Deployment Challenges of Industrial Intrusion Detection Systems

Konrad Wolsing[1,2][0000−0002−7571−0555], Eric Wagner[1,2][0000−0003−3211−1015], Frederik Basels[2], Patrick Wagner[1], and Klaus Wehrle[1][0000−0001−7252−4186]

[1] RWTH Aachen University, Aachen, Germany, *last*@comsys.rwth-aachen.de
[2] Fraunhofer FKIE, Wachtberg, Germany *first.last*@fkie.fraunhofer.de

Abstract. With the escalating threats posed by cyberattacks on Industrial Control Systems (ICSs), the development of customized Industrial Intrusion Detection Systems (IIDSs) received significant attention in research. While the existing literature proposes effective IIDS solutions evaluated in controlled environments, their deployment in real-world industrial settings poses several challenges. Adding to known obstructions, this paper highlights two critical aspects that significantly impact IIDSs' practical deployment, i.e., the need for sufficient amounts of data to train the IIDS models and the challenges associated with finding suitable hyperparameters, especially for IIDSs training only on normal ICS data. Through empirical experiments conducted on multiple state-of-the-art IIDSs and diverse datasets, we establish the criticality of these issues in deploying IIDSs in ICS environments. Our findings show the necessity of extensive malicious training data for supervised IIDSs, which can be impractical considering the complexity of recording and labeling attacks in actual ICSs. Furthermore, while other IIDSs circumvent the previous issue by requiring only benign training data, these can suffer from the difficulty of setting appropriate hyperparameters, which likewise can diminish their performance. By shedding light on these challenges, we aim to enhance the current understanding of limitations and considerations necessary for deploying effective cybersecurity solutions in ICSs, which might be one reason why IIDSs see few deployments.

Keywords: Industrial Intrusion Detection Systems · Cyber-Physical Systems · Industrial Control Systems · Hyperparameter · Deployment.

1 Introduction

Industrial Control Systems (ICSs), ranging from manufacturing over power grids to water and gas distribution, are facing harmful consequences due to cyberattacks [3, 17]. The protection of such facilities is, however, not trivial as many systems rely on insecure legacy communication protocols, replacement of which is cumbersome, expensive, and often unrealistic due to high uptime requirements [15]. Consequently, recent research focuses on easily retrofittable Industrial Intrusion Detection Systems (IIDSs) specifically designed to take advantage of the unique characteristics of each ICS by searching for anomalous behavior in largely predictable networking patterns and physical processes [22].

The foundation of these detection mechanisms is mostly rooted in classical *supervised* machine-learning or *One-Class Classifiers (OCCs)* [22, 34]. In supervised approaches, the IIDS is trained on labeled samples of normal behavior *and* attacks to learn classifiers, e.g., Random Forests (RFs) or Support Vector Machines (SVMs) [27]. Meanwhile, OCCs are trained only on normal ICS behavior, e.g., to identify the operational boundaries of physical measurements [33], and deviations from this learned behavior are classified as potential attacks.

Research demonstrates the alleged effectiveness of hundreds of newly proposed IIDSs by evaluating them on dedicated datasets and publishing achieved detection performances [8, 22]. In vitro, these IIDSs achieve excellent results [12, 20, 24, 33]. However, when it comes to real-world deployments, these solutions are challenging to configure [10] and then cannot perform as promised [2, 30] and thus do not find their way into practice [29]. Consequently, the performance derived by current evaluation methodologies seems hardly representative of the actual quality of an IIDS if deployed in the real world [6]. While the scientific literature already identifies challenges for transferring IIDSs from research into practice [2, 6, 30], we proclaim that two aspects impacting IIDSs' deployability in ICS environments require additional attention.

First, it remains unclear how much training data is required to maximize detection performance. This question is especially critical in the case of supervised IIDSs, where the generation of attack samples in a testbed might still be relatively easy, but collecting or generating real-world attack samples is much harder [7]. OCCs' training data, on the other hand, is easily collectable, but they still require hyperparameter tuning [18]. Yet, hyperparameter tuning is rarely intuitive, especially with often-employed custom classifiers, and it remains unknown whether it is possible to transfer good hyperparameters between ICS deployments as considered feasible in other machine-leraning domains [28]. In research, the authors thus may optimize them for a given dataset (with attacks), which is, however, unfeasible in practice due to lack of attack samples.

Intrusion detection for ICS is especially challenging because of hard to obtain and labeled *attack samples* from cyberattacks in real systems, as their generation could expose, disrupt, and potentially damage sensitive infrastructure or facilities. For artificial datasets and testbeds as used in research [8], on the other hand, it is relatively easy to generate such attack samples. Thus far, IIDS proposals do, however, all require custom training phases for the concrete deployment with hardly any model transferability across scenarios [11, 34]. We thus observe a large discrepancy between training data availability for research activities and real-world deployments, which may be the cause for the reported challenging deployment of current research proposals [2, 29]. In this publication, we measure the severity of these factors on the detection performance of diverse IIDSs to understand how big their influence in potential deployments can become.

Contributions. To investigate the potential influence of training data availability on IIDSs' deployability, we make the following contributions:

- We demonstrate that the amount of attack samples in training significantly influences the performance of IIDSs based on supervised machine-learning.
- We show that the influence of hyperparameters for OCC-based IIDSs varies tremendously. While some IIDSs are susceptible to even tiny changes, others are largely hyperparameter-agnostic and even generalize across deployments.
- Based on our findings, we advocate for more expressive IIDS evaluation procedures to narrow the gap between research and real-world IIDS deployments.

Availability Statement. To facilitate further research, we publish the artifacts from our paper: https://zenodo.org/records/10728074

2 Background on Industrial Intrusion Detection

For readers unfamiliar with the topic of *industrial* intrusion detection, we motivate the rationale of retrofitting such solutions to ICS. We then present one IIDS from the literature in detail, which we evaluate in this publication.

Industrial Control Systems (ICSs) are the foundation of diverse applications such as manufacturing, production, distribution of water, gas, or electricity, and autonomous vehicles [17]. One typical architecture that all these applications rely on are digital control loops measuring the environment with sensors and influencing it through actuators usually interconnected with industrial control networks [15]. Consequently, ICSs are likewise susceptible to regularly occurring threats from cyberspace [3]. For their mitigation, either preventive measures such as authenticated and encrypted communication channels [9] or detective approaches like IIDSs [22] can be implemented. This publication focuses on the latter, which aim at timely indicating malicious behavior to ICS operators before actual harm can be conducted and avoid attacks remaining uncovered.

To detect unwanted behavior, the detection methodologies underlying *industrial* IDSs make great use of domain knowledge and ICS-specific behavior [34]. One key attribute is ICSs' notorious predictability, as they usually perform repetitive tasks [15]. Based on a set of training data, a detection model can be trained and tuned with hyperparameters to indicate unexpected deviations, such as cyberattacks. Note that supervised IIDSs require attack samples while OCC methods solely train on benign data. The goal of each approach and their tuning is to detect as many cyberattacks as possible while emitting few false positive alerts, which would have to be falsified by operators afterward. The performance of an IIDS is ultimately measured with metrics [22] like the F1 score.

One approach to implementing such an OCC-based IIDS is MinMax (cf. Fig. 1) [33]. It is based on the fact that physical values measured by sensors usually reside within precise limits, e.g., a boiler inside an ICS has a lower and upper operational temperature. MinMax extracts these limits from a set of benign training data. Then, since physical measurements can underlie natural variation and noise, the approach enlarges these limits by a configurable hyperparameter to avoid too many false positive alarms. In the end, an alarm is raised if a measurement exceeds or undercuts the trained threshold. Note that MinMax

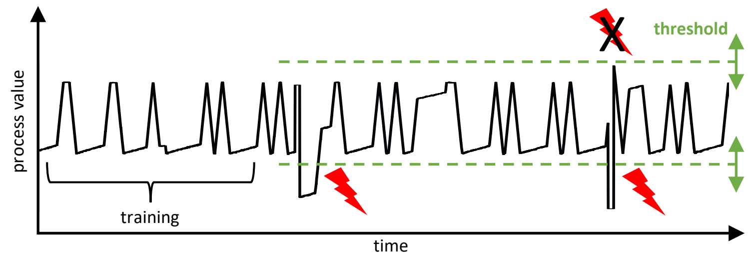


Fig. 1. An IIDS learns the repetitive patterns of an ICS's behavior to indicate anomalies. This requires finding a suitable hyperparameter, such as the threshold for MinMax visualized here [33], which influences the alert decision of an IIDS.

serves as an example and IIDSs generally exhibit complex decision-making algorithms. Finally, the ICS operators are in charge of analyzing the raised alarm and initiating countermeasures.

For OCC-based IIDSs, as depicted here, the training requires benign data recorded, e.g., during normal ICS behavior. Still, for deployment, hyperparameters, i.e., the threshold, have to be adequately selected to reduce the number of false-positives and not miss attacks (cf. Fig. 1). Contrary, while supervised IIDSs can find adequate hyperparameters themselves during training as they also learn on malicious samples, this requires obtaining or generating such (attack) data in an ICS potentially involving actual physical processes [7].

3 Open IIDS Deployment Challenges in ICS

After a short primer on IIDSs, we now highlight deployment challenges of IIDSs along recent related work, reproducibility studies, and meta-reviews (Sec. 3.1). Afterward, we formulate the research questions addressed in this paper (Sec. 3.2).

3.1 Related Work

For IDS research, there exists a body of meta-studies that critically reflect their effectiveness and suitability. In that regard, Sommer et al. [30] argue, not specifically focusing on industrial networks, that machine-learning is better suited for finding similarities than differences, which complicates their application in anomaly detection. Moreover, it is challenging to conduct sound evaluations, which they presume to be the reason why most approaches cannot keep up with expectations in real deployments. Adding to these issues, Ahmed et al. [2] identify scalability, exhaustive system modeling during training, and noisy input data as challenges seldom evaluated in live deployments. Also challenges like operational drift and component aging become only apparent in real deployments [25]. While issues of applying general machine-learning in practice are well-known [6], the effects of training *industrial* IDSs in artificial scenarios and their implications for potential deployments have thus far not been experimentally analyzed.

Moreover, general machine learning research has examined the importance of hyperparameter tuning [6, 18]. Here, we are mostly concerned with second-level hyperparameters, i.e., hyperparameters that must be set prior to training [28]. To obtain a general understanding of the tunability of these second-level hyperparameters, Probst et al. [28] analyzed six supervised machine-learning algorithms. They found good default values working on many datasets and identified those hyperparameters worth considering for tuning. In a similar study, Weerts et al. [32] found out that leveraging default hyperparameters was non-inferior to tuning them. However, all these works mostly consider supervised machine-learning and neither look at OCC nor tackle the peculiarities in ICS. Regarding the latter, default values found in these works do not apply to the entirely different and custom OCC-based IIDS algorithms usually found in ICS research. Focusing on ICS, Fung et al. [14] show exemplarily that three considered IIDSs deliver mostly stable performance under different hyperparameters. However, the set of tested hyperparameters is derived from attack samples, which may not be available (in high quantity or quality) for real deployments.

3.2 Research Questions

The deployment of IIDSs in real industrial networks proves challenging, with experimental deployments failing to keep up with promising results from artificial scenarios. We suspect training data availability, especially samples of attacks, to be one potential cause for this situation. Detection algorithms themselves are often applicable to multiple industrial domains [34]. However, they assume to be trained separately for each deployment to learn the expected behavior. For example, the learned boundaries of a water tank's maximum acceptable fill level differ for each IIDS deployment. Consequently, it is inevitable to train an IIDS for a specific target use case. Yet, this challenge of training an IIDS is not critically reflected in research where simply another (existing) dataset can be leveraged. To verify our suspicion and improve future evaluation methodologies of IIDSs to reflect their actual deployability into real-world scenarios, we answer four key research questions within this paper.

Q1 – How many attack samples do supervised IIDSs need? The training of supervised IIDSs requires benign *and* malicious data samples. E.g., one of the most commonly used dataset [22], the Morris Gas dataset [26], consists of 274.628 samples, of which 22% are attacks. For evaluations, authors usually randomly shuffle and split this dataset, leveraging 80% for training and the rest for evaluations [4, 27]. With this split, the training data still contains around 48.000 attack samples. Yet, obtaining this amount of labeled attack samples from each ICS an IIDS should be deployed is unrealistic considering the costs and risks associated with their generation leading to our question.

Q2 – How much training data do OCC-based IIDSs need? IIDSs requiring only benign training data can be trained with less difficulty, e.g., even during the regular operation of an ICS. However, this training data must still be collected, and it must be ensured that it reflects *all* possible normal behavior.

Hence, we want to understand how much training data is actually necessary and whether large variances exist across detection methods.

Q3 – What is the influence of hyperparameters on performance? Beyond training data, OCC-based IIDSs request hyperparameters, which may significantly impact detection performance. Here, the MinMax IIDS introduced in the background (cf. Sec. 2) uses a fixed threshold across datasets, whereas an optimized threshold could drastically influence detection performance, as evidenced in Fig. 1. However, such hyperparameter tuning is only possible if attack samples for the concrete deployment scenario are available.

Q4 – Can we transfer good hyperparameters across scenarios? To unlock the benefits of tuned hyperparameters in OCC-based IIDSs, we consider the previously proposed concept of transferring good configurations across deployment scenarios [28, 32]. Such a step would also allow us to use the extensively available attack samples from artificial scenarios to tune real-world deployments. However, thus far, it remains unclear to what extent such transferability is possible and to what extent this is scenario and IIDS dependent.

4 Deployability of Supervised IIDS

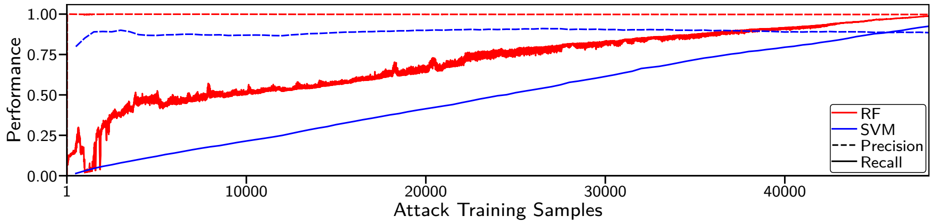
Our initial analysis concerns the deployability of supervised IIDSs w.r.t. the amount of required attack samples. We first describe our experiment design, then analyze our results, and finally summarize the implications of our findings.

4.1 Experiment Setup

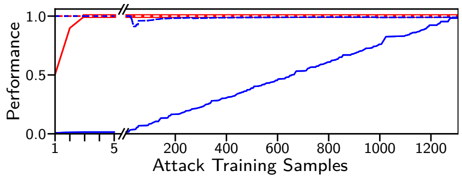
For our experiments on supervised IIDSs, we consider a RF and a SVM classifier as used in several proposed IIDSs [4, 19, 27]. As independently examined by Perez et al. [27] and Anton et al. [4], these classifiers can be adapted to operate on Modbus network traffic via derived features such as the function code or transmitted process values. The classifiers are trained and evaluated on a set of benign and malicious Modbus packets. Our experiment is based on existing re-implementations of these two IIDSs made available in the IPAL IDS Framework [34]. We took care to use the same data preprocessing and hyperparameters as mentioned in the publication [27] (cf. Availability Statement).

Concerning the datasets, we leverage the same dataset originally used to evaluate the two IIDSs [4, 27], which is also the most commonly used dataset for supervised IIDSs [22]. This dataset has been recorded in a miniature gas-pipeline ICS lab-environment leveraging Modbus as communication protocol. Within this setup, a total of 60048 attack samples across 35 types of attacks with varying complexity, such as reconnaissance or modifying setpoints, have been collected. Whether accumulating and labeling this amount of attack data outside a lab in the field is actually possible remains questionable.

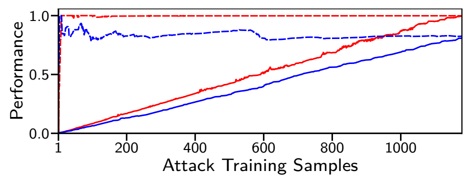
To understand how many attack samples are necessary to train a supervised IIDS, we reduce the number of attacks contained in the training dataset while keeping the number of benign training data constant. We start with a random



(a) After an initialization phase, the recall increases linearly with more attack samples, while changes in precision are only minimal. To yield high detection scores, more than 40.000 malicious packets are required in training for both supervised IIDSs. Note that the data for SVM was sampled in steps of 500 attacks due to long training times.



(b) The RF requires few training samples on attack number 19 (clean registers).



(c) Attack 10 (change physical value) requires many samples to be trained.

Fig. 2. Gradually increasing the amount of attack samples within the training data reveals that both RF and SVM require lots of data to yield satisfying detection performance. For a simple attack, cf. Fig. 2(b), the RF requires only about three samples.

80/20 train/test split as in the original evaluation [4, 27] and five folds. We then remove all but one attack sample from the training data and retrain the IIDS while gradually increasing the amount of attacks in the training data. The 20% of the test set remain unchanged after the random train/test split. For each number of learned attack samples, we calculate the average recall (fraction of identified attacks) and precision (fraction of correct alerts) over all folds.

4.2 Q1 – How many attack samples do supervised IIDSs need?

Having established our methodology, we can now exemplarily assess the deployability of supervised IIDSs w.r.t. to the amount of attack samples. To this end, Fig. 2 depicts the detection performance of the RF and SVM classifiers. In Fig. 2(b) and Fig. 2(c), we show two exemplary attack types in isolation.

Starting with a broad overview in Fig. 2(a), if the RF is trained on all available attack samples ($x = 48.049$), it reaches a precision of 0.998 and a recall of 0.987. Likewise, the SVM achieves a score of 0.885 in precision and 0.924. As we expect, both IIDSs achieve the best detection performance when trained on all (those in the original train set) available attack samples.

As we reduce the number of attack samples, we observe a nearly linear reduction in recall of both IIDSs. For RF, the performance drops from 0.987 to about 0.44 if provided with just 5000 attack samples. Below this threshold, the recall for

RF drops even more drastically. The SVM shows a similar trend with fewer fluctuations. Interestingly, precision remains largely unaffected in both cases, which we presume results from not changing the amount of benign training data.

To better understand these effects, we repeat this experiment only considering a single attack type during training and testing. Exemplarily, we show attack type 19 from the dataset [26] in Fig. 2(b) and attack type 10 in Fig. 2(c). For attack type 19, we observe a vastly diverging behavior between RF and SVM. The RF achieves optimal detection rates after just three attack samples. The IIDS has likely learned to identify that this attack uses a Modbus function code not occurring during normal behavior. In contrast, this generalization does not apply to the SVM. Attack type 10, shown in Fig. 2(c), which manipulates reported sensor readings, proves difficult to learn for both IIDSs. Here, the recall continues to grow linearly as more attack samples are available for training. Overall, we see that only with a high number of malicious training samples can the IIDSs score the excellent detection results reported in the respective publications.

4.3 Conclusion

Looking back at our results, we see that supervised IIDSs can generalize an attack pattern in some cases as observed, for example, for the RF classifier for attack type 19, which introduces an otherwise unused Modbus function code. This attack should thus also easily be detected by simple rule-based IIDSs [13]. In general, however, we observe relatively little generalization for both IIDSs. The linearly increasing recall scores with increasing the number of attack samples rather indicate an overfitting behavior of the classifiers, i.e., only the precise misbehavior observed during training is also later classified as such. These results provide further evidence for prior work by Kus et al. [21], who already identified a lack of generalization during supervised IIDS training.

As proclaimed by Etalle et al. [11], we also find that supervised IIDSs are rather unsuited for deployments in diverse ICS environments due to only performing well with many attack samples, potentially due to overfitting, which aligns with prior research [2, 11, 21, 25]. Yet, these issues are hardly discussed in prior work as publications promoting the use of machine-learning in ICS mainly focus on the final achieved detection performance [4, 19, 27]. Consequently, novel designs for supervised IIDSs must be critically reviewed to be considered suitable for real ICS deployments.

5 Deployability of OCC-based IIDS

OCC-based IIDSs promise to avoid these issues of supervised IIDSs by requiring only training data from benign ICS operations. Getting such benign data is easier than collecting or generating attack samples, but it must still be collected, processed, and verified, such that requiring less training data makes an OCC-based IIDS easier to deploy. Moreover, hyperparameter tuning, especially if hyperparameters cannot be transferred across scenarios, can still unrealistically boost

Table 1. We analyze four state-of-the-art IIDSs with diverse hyperparameters, on three datasets. We aim at 10.000 random samples for each IIDS's hyperparameter space, yet we have reached computational limits, resulting in fewer samples for some.

IIDS	SWaT [16]	WADI [1]	BATADAL [31]	Parameter
MinMax [33]	10 000	10 000	10 000	2
Invariant [12]	703	10 000	1088	10
TABOR [24]	10 000	10 000	10 000	7
Seq2SeqNN [20]	231	182	500	6

an OCC-based IIDSs' performance in research. To understand these effects, we first lay out the evaluation setup underlying our measurements (Sec. 5.1) to then tackle the research questions Q2 to Q4. In the end, we summarize our findings on the deployability challenges of OCC-based IIDSs (Sec. 5.5).

5.1 Experiment Setup

We examine four IIDSs, which were published at top security conferences – MinMax [33], Invariant [12], TABOR [24], and Seq2SeqNN [20]. We make use of available open-source implementations or validated re-implementations within the IPAL IIDS framework [34]. For further details, we refer the reader to Appx. A, the respective publications, or IPAL's public implementation [34]. These four IIDSs, which also feature vastly different numbers of hyperparameters for their configuration, cf. Tab. 1, build the foundation for our analysis.

To generalize our results, we analyze three datasets, namely the SWaT [16], WADI [1], and BATADAL [31], which are among the most commonly used datasets in this research area [22]. All three datasets come with dedicated training data that is free of attacks. The testing part of SWaT and WADI contains 36 and 14 different cyberattacks respectively while BATADAL provides 12 attacks.

While the previous experiment's design decisions coincide with usual IIDS evaluation methodologies [22], our work differs within the hyperparameter selection we aim to study. Although three of the examined IIDSs' publications contain short discussions about (some) hyperparameters [12, 20, 33], none defines the precise acceptable range of the hyperparameter space. To this end, we have to come up with our own definition. For nominal and ordinal hyperparameters, we simply enumerated all possible values, and for rational numbers, we had to define a custom range based on our understanding of the proposed system. During their definition, we took special care that the values proposed in the original publications are contained in our analyzed ranges.

Finally, to conduct a parallelized examination of the hyperparameter in a repeatable manner, we leveraged Ray Tune [23], a library to scale hyperparameter search and tuning. Provided with a definition of a hyperparameter search space, Ray Tune selects one hyperparameter configuration uniformly at random at a time and then trains and evaluates the respective IIDS on the dataset. We then calculate the precision, recall, and F1 score metrics, as these are among the most common performance metrics in IIDS research [22].

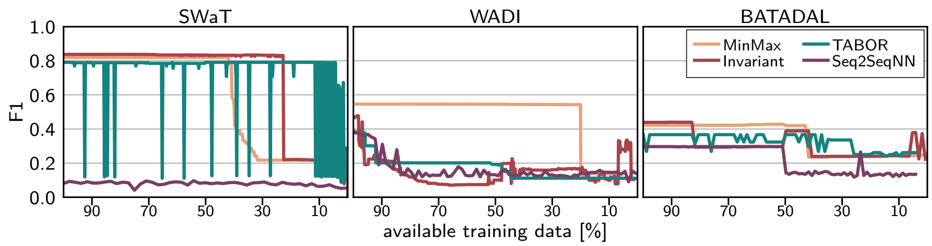


Fig. 3. Reducing the amount of benign training data likewise diminishes the detection rate but not all IIDSs experience an equal performance reduction. The training data Seq2SeqNN on SWaT and WADI is reduced in steps of 10.000 in contrast to 1.000 for the others whereas all IIDSs on BATADAL are sampled in steps of 100.

We aimed to achieve up to 10.000 samples for all IIDS and dataset combinations, building a solid foundation for our subsequent analyses, cf. Tab. 1. In some cases, such as evaluating the Invariant IIDS on SWaT, training a single configuration takes up to eleven days, which explains the reduced number of samples. Similarly, the training of the Seq2SeqNN IIDS requires exclusive access to potent GPUs to train a neural network. To grant other researchers access to the result of these extensive computations for further analyses, we made all collected data publicly available, cf. Availability Statement.

5.2 Q2 – How much training data do OCC-based IIDSs need?

First, we want to understand the impact of the amount of (benign) training data on IIDS performance. Here, we only consider the best hyperparameters found w.r.t. the F1 score for each IIDS and dataset combination. Beginning with the entire training data (100%), we gradually reduced the training data and evaluated the IIDS after each training against the entire test dataset.

As shown in Fig. 3, the amount of training data impacts the detection performance of each IIDS differently. E.g., the performance of MinMax on SWaT and BATADAL initially stays high. Only when the data is reduced to about less than 40% does the performance drop significantly. On WADI, this drop occurs much later at about 20% of the overall training data. For Invariant, we observe a similar pattern on SWaT achieving top scores even with about 25% of the data. Yet, on WADI, this approach requires nearly all training data to get close to its optimal score. TABOR on SWaT shows another interesting behavior where instead of a slow reduction, we observe occasional drops in performance, which accumulate toward the end. Upon investigation of TABOR's trained model, we noticed that the drops in between are caused by learning a different model, showing the unstable nature of the trained model. This also occurs in reduced form for BATADAL but not on WADI, where TABOR shows a more continuous reduction as less training data is made available. Seq2SeqNN performs poorly on SWAT and on the other datasets its performance drops significantly as training data is reduced to 50%. Overall, we observe that all IIDSs perform nearly optimally on SWaT and BATADAL with just about half of the training data, while performance on WADI often quickly drops off.

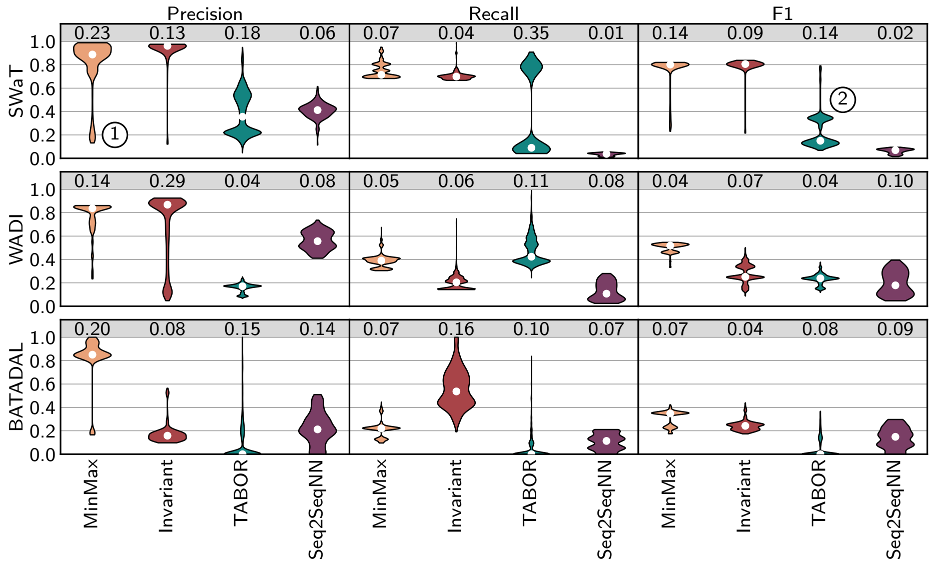
Takeaway. Our data shows that judging upfront whether one has acquired enough training data in a deployment scenario can be challenging. The amount of training data seems to be neither directly dependent on the IIDS nor on the complexity of the concrete scenario. As Invariant and Seq2SeqNN on WADI experience a substantial increase close to 100% training data, this may be an indication that these IIDSs would benefit from even more training data than contained in the dataset. We also see that the performance of the different IIDSs drops suddenly after a certain point, indicating that not observing some specific event during training can be responsible for much of the performance loss. Interestingly, the IIDSs seem to have different events triggering their performance loss. When interpreting these results, dataset characteristics should also be kept in mind. E.g., SWaT contains one attack that is significantly longer than the others, which significantly worsens the F1-score if it is not detected anymore. Hence, the sudden drops of MinMax and Invariant on SWaT could be explained by the sudden inability to identify that specific attack. Overall, determining the amount of training data varies across IIDSs and scenarios, such that a final assessment can only be made on a case-by-case basis. Nonetheless, over all datasets, utilizing more training data does not diminish the detection performance.

5.3 Q3 – What is the influence of hyperparameters on performance?

Next, for our investigation on the significance of hyperparameters (Q3), we take a broad view of the obtained measurements (cf. Sec. 5.1). To this end, Fig. 4 depicts every IIDS's performance distribution along several metrics and datasets.

At first glance, we observe that hyperparameters have a tremendous effect on the performance of IIDSs. E.g., considering the precision of the MinMax IIDS on the SWaT dataset (cf. ① in Fig. 4), the performance varies between 0.99 at best and 0.13 at worst, which implies that, depending on the chosen configuration, the approach performs close to optimal or is inapplicable. But looking at the entire distribution, it becomes apparent that low values in recall are outliers as the median performance (white dots) is still high at 0.89. Still, the standard deviations around the median is relatively high at 0.23, and thus, performance penalties can be expected for MinMax in recall if not parameterized correctly.

Taking a broader look at the precise distribution of different approaches, not all IIDSs exhibit the same patterns. When considering MinMax and Invariant for SWaT in the F1 score, the majority of configurations perform decently, and bad results are mostly outliers. We call this type of distribution *stable* as it is quite likely to pick a good-performing configuration without having to invest great efforts. In contrast, the opposite is true for TABOR ②, with a median of just 0.15, which is far from what could be achieved at best (0.79) with this approach. Here, unlike MinMax, it is quite unlikely to hit such a good-performing configuration even with expert knowledge. Therefore, there is a qualitative difference between the presumably stable MinMax, which promises to have a straightforward configuration process [33], and TABOR. Note that for MinMax, these



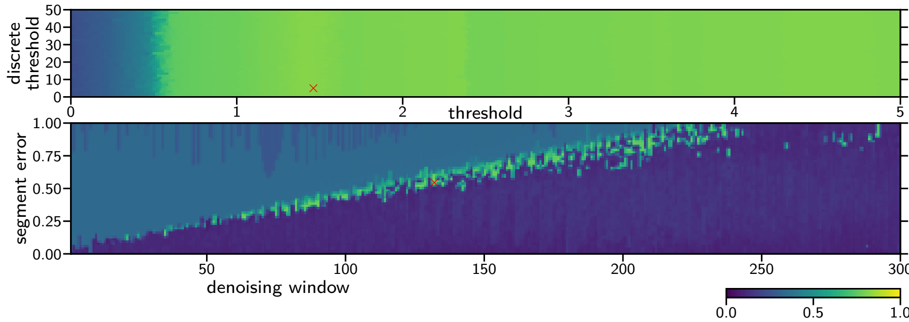
The white dot represents the median and the numbers on top are the standard deviation of each distribution.

Fig. 4. An IIDSs' performance depends on an optimal choice of hyperparameter. While MinMax or Invariant yield satisfying results in F1 score on SWaT for the majority of configurations, obtaining a good configuration for TABOR is challenging. Thus, judging the expectable performance of an IIDS by a single number can be misleading.

observations may be affected by only having two hyperparameters in the first place (cf. Tab. 1). Still, Invariant, despite having the most parameters, features a similar stable distribution as MinMax, at least w.r.t. the F1 score.

Next, we want to understand whether the (in-)stability property is inherent to a specific IIDS. First and foremost, note that the absolute scores achieved between the datasets (cf. lower part of Fig. 4) are sometimes lower compared to SWaT as not all IIDSs were primarily designed for the other datasets. Hence, we only focus on the distributions here. In general, the distributions are loosely similar in each setting. The performance distributions of most IIDSs have roughly the same features on WADI and BATADAL, with some exceptions, such as WADI missing the outliers to the top in some cases. This observation indicates that the stability of an IIDS may be dominantly determined by the underlying detection mechanism rather than the scenario. Consequently, stability seems to be an inherent feature of an IIDS, which could act as a proxy for determining how easy or difficult deploying an IIDS in a new, real application may be.

Considering MinMax, the authors publish their IIDS with a F1 score of 0.78 for SWaT and 0.52 for WADI [33]. W.r.t. our evaluation, these numbers are close to the median (SWaT 0.8 and WADI 0.52) and leave headroom to the maximum (0.82 respectively 0.55). Thus, the published numbers are representative of the expectable performance, which comes as no surprise as the authors stated not



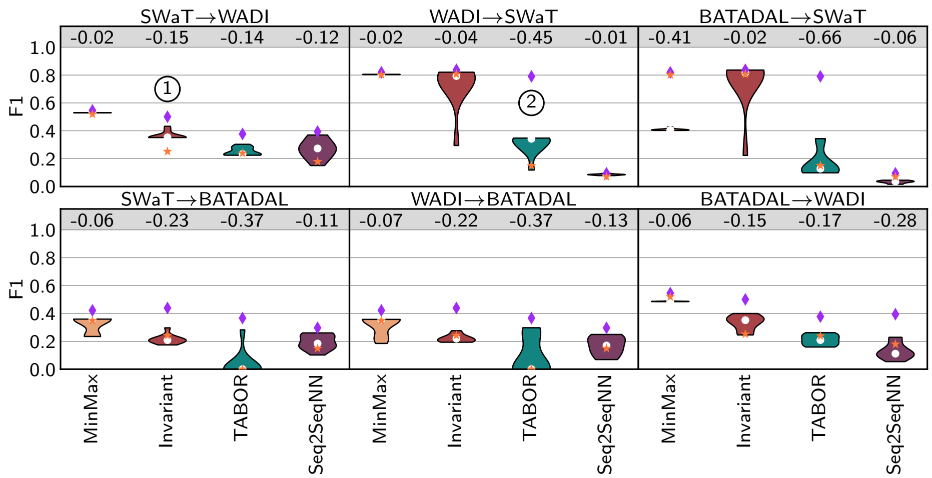
The red cross (×) indicates the optimum found during the experiment.

Fig. 5. The impact of hyperparameters can vary significantly between approaches. While on the SWaT for MinMax (upper plot), one parameter is decisive for the entire performance, suitable configurations for TABOR (lower plot) are more challenging to obtain as several parameters influence each other.

to have performed any parameter optimization [33]. In contrast, Feng et al. [12] promote the Invariant IIDS with a recall of 0.79 for SWaT and 0.47 for WADI. Compared to the median performance (SWaT 0.7 and WADI 0.2), the published values are outliers by multiple standard deviation. Therefore, it can be assumed that Feng et al. published optimized performance statistics. Such fine-tuning certainly has value in examining what maximal performance can be achieved by a proposed approach. However, it risks misrepresentation how good a system may perform in a real deployment and may prevent fair comparisons of approaches.

Given the distinct behaviors IIDSs show under varying hyperparameters, we also asked what the reasons for these behaviors might be. Therefore, we take a closer look at MinMax and TABOR on the SWaT dataset and F1 score, as depicted in Fig. 5, where we visualize the detection performance as a heatmap in dependence of two relevant hyperparameters. For MinMax, we identify that the final result mostly depends only on the threshold parameter (cf. Sec. 2). For small values (below 0.6), there is a significant drop in detection performance, but afterward, there are only subtle changes and the threshold has no significant impact anymore. In contrast, for TABOR, we observe more interdependence in two of the seven hyperparameters. Both parameters influence the performance, and changes to one parameter alter the optimal value of the other parameter. Thus, only a combination of correctly set hyperparameters yields good configurations, which complicates setting up TABOR and explains our previous observation where only a few configurations yielded good performance.

Takeaway. We observed that hyperparameters have a tremendous impact on the measured performance of OCC-based IIDSs. Moreover, there exist considerable differences in IIDS stability. The MinMax or Invariant IIDSs yield results that are close to their optimal in a majority of configurations. At the same time, TABOR only achieves optimal performance if multiple hyperparameters are fine-tuned, cf. Fig. 5. Our results stand in contrast to Fung et al., who



The numbers on top are the difference between the median (white dot) and maximum achievable performance (♦).

Fig. 6. Transferring the top ten configurations found for one dataset to another promises to avoid the problem of parameter optimization in new settings. But, this methodology usually lacks behind the achievable optimum (♦) and does not systematically exceed randomly selected hyperparameters, cf. ⋆ marking the median of all hyperparameters in the target dataset.

claimed that reconstruction-based IIDSs can have a good performance over a broad spectrum of hyperparameters [14], likely because our evaluation covered a more diverse set of IIDSs. This (in-)stability w.r.t. hyperparameters complicates scientific comparisons and real-world applicability if the performance of an IIDS is only acceptable for a very confined parameter space. Consequently, we warn that judging an IIDSs' performance by a single configuration, as done currently throughout the literature, can be misleading.

5.4 Q4 – Can we transfer good hyperparameters across scenarios?

As we discussed in the previous section, it can be difficult to obtain suitable hyperparameters for an IIDS for a given deployment or dataset. For the selection of suitable hyperparameters, we do, however, not need to start from scratch in most cases. Instead, published parameters or guidelines from previous deployments may help to identify good parameters. Thus, one idea is to reuse these already known configurations and transfer them to a new scenario to hopefully achieve adequate performance. If such hyperparameter transfers are feasible, it would alleviate the problem of (in-)stability discussed before. Previously, Probst et al. [28] found universally good-performing default hyperparameters for supervised IIDSs. However, we consider OCCs-based IIDSs with potentially more intricate hyperparameters that may hinder such transferability.

As the first step in that direction and to examine whether a known, good-performing configuration is also suitable on a different dataset, we conducted the following evaluation. First, we select the top ten configurations according to the F1 score of an IIDS and dataset, e.g., MinMax on SWaT, and measure the performance of these hyperparameters applied to the other datasets, i.e., WADI and BATADAL. Applying this methodology, Fig. 6 depicts the distribution of the obtained ten results on the new datasets. In addition, we mark the globally achievable optimum in a given setting found in the previous analysis from Sec. 5.3 with an ♦ and the median performance of randomly selected hyperparameters with an ⋆. This analysis enables assessing how likely a configuration is transferable to another scenario without tuning but also depicts the potential losses in performance along the way. Note that since we sampled all hyperparameters randomly (cf. Sec. 5.1), it is not guaranteed that we measured the precise configuration on the respective other datasets. In that case, we selected the measurement closets to the selected default configuration.

We start considering the Invariant IIDS transferred from the SWaT to the WADI dataset as a case study first (cf. ① in Fig. 6). We see that the transferred configurations achieve decent performance relative to the achievable maximum of 0.5. The median expectable performance from transferring the configurations (white dot) is −0.15 points lower than the maximum achievable. Given that no effort was required to find these configurations, this median of the transferred configurations (0.35) is an improvement over the previous random median performance (0.25, cf. Fig. 4). In contrast to this example, there also exist cases where hardly any transferability is possible. When considering TABOR transferred from the WADI to the SWaT dataset (cf. ② in Fig. 6), there is a large gap between the median of the transferred configurations (0.23) and the achievable optimum (0.79). While in that case, transferring the results is still better than picking a random configuration (0.15), a large potential is left on the table. More generally, the median transferred performance (white dots) is, on average, 0.18 lower than the respective achievable optimum (♦). At the same time, this method is equal to randomly selecting a configuration (no difference to ⋆ on average). Thus, while transferring configurations from one scenario to another seems promising, this concept still proves not be be that advantageous.

Takeaway. On the one hand, OCC-based IIDSs lack guidelines for hyperparameter configuration. On the other hand, if known configurations exist, they only offer limited transferability. On average, good hyperparameters on one dataset do not perform better than randomly chosen hyperparameters on another dataset. Thus, if an IIDS is challenging to configure in the first place, even default configurations or templates from other scenarios do not help much, and, in the worst case, manual efforts are required to tune the approach individually.

5.5 Conclusion

We began with the observation that the amount of data required for training differs significantly between IIDSs (Q2), which complicates providing concrete advice for deployments. Still, as one redeeming feature, more training data does not seem to negatively impact the detection performance. Next, we studied the hypothesis that hyperparameters are a crucial factor for IIDS performance and again observed vastly different behaviors w.r.t. stability. Indeed, it proved difficult for some IIDSs to yield good results on average (Q3), and quick solutions such as generic default configurations that generalize to new scenarios or datasets did not prove promising (Q4). In contrast to the works by Probst et al. [28] and Weerts et al. [32], we found that deriving default values for hyperparameters of OCC-based IIDSs for the ICS domain is challenging for our three analyzed datasets and tuning them manually based on attack samples still brings an enormous performance benefit. Therefore, these effects can explain previously reported problems from related work, e.g., failed reproducibility studies or deploying such approaches in practice [2, 11, 30]. I.e., Erba et al. [10] tried to reproduce the Invariant IIDS and had troubles finding the hyperparameters to match the publications result. This is in line with our assumption from Sec. 5.3 where we presumed that the authors of the Invariant IIDS have tuned their published parameters. Consequently, current evaluation methodologies in research omit a relevant attribute of IIDSs that is currently not easily measurable.

In general, obtaining a quantitative intuition on an IIDS's training and tuning demands can provide valuable data on the one hand, for ICS operators having to select, set up, and configure an IIDS and, on the other hand, for research to establish fairer and easier comparisons. Note that we do not want to prioritize an IIDS with low training and low tuning demands over ones with excellent detection performance. Instead, we want to create awareness for these challenges and advocate for researchers to scrutinize their work more w.r.t. their deployability.

6 Open Deployment Issues and Limitations

Our results regarding the analysis of research questions Q1 to Q4 prominently show that there exist complex challenges to transferring an IIDS developed in research to an actual ICS that are not captured accurately by the current standard in IIDS evaluations. Hence, the standard procedure of publishing the detection performance for one or multiple datasets [22] is insufficient to capture an IIDS' true value. Concerning these issues, we now discuss new strategies to assess the ease and limitations of an IIDS' deployment already during the research stage.

One significant obstruction in deploying IIDSs is acquiring sufficient training data (Q1 and Q2) whilst avoiding overfitting of supervised IIDS models. From a research perspective, an adopted evaluation methodology that more deeply assesses the capabilities and especially training properties of an IIDS in the lab may be suited to estimate its training demand upfront. In that regard, the evaluation methodologies we presented enable, one the one hand, inferring the learning rate from which the amount of required training data for a deployment can be estimated. On the other hand, by visualizing the learning rate of individual attacks, first signs of overfitting can be revealed. In addition, the methodologies proposed by in related work [5, 21] can answer how well a supervised approach generalizes to unknown attacks, e.g., found during live operations, which are not part of the training data. Together, such enhanced evaluation methodologies can reveal IIDSs that a) require little training samples and b) generalize to a wide variety of (zero-day) cyberattacks beyond the ones seen in training.

With the previous issues addressed, the challenge of configuring an approach (Q3 and Q4) remains. For research, analyzing (new) IIDSs w.r.t. their stability in hyperparameters or ease of configuration, as done by us, can provide additional information for ICS operators on which IIDS may be best suited for a given deployment. Therefore, we ideally need a compact metric that expresses the average performance or stability of performance results. While the data generated in our publication would allow us to compute such values (cf. Fig. 4), how to arrive at a holistic metric that is adequate for scientific purposes is still unclear to us. Another idea for better understanding OCC-based IIDSs is to use a few attack samples from reference attacks to configure hyperparameters. Whether this yields good hyperparameters to detect other attacks remains to be seen.

More generally, while the previously sketched concepts may work well for research, it is not directly apparent how their insights transfer to actual deployments. Also, deployability, in general, involves more than recording training datasets and configuring hyperparameters [6]. E.g., the issue of operational drifts such as wear and tear, which can invalidate once-trained models over time, has been neglected by us [25]. Answering whether an IIDS is ultimately deployable in an actual system thus likely has to involve the expertise of ICS stakeholders as already demanded in meta-surveys, e.g., by Lamberts et al. [22]. Regarding our work, we can, therefore, not finally argue how much training data would still be acceptable or how many false positives and false negatives are tolerable without conducting experiments together with ICS experts within an actual ICS.

7 Conclusion

ICSs become an indispensable building block for our modern society and, with their high level of digitalization, face potentially disastrous cyberattacks. As a reaction, research to automatically detect such intrusions took off within the last decade [22], and nowadays, with plenty of promising IIDSs, the transition to deploying those solutions in real-world ICSs is urgently needed. Yet, this step involves its own challenges, of which we assess and quantify two in detail in this paper. Especially in industrial settings, the acquisition of adequate training data, avoiding overfitting during training, and the configuration of hyperparameters for IIDSs to match their excellent detection performance found in (synthetic) research environments is challenging. As we show, too little training data or tuning of hyperparameters can lead to devastating performance penalties.

While finding solutions to those issues would require the involvement of ICS stakeholders that ultimately deploy IIDSs, we, from a research perspective, recommend taking those properties into account while evaluating novel approaches. Thereby, we can hopefully shift these deployability challenges more into the focus of researchers who design new intrusion detection methods.

Acknowledgments. This work is part of the project MUM2 and was partially funded by the German Federal Ministry of Economic Affairs and Climate Action (BMWK) with contract number 03SX543B managed by the Project Management Jülich (PTJ). This paper was partially supported by the EDA Cyber R&T project "Cyber Electromagnetic Resilience Evaluation on Replicated Environment (CERERE)", funded by Italy and Germany. The authors are responsible for the publication's contents.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. C. Ahmed, V. R. Palleti and A. P. Mathur: WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In: CySWATER. ACM (2017). https://doi.org/10.1145/3055366.3055375
2. C. M. Ahmed, G. R. M. R. and A. P. Mathur: Challenges in Machine Learning Based Approaches for Real-Time Anomaly Detection in Industrial Control Systems. In: CPSS. ACM (2020). https://doi.org/10.1145/3384941.3409588
3. T. Alladi, V. Chamola and S. Zeadally: Industrial control systems: Cyberattack trends and countermeasures. Comput. Commun. **155** (2020). https://doi.org/10.1016/j.comcom.2020.03.007
4. S. D. D. Anton, S. Sinha and H. Dieter Schotten: Anomaly-based intrusion detection in industrial data with svm and random forests. In: SoftCOM (2019). https://doi.org/10.23919/SOFTCOM.2019.8903672
5. G. Apruzzese, P. Laskov and J. Schneider: Sok: Pragmatic assessment of machine learning for network intrusion detection. In: IEEE EuroS&P (2023). https://doi.org/10.1109/EuroSP57164.2023.00042
6. D. Arp, E. Quiring, F. Pendlebury et al.: Dos and Don'ts of Machine Learning in Computer Security. In: USENIX Security Symposium (SEC) (2022)
7. L. Bader, M. Serror, O. Lamberts et al.: Comprehensively Analyzing the Impact of Cyberattacks on Power Grids. In: IEEE EuroS&P (2023). https://doi.org/10.1109/EuroSP57164.2023.00066
8. M. Conti, D. Donadel and F. Turrin: A Survey on Industrial Control System Testbeds and Datasets for Security Research. IEEE Communications Surveys & Tutorials **23**(4) (2021). https://doi.org/10.1109/COMST.2021.3094360
9. M. Dahlmanns, J. Lohmöller, J. Pennekamp et al.: Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things. In: ASIACCS. ACM (2022). https://doi.org/10.1145/3488932.3497762
10. A. Erba and N. O. Tippenhauer: Assessing Model-free Anomaly Detection in Industrial Control Systems Against Generic Concealment Attacks. In: ACSAC (2022). https://doi.org/10.1145/3564625.3564633
11. S. Etalle: From Intrusion Detection to Software Design. In: ESORICS. Springer (2017). https://doi.org/10.1007/978-3-319-66402-6_1
12. C. Feng, V. R. Palleti, A. Mathur et al.: A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In: NDSS. Internet Society (2019). https://doi.org/10.14722/ndss.2019.23265
13. I. N. Fovino, A. Carcano, T. De Lacheze Murel et al.: Modbus/DNP3 State-Based Intrusion Detection System. In: AINA. IEEE (2010). https://doi.org/10.1109/AINA.2010.86

14. C. Fung, S. Srinarasi, K. Lucas et al.: Perspectives from a Comprehensive Evaluation of Reconstruction-based Anomaly Detection in Industrial Control Systems. In: ESORICS. Springer (2022). https://doi.org/10.1007/978-3-031-17143-7_24
15. B. Galloway and G. P. Hancke: Introduction to industrial control networks. IEEE Communications Surveys & Tutorials **15**(2) (2013). https://doi.org/10.1109/SURV.2012.071812.00124
16. J. Goh, S. Adepu, K. N. Junejo et al.: A Dataset to Support Research in the Design of Secure Water Treatment Systems. In: CRITIS. Springer (2016). https://doi.org/10.1007/978-3-319-71368-7_8
17. A. Humayed, J. Lin, F. Li et al.: Cyber-physical systems security—a survey. IEEE Internet of Things Journal **4**(6) (2017). https://doi.org/10.1109/JIOT.2017.2703172
18. F. Hutter, L. Kotthoff and J. Vanschoren: Automated machine learning: methods, systems, challenges. Springer Nature (2019). https://doi.org/10.1007/978-3-030-05318-5
19. K. N. Junejo and J. Goh: Behaviour-based attack detection and classification in cyber physical systems using machine learning. In: CPSS (2016). https://doi.org/10.1145/2899015.2899016
20. J. Kim, J.-H. Yun and H. C. Kim: Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks. In: CyberICPS. Springer (2020). https://doi.org/10.1007/978-3-030-42048-2_1
21. D. Kus, E. Wagner, J. Pennekamp et al.: A False Sense of Security? Revisiting the State of Machine Learning-Based Industrial Intrusion Detection. In: CPSS. ACM (2022). https://doi.org/10.1145/3494107.3522773
22. O. Lamberts, K. Wolsing, E. Wagner et al.: Sok: Evaluations in industrial intrusion detection research. Journal of Systems Research **3**(1) (2023). https://doi.org/10.5070/SR33162445
23. R. Liaw, E. Liang, R. Nishihara et al.: Tune: A Research Platform for Distributed Model Selection and Training (2018). https://doi.org/10.48550/arXiv.1807.05118
24. Q. Lin, S. Adepu, S. Verwer et al.: TABOR: A Graphical Model-based Approach for Anomaly Detection in Industrial Control Systems. In: ASIACCS. ACM (2018). https://doi.org/10.1145/3196494.3196546
25. G. R. M. R., C. M. Ahmed and A. Mathur: Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. Cybersecurity **4** (2021). https://doi.org/10.1186/s42400-021-00095-5
26. T. H. Morris, Z. Thornton and I. Turnipseed: Industrial control system simulation and data logging for intrusion detection system research. In: SCSS. CAE in Cybersecurity Community (2015)
27. R. L. Perez, F. Adamsky, R. Soua et al.: Machine Learning for Reliable Network Attack Detection in SCADA Systems. In: IEEE TrustCom. IEEE (2018). https://doi.org/10.1109/TrustCom/BigDataSE.2018.00094
28. P. Probst, A.-L. Boulesteix and B. Bischl: Tunability: Importance of hyperparameters of machine learning algorithms. Journal of Machine Learning Research **20**(53) (2019), http://jmlr.org/papers/v20/18-444.html
29. S. Seng, J. Garcia-Alfaro and Y. Laarouchi: Why anomaly-based intrusion detection systems have not yet conquered the industrial market? In: Foundations and Practice of Security (2022). https://doi.org/10.1007/978-3-031-08147-7_23
30. R. Sommer and V. Paxson: Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In: SP. IEEE (2010). https://doi.org/10.1109/SP.2010.25

31. R. Taormina, S. Galelli, N. O. Tippenhauer et al.: Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. Journal of Water Resources Planning and Management **144**(8) (2018). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000969
32. H. J. P. Weerts, A. C. Mueller and J. Vanschoren: Importance of tuning hyperparameters of machine learning algorithms (2020). https://doi.org/10.48550/arXiv.2007.07588
33. K. Wolsing, L. Thiemt, C. van Sloun et al.: Can Industrial Intrusion Detection Be SIMPLE? In: ESORICS. Springer (2022). https://doi.org/10.1007/978-3-031-17143-7_28
34. K. Wolsing, E. Wagner, A. Saillard et al.: IPAL: Breaking up Silos of Protocol-dependent and Domain-specific Industrial Intrusion Detection Systems. In: RAID. ACM (2022). https://doi.org/10.1145/3545948.3545968

A IDS Description

To measure the deployability of OCC-based IIDSs, we examined four existing approaches. In the following, we provide a short description of their concept:

MinMax. The first IIDS, MinMax, learning the minimum and maximum bounds of a sensors' normal values (cf. Sec. 2), serves as a representative for a class of lightweight IIDSs that aim to implement straightforward detection methodologies that do not require complex configuration, technical understanding, or computational resources [33]. Any violation against the learned minimum and maximum values is indicated as an alert to the ICS operators.

Invariant. This IIDS [12] leverages data mining techniques to find mathematical equations that must be fulfilled at all times. E.g., if the inlet valve of a water tank is opened, its water level is expected to rise. Since an invariant is fulfilled all the time during normal behavior, any violation is then reported.

TABOR. This IIDS fuses three detection approaches based on timed automata, Bayesian networks, and out-of-bounds checks [24]. The timed automata component considers a single sensor value and learns a model of its behavior. E.g., the water levels of a tank usually rise for 30 minutes and then decrease over several hours. Together with the Bayesian network, unknown process states can be determined, such as the inlet valve being still opened despite the water level rising for more than 30 minutes. To complement their method, an alert is also raised with an out-of-bounds check working similarly to the MinMax IIDS.

Seq2SeqNN. Lastly, Seq2SeqNN [20] trains a neuronal network on GPUs to understand the ICS's behavior and perform predictions for the future. Given a recent history of physical values, the neuronal network is able to perform a prediction for the near future. If these predictions deviate too much from the observed behavior, an alarm is raised.