Can Industrial Intrusion Detection Be SIMPLE?

Konrad Wolsing[1,2], Lea Thiemt[2], Christian van Sloun[2],
Eric Wagner[1,2], Klaus Wehrle[2], and Martin Henze[3,1]

[1] Cyber Analysis & Defense, Fraunhofer FKIE, Germany
{firstname.lastname}@fkie.fraunhofer.de
[2] Communication and Distributed Systems, RWTH Aachen University, Germany
{lastname}@comsys.rwth-aachen.de
[3] Security and Privacy in Industrial Cooperation, RWTH Aachen University,
Germany henze@cs.rwth-aachen.de

Abstract. Cyberattacks against industrial control systems pose a serious risk to the safety of humans and the environment. Industrial intrusion detection systems oppose this threat by continuously monitoring industrial processes and alerting any deviations from learned normal behavior. To this end, various streams of research rely on advanced and complex approaches, i.e., artificial neural networks, thus achieving allegedly high detection rates. However, as we show in an analysis of 70 approaches from related work, their inherent complexity comes with undesired properties. For example, they exhibit incomprehensible alarms and models only specialized personnel can understand, thus limiting their broad applicability in a heterogeneous industrial domain. Consequentially, we ask whether industrial intrusion detection indeed has to be complex or can be SIMPLE instead, i.e., Sufficient to detect most attacks, Independent of hyperparameters to dial-in, Meaningful in model and alerts, Portable to other industrial domains, Local to a part of the physical process, and computationally Efficient. To answer this question, we propose our design of four SIMPLE industrial intrusion detection systems, such as simple tests for the minima and maxima of process values or the rate at which process values change. Our evaluation of these SIMPLE approaches on four state-of-the-art industrial security datasets reveals that SIMPLE approaches can perform on par with existing complex approaches from related work while simultaneously being comprehensible and easily portable to other scenarios. Thus, it is indeed justified to raise the question of whether industrial intrusion detection needs to be inherently complex.

1 Introduction

Cyberattacks against Industrial Control Systems (ICSs) with the goal of financial gains, damaging equipment, or even risking human lives by blocking normal operations or injecting false data are becoming more prevalent [7]. Recent examples of such attacks include the attempted poising of a Florida city's water supply by increasing its sodium hydroxide concentration [59]. Increased connectivity to the Internet is one driver behind this development, but practice shows that even air-gapped systems are not secure against sophisticated attacks anymore [7].

Besides preventive security mechanisms, e.g., integrity protection, recent research has seen a rising interest in detecting intrusions into industrial networks. Such Intrusion Detection Systems (IDSs) passively monitor processes to alert about anomalous behavior before any real damage can occur and promise to provide a non-intrusive, retrofittable, and easily deployable security solution. In contrast to traditional IDSs known from office or data center environments, Industrial Intrusion Detection Systems (IIDSs) have the unique advantage that they can leverage the predictability and repetitiveness of ICSs to identify even advanced and stealthy attacks [11]. Auspicious results are reported by process-aware IIDSs, which incorporate the physical state of the monitored ICS into their decision-making. Consequently, they received tremendous interest from the research community, with state-of-the-art IIDSs based mainly on machine learning, e.g., artificial neural networks [50], graph theory [57], or linear algebra [11].

The powerful underlying IIDS methodologies yield promising detection performances, however, at the cost of complexity, requiring resource-intensive operations and hindering generalizability [51, 76]. Furthermore, the alarms raised by, e.g., artificial neural networks, are often not explainable, making it challenging to derive concrete actions for mitigating attacks [28]. Meanwhile, we observe that attacks, like the one on Florida's water supply [59], lead to apparent deviations from normal operations. Therefore, in this paper, we pose the question: Do IIDSs indeed have to be complex to reliably detect attacks on industrial systems?

To answer this question, we study to what extent IIDSs can be simple, e.g., merely keeping track of the minimum and maximum of observed process values, and whether they perform on par with complex related work. Surprisingly, such approaches have obtained no attention so far, likely as they have never been considered suitable in traditional networks, e.g., data centers. However, as we show in this paper, this conclusion is not necessarily true for industrial networks due to the repetitive and predictable nature of their underlying physical processes. SIMPLE IIDSs avoid many of the drawbacks of complex solutions as they are *Sufficient* to detect most attacks, operate *Independently* of parameters, provide *Meaningful* alerts, are *Portable* to other industrial scenarios, require only *Local* process knowledge, and can be realized using *Efficient* computational operations.

Contributions. More precisely, we present the following contributions to determine whether the complexity of state-of-the-art IIDSs is indeed needed:

- We analyze the current state of IIDS research. Our study of 70 approaches unveils limitations w.r.t. deployability, computational complexity, generalizability, focus on non-stealthy attacks, and incomprehensibility of alarms (Sec. 3).
- To assess whether industrial intrusion detection needs to be inherently complex, we design four intentionally SIMPLE IIDSs[1] characterized by straightforward, relatable, and easy-to-compute concepts (Sec. 4).
- We then compare the performance of our SIMPLE IIDSs against state-of-the-art complex related work alongside four industrial datasets. Our results show that SIMPLE IIDSs detect more attacks than complex related work detects on average and can be ported effortlessly across industrial domains (Sec. 5).

[1] Implementation available at: https://github.com/fkie-cad/ipal_ids_framework

2 Intrusion Detection in Industrial Control Systems

Industrial networks are responsible for operating today's manufacturing plants and critical infrastructure. Due to their high degree of automation, industrial communication almost exclusively relies on machine-to-machine communication between sensors measuring the current physical environment and actuators interacting with the external world. In contrast to the unpredictable behavior of traditional networks induced by spontaneous human interactions, industrial networks exhibit regularly repeating and predictable behavior [76]. These patterns only change due to failures or after seldom structural changes to the physical processes, e.g., a manufacturing plant being configured for a new product.

In the past, industrial networks were isolated from the Internet and therefore assumed secure; Hence no protection mechanisms, like authentication or encryption, were integrated. Nowadays, as more connectivity is demanded, e.g., for remote monitoring or cross-production plant optimization, these networks can no longer be considered secure [7]. While retrofitting preventive security mechanisms requires expensive downtime and is often inapplicable due to legacy hardware and resource constraints, IIDSs offer a unique alternative opportunity.

IDSs for traditional office and server networks, e.g., Zeek and Snort, usually define rules for typical malware and attack patterns that trigger an alarm indicating known suspicious activities. However, due to the industry's diversity, attacks are usually unique and targeted, significantly reducing their efficiency.

Contrary, IIDSs can take advantage of the abundance of sensor and actuator data exchanged over the network. The fact that processes behave predictably according to physical constraints enables a great potential for anomaly detection training on benign data and alerting deviations. Specifically, *process-aware* IIDSs report excellent detection capabilities, as recent surveys emphasize [23, 37]. However, their effectiveness is still questionable, as many detection methodologies are over-engineered to detect specific attacks in specific systems and are thus not suitable to detect new and tailored attacks as often observed in industrial networks [51, 76]. Still, process-aware anomaly detection offers the opportunity to passively and retroactively protect manufacturing plants and critical infrastructure against powerful attacks.

3 The State of Industrial Intrusion Detection Research

Given the promise of IIDSs to offer an easily retrofittable solution to secure industrial networks, the research landscape around industrial intrusion detection has experienced huge attention across all industrial domains. Different surveys put significant effort into providing a holistic overview of this scattered research field [23, 37]. Surprisingly and contradicting the initial promise of an *easily* retrofittable solution, the current state-of-the-art, governed by all kinds of machine learning, comes at the cost of a complexity overhead, e.g., in terms of demanded computational resources, limited generalizability across industrial domains, or incomprehensiveness of the detection models and emitted alerts. In the end, this hinders the widespread deployment of security mechanisms.

Table 1. Complex approaches govern the current state of industrial anomaly detection research, and only a few evaluation datasets, like SWaT, are being dominantly used.

	Detection Method (unique publications)	SWaT (63)	WADI (24)	HAI (6)
Artificial NNs (44)	Autoencoder (15)	[12, 16, 26, 36, 40, 41, 47, 52, 58, 62, 68, 75, 77, 83]	[12, 62]	[45]
	Other NN (12)	[1, 20, 22, 29, 34, 49, 50, 65, 70, 71, 73, 78]	[1, 22, 29, 34, 50, 70, 73]	–
	RNN (9)	[6, 30, 39, 53–56]	[30, 56]	[13, 42]
	GAN (5)	[5, 14, 44, 60, 64]	[14, 60]	–
	DNN (3)	[43, 46, 66]	–	–
Graphs (6)	Automata (3)	[15, 57, 79]	[79]	–
	Other (3)	[17, 35]	[17, 35, 69]	–
Miscellaneous (20)	Invariants (3)	[33, 74, 82]	[33, 82]	–
	Linear algebra (3)	[11, 24, 63]	–	–
	Classifier (2)	[9, 25]	[25]	–
	Fingerprinting (2)	[2, 4]	[2]	–
	Matrix Profiles (2)	[8, 10]	–	–
	Other (8)	[18, 32, 48, 80, 81]	[67, 80, 81]	[21, 48, 61]

To shine light on this issue and precisely understand the degree of complexity in related work, we systematically analyze the IIDS research landscape. We set out to assess IIDSs that implement anomaly detection, i.e., train models on benign data, as they are especially suited for industries (cf. Sec. 2). To this end, we systematically review *all* papers citing one of the three datasets commonly used in research [19] (SWaT [38], WADI [3], and HAI [72]) according to Scopus and Semantic Scholar as of April 20, 2022, resulting in 215 publications for SWaT, 92 for WADI, and 18 for HAI. We then manually filter for anomaly detection IIDSs, thus especially excluding supervised machine learning, position papers, and surveys. As summarized in Tab. 1, 70 unique publications fulfill these requirements (some papers use more than one dataset).

We structure found approaches alongside their underlying detection methodologies into three broader classes (cf. Tab. 1). *Artificial neural networks* (63% of publications) are usually trained to predict the physical state based on recent historical samples. They then define a difference measure, e.g., between predicted and observed state, and raise an alarm if a threshold is surpassed. In contrast, *graph-based* IIDSs (9%) aggregate similar expected behavior into (physical) states of the system with transitions between these states. Unknown states, transitions, or irregularities in their occurrence indicate an anomaly. A large class of *miscellaneous* approaches (28%) shows that the research community has not settled on a preferred direction even in this confined domain.

Interestingly, we found that *all* approaches from Tab. 1 rely on complex methods: While we occasionally observed related work supplemented with straightforward methods, e.g., out-of-bound checks [57], to the best of our knowledge, such simple approaches have not yet been evaluated in isolation. In the following, we detail our survey's findings by focusing on issues resulting from their complexity.

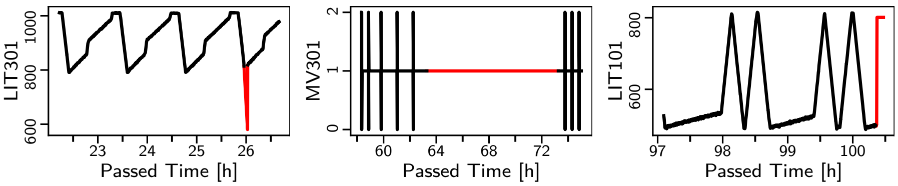
Computational Complexity. The implementation of any detection methodology should be quick enough to be deployable in real-time environments, i.e., detection should not be significantly delayed by processing overhead. E.g., even if adequate hardware is available, requirements such as GPUs in 23% of the publications drastically limit deployability. Although artificial neural network model sizes of about 1.5MB are claimed to be lightweight enough to be processed by industrial hardware [30] for other deployments, e.g., on programmable network switches (as commonly done for traditional IDSs), this is still infeasible.

Hindered Generalizability. IIDS research is characterized by an inherent heterogeneity across deployment domains, although underlying fundamental principles remain similar (cf. Sec. 2). While it would make sense to transfer the achievements of IIDS research conducted for one domain to another, it is known that most published approaches (75% [76]) evaluate a single use-case. Also, in our survey, we find that IIDSs are evaluated only on a median of 1.5 (2 on average) different datasets, and since complex IIDSs are fine-tuned to one specific scenario, they are rarely applied elsewhere [27, 76]. Even though some papers claim generalizability to other domains [46], this claim is not proven.

Incomprehensible Alarms. After an IIDS has indicated a potential threat by emitting an alarm, further (manual) investigation is necessary to find and ultimately mitigate its cause. This could include determining the affected part of the process and isolating it from the rest of the network. While a few IIDSs' alarms are reasonably descriptive [33, 42, 57] and would ease in-depth investigation, such works are rare in our survey. In most cases, the decisions of machine learning classifies are often incomprehensible or only accessible to highly-trained specialists. For instance, feeding a vector of process values into an artificial neural network [46], it is not clear why the vector would be classified as benign or malicious, preventing timely tracing of an alarm back to its source.

Difficult Deployment. Training an IIDS to a process usually requires configuring plenty of hyperparameters, especially for machine learning, which relies on experts knowing the details of a model. As scientific reproduction studies indicate [27], even IDS experts fail when trying to configure an already published approach (with source code available) to match the original publication's results. While setting up an IIDS is usually done once and, therefore, the training overhead might be justifiable, industrial processes are subject to change if the process is adapted or optimized from time to time. In the worst case, this makes the trained model obsolete and requires redoing the entire process. Also, verifying a retrained model is difficult if the model is not humanly comprehensible [28].

Non-Stealthy Attacks. One common argument to justify complexity is the goal to unveil stealthy attacks. While approaches evaluated on specifically crafted datasets exist [11], a closer inspection of the *commonly* used datasets reveals that many attacks are not difficult to detect. As shown in Fig. 1 for SWaT, overshooting or undershooting regular process values, remaining for too long in a single state (flat line), or unusual steep inclines or declines do not necessarily require complex detection mechanisms. Still, this is not a flaw of the datasets, as recent real-world incidents prove. E.g., the sodium hydroxide concentration of an insecure water treatment plant in Florida was set to hazardous levels (about 100 times increased) [59], constituting a potentially easily detectable real attack.



(a) Over- or undershooting (b) Flat line (c) Steep in- or declines

Fig. 1. Manual investigation of the SWaT dataset reveals that attacks (red) commonly used for evaluating IIDSs in research are often not stealthy and thus easy to detect.

In conclusion, current research on industrial intrusion detection is primarily driven by complex approaches, while attacks in evaluation datasets and examples from real-world incidents seem to be detectable relatively straightforward. Given further drawbacks, i.e., incomprehensible alarms, computational complexity, hindered generalizability, and difficult deployment, it is unknown whether this complexity is necessary or whether IIDSs could not be (more) simple instead.

4 SIMPLE Industrial Intrusion Detection

To study the question of whether IIDSs indeed have to be complex, we first define properties that characterize a SIMPLE IIDS (Sec. 4.1). We then present our four IIDSs (Sec. 4.2) derived from typical attack patterns and natural ICS behaviors, e.g., that physical and operational limits constrain possible value ranges.

4.1 Sufficient, Independent, Meaningful, Portable, Local & Efficient

The focus of research on complex IIDSs leads to inherent drawbacks as laid out in Sec. 3, and to address these issues, we propose six properties for Sufficient, Independent, Meaningful, Portable, Local, and Efficient (SIMPLE) IIDSs:
Sufficient. Although simpler in design, an IIDS should be sufficient to detect most attacks while emitting few false alarms (compared to complex approaches).
Independent. Since training an IIDS to specific scenarios is currently complicated due to plenty of hyperparameters influencing the training process, SIMPLE models should be independent of parameters and specialized personnel required to find parameters or re-evaluate a trained model after any modification.
Meaningful. As IIDSs protect physical processes, providing operators with meaningful alerts is essential. They allow determining which sensors/actuators behave anomalously and thus take appropriate measures in a timely manner.
Portable. Since the industrial domain is inherently heterogeneous, an IIDS should be portable to various industrial scenarios. I.e., it needs to be adaptable to different ICS processes and kinds of sensors/actuators types.
Local. Detection methodologies should be local to individual sensors/actuators so that they can be adjusted to their particular distinct behavior. Furthermore,

locality enables partially adjusting an IIDS when the ICS is modified, and sensors/actuators are added or removed without obsoleting other models.
Efficient. The detection methodology should be computationally efficient during training and live detection. Since an ICS's process may change, quick retraining avoids extensive periods in which an obsolete model is used. Efficiency during live detection enlarges hardware and deployment choices and eases timely responses.

Besides SIMPLE, related work already postulated a similar set of requirements [28]. Overall, our six properties address the challenges of complex detection approaches widely found in literature and thus provide the foundation for easily understandable, lightweight, generalizable, and effective intrusion detection.

4.2 Designing SIMPLE IIDSs

To turn the postulated properties into reality and thus lay the foundation for answering whether industrial intrusion detection indeed has to be inherently complex, we design four SIMPLE IIDSs. Our approaches are inspired by typical attack patterns found in scientific datasets (cf. Fig. 1), natural ICS behaviors, and share a set of common characteristics as explained in the following.

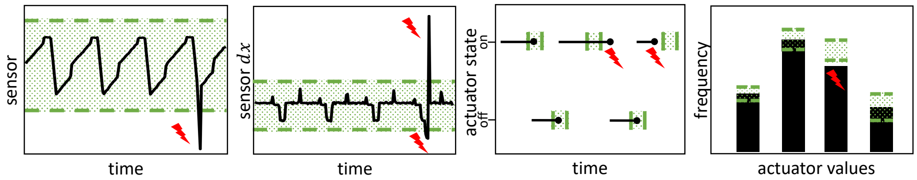
At the core, a SIMPLE IIDS learns a single model per sensor/actuator of the ICS and is trained in a single pass. Not only do separate models fulfill locality, but they are even necessary as process variables exhibit different value ranges, i.e., sensors (float) and actuators (discrete), obviating the need for additional normalization known from complex related work (e.g., [1, 6, 43]). Simultaneously, we avoid introducing process dependencies into the model, which would inherently increase complexity. By iterating over the data only once, we significantly reduce the computational complexity of the training process.

All our detection models train a lower (min) and an upper (max) threshold of a certain, easily computable property and emit an alarm if one of these thresholds is exceeded. To account for variability in physical values and between process cycles due to noise or the fact that training data might not cover all expected data ranges, we introduce an error margin to the learned thresholds as follows:

$$min_{\text{err}} := min - \frac{max - min}{2} \qquad max_{\text{err}} := max + \frac{max - min}{2}$$

The resulting thresholds min_{err} and max_{err}, which are then used for emitting alarms, effectively double the trained range. While this approach is highly opportunistic, it is universally applicable and could be theoretically adjusted easily, yet we refrain from doing so in the spirit of simplicity.

In the following, we present the design of our four SIMPLE IIDSs (MinMax, Gradient, Steadytime, and Histogram) based on these common characteristics. As visualized in Fig. 2, each approach is inspired by natural ICS behaviors or typical attack patterns. We do not claim that our set of SIMPLE IIDSs is exhaustive; theoretically, many others exist. Nevertheless, the following four approaches are adequate to study whether the inherent complexity of IIDSs observed in related work is required. Still, it is essential to note that not every approach is equally well suited for each type of sensor/actuator.



(a) MinMax (b) Gradient (c) Steadytime (d) Histogram

Fig. 2. We introduce four SIMPLE IIDS ideas detecting over- or undershooting with a MinMax approach, steep in- or declines with Gradient, flat lines with Steadytime, and unnatural process fluctuations with a Histogram. Each IIDS trains an allowed range (green area). If the threshold (green line) is surpassed, an alarm (red arrow) is emitted.

MinMax. The minimum and maximum (MinMax) approach (cf. Fig. 2a) detects whether a sensor's/actuator's current value exceeds the range observed in the training data and raises an alarm if any observation falls outside that range (± error margin). This approach is motivated by the intuition that process values of industrial systems relate to physical measurements or setpoints and thus usually obey certain limits. E.g., temperatures below the freezing point of a liquid are not desirable for pumping it through pipes. Even if the physical setup does not limit the value range, operational requirements may impose restrictions on the allowed data range, e.g., the pH value of a liquid may not exceed a specific range to be non-hazardous. Thus, we assume that an industrial system exhibits a class of values inside well-defined minimum and maximum limits.

Gradient. Following a similar intuition, the Gradient approach (cf. Fig. 2b) detects whether a sensor's/actuator's slope exceeds the minimum and maximum observed during training (± error margin). While MinMax observes global changes, more subtle attacks occurring within these limits may remain unnoticed. E.g., as shown in Fig. 1c, the sensor is set to a high value within the operational limits, yet far too abrupt, thus introducing a noticeable discontinuity. Hence, the Gradient approach assumes that ICSs have continual character, i.e., physical values such as temperatures cannot change at arbitrary speed.

Steadytime. Focusing on another temporal aspect, the Steadytime approach (cf. Fig. 2c) detects whether a sensor/actuator remains static, i.e., does not change its value, for a shorter or longer time than seen during training (± error margin). This approach is motivated by the observation that an attack, e.g., freezing a sensor/actuator (cf. Fig. 1b) such as a pressure relief valve, cannot be detected by checking whether a value or the velocity of a value change remains within certain boundaries (MinMax/Gradient). Since a steady state is difficult to define for noisy sensor data, Steadytime takes only process values into account if the number of distinct values during training is sufficiently small (≤ 10).

Histogram. Specifically targeting the occurrence of values, the Histogram approach (cf. Fig. 2d) tracks their distribution within a fixed-sized window and tests whether it is in line with a histogram seen during training (± error margin). The underlying intuition expects a similar distribution of reoccurring values between process cycles. This approach can detect the existence and absence of frequent value changes, which the other three approaches cannot detect. The histograms are created by counting the number of times each distinct value appears in a sliding window. We merge them into a single histogram that covers each value's minimum and maximum occurrences across all distinct fixed-sized windows. The window size should match the duration of a process cycle, which could be automatically determined in an additional run over the dataset prior to training the histograms. Like Steadytime, Histogram only applies for process values with a few distinct values (≤ 10), as comparing two histograms value-by-value is unfeasible for noisy sensor data.

These four proposals stand in stark contrast to related work, which focuses on inherently complex approaches such as leveraging multiple Autoencoders [14] or fusing two IIDS directions into one solution [26]. While further refinements to our IIDSs are possible, we explicitly focused on fundamental and minimalistic approaches to understand their effectiveness and assess whether industrial intrusion detection really needs to be complex or can be more SIMPLE instead.

5 Industrial Intrusion Detection Can Indeed Be SIMPLE

Using our four SIMPLE IIDSs, we can now study the fundamental question of whether industrial intrusion detection inherently needs to be complex or whether and to which extent SIMPLE approaches provide a viable alternative. To answer this question, we specifically study whether they are (i) sufficient to detect most attacks, (ii) competitive to complex approaches from related work, and (iii) portable across industrial scenarios. To this end, we first provide an overview of our evaluation setup (Sec. 5.1) before we analyze how our approaches perform on the widely-used reference dataset SWaT (Sec. 5.2), their portability to three additional industrial datasets (Sec. 5.3), and ultimately discuss the prospects of SIMPLE IIDSs for industrial intrusion detection (Sec. 5.4).

5.1 Evaluation Setup

We begin our analysis by describing the implementation, datasets, and evaluation metrics underlying our evaluations.

Implementation. We implemented our four SIMPLE IIDSs (cf. Sec. 4.2) in Python on top of the IPAL framework [76], which offers a holistic scientific platform to implement, evaluate, and compare industrial intrusion detection approaches. Most importantly, IPAL introduces a unified representation for the data input, which facilitates the seamless application of IIDSs to many datasets. Furthermore, it provides (re-)implementations of state-of-the-art IIDSs from related work [76], which we use as a comparison benchmark. To facilitate further research on industrial intrusion detection, we make the implementations of our SIMPLE IIDSs publicly available[2] within the IPAL framework.

[2] Implementation available at: https://github.com/fkie-cad/ipal_ids_framework

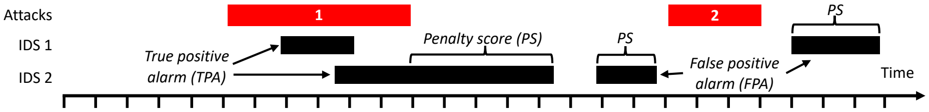


Fig. 3. When evaluating IIDSs, a true positive alarm (TPA) overlaps with the attack label from the dataset (red), while a false positive alarm (FPA) does not overlap with any attack. The penalty score (PS) measures the “overshooting” of all raised alarms.

Datasets. We evaluate our IIDSs on four state-of-the-art industrial datasets based on physical testbeds and including attacks against the industrial process: SWaT [38], the most widely-used dataset, represents a multi-staged water treatment system. Similarly, WADI [3] serves as an example of portability to a water distribution scenario. Additionally, we consider the novel WDT dataset [31] since it includes network and physically induced attacks, and finish with HAI [72] modeling power generation and storage – an entirely different industrial domain. **Evaluation Metrics.** To objectively quantify the performance of both SIMPLE and complex IIDSs, we refer to a set of performance metrics. As visualized in Fig. 3, datasets contain labels (in red) indicating a time range when an attack took place. An IIDS indicates these attacks by emitting alarms (in black). As traditional metrics, we utilize *accuracy*, *precision*, *recall*, and *F1-score* – the defacto standard for evaluating classifiers. Yet, as they focus on the label coverage, they do not express how many attacks are detected and are skewed if attacks are of different lengths. Furthermore, effects unique to industrial settings, such as the stabilization time required after an attack, are not considered. Thus, we additionally calculate the percentage of *detected attacks*, the number of *true positive alarms* (TPA), i.e., alarms overlapping with an attack, false-positive alarms (FPA), i.e., non-overlapping alarms, and the *penalty score* (PS) aggregating the non-overlapping time span [57] to provide a more holistic perspective.

5.2 Sufficiency: SIMPLE IIDSs on Par With Complex Approaches

First, we study whether SIMPLE IIDSs are *sufficient* to detect most attacks (cf. Sec. 4.1) and whether industrial intrusion detection must be inherently complex. To this end, we compare our approaches' detection performance to related work in an in-depth evaluation based on SWaT [38], as it is the most widely-used dataset in literature (90% of publications according to our analysis in Tab. 1).

SWaT consists of a training part of normal ICS behavior and a test part containing 36 attacks. We trained our four IIDSs on the training data omitting the first ∼22 hours during which the system stabilizes after activation. As seven out of SWaT's 51 sensors and actuators do not maintain the regular patterns observed during training, we excluded those from further evaluation. Skipping the stabilization phase [49, 55, 58, 63, 66, 75, 79] and omitting process values [50, 52, 62, 66, 75] in SWaT are common practices in related work. Notably, instead of excluding the process values, a process expert could manually adapt a pretrained SIMPLE model, which is impossible for complex approaches (cf. Sec. 3).

Table 2. Already a high-level analysis reveals that our SIMPLE IIDSs (green) in combination can detect 75% of attacks from the SWaT dataset, thus performing comparably to complex approaches from related work (blue).

Attack	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	Σ
MinMax		🟩				🟩	🟩	🟩	🟩			🟩					🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩		🟩		🟩	🟩	🟩	🟩	🟩	🟩	23
Gradient		🟩	🟩			🟩	🟩	🟩	🟩			🟩				🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩		25
SteadyTime		🟩					🟩	🟩	🟩				🟩				🟩		🟩	🟩	🟩	🟩	🟩		🟩				🟩						🟩		14
Histogram		🟩					🟩	🟩	🟩				🟩				🟩		🟩	🟩	🟩		🟩		🟩				🟩						🟩		13
SIMPLE		🟩	🟩			🟩	🟩	🟩	🟩			🟩	🟩			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩	🟩	27
DIF [25]			🟦		🟦	🟦	🟦	🟦	🟦								🟦	🟦		🟦			🟦	🟦		🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦		19
1D-CNN [26]	🟦	🟦	🟦				🟦		🟦		🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦			🟦		🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	26
SVM [43]				🟦	🟦	🟦	🟦	🟦	🟦					🟦	🟦	🟦	🟦	🟦		🟦			🟦		🟦		🟦				🟦	🟦	🟦		🟦	🟦	20
DNN [43]					🟦		🟦	🟦	🟦					🟦	🟦		🟦	🟦					🟦									🟦	🟦	🟦	🟦		13
Seq2SeqNN [46]	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦			🟦	🟦		🟦	🟦	🟦	🟦			🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	29
1D-CNN [49]	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	31
TABOR [57]	🟦	🟦		🟦	🟦		🟦	🟦	🟦				🟦	🟦	🟦	🟦	🟦				🟦	🟦	🟦		🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		24
GAN [64]		🟦	🟦		🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	32
MADICS [66]		🟦				🟦	🟦	🟦	🟦				🟦				🟦	🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	23
NN [71]		🟦			🟦	🟦	🟦	🟦	🟦			🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦		25
Com-AE [75]	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦				🟦		🟦		🟦	🟦	🟦		🟦		🟦		🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	26
1D-CNN [78]	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	32

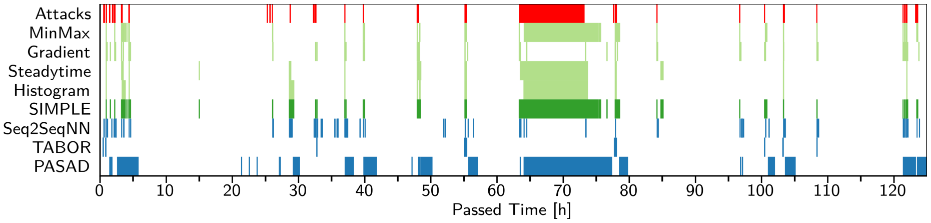
High-level Ability to Detect Attacks. We use SWaT to gain a first assessment of the ability of our SIMPLE IIDSs to detect attacks and to compare them to complex approaches from related work. To this end, we analyzed all 63 publications evaluating on SWaT (cf. Sec. 3) to obtain those that provide sufficiently detailed information on which specific attacks they detect, resulting in eleven publications covering twelve complex IIDSs. Tab. 2 visualizes which SIMPLE (green) and complex (blue) IIDSs can detect which of the 36 attacks in SWaT.

Our four combined approaches (denoted with "SIMPLE") detect a majority of attacks (75%/27 attacks). Our arguably most simple approach (MinMax) alone can detect 23 attacks, and Gradient performs best by detecting 25 attacks. For comparison, the average number of detected attacks by related work is 25.0. Aggregating related work's capabilities, they detect all except a single scenario. However, in the twelve scenarios that *all* SIMPLE approaches detect, seven of the twelve complex IIDSs do not fully cover these seemingly easy-to-detect attacks. Regarding the nine attacks that are not detected by any SIMPLE approach, four (4, 10, 11, 14) have repeatedly been reported as not-detectable [46, 71, 78], and we observe inefficiencies in complex approaches too. Notably, not a single attack is detected by all complex approaches but not by our SIMPLE methods.

Thus, SIMPLE IIDSs seem on par with their complex counterparts, detecting more attacks on average but less in total for the benefits of increased simplicity.

In-depth Comparison. Besides high detection rates, IIDSs should have a low false-positive rate [28]. To study how SIMPLE IIDSs fare against selected complex related work, we study their alert behavior in-depth visually (Fig. 4) and alongside metrics (Tab. 3). We again provide combined results for "SIMPLE", where an alarm is emitted whenever any IIDS emits an alarm. Notably, this may fuse alerts into larger ones, which can result in fewer overall TPAs and FPAs.

Since this evaluation requires access to complex IIDSs, we selected one representative approach for each of the three classes (cf. Tab. 1) for which we have implementations available (cf. Sec. 5.1): (i) Seq2SeqNN [46] (representing artificial neural networks) predicts the following expected output based on samples



Short attacks and alarms are enlarged to a minimum width of 0.15% for visibility.

Fig. 4. A visual inspection of the alerts of SIMPLE IIDSs (green) shows thorough coverage w.r.t. the attacks for the SWaT dataset (red). The alerts of the three representative complex approaches (blue) are less expressive and contain more false positives.

of recent process history, and alerts if the prediction deviates long enough from the observed behavior, (ii) TABOR [57] (graph-based) combines an automaton, Bayesian network, and out-of-bounds check into a single solution (while we could only reproduce one of TABOR's 16 models for SWaT, this model still suffices for our analysis), and (iii) PASAD [11] (miscellaneous) leverages a singular spectrum analysis to identify recurring process patterns on a per-sensor basis.

Upon visual inspection based on Fig. 4, the alerts emitted by the SIMPLE IIDSs coincide with the attacks to be detected to a large extent. Furthermore, during non-attack periods, they do not emit large amounts of false alarms. Overall, the three complex approaches contain more false alarms and detect fewer attacks. Thus, while from a high-level view, the complex IIDSs under study appeared to detect slightly more attack scenarios (cf. Tab. 2), they come at the price of more false positives and consequently reduced utility.

Moreover, most related complex approaches' alarms are incomprehensible as they often exhibit multiple alarms around an attack (Seq2SeqNN) or alert over a long time range covering many attacks (PASAD). Our IIDSs, on the other hand, precisely overlap with the attacks and additionally allow to determine the potential malicious sensors through their locality property (cf. Sec. 4.1). E.g., for 21 of the attacks detected by Gradient, the alerts stem from the process value indicated as the attack point in the SWaT dataset, thus providing a reliable starting point for subsequent incident response.

The individual metrics summarized in Tab. 3 confirm our previous observation that SIMPLE approaches detect large amounts of attacks (detected attacks and TPA), emit few false alarms (FPA), and perform on par with related work. Notably, while Steadytime and Histogram detect fewer attacks, they simultaneously have the lowest FPA score of all IIDSs under study. In terms of accuracy, precision, recall, and F1 score, the MinMax, Steadytime, and Histogram IIDS outperform Seq2SeqNN and PASAD and perform roughly equivalent to TABOR. These metrics are surprisingly good, considering the simplicity of the detection methods, which are not optimized to any metrics (a problem common for machine learning approaches [51]). While Gradient showed auspicious detection performance in the visual comparison, it fares poorly for the individual metrics.

Table 3. Across all relevant quantifiable evaluation metrics, SIMPLE IIDSs (especially in combination) are competitive to the studied complex approaches from related work.

IIDS	Detected Attacks [%]	TPA	FPA	PS	Acc.	Prec.	Rec.	F1
MinMax	63.89	22	9	14647	0.94	0.75	0.81	0.78
Gradient	69.44	47	64	352	0.88	0.3	0.00	0.01
Steadytime	38.89	16	4	5033	0.96	0.89	0.75	0.81
Histogram	36.11	12	0	6794	0.95	0.85	0.72	0.78
SIMPLE	75.0	26	23	19621	0.94	0.71	0.87	0.78
Seq2SeqNN	72.22	30	37	7559	0.88	0.44	0.11	0.17
TABOR*	66.67	–	–	–	–	0.86	0.79	0.82
PASAD	44.44	10	14	81604	0.78	0.32	0.72	0.45

* Results taken from the publication [57] as not all model parameters were reproducible.

The main reason for this phenomenon is that these metrics favor long attack coverage, a phenomenon we study in more detail in the appendix.

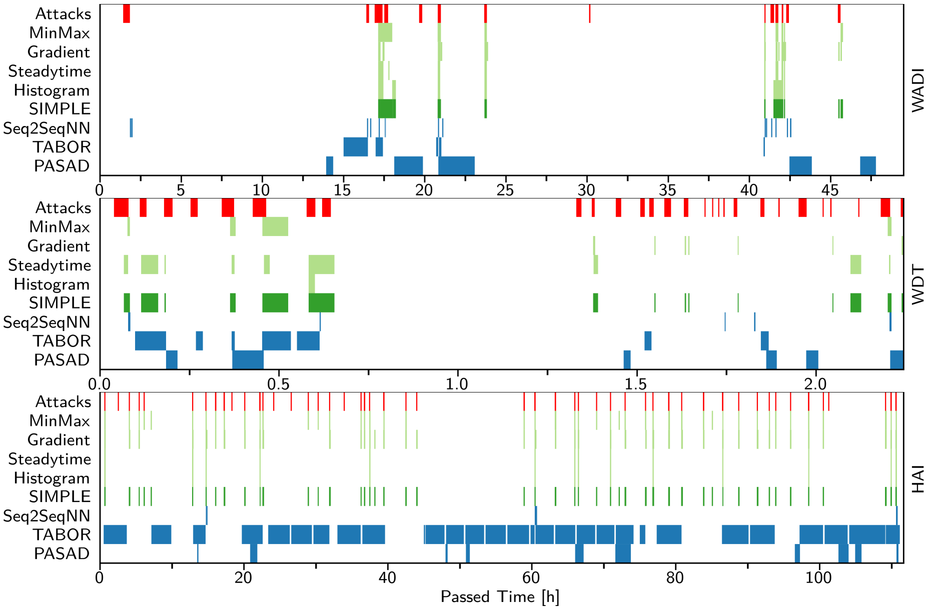
Takeaway: All four SIMPLE IIDSs detect a sufficient number of attacks in the SWaT dataset. Combining the SIMPLE approaches allows detecting 75% of all attacks while visually emitting only a few false alerts. Moreover, raised (false) alarms are meaningful due to their local design and comprehensible models. Compared to related work, SIMPLE IIDSs can keep up with complex approaches in terms of the number of detected alarms, and especially false positives.

5.3 Portability: SIMPLE IIDSs Work Effortlessly in New Settings

To ensure that IIDSs are widely applicable, they must be portable to various industrial scenarios and processes with a short training phase and without requiring (re-)inventions (cf. Sec. 4.1). Consequently, we show the portability of our IIDSs by applying them to three additional industrial datasets (WADI [3], WDT [31], and HAI [72], cf. Sec. 5.1). We again compare our IIDSs to the three complex representatives from related work (Seq2SeqNN, TABOR, and PASAD, cf. Sec. 5.2). Unlike ours, porting the complex IIDSs to the new datasets required extensive manual work to find suitable models and parameters. Once more, we analyze the results both visually (Fig. 5) and for various metrics (Tab. 4).

WADI. At first glance, with 64% of detected attacks overall, the SIMPLE IIDSs do not perform as strongly on WADI as on SWaT. However, even our worst-performing IIDS, Steadytime (43%), still outperforms PASAD (14%) and TABOR (29%). More importantly, we visually observe only two false alarms not closely related to an attack (at around 18h and 42h). While the complex IIDSs exhibit similarly few false alarms, their penalty score (PS) is exceptionally high, indicating that their alarms are too imprecise to match a single attack.

WDT. The WDT dataset proves challenging for all IIDS types, with SIMPLE approaches detecting up to 22% of the attacks, compared to 4% (Seq2SeqNN) up to 18% (TABOR) for the complex approaches. Upon closer inspection, we identified one cause to be attacks not targeting the industrial processes, e.g., network scanning. Since complex approaches are likewise incapable of finding these attack types, SIMPLE IIDSs provide an equally performing alternative.



Short attacks and alarms are enlarged to a minimum width of 0.15% for visibility.

Fig. 5. Porting SIMPLE IIDSs (green) to three additional datasets shows their generalizability to various industrial scenarios, while complex IIDSs (blue) perform worse. Note that the results on the SWaT dataset have previously been discussed in Fig. 4.

HAI. The SIMPLE IIDSs perform well for the HAI dataset with 86% of detected attacks, a low PS of only 531, and nearly no FPA for MinMax, Steadytime, and Histogram. Notably, MinMax and Gradient perform especially well on HAI, thus showing that attacks can be detected reliably even with the simplest approaches. Complex related work, in contrast, falls far behind, and TABOR is even largely inapplicable, likely due to HAI's less pronounced regular patterns.
Takeaway: SIMPLE IIDSs, unlike their complex counterparts, can be ported to new datasets without manual effort. Furthermore, considering that our approaches, in contrast to complex approaches, performed best on HAI (representing an entirely new industrial domain), this perfectly proves their portability.

5.4 Discussion: Industrial Intrusion Detection Can Be SIMPLE

Wrapping up our evaluation, we recapitulate the promised properties of SIMPLE IIDSs (sufficient, independent, meaningful, portable, local, and efficient) and discuss to which extent our proposed IIDSs capitalize on them.

Although we relied on straightforward detection methods and chose an opportunistic error threshold, our approaches proved to be on par with significantly more complex detection methods. Most attacks are detected for the SWaT and HAI datasets, and across all four examined datasets, our IIDSs are *Sufficient* compared to related work (cf. Sec. 5.2 and 5.3).

Table 4. For all relevant metrics, e.g., detected attacks, our SIMPLE IIDSs generalize better to new industrial settings than complex related work (an in-depth analysis, e.g., regarding FPA of the Gradient IIDS, is provided in the appendix). Note that the results on the SWaT dataset have previously been discussed in Tab. 3.

	IIDS	Detected Attacks [%]	TPA	FPA	PS	Acc.	Prec.	Rec.	F1
WADI	MinMax	50.0	6	4	1751	0.96	0.7	0.41	0.52
	Gradient	50.0	44	12	12	0.94	0.79	0.0	0.01
	Steadytime	42.86	7	2	413	0.96	0.88	0.3	0.44
	Histogram	50.0	6	6	1775	0.95	0.64	0.32	0.42
	SIMPLE	64.29	6	9	3120	0.95	0.58	0.44	0.5
	Seq2SeqNN	57.14	8	6	1293	0.94	0.52	0.14	0.22
	TABOR	28.57	5	0	5792	0.92	0.3	0.25	0.27
	PASAD	14.29	2	3	23197	0.82	0.05	0.13	0.07
WDT	MinMax	7.84	4	0	268	0.72	0.3	0.08	0.13
	Gradient	5.88	8	3	3	0.75	0.73	0.01	0.01
	Steadytime	17.65	9	0	411	0.74	0.45	0.23	0.31
	Histogram	1.96	1	0	0	0.76	1.0	0.04	0.09
	SIMPLE	21.57	16	3	639	0.71	0.38	0.27	0.32
	Seq2SeqNN	3.92	2	3	58	0.74	0.19	0.01	0.02
	TABOR	17.65	7	0	762	0.67	0.3	0.22	0.26
	PASAD	11.76	4	2	639	0.68	0.27	0.16	0.2
HAI	MinMax	86.0	73	7	496	0.98	0.87	0.38	0.53
	Gradient	78.0	209	48	96	0.98	0.89	0.09	0.16
	Steadytime	28.0	15	0	161	0.98	0.86	0.11	0.19
	Histogram	28.0	15	0	161	0.98	0.86	0.11	0.19
	SIMPLE	86.0	145	26	531	0.99	0.87	0.4	0.55
	Seq2SeqNN	4.0	2	5	936	0.98	0.29	0.04	0.07
	TABOR	70.0	22	7	271159	0.32	0.02	0.6	0.04
	PASAD	4.0	2	11	29839	0.9	0.01	0.04	0.02

Contrary to related work, which, e.g., requires up to 16 models for a single dataset [57] or unique parameterization for each process value [11], our SIMPLE approaches are *Independent* of parameters. While theoretically, the margin of error or the Histogram's window size could be fine-tuned for better performance, even their default values, as evaluated by us, yield a competitive performance.

The alerts emitted by our approaches largely coincide with the attacks (cf. Fig. 4 and 5). Furthermore, these carry *Meaningful* insights for incident response, e.g., to which extent the trained threshold is exceeded (MinMax and Gradient).

As our SIMPLE approaches generalize to four diverse datasets (cf. Sec. 5.3), they have successfully proved to be *Portable* across various industrial settings.

Already by design (cf. Sec. 4.2), all our SIMPLE approaches are *Local*, i.e., operate on a per-sensor basis. As such, they are inherently able to identify the triggering value directly. To illustrate the resulting advantages exemplary for the SWaT dataset, which provides precise information on the attack location, MinMax and Gradient, e.g., could easily identify 18 respectively 21 attack locations correctly, significantly easing attack identification and hence incident response.

Finally, the IIDSs are *Efficient* w.r.t. computing resources as they rely on elementary computational operations both during model creation and detection.

E.g., MinMax and Steadytime only perform an interval test, Gradient requires an additional subtraction for the slope computation, and Histogram counts and compares the last recently occurring process values. Besides computational efficiency, they are also optimized for a low memory footprint, thus easing their applicability in resource-limited industrial settings: MinMax and Gradient only store the minimal and maximal bounds on a per-sensor basis, while Steadytime and Histogram only require the thresholds per occurring value for each sensor.
Takeaway: The four exemplary approaches presented in this paper satisfy the properties of a sufficient, independent, meaningful, portable, local, and efficient IIDS. Thus, we show that industrial intrusion detection can indeed be SIMPLE, challenging the necessity of inherent complexity found across related work.

6 Conclusion

Industrial intrusion detection constitutes a retrofittable solution to counteract harmful cyberattacks against increasingly threatened industrial control systems. Striving to achieve (close to) optimal detection of attacks, the research community proposed a wide variety of approaches to detect anomalies in the process state across different industrial domains. However, as we identify based on a systematic analysis of 70 proposals from related work, these approaches show an inherent complexity where high detection performance is accompanied by dearly bought consequences such as a lack of model and alert comprehensibility or a high demand for computing resources. Considering that IIDSs leverage the repetitive nature of physical processes, we wonder why simpler detection methods have not been considered so far. To overcome this gap, we study whether IIDSs can be SIMPLE (Sufficient, Independent, Meaningful, Portable, Local, and Efficient) instead of having to rely on complex models with all their disadvantages. Thus, we designed four exemplary minimalistic approaches, such as straightforward range checks. Surprisingly, as we show across four distinct industrial datasets, simplicity does not result in reduced detection capabilities, as simple methods are on par with significantly more complex related work. Simultaneously, simple approaches offer highly beneficial properties such as eased configuration, model and alert comprehensibility, and reduced computational overhead. Thus, simple IIDSs provide a viable alternative to complex approaches, raising the question whether slight increases in detection capabilities justify computational overheads and reduced utility. Still, it remains open whether our results are constrained by the studied datasets (i.e., the included attacks that are too “easy” to detect) or whether SIMPLE IIDSs are inherently sufficient to detect cyberattacks. Consequently, future research has to investigate the raison d’être for complex IIDS w.r.t. advanced and stealthy attacks, beyond limiting their evaluations to the datasets currently in widespread use, for which simple approaches suffice.

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

1. Abdelaty, M.F., et al.: DAICS: A Deep Learning Solution for Anomaly Detection in Industrial Control Systems. IEEE Trans. Emerg. Topics Comput. (2021)
2. Ahmed, C., et al.: *NoisePrint*: Attack Detection Using Sensor and Process Noise Fingerprint in Cyber Physical Systems. In: ACM ASIACCS (2018)
3. Ahmed, C., et al.: WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In: CySWATER (2017)
4. Ahmed, C., et al.: Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In: ACSAC (2018)
5. Alabugin, S.K., et al.: Applying of Generative Adversarial Networks for Anomaly Detection in Industrial Control Systems. In: GloSIC (2020)
6. Alabugin, S.K., et al.: Applying of Recurrent Neural Networks for Industrial Processes Anomaly Detection. In: IEEE USBEREIT (2021)
7. Alladi, T., et al.: Industrial control systems: Cyberattack trends and countermeasures. Computer Communications **155** (2020)
8. Anton, S.D.D., et al.: Using Temporal and Topological Features for Intrusion Detection in Operational Networks. In: ARES (2019)
9. Anton, S.D.D., et al.: Security in Process: Detecting Attacks in Industrial Process Data. In: CECC (2019)
10. Anton, S.D.D., et al.: Intrusion Detection in Binary Process Data: Introducing the Hamming-Distance to Matrix Profiles. In: IEEE WoWMoM (2020)
11. Aoudi, W., et al.: Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. In: ACM CCS (2018)
12. Audibert, J., et al.: USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In: ACM SIGKDD (2020)
13. Bae, S., et al.: Research on Improvement of Anomaly Detection Performance in Industrial Control Systems. In: WISA (2021)
14. Cao, D., et al.: Self-Adaption AAE-GAN for Aluminum Electrolytic Cell Anomaly Detection. IEEE Access **9** (2021)
15. Castellanos, J.H., et al.: A Modular Hybrid Learning Approach for Black-Box Security Testing of CPS. In: ACNS (2019)
16. Chen, X., et al.: DAEMON: Unsupervised Anomaly Detection and Interpretation for Multivariate Time Series. In: IEEE ICDE (2021)
17. Chen, Z., et al.: Learning Graph Structures With Transformer for Multivariate Time Series Anomaly Detection in IoT. IEEE IoT-J (2021)
18. Clotet, X., et al.: A Real-Time Anomaly-Based IDS for Cyber-Attack Detection at the Industrial Process Level of Critical Infrastructures. IJCIP **23** (2018)
19. Conti, M., et al.: A Survey on Industrial Control System Testbeds and Datasets for Security Research. IEEE Commun. Surv. Tutor. **23**(4) (2021)
20. Dai, E., et al.: Graph-Augmented Normalizing Flows for Anomaly Detection of Multiple Time Series. In: ICLR (2022)
21. Demertzis, K., et al.: Variational Restricted Boltzmann Machines to Automated Anomaly Detection. Neural Computing and Applications (2022)
22. Deng, A., et al.: Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In: AAAI (2021)
23. Ding, D., et al.: A Survey on Security Control and Attack Detection for Industrial Cyber-Physical Systems. Neurocomputing **275** (2018)
24. Dutta, A.K., et al.: CatchAll: A Robust Multivariate Intrusion Detection System for Cyber-Physical Systems Using Low Rank Matrix. In: CPSIoTSec (2021)

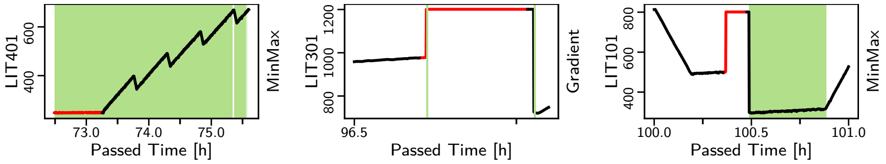
25. Elnour, M., et al.: A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems. IEEE Access **8** (2020)
26. Elnour, M., et al.: Hybrid attack detection framework for industrial control systems using 1d-convolutional neural network and isolation forest. In: CCTA (2020)
27. Erba, A., et al.: No Need to Know Physics: Resilience of Process-Based Model-Free Anomaly Detection for Industrial Control Systems. arXiv:2012.03586 (2020)
28. Etalle, S.: From intrusion detection to software design. In: ESORICS (2017)
29. Faber, K., et al.: Ensemble Neuroevolution-Based Approach for Multivariate Time Series Anomaly Detection. Entropy **23**(11) (2021)
30. Fährmann, D., et al.: Lightweight Long Short-Term Memory Variational Auto-Encoder for Multivariate Time Series Anomaly Detection in Industrial Control Systems. Sensors **22**(8) (2022)
31. Faramondi, L., et al.: A Hardware-in-the-Loop Water Distribution Testbed Dataset for Cyber-Physical Security Testing. IEEE Access **9** (2021)
32. Farsi, H., et al.: A Novel Online State-Based Anomaly Detection System for Process Control Networks. IJCIP **27** (2019)
33. Feng, C., et al.: A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In: NDSS (2019)
34. Feng, C., et al.: Time Series Anomaly Detection for Cyber-Physical Systems via Neural System Identification and Bayesian Filtering. In: ACM SIGKDD (2021)
35. Francisquini, R., et al.: Community-Based Anomaly Detection Using Spectral Graph Filtering. Applied Soft Computing **118** (2022)
36. Gauthama Raman, M., et al.: Deep Autoencoders as Anomaly Detectors: Method and Case Study in a Distributed Water Treatment Plant. Comput. Secur. **99** (2020)
37. Giraldo, J., et al.: A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. ACM Computing Surveys **51**(4) (2018)
38. Goh, J., et al.: A Dataset to Support Research in the Design of Secure Water Treatment Systems. In: CRITIS (2016)
39. Goh, J., et al.: Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In: IEEE HASE (2017)
40. Gong, S., et al.: A Prediction-Augmented AutoEncoder for Multivariate Time Series Anomaly Detection. In: ICONIP (2021)
41. Guo, Y., et al.: Unsupervised Anomaly Detection in IoT Systems for Smart Cities. IEEE TNSE **7**(4) (2020)
42. Hwang, C., et al.: E-SFD: Explainable Sensor Fault Detection in the ICS Anomaly Detection System. IEEE Access **9** (2021)
43. Inoue, J., et al.: Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In: DMCIS (2017)
44. Intrator, Y., et al.: MDGAN: Boosting Anomaly Detection Using Multi-Discriminator Generative Adversarial Networks. arXiv:1810.05221 (2018)
45. Kim, D., et al.: Stacked-Autoencoder Based Anomaly Detection with Industrial Control System. In: SNPD (2021)
46. Kim, J., et al.: Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks. In: CyberICPS (2020)
47. Kim, S., et al.: APAD: Autoencoder-Based Payload Anomaly Detection for Industrial IoE. Applied Soft Computing **88** (2020)
48. Kim, Y., et al.: Anomaly Detection Using Clustered Deep One-Class Classification. In: AsiaJCIS (2020)
49. Kravchik, M., et al.: Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. In: CPS-SPC (2018)

50. Kravchik, M., et al.: Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA. IEEE TDSC (2021)
51. Kus, D., et al.: A False Sense of Security? Revisiting the State of Machine Learning-Based Industrial Intrusion Detection. In: ACM CPSS (2022)
52. Kwon, H.Y., et al.: Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods. Electronics **11**(6) (2022)
53. Lavrova, D., et al.: Using GRU Neural Network for Cyber-Attack Detection in Automated Process Control Systems. In: IEEE BlackSeaCom (2019)
54. Lee, C.K., et al.: Studies on the GAN-Based Anomaly Detection Methods for the Time Series Data. IEEE Access **9** (2021)
55. Li, D., et al.: Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. In: KDD BigMine (2018)
56. Li, D., et al.: MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In: ICANN (2019)
57. Lin, Q., et al.: TABOR: A Graphical Model-Based Approach for Anomaly Detection in Industrial Control Systems. In: ACM ASIACCS (2018)
58. Macas, M., et al.: An Unsupervised Framework for Anomaly Detection in a Water Treatment System. In: IEEE ICMLA (2019)
59. Margolin, J.: Outdated Computer System Exploited in Water Treatment Plant Hack (2021), https://abc7news.com/story/10328196/, accessed: 2022-04-24
60. Maru, C., et al.: Collective Anomaly Detection for Multivariate Data Using Generative Adversarial Networks. In: CSCI (2020)
61. Mokhtari, S., et al.: Measurement Data Intrusion Detection in Industrial Control Systems Based on Unsupervised Learning. AIMS-ACI **1**(1) (2021)
62. Naito, S., et al.: Anomaly Detection for Multivariate Time Series on Large-Scale Fluid Handling Plant Using Two-Stage Autoencoder. In: ICDMW (2021)
63. Nedeljkovic, D.M., et al.: Detection of Cyber-Attacks in Systems With Distributed Control Based on Support Vector Regression. TELFOR Journal **12**(2) (2020)
64. Neshenko, N., et al.: A Behavioral-Based Forensic Investigation Approach for Analyzing Attacks on Water Plants Using GANs. FSI Digital Investigation **37** (2021)
65. Oliveira, N., et al.: Anomaly Detection in Cyber-Physical Systems: Reconstruction of a Prediction Error Feature Space. In: SINCONF (2021)
66. Perales Gomez, A.L., et al.: MADICS: A Methodology for Anomaly Detection in Industrial Control Systems. Symmetry **12**(10) (2020)
67. Pranavan, T., et al.: Contrastive Predictive Coding for Anomaly Detection in Multi-Variate Time Series Data. arXiv:2202.03639 (2022)
68. Pyatnisky, I., et al.: Assessment of the applicability of autoencoders in the problem of detecting anomalies in the work of industrial control Systems. In: GloSIC (2020)
69. Ray, S., et al.: Learning Graph Neural Networks for Multivariate Time Series Anomaly Detection. arXiv:2111.08082 (2021)
70. Schneider, T., et al.: Detecting Anomalies Within Time Series Using Local Neural Transformations. arXiv:2202.03944 (2022)
71. Shalyga, D., et al.: Anomaly Detection for Water Treatment System Based on Neural Network With Automatic Architecture Optimization. arXiv:1807.07282 (2018)
72. Shin, H., et al.: HAI 1.0: HIL-based Augmented ICS Security Dataset. CSET (2020)
73. Tuli, S., et al.: TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. In: VLDB (2022)
74. Umer, M.A., et al.: Generating Invariants Using Design and Data-Centric Approaches for Distributed Attack Detection. IJCIP **28** (2020)
75. Wang, C., et al.: Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network. WCMC **2020** (2020)

76. Wolsing, K., et al.: IPAL: Breaking up Silos of Protocol-dependent and Domain-specific Industrial Intrusion Detection Systems. In: Proc. of RAID (2022)
77. Xiao, Q., et al.: Memory-Augmented Adversarial Autoencoders for Multivariate Time-Series Anomaly Detection With Deep Reconstruction and Prediction. arXiv:2110.08306 (2021)
78. Xie, X., et al.: Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU. IEEE Access **8** (2020)
79. Xu, Q., et al.: Digital Twin-Based Anomaly Detection in Cyber-Physical Systems. In: IEEE ICST (2021)
80. Yan, T., et al.: TFDPM: Attack Detection for Cyber-Physical Systems With Diffusion Probabilistic Models. arXiv:2112.10774 (2021)
81. Yang, L., et al.: Iterative Bilinear Temporal-Spectral Fusion for Unsupervised Representation Learning in Time Series. arXiv:2202.04770 (2022)
82. Yoong, C.H., et al.: Deriving Invariant Checkers for Critical Infrastructure Using Axiomatic Design Principles. Cybersecurity **4** (2021)
83. Zhang, K., et al.: Federated Variational Learning for Anomaly Detection in Multivariate Time Series. In: IEEE IPCCC (2021)

Appendix

To better understand the SIMPLE IIDSs mechanics, we take a detailed look at their detection phase. We occasionally see alerts stretching significantly further (with interruptions) than the actual attack. In Fig. 6a, the MinMax IIDS raises an alarm throughout the ICS's recovery phase since the process values still deviate from their normal values and fluctuate until stabilizing. The Gradient IIDS reveals another phenomenon in Fig. 6b, leading to supposedly false alerts inherent to its design. As it indicates in- or declines, its alerts are short, which results in a poor performance w.r.t. to metrics evaluating the attack coverage. While this method is precise in finding the actual beginnings and endings of attacks, it often raises an alarm shortly after an attack when the process quickly returns to normal operation. Finally, in Fig. 6c, we observe effects that can occur after the actual attack ended (or where datasets are not precisely labeled). All of these effects result in insufficient attack coverage and false alarms, such that the good performance of IIDSs is not captured well by the available metrics.



(a) Alerts may arise during process stabilization.

(b) Gradient does not raise continuous alerts.

(c) Attacks may have a delayed impact.

Fig. 6. IIDS performance metrics can show a skewed picture when to be detected physical anomalies (green) are misaligned with the actual attack timing (red).