B^T + \Sigma_w$$

Multiplying $q_k^{(i)}$ with $\boldsymbol{v}_{k+1} + B\boldsymbol{q}_k + \boldsymbol{w}_{k+1}$, we drive a second test on every actuator.

Test 4.

$$\lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} q_k^{(i)}(\hat{\boldsymbol{x}}_{k+1} - A\hat{\boldsymbol{x}}_k - Bg_k(\hat{\boldsymbol{x}}_k)) = B_{\cdot,i} {\sigma_q^{(i)}}^2$$

Theorem 2. *If $\hat{\boldsymbol{x}}$ satisfies both Tests 3 and 4, B is of rank n, and $\hat{\boldsymbol{x}}_0 \equiv \boldsymbol{x}_0$, then $P = 0$. In other words, the sensors are honest.*

The proof for this theorem closely follows the arguments presented in Theorem 6 of [8]. We apply a similar methodology while also considering the unique characteristics of our problem scenario. This concludes our analysis under linear cases, where we have inserted noises into setpoints and developed a set of tests to validate the honesty of sensors under PID control. The statistical tests utilises the principles of dynamic watermarking and leverages the relationships between the setpoint, manipulated variables, and process variables. The proofs here provide the basis for transferring setpoint modification into general cases for intrusion detection, where system models remain inaccessible or involve non-linearity.

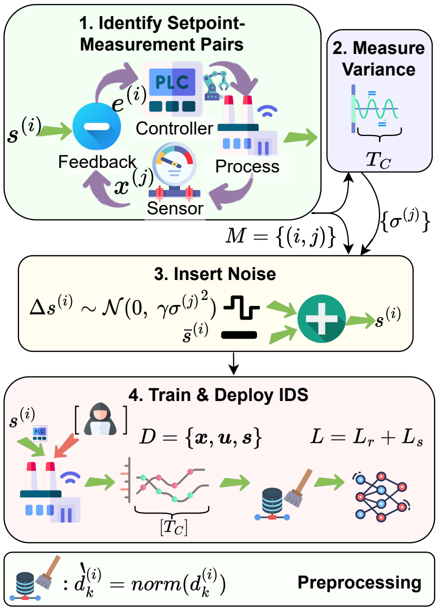


Fig. 2. Changing setpoint-based intrusion discovery framework.

C. General Cases

The changing setpoint scheme is extended to arbitrary closed-loop systems to make the setpoint-measurement correlations learnable to machine learning-based IDSs and boost the detection performance. Figure 2 depicts an overview of the scheme and can be narrated as follows. a) Setpoints and their corresponding sensors are first identified from control policy and the measurement variances measured under normal operations. The goal of this step is to identify setpoint candidates for noise insertion and scale the noise variances accordingly. b) Sequences of zero-mean Gaussian variables are then generated independently and added onto constant setpoints. This is identical to the noise insertion step in linear cases, with subset being applied to satisfy operational constraints. c) During training, process data under normal operations are collected for training IDSs in an unsupervised manner. During inference, IDSs output anomaly scores and generate alerts based on real-time process data.

1) Setpoint Identification: We identify each of the setpoints $s^{(i)}$ and its corresponding process variable $x^{(j)}$ in the ICS, where $x^{(j)}$ is directly controlled by $s^{(i)}$ via a control loop, along with the constant value $\bar{s}^{(i)}$ of $s^{(i)}$. The bijection relationships are stored in $M = \{(i,j) |\ x^{(j)}$ is controlled by $s^{(i)}\}$. All the information are extricable from the control policy implemented in the system.

2) Variance Measurement: The system is then placed under normal operations and constant setpoint values for T_C hours. For each $(i,j) \in M$, the variance $\sigma_{x^{(j)}}^2$ of $x^{(j)}$ of is measured. This helps determine the variances of noises added onto setpoints since large variances destabilise the system while small variances leave hidden the setpoint-measurement correlations.

3) Noise Insertion: Unlike in Sections IV-A and IV-B where all setpoints are being changed, a subset of setpoints $M_C \subseteq M$ are selected for changes. This is due to the fact that some setpoints might be safety- or quality-sensitive and thus prohibited from changes. To change setpoint values dynamically, a sequence of i.i.d. and Gaussian variables $\Delta s^{(i)} \sim \mathcal{N}(0,\ \gamma\sigma_{x^{(j)}})$ is generated for each $s^{(i)}$ where $(i,j) \in M_C$. Noise sequence $\Delta s^{(i)}$ is independent of every other noise sequence $\Delta s^{(q)}$ where $q \neq i$ or the measurement noises. For each setpoint, the generated noise is then added onto $\bar{s}^{(i)}$ every T hours, such that $s_k^{(i)} = \bar{s}^{(i)} + \Delta s_q^{(i)}$ for $(q-1)T/\Delta t \leq k < qT/\Delta t$ where $q \in \mathbb{N}$. Unlike in Sections IV-A and IV-B where setpoints change at every step k, setpoints are only changing periodically to prevent oversaturating the control algorithm as real-world systems might react slowly to setpoint changes.

4) Data Collection and Pre-processing: During both training and detection stages, process data $D = \{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{s}\}$ is collected where $\boldsymbol{x}$ is the time series of all process variables, $\boldsymbol{u}$ of all manipulated variables, and $\boldsymbol{s}$ of all setpoints. All variables in D are normalised to zero mean and unit variance using the training data. At each time step k, the IDS has only access to data collected up until k, that is, $D_{1:k} = \{\boldsymbol{x}_{1:k}, \boldsymbol{u}_{1:k-1}, \boldsymbol{s}_{1:k-1}\}$.

5) Unsupervised Learning: For training, the system is placed under normal operations and changing setpoints for T_C hours. Data D is collected for training the IDS in an unsupervised fashion since no attack labels are provided.

The IDS models the normal behaviours of the system and learns to score the anomalies given the current state. For tree-based methods such as Isolation Forest and RRCF, anomaly scores are measured by depths to the root or shifts in complexity of data points. For reconstruction-based methods, anomaly scores are the mean squared reconstruction errors. To generate alerts from raw scores, threshold selection methods such as Peaks-Over-Threshold [20] can be applied and extreme anomaly scores are subsequently flagged as attacks.

V. Experiments

A. System Characteristics

A chemical production benchmark [21] is simulated for the evaluation of the framework. The Tennessee Eastman Process (TEP) [9] includes two gas-liquid exothermic reactions occurring concurrently to produce a product mix, as depicted in Figure 3. It has 41 process variables and 12 manipulated variables. Control strategy in [22] is adopted with 19 control loops and constant setpoint values, each ran by a PI controller.

Seven setpoints and their corresponding process variables are identified within the control policy. The process data of each identified variable under normal operations for $T_C = 72$ hours is collected with the default sampling period $\Delta t = 0.01$ hour. Along with the constant setpoint values, the standard deviations of the process variables are reported in Table I.

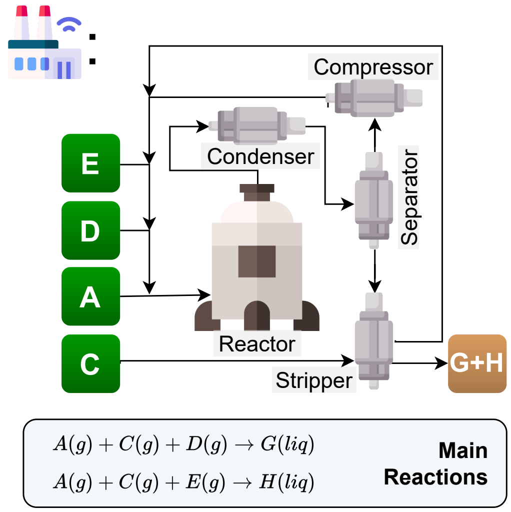

Compressor
Condenser
Separator
E
D
A
C
Reactor
Stripper
G+H
$A(g) + C(g) + D(g) \rightarrow G(liq)$
$A(g) + C(g) + E(g) \rightarrow H(liq)$
Main Reactions

Fig. 3. Tennessee Eastman process.

TABLE I
Setpoints and Corresponding Process Variables.

Setpoint	$\bar{s}^{(i)}$	Variable	$\sigma_{x^{(j)}}$	Unit
$s^{(1)}$: Production	22.89	$x^{(17)}$	0.11	m^3/hour
$s^{(2)}$: Stripper Level	50.00	$x^{(15)}$	1.03	%
$s^{(3)}$: Separator Level	50.00	$x^{(12)}$	1.00	%
$s^{(4)}$: Reactor Level	65.00	$x^{(8)}$	0.50	%
$s^{(5)}$: Reactor Pressure	2800.00	$x^{(7)}$	0.94	kPa
$s^{(6)}$: Component G in Product	53.80	$x^{(40)}$	0.52	Mole %
$s^{(7)}$: Reactor Temperature	122.90	$x^{(9)}$	0.01	°C

B. Reactor Level Attacks

1) Attack Design: Attacks against the reactor level sensor $x^{(8)}$ are first considered. Table II lists nine attack instances of three different types as in Section III-C, where $T_a = k_a\Delta t$ is the starting time in hours. During testing, only one attack instance is launched at a time, hence a total of nine runs are collected as test cases. The attack concludes when the process ends at T_C's-th hour or the safety constraints are breached, whichever is sooner.

2) Noise Insertion and Data Collection: Noises are randomly generated and inserted into $s^{(4)}$, as described in Section IV-C3. Parameters $\gamma = 1$ and $T = 1$ are chosen to accommodate the noise and the time scale of the system dynamics. Process data is collected and normalised both under normal operations and under individual attack instances.

3) Unsupervised Training: Five unsupervised IDSs are trained on normal process data and their performance under attacks are compared with and without changing setpoint enabled. Both tree-based methods (Isolation Forest/if and RRCF/rr) and deep learning-based methods

TABLE II
Reactor Level Attack Set.

No.	Sensor	Type	T_a	Parameters
1	$x^{(8)}$	DoS	59	
2	$x^{(8)}$	DoS	30	
3	$x^{(8)}$	DoS	42	
4	$x^{(8)}$	Replay	28	$T_D = 4$
5	$x^{(8)}$	Replay	35	$T_D = 8$
6	$x^{(8)}$	Replay	44	$T_D = 17$
7	$x^{(8)}$	Injection	29	$\alpha = 1.00, \beta = 1$
8	$x^{(8)}$	Injection	57	$\alpha = 1.05, \beta = 1$
9	$x^{(8)}$	Injection	47	$\alpha = 1.00, \beta = \sqrt{2}$

TABLE III
IDS Performance under Reactor Level Attacks.

	DoS			**Replay**			**Injection**		
IDS	AUC	F1	MTTD	AUC	F1	MTTD	AUC	F1	MTTD
if	43.25	0.91	1.32	56.19	**2.81**	1.92	66.42	**9.17**	0.23
if*	**50.69**	**2.26**	3.06	**58.65**	2.62	1.89	**70.07**	8.00	1.93
rr	43.80	1.03	1.94	51.22	1.09	0.65	55.39	1.77	1.90
rr*	**46.01**	**2.09**	0.74	**52.32**	**1.44**	1.90	**57.98**	**2.17**	1.94
mg	92.46	**85.70**	3.52	87.65	75.79	5.62	95.04	89.71	3.12
mg*	**93.30**	82.35	3.70	**98.40**	**87.75**	2.10	**99.58**	**93.19**	0.70
gd	92.59	**86.06**	3.57	87.15	74.64	6.17	95.27	90.04	2.63
gd*	**93.04**	83.23	3.59	**98.48**	**88.18**	2.05	**99.57**	**93.55**	0.61
ta	92.53	**86.21**	3.57	87.06	74.78	5.71	95.23	90.13	2.63
ta*	**93.13**	82.85	3.68	**98.46**	**89.52**	1.42	**99.56**	**93.37**	0.62

(MAD-GAN/mg, GDN/gd, and TranAD/ta) are included in the comparison. Publicly available implementations [18], [23]–[25] are adapted. Default hyper-parameters are used unless otherwise specified.

4) IDS Comparison: Table III provides the Area Under the Receiver Operating Characteristic Curve (AUC), F1 scores in percentages, and Mean Time to Detect (MTTD) in the unit of hours for the five IDSs under both settings. Training and evaluation with changing setpoint enabled is marked with *. Results are averaged across three different seeds.

AUC is computed as the area under of curve of True Positive Rates vs False Positive Rates across different thresholds. The True Positive Rate is defined as $TP/(TP+FN)$ and False Positive Rate is $FP/(FP+TN)$ where TP, TN, FP, and FN are the true positives, true negatives, false positives, and false negatives of the intrusion alerts. Precision is defined as $TP/(TP+FP)$ and recall is $TP/(TP+FN)$. The F1 score is the harmonic mean of the precision and the recall. MTTD is the average duration between T_a and the timestamp of the first generated alert since. In the case of no alerts generated, T_C is used.

To offer a fair comparison of IDSs, the best performance under all settings are considered. In terms of AUC, deep learning based methods consistently outperform the tree-based methods by at least 42.61% (MAD-GAN vs Isolation Forest), 39.83% (GDN vs Isolation Forest), and 29.51% (MAD-GAN vs Isolation Forest) in DoS, Replay and Injection attacks. The results are not surprising given the complexity of deep learning methods, though some recent studies [26], [27] found that shallow techniques could outperform deep models on large-scale anomaly archives.

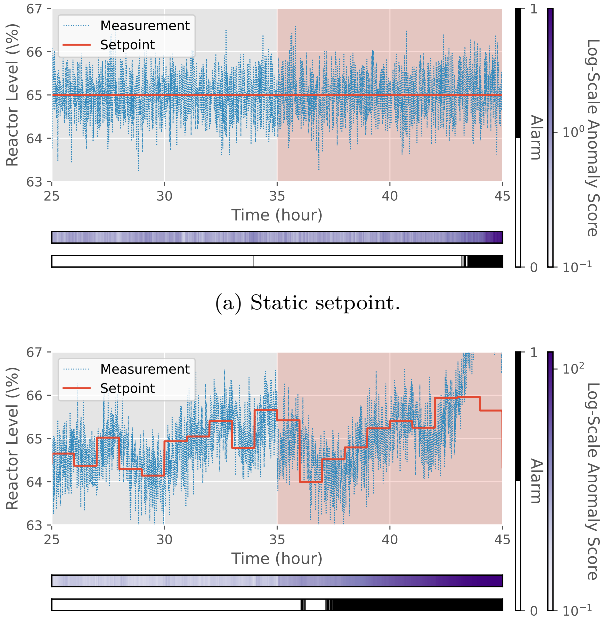
All deep learning methods deliver the same level of performance in both AUC and F1 scores. Despite the performance similarity, the complexity of deep learning models differ greatly with MAD-GAN, GDN, and TranAD each employing approximately 16K, 7K, and 330K trainable parameters. We also note that the complexity do not necessarily align with the training and inference time of the models, with GDN taking 165 seconds to train, compared to only 22 seconds by TranAD. For this reason, TranAD is used as the default IDS in subsequent sections for experiments unless specified otherwise.

5) Static versus Changing Setpoint: With the changing setpoint in place, improvements in AUCs are observed across attack categories regardless of the IDS used. The changing setpoint boosts the previously best AUCs by 10.83% (GDN vs MAD-GAN) in Replay, 4.31% (MAD-GAN vs GDN), and 0.71% (MAD-GAN vs GDN) in DoS. Improvements in F1 scores are also observed with 13.73% (TranAD vs MAD-GAN) in Replay and 3.42% (GDN vs TranAD) in Injection, despite a small 2.98% (GDN vs TranAD) dip in DoS. The last dip suggest that the current threshold selection scheme might not be optimal and highlights the fact that the threshold-independent AUC is a more robust metric for detection performance. Note that MTTD is also influenced by threshold selection as a lowest possible threshold would generate alerts on all data points and thus zero MTTD and full recall but a threshold like this becomes less useful for deployment with extremely high false positives.

For TranAD specifically, AUCs increase by 0.60%, 11.40%, and 4.33% in DoS, Replay, and Injection. AUC improvements are reported in seven out of nine attacks (not shown here due to page limit). Our approach is effective in boosting the intrusion detection performance.

Figure 4 depicts the system under an attack starting at the 35-th hour. Readings of reactor level with a delay of 8 hours are replayed to the controller. With a static setpoint, the setpoint-measurement correlation remains hidden to the IDS, preventing a timely detection of the attack. With a changing setpoint, however, the correlation is apparent before the attack and is soon broken after the attack has started. That helps the IDS to identify the attack in time.

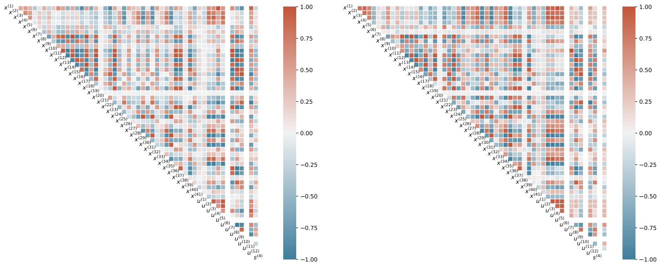
To explore the detection improvement, we consider the Pearson bi-variate correlations. Figure 5 presents the correlation coefficients between each pair of variables in D in matrix form. Since the correlation coefficient matrix is symmetric with $r^{(i,j)} = r^{(j,i)}$, only the upper triangle of the matrix is shown. All setpoints except $s^{(4)}$ are omitted here as they remain constant regardless of the control strategy in this case. By its nature, the changing setpoint strategy produces one more column, i.e., the rightmost $s^{(4)}$, of correlations. Additionally, stronger correlations are observed among pre-existing bi-variate relationships. The



(a) Static setpoint.

(b) Changing setpoint.

Fig. 4. Reactor level and intrusion detection under Attack 5.



(a) Static setpoint.

(b) Changing setpoint.

Fig. 5. Pearson correlations under Attack 5.

changing setpoint approach makes the latent setpoint-measurement correlations visible and magnifies the inter-measurement correlations. This allows IDSs to model the correlated patterns in unlabelled data during training and exploit such relationships for detection.

C. *Arbitrary Sensor Attacks*

1) Attack Design and Noise Insertion: Arbitrary sensor attacks are considered where the attacker can manipulate the readings by any sensor in the system. Table IV lists 15 sensor attacks that might be performed by an attacker, spanning seven process variables directly involved in the control policy. To implement our changing setpoint strategy, Gaussian noises are inserted onto $s^{(i)}$ with $\gamma = 0.3$ and $T = 1$ for all $(i, j) \in M_C$ and $M_C = M$.

2) IDS Performance: Table IV also shows the AUCs of TranAD with and without the changing setpoints. AUC improvements up to 27.49% are observed in 11 out of 15 attacks. Overall, the AUC increases by 6.01% from 62.87% without the changing setpoint.The results are in line with the improvements we reported in Section V-B, where inserting noises into one setpoint helps detect attacks to various degrees. Similar to Figure 5, we observe stronger bi-variate correlations between variables and that latent setpoint-measurement correlations are uncovered.

TABLE IV
Arbitrary attack set and IDS performance.

No.	Sensor	Type	T_a	Parameters	AUC	AUC*
1	$x^{(8)}$	DoS	25		**85.38**	82.01
2	$x^{(12)}$	DoS	58		60.10	**60.57**
3	$x^{(12)}$	Replay	45	$T_D = 3$	83.52	**90.05**
4	$x^{(12)}$	Injection	50	$\alpha = 1.00, \beta = \sqrt{2}$	42.25	**69.74**
5	$x^{(15)}$	DoS	51		**57.27**	42.59
6	$x^{(15)}$	Replay	48	$T_D = 4$	83.44	**92.71**
7	$x^{(15)}$	Injection	36	$\alpha = 1.00, \beta = \sqrt{2}$	54.32	**60.75**
8	$x^{(17)}$	Replay	56	$T_D = 11$	75.72	**99.11**
9	$x^{(23)}$	Replay	60	$T_D = 15$	**75.60**	69.73
10	$x^{(23)}$	Injection	29	$\alpha = 1.00, \beta = \sqrt{2}$	**84.59**	80.25
11	$x^{(25)}$	DoS	54		42.20	**52.00**
12	$x^{(25)}$	Replay	49	$T_D = 13$	76.48	**91.32**
13	$x^{(40)}$	DoS	30		34.41	**45.66**
14	$x^{(40)}$	Replay	48	$T_D = 3$	50.57	**52.17**
15	$x^{(40)}$	Injection	47	$\alpha = 1.00, \beta = 1$	37.19	**44.48**

D. *Performance Impact*

This section investigates the impact of the changing setpoint on ICS's performance. Concretely, we compare the IDS overheads, the production rates, and economic costs of the system before and after the changing setpoint is enabled. Settings in Section V-B are adopted.

1) IDS Overheads: Since the framework does not perturb the training and deployment pipeline for IDS but only the data, the overheads incurred will be minor. Indeed, the training of TranAD with the changing setpoint takes an average of 22.07 seconds on T_C hours of process data, a 0.59% increase compared to 21.94 seconds without the changes. Likewise, inference on each sampling point takes an average of 8.79×10^{-4} seconds, compared to 7.97×10^{-4} seconds without the changes. Inference times are negligible compared to the sampling period $\Delta t = 0.01$ hour.

2) Production: We compare the production rates under normal operations for T_C hours. The production rates share the same mean value of 22.89 m^3/hour, being the control objective in Table I. The changing setpoint incurs a marginally larger standard deviation (0.12 versus 0.11) for production. That could be attributed to the its dynamic nature.

3) Running Costs: Similarly, we measure and compare the economic costs of running the chemical production system under normal operations for T_C hours. The calculation formula is given in the TEP paper [9]. The average costs with the changing setpoint are \$113.94 USD/hour, a 2.42% increase over \$113.92 USD/hour without the changes. Such increase is minimal and should not result in large financial penalty for implementing our framework.

VI. Conclusion

In this work, we proposed a novel active defence framework that superimposes Gaussian noise signals on static setpoint values and transforms passive IDSs into proactive detectors. The approach reveals latent setpoint-measurement correlations, significantly enhancing the detection capabilities of existing IDSs. The overarching contribution lies in the independence from explicit system dynamics modelling, a requirement prevalent in existing methods. The effectiveness has been proven both theoretically and empirically. In a linear system, setpoint modification was shown to enable perfect intrusion detection. In a non-linear system, experiments demonstrated significant detection improvements over static setpoint control with only marginal overheads. The exploration of concurrent setpoint changes has also shown potential for detection enhancement and opened new research avenues.

References

[1] K. I. CERT, "Threat landscape for industrial automation systems. statistics for h1 2022," 2022. [Online]. Available: https://ics-cert.kaspersky.com/publications/reports/2022/09/08/threat-landscape-for-industrial-automation-systems-statistics-for-h1-2022/

[2] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100516, 2022.

[3] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93 083–93 108, 2020.

[4] J. Mern, K. Hatch, R. Silva, C. Hickert, T. Sookoor, and M. J. Kochenderfer, "Autonomous attack mitigation for industrial control systems," in *Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, 2022, pp. 28–36.

[5] G. L. Babineau, R. A. Jones, and B. Horowitz, "A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions," in *Proceedings of the IEEE Conference on Technologies for Homeland Security*, 2012, pp. 99–104.

[6] P. Griffioen, S. Weerakkody, and B. Sinopoli, "A moving target defense for securing cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2016–2031, 2021.

[7] Z. Dai, M. Leeke, Y. Ding, and S. Yang, "A heterogeneous redundant architecture for industrial control system security," in *Proceedings of the 27th IEEE Pacific Rim International Symposium on Dependable Computing*. Los Alamitos, CA, USA: IEEE Computer Society, December 2022, pp. 89–97.

[8] B. Satchidanandan and P. R. Kumar, "Dynamic watermarking: Active defense of networked cyber–physical systems," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 219–240, 2017.

[9] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, no. 3, pp. 245–255, March 1993.

[10] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, 2009, pp. 911–918.

[11] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 93–109, 2015.

[12] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.

[13] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4027–4035, May 2021.

[14] A. Jhumka and M. Leeke, "Issues on the design of efficient fail-safe fault tolerance," in *Proceedings of the 20th IEEE International Symposium on Software Reliability Engineering*, 2009, pp. 155–164.

[15] ——, "The early identification of detector locations in dependable software," in *Proceedings of the 22nd IEEE International Symposium on Software Reliability Engineering*, 2011, pp. 40–49.

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[17] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *Proceedings of the 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. JMLR, June 2016, pp. 2712–2721.

[18] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," *Proceedings of VLDB*, vol. 15, no. 6, pp. 1201–1214, 2022.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. California, USA: Curran Associates, Inc., December 2017.

[20] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2828–2837.

[21] A. Bathelt, N. L. Ricker, and M. Jelali, "Revision of the tennessee eastman process model," in *Proceedings of the 9th IFAC Symposium on Advanced Control of Chemical Processes*, vol. 48, no. 8, 2015, pp. 309–314.

[22] N. Lawrence Ricker, "Decentralized control of the tennessee eastman challenge process," *Journal of Process Control*, vol. 6, no. 4, pp. 205–221, 1996.

[23] A. Bhatnagar, P. Kassianik, C. Liu, T. Lan, W. Yang, R. Cassius, D. Sahoo, D. Arpit, S. Subramanian, G. Woo, A. Saha, A. K. Jagota, G. Gopalakrishnan, M. Singh, K. C. Krithika, S. Maddineni, D. Cho, B. Zong, Y. Zhou, C. Xiong, S. Savarese, S. Hoi, and H. Wang, "Merlion: A machine learning library for time series," 2021.

[24] ——, "Merlion: A machine learning library for time series," https://github.com/salesforce/Merlion, 2021.

[25] S. Tuli, G. Casale, and N. R. Jennings, "Tranad," https://github.com/imperial-qore/TranAD, 2022.

[26] F. Rewicki, J. Denzler, and J. Niebling, "Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series," *Applied Sciences*, vol. 13, no. 3, p. 1778, January 2023.

[27] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," in *Proceedings of the 36th Neural Information Processing Systems*, 2022.