A False Sense of Security? Revisiting the State of Machine Learning-Based Industrial Intrusion Detection

Dominik Kus[*,†], Eric Wagner[†,*], Jan Pennekamp[*], Konrad Wolsing[†,*],
Ina Berenice Fink[*], Markus Dahlmanns[*], Klaus Wehrle[*,†], and Martin Henze[‡,†]
[*] *Communication and Distributed Systems*, RWTH Aachen University, Germany · {lastname}@comsys.rwth-aachen.de
[†] *Cyber Analysis & Defense*, Fraunhofer FKIE, Germany · {firstname.lastname}@fkie.fraunhofer.de
[‡] *Security and Privacy in Industrial Cooperation*, RWTH Aachen University, Germany · henze@cs.rwth-aachen.de

ABSTRACT

Anomaly-based intrusion detection promises to detect novel or unknown attacks on industrial control systems by modeling expected system behavior and raising corresponding alarms for any deviations. As manually creating these behavioral models is tedious and error-prone, research focuses on machine learning to train them automatically, achieving detection rates upwards of 99 %. However, these approaches are typically trained not only on benign traffic but also on attacks and then evaluated against the same type of attack used for training. Hence, their actual, real-world performance on unknown (not trained on) attacks remains unclear. In turn, the reported near-perfect detection rates of machine learning-based intrusion detection might create a false sense of security. To assess this situation and clarify the *real* potential of machine learning-based industrial intrusion detection, we develop an evaluation methodology and examine multiple approaches from literature for their performance on unknown attacks (excluded from training). Our results highlight an ineffectiveness in detecting unknown attacks, with detection rates dropping to between 3.2 % and 14.7 % for some types of attacks. Moving forward, we derive recommendations for further research on machine learning-based approaches to ensure clarity on their ability to detect unknown attacks.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; • **Computing methodologies** → *Machine learning*; • **Networks** → Cyber-physical networks.

KEYWORDS

anomaly detection; machine learning; industrial control system

ACM Reference Format:
Dominik Kus, Eric Wagner, Jan Pennekamp, Konrad Wolsing, Ina Berenice Fink, Markus Dahlmanns, Klaus Wehrle, and Martin Henze. 2022. A False Sense of Security? Revisiting the State of Machine Learning-Based Industrial Intrusion Detection. In *Proceedings of the 8th ACM Cyber-Physical System Security Workshop (CPSS '22), May 30, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3494107.3522773

CPSS '22, May 30, 2022, Nagasaki, Japan
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9176-4/22/05…$15.00
https://doi.org/10.1145/3494107.3522773

1 INTRODUCTION

With ongoing digitization, Industrial Control Systems (ICS) are increasingly networked and connected to the Internet [43, 48], thus suspending the long-deployed air-gap principle as primary protection against intrusion. However, legacy ICS devices were usually not designed to implement adequate network security and are rarely replaced due to high costs and long device lifetimes [32, 48]. Consequently, ICS are increasingly targeted by cyberattacks with potentially severe damage [24, 27], exposing the glaring security deficits of ICS, which stem most importantly from weak or missing protection mechanisms [1, 37]. To alleviate this situation, security mechanisms must be retrofitted for Internet-connected ICS devices.

Network-based intrusion detection systems (IDSs) [40, 49] constitute a promising approach for such retrofitting. While *signature*-based IDSs detect attacks using pre-configured signatures, *e.g.*, a specific sequence of network packets, *anomaly*-based IDSs model the expected behavior of a system and consider deviations as potential intrusion, *e.g.*, a control parameter outside physical bounds. Thus, while signature-based IDSs can only identify known attacks, anomaly-based IDSs promise to also detect novel attacks [29, 51].

Anomaly-based intrusion detection is particularly well-suited for ICSs, as industrial devices usually exhibit regular and predictable communication patterns [25] that remain largely unchanged over time and ease the creation of behavioral models. However, the individual and application-specific use of industrial devices requires tailoring IDSs to the deployed system, involving high effort [5].

A promising approach to address this issue for *industrial* IDSs (IIDSs) is the application of machine learning (ML). ML algorithms can be trained on historic ICS data, thereby learning properties of the physical system and attacks on it. Hence, ML algorithms supersede the manual crafting of system models in anomaly detection and signatures in signature-based detection. Furthermore, the ability of ML to generalize and abstract patterns allows even operating on "new" (*i.e.*, unseen) data. However, classifying ML-based IDSs as signature- or anomaly-based [50], *i.e.*, determining whether an IDS learns normal behavior, attack signatures, or both, is non-trivial due to the intransparency of the learning process within ML.

Related work has proposed various ML-based IIDSs, either by training exclusively on benign network traffic from the ICS [20] or by training on a mix of benign and malicious traffic [28], indicating almost perfect detection performance in excess of 99 % [35]. However, widely-used performance evaluation methods, *e.g.*, metrics such as precision, recall, or F_1-score, only cover the ability of an IIDS to detect known attacks and do *not* capture their ability to detect new variations or even entirely new types of attacks, which is the core promise of anomaly-based over signature-based detection.

We argue that, especially for systems trained on a randomly sampled mix of benign and malicious traffic, it is debatable whether those IIDSs can actually realize anomaly detection as opposed to only learning signatures of trained attacks. Thus, also taking unseen attacks into account, the *actual* performance of ML-based IIDSs remains unclear to this point. By only providing evaluation results on known attacks while claiming to perform anomaly detection, such approaches create the impression of almost perfect protection and lead to a false sense of security in real-world deployments.

Thus, clarification of the *real* potential of ML-based IIDSs to offer comprehensive protection is urgently needed. While literature on attacking specific IIDSs [19] and on performing stealthy attacks against ICS [21] exists, there is a lack of research w.r.t. the performance of IIDSs on unknown attacks. In this paper, we address this issue by examining and evaluating multiple ML-based IIDSs from literature for their performance on unknown attacks.

Contributions. Our main contributions are as follows.

- We derive a methodology to evaluate the generalizability of ML-based IIDSs to detect attacks they have not been trained on.
- Using our methodology, we evaluate existing ML-based IIDSs on attacks that were deliberately excluded from training. Our results show that these approaches only perform well on attacks they have been trained on, leading to a false sense of security.
- By further training ML-based IIDSs on only *one* attack type, we assess their ability to work in scenarios where the amount of available training data is limited. We show that ML-based IIDSs do not generalize well enough to capture new (unknown) attacks.

Availability Statement. To foster further research and ensure reproducibility, our evaluation framework and evaluation artifacts are available at: https://github.com/COMSYS/ML-IIDS-generalizability.

2 INDUSTRIAL INTRUSION DETECTION

Complementing preventive security measures such as firewalls, encryption, and authentication, intrusion detection acts as an important additional safeguard to discover remaining attacks [32, 57]. In the following, we provide the necessary background on intrusion detection and argue that its passive nature eases retrofittability and thus makes it especially attractive for industrial networks (Section 2.1). We then specifically focus on ML-based intrusion detection (Section 2.2), as it promises strong detection capabilities for advanced attacks on industrial networks [12, 28, 32].

2.1 Traditional vs. Industrial IDSs

Intrusion detection systems (IDSs) passively monitor the behavior of individual devices (host-based) and/or communication between devices (network-based) [32] to discover attacks or suspicious behavior. To this end, the core idea behind IDSs is that attacks lead to observably different behavior than normal system operation and can thus be detected. In traditional IDS settings such as office or server networks [53], attacks are typically spread out widely, *e.g.*, through Internet-scale malware. Rule-based IDSs, such as YARA [6], Suricata [40], or Zeek [52] (previously Bro [49]), can often reliably detect them. Consequently, state-of-the-art IDSs in these traditional settings often rely on signatures or rules as attack indicators. They raise an alarm whenever device or communication behavior matches any of the predefined rules or signatures [31].

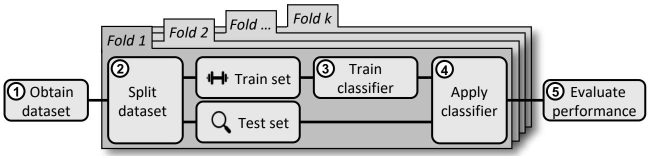


Figure 1: k-fold cross-validation allows to accurately evaluate the performance of a machine learning classifier. For each fold, different parts of the dataset are used for training and testing. Eventually, the full dataset is classified (tested) once.

Especially in industrial settings, IDSs promise to be an easily deployable safety net for otherwise often insufficiently secured industrial networks [17, 18, 48]. In particular, network-based IDSs can easily be integrated into existing infrastructure without time-consuming and costly changes to deployed devices or software. Furthermore, the unique characteristics of industrial networks and processes facilitate the use of intrusion detection: In contrast to traditional IT networks (office or server), communication in industrial networks follows a significantly more regular and predictable pattern [25, 59], *e.g.*, sensor readings that are refreshed with a fixed periodicity [3]. As such, an underlying assumption of industrial IDSs is that attacks likely lead to a distinctly different communication behavior. At the same time, the strong interdependence between industrial processes and the communication necessary to monitor and control them allows IDSs to detect even subtle attacks, such as minor manipulations to the water's acidity in a water treatment plant [56]. Therefore, the industrial context is uniquely suitable for the deployment of, especially anomaly-based, IDSs.

However, while industrial networks provide ample additional opportunities to detect attacks, these attacks typically cannot easily be described using rules or signatures (as it is possible for traditional IT networks) [61]. For example, subtle attacks might send unsuspicious network packets (*i.e.*, those also appearing in legitimate communication), but cleverly time these to bring a supervised process into an unsafe state [56]. As detecting such attacks pushes rule- or signature-based IDSs beyond their limits [13, 28], a large research community has gathered around ML approaches to also detect advanced attacks on industrial networks [12, 36, 60].

2.2 Machine Learning for Industrial IDSs

The premise of machine learning (ML) for industrial intrusion detection is simple: "Automatically" learn what constitutes benign and malicious behavior to later classify observed behavior without having to care about details [50] of the underlying physical process and communication. We explain the training and evaluation process of an ML-based industrial IDS in the following alongside Figure 1. After successful training, the IDS can then be deployed to *predict* whether observed samples (*e.g.*, a sequence of network packets) constitute an attack.

The first step to creating an ML-based IIDS is to obtain a suitable ICS dataset covering nominal operation *and* labeled attack patterns (Step ①). In Step ②, this dataset is split into a training (used for creating the classifier) and a testing dataset (used for evaluating the classifier), *e.g.*, according to a predefined ratio, such as 80/20. The respective training process (Step ③) depends on the specific classifier, but usually relies on iterative optimization, *e.g.*, using gradient descent. After the classifier is trained, in Step ④, it is applied on the test set to predict whether the contained samples are malicious or not. Finally, in Step ⑤, the prediction results are validated against the known labels (from the dataset) to evaluate the classifier's performance (*i.e.*, how "well" it detects attacks).

In our simplified example, so far, the results are obtained using a single train/test split, *i.e.*, testing is performed on a single test set, providing only limited insights into the performance of the IIDS on different datasets. To mitigate this issue, cross-validation (CV) [47] is a structured approach for evaluating an IIDS on multiple train/test splits and thus increases confidence in the results. For k-fold cross-validation, the dataset is partitioned into k equal-sized parts. As shown in Figure 1, Steps ②–④ are then repeated k-times using a non-random train/test split where one of the k parts is used as the test set while the remaining $k-1$ parts form the train set. For the final evaluation of the approach, the results from each of those k *folds* are then aggregated, *e.g.*, by calculating the arithmetic mean.

While the used classifiers vary widely across approaches (*cf.* Section 3), the process to measure their performance (Step ⑤) remains the same. A classifier's predictions over the test set (*cf.* Step ④) are interpreted alongside four possible outcomes: *True positives (TP)* count the number of correctly identified malicious samples, while *false positives (FP)* count the number of benign samples incorrectly classified as malicious. Analogously, *true negatives (TN)* count the number of correctly classified benign samples, whereas *false negatives (FN)* count the number of malicious samples falsely classified as benign. A perfect classifier has only true positives and true negatives, *i.e.*, no false positives or false negatives.

Two widely used metrics based on these outcomes are *precision* and *recall* [11]. Precision represents the ratio of correctly identified malicious samples among all samples predicted as malicious ($^{TP}/_{(TP+FP)}$). Hence, the optimal value of 1 implies that all samples predicted as malicious indeed correspond to malicious behavior (*i.e.*, no false alarms). Recall represents the ratio of correctly identified malicious samples among all actually malicious samples ($^{TP}/_{(TP+FN)}$). For an optimal value of 1, all malicious samples were correctly identified (*i.e.*, no malicious sample is missed). Thus, these metrics complement each other and provide good insights into a classifier's performance, even with unbalanced datasets [11].

Optimally, both precision and recall would be 1; however, there is often a trade-off between the two values. Both metrics can be aggregated into the F_1-score ($^{2}/_{(Precision^{-1}+Recall^{-1})}$), which provides a single measure for a classifier's performance. Ergo, an F_1-score of 1 would be optimal and imply that both precision and recall are 1.

While employing ML for industrial intrusion detection offers various benefits, such as easier deployment [12], one fundamental drawback is that classifiers can only "learn" information and patterns for which corresponding training data exists [50]. Consequently, while ML promises to also detect advanced attacks on industrial networks that rules or signatures cannot cover, there is a latent apprehension that such approaches can only detect attacks they have been trained on and thus remain oblivious of all other, especially evolving and newly developed, attacks.

Takeaways. Intrusion detection, especially based on ML, is promising to easily retrofit industrial networks with capabilities to timely uncover attacks. However, there is an inherent risk that ML-based IDSs for industrial networks cannot generalize to detect attacks they have not been trained on.

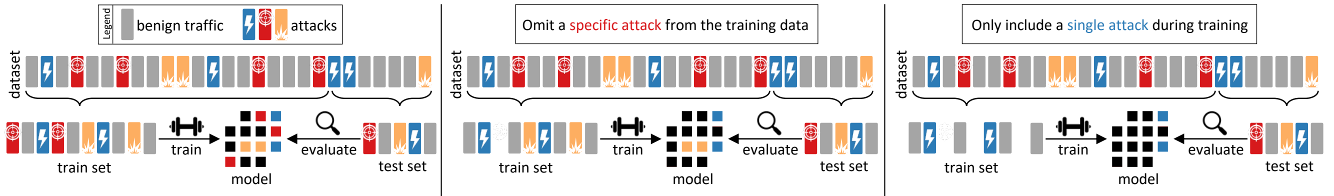
3 RELATED WORK ON ML-BASED IIDSS

In recent years, ML, with all its benefits and potential pitfalls, has seen widespread application in industrial intrusion and anomaly detection, and a plethora of ML-based anomaly detectors have been proposed [12, 36, 55]. Notably, supervised learning-based approaches gained significant interest for their high attack and anomaly detection rates [45]. Such supervised classifiers include random forests (RFs) [4, 7, 8, 10, 15, 28, 33, 35, 38], which utilize multiple decision trees to split a dataset's features into similar classes. Contrary, support vector machines (SVMs) [7, 8, 10, 13, 28, 30, 33, 35] map all features into a vector space and derive decision boundaries to separate individual classes. From a different angle, neural networks (NNs) [4, 14, 20, 23, 26, 28, 30, 35, 44, 46] mimic human brains and can be trained to model any function, e.g., classifying input features as benign or malicious. Besides this plurality in ML-based approaches, their common motivation is to increase utility compared to deterministic signature-based intrusion detection through (i) generalizability across domains and (ii) the ability to identify novel, not previously seen, anomalies.

However, whether machine-learning intrusion detection can indeed live up to these promises has been challenged recently [2, 19, 45, 60]. Concerning generalizability, recent investigations [19, 60] show that applying IIDSs to novel domains is not as straightforward, and its success depends on the underlying detection methods. With respect to the ability to detect unknown, *i.e.*, not trained on, attacks, even if detection scores for individual attack types are calculated [7, 13, 14, 20, 28, 35, 46], it remains unclear how classifiers handle unseen attacks or variations. The claimed strength of many ML-based IIDSs can be further questioned when considering that classifiers that only train on benign data [9, 22, 34] generally suffer from worse detection performance and higher false-positive rates [28]. While these discussions point out problems w.r.t. claimed generalizability of existing results beyond specific evaluation scenarios, they do not further focus on the underlying evaluation methodology in initially proposed scenarios.

To this end, problems with the evaluation methodology of ML-based systems in general and anomaly detectors in specific were raised by various related work [2, 11, 41, 45, 50, 51, 55]. For one, the interpretation of evaluation results is not straightforward, leading to false conclusions from misinterpreted data [11, 41]. Moreover, related work questions whether ML is actually suitable for anomaly detection, or if it is rather only able to detect variations of already seen attacks in office networks [50] as well as for ICSs [2, 45]. However, these statements are not backed with detailed analysis of the actual ability of ML to perform anomaly detection for ICS networks, where the predictable behavior under normal operation potentially allows to still detect anomalies. In a similar vein, the enhancement of existing datasets with additional, artificial attack signatures to more accurately present what constitutes anomalous behavior is proposed [54, 58]. However, the exact implications for ML-based IIDSs that train on those enhanced datasets remain unexplored.

Takeaways. Machine learning-based industrial intrusion detection received broad attention from research due to the promise to generalize across domains and detect previously unseen anomalies. Simultaneously, it becomes clear that such claims require much more scrutiny. While first detailed analyses of the generalizability



(a) Traditional ML approach, using a random sample of the dataset for classifier training.

(b) Excluding a specific attack during the training of the classifier to test its generalizability.

(c) Focusing on a single attack only to reveal interrelations between different trained attacks.

Figure 2: Traditional evaluations source their training data from a random sample. Thus, they fail to properly test the classifier's generalizability. By proposing a methodology consisting of two experiments, we intend to address this gap in IIDS research.

of industrial intrusion detection have been conducted [19, 60], to the best of our knowledge, the potential of (supervised) ML to detect unknown, *i.e.*, not trained on, anomalies remains unexplored. Thus, it remains unclear whether novel anomalies and variations of known attacks can be detected reliably by ML, demanding a methodology to perform such an analysis.

4 DISSECTING ML-BASED IIDS EVALUATIONS

The main goal of evaluating an IDS is to gain insight into its capabilities [50]. While performance metrics help to express achieved results, the underlying evaluation methodology is far more important to accurately rate IDS performance, as it dictates how to interpret generated metrics and what meaning they convey for practical, real-world performance. Given safety-related considerations in industrial settings, this issue is crucial for deployed IIDSs.

Due to the specific abstracting properties of ML, the underlying methodology is especially relevant for ML-based IIDSs. When surveying ML-based security evaluations, research discovered various prevalent pitfalls in a multitude of papers [11]. In our context, the selection of the test dataset is most concerning as it determines what scenarios the system is actually tested for and what scenarios are not credibly covered by the proposed IIDS.

To shine a light on this issue, we discuss the traditional, state-of-the-art evaluation methodology and its deficits w.r.t. the IIDSs' ability to generalize to new attacks (Section 4.1). We address those shortcomings by proposing an improved evaluation methodology that is specifically tailored to evaluating ML-based IIDSs (Section 4.2).

4.1 Today's Traditional Evaluation Approach

A commonly used approach to evaluate ML-based IIDSs is to apply a traditional evaluation methodology from the field of ML, as we illustrate in Figure 2(a). Here, the final evaluation is performed on the test set, which corresponds to a sample of the original dataset that was withheld during training. Thus, the classifier has not seen this exact data during the training step. Based on the classification results of the test set, metrics, such as precision and recall, express the achieved performance. We refer to Section 2.2 for more elaborate details on the general training and evaluation processes.

While precision- and recall-based metrics are generally recommended to address an imbalanced dataset and to prevent the base rate fallacy [11], the selection of the test set warrants further attention. When randomly sampling both training and testing data from the ICS source dataset, they most likely cover similar observations. Hence, they are very homogeneous (visualized by the same types of attacks in both train and test set in Figure 2(a)). Thus, assuming the measured performance to be indicative of real-world performance contains the implicit assumption that the real-world environment in which the IIDSs will be deployed is homogeneous to the train set—an assumption with severe implications.

While this assumption is reasonable for traditional ML applications, such as image classification or natural language processing, it is not suitable for security-related applications, and more specifically IIDSs, for the following reasons: (i) Identifying and predicting all perceivable risks is highly unlikely, and (ii) attackers are constantly adapting and improving their tools. Consequently, IIDSs are most likely confronted with novel attacks or variants of existing attacks during their deployment. Regardless, those scenarios are not captured by the above methodology: Given that the sampled test and train sets are homogeneous, the likelihood of new, unknown attacks is not accounted for while testing the classifier.

Unfortunately, for this reason, corresponding evaluations leave many questions unanswered. In particular, they fail to explain how systems react to new attacks and what their overall classification limitations are. However, these aspects are important to provide accurate evaluations [50]. Thus, today's inadequate evaluation methodology can lead to a warped perception of IIDS performance and create a false sense of security. To tackle this problem and to develop further insight into the IIDSs' capabilities, especially concerning their ability to detect unknown attacks in the wild, we propose an improved evaluation methodology for ML-based IIDSs.

4.2 Our Proposed Evaluation Methodology

In the following, we introduce our new methodology to allow for expressive performance evaluations in the context of ML-based IIDSs and thus address today's methodical shortcomings.

4.2.1 Methodology Overview. Overall, we propose a methodology to evaluate the performance of ML-based IIDSs that sources from two distinct experiments. Thereby, we intend to provide additional insights into the real-world performance of deployed IIDSs. In the first experiment, we investigate the IIDSs' ability to detect new attacks by deliberately omitting attacks from the training set. This setting synthetically simulates a situation where the IIDS is confronted with a new, unknown kind of attack during its deployment. For the second experiment, we only train the classifier for a single kind of attack (as well as benign data) by omitting other (known) attacks from the training set. This setting can help to put the results from the first experiment into perspective. It further highlights interrelations between different attacks present in the dataset. Finally, for settings where the intended generalization failed to reliably detect (new) attacks, it answers the question of whether "specialized" classifiers that only focus on single attacks are a promising way.

Our methodology is independent of the used classifier (*i.e.*, used IIDS) and the input dataset (which only must be segmentable into different classes of attacks). In this paper, we systematically analyze and compare the performance of three different ML-based IIDS classifiers (*cf.* Section 5). To ensure generalizable and significant results, we also mandate the use of cross-validation for our methodology. In Figures 2(b) and 2(c), we visualize how the train and test sets are prepared. Now, we elaborate on the individual experiments.

4.2.2 Detection of New, Unknown Attacks. Overall, we propose a method to analyze a classifier's ability to detect novel attacks during its deployment. To this end, we specifically adapt the dataset splitting (Step ② in Figure 1) prior to the classifier's training (Step ③). We illustrate the general idea for a single fold in Figure 2(b). The subsequent performance evaluation of the used classifier is then based on known metrics, *i.e.*, precision and recall.

As a prerequisite, the dataset is split into k parts of equal size, which corresponds to a k-fold cross-validation, *i.e.*, we use a single part for the test set while the remaining parts constitute the train set. When using $k = 5$, the resulting train set corresponds to approximately 80 % of the input data, and the remaining 20 % are used for the test set. Subsequently, for each type of attack, we conduct an evaluation. More specifically, we consider every attack to be unknown once, *i.e.*, we filter all instances of this attack from the train set and move them to the test set. While this approach alters the ratio between train and test set, its implications are usually negligible in practice due to the inherent imbalance between benign and malicious packets (in common ICS datasets). Thus, we prepare k folds for each type of attack where the instances related to the attack are only part of the test set, *i.e.*, overall, we repeat the process of Figure 2(b) ($k \times$ #attack) times to evaluate a classifier.

After training and testing on all these folds, we individually compute the performance metrics (typically, precision and recall) for each type of attack. Inspecting those results can provide an overall impression of how well the classifier abstracts from the specifics of the dataset, *e.g.*, by learning process-specific parameters or communication patterns of the ICS, to detect attacks without relying on specific pre-trained attack patterns. Thus, our methodology provides an understanding of whether the ML-based IIDS *only* detects known attacks (*i.e.*, being limited in its practical use), or whether it is also able to generalize the input data—a central promise of ML-based approaches—*i.e.*, to also reliably detect "anomalies" in safety-critical, real-world deployments.

4.2.3 Independently Evaluating Attacks. Following this first experiment, we intend to obtain an improved understanding of how the classifier learns, deals with, and abstracts from seen (trained on) attacks. For evaluations, these insights are instrumental in different ways: (i) They underline why we observe a generalizability for specific types of attacks, (ii) they allow us to identify interrelations between attacks, and (iii) thus, they also assist in understanding results that we obtained by performing the first experiment.

We propose a second experiment that focuses only on individual attacks. To this end, we again adapt the dataset splitting (Step ② in Figure 1). When compared to the first method, the overall setup is very similar: We only modify the applied filter as we detail in Figure 2(c) to only keep one attack at a time in the train set. In particular, for each attack, we fully exclude all other attacks from the train set, *i.e.*, we filter all malicious instances not belonging to the respective attack from the train set and move them to the test set. Thus, we train a classifier on a single type of attack. For a complete evaluation, we repeat this process ($k \times$ #attack) times.

Overall, the first experiment corresponds to an "all-but-one" approach and the second experiment follows an "only-one" evaluation.

While we primarily want to study the classifier concerning specific attacks, the observed results might also reveal interrelations between different types of attacks in the dataset. In addition, this experiment can highlight issues of the classifier related to underfitting, *i.e.*, the used classifier is not able to properly handle all types of attacks in the dataset simultaneously. Thus, it provides important insights into today's challenges with ML-based IIDSs. To address the issue of underfitting, a potential approach could be to also train a "specialized" classifier for this specific, challenging attack only.

Takeaways. Today's state-of-the-art evaluation methodology for ML-based IIDSs fails to consider the approaches' ability to detect new, unknown attacks. As this property is crucial to properly assess their capabilities (especially in light of generalization), we propose a methodology consisting of two experiments. Thereby, we are able to obtain further insights into how IIDSs deal with attacks and how well they generalize to novel forms of attacks, *i.e.*, how much security they can provide in real-world settings.

5 REVISITING THE EVALUATION OF IIDSS

To assess the impact of shortcomings in today's evaluation methodology for ML-based IIDSs (*cf.* Section 4.1) and shine a light on their true capability to detect novel attacks, we apply our methodology proposed in Section 4.2 to three ML-based IIDSs from literature.

To this end, we first discuss our experimental setup, including the examined IIDSs, the used dataset, and the specific application of our methodology (Section 5.1). The results of omitting certain attacks or attack categories from training show that the examined approaches are largely unable to detect novel attacks (Section 5.2). Training the IIDSs on single attacks or categories reveals that cross-learning between attacks and categories is limited to some special cases. Classifiers mostly learn the attacks on which they are explicitly trained on. Combining the results from both experiments details that the examined IIDSs, despite contrary promises, behave much more like signature-based than anomaly-based IDSs (Section 5.3).

5.1 Experimental Setup

Over the last years, a plethora of ML-based IIDSs have been proposed and evaluated without much concern about how these generalize to new attacks (*cf.* Section 3). In this paper, we specifically revisit the performance of three recently proposed IIDSs based on different ML algorithms: RFs, SVMs, and BLSTMs [35]. These IIDSs constitute prime candidates for our analysis as (i) they feature official open-source implementations, and (ii) research has independently validated and reproduced their reported performance [60]. Specifically, for our evaluation, we rely on existing IIDS implementations that utilize the industrial abstraction layer IPAL [60].

Table 1: Attack categories as introduced for the gas pipeline dataset [39] that we use during our methodology evaluation.

ID	Abbr.	Descriptive Name	#Attacks
1	NMRI	Naïve Malicious Response Injection	4
2	CMRI	Complex Malicious Response Injection	7
3	MSCI	Malicious State Command Injection	5
4	MPCI	Malicious Parameter Command Injection	12
5	MFCI	Malicious Function Code Injection	3
6	DoS	Denial of Service	1
7	Recon	Reconnaissance	3

While our experiments arguably would benefit from a larger set of IIDSs, we were unable to consider additional implementations. Due to the lack of publicly available artifacts, we initially tried to contact the authors of four recent publications. Unfortunately, we only received a single negative response, indicating that the main author no longer works at the corresponding lab. Subsequently, we attempted to re-implement these approaches on our own. However, despite this effort, we were unable to reproduce the reported results. Regardless, given the variety of our considered ML algorithms, covering both traditional ML (RFs and SVMs) as well as deep learning techniques (BLSTMs), we are confident to report representative results in this paper. In the future, other researchers can easily repeat our experiments for other classifiers due to the use of IPAL.

For our evaluation, we require datasets that ideally contain multiple instances of labeled attack types (*cf.* Section 4.2). In particular, in this paper, we rely on an established dataset from a gas pipeline ICS that has been specifically designed for cybersecurity research [39]. At the hardware level, the ICS consists of a pressure sensor and two actuators (a pump and a solenoid valve) that are automated by a control system to regulate the pipeline's pressure. In total, the dataset contains 274,628 network packets, out of which about 22 % belong to labeled attacks. These attacks stem from 35 different attack types designed for this ICS, which are grouped into 7 categories, as we detail in Table 1.

For *NMRI*- and *CMRI*-related attacks, the attacker injects malicious sensor readings and setpoints to manipulate control algorithms. Attacks that are included in *MSCI*, *MPCI*, and *MFCI* send manipulated commands to actuators to change the state of devices or interfere with their communication. The *DoS* category consists of a single type of attack in which bad CRC checksums are used to disrupt a device's functionality by overloading it with invalid messages. Finally, the *Recon*-related attacks attempt to obtain insights into the ICS's operation by scanning for devices and their functionality. For the ordering of attacks within a category, we follow the original publication.

For both described experiments (*cf.* Section 4.2), we conducted our evaluation on two different aggregation levels: (a) by omitting/training all attacks from a specific attack category (*cf.* Table 1), and (b) by focusing our analysis on the individual 35 attack types to analyze the generalization *within* and *between* attack categories. In line with best practices, we use 5-fold cross-validation throughout our evaluation to retain the 80/20-split between train and test set from the original publication [39]. We report on the average recall and precision values across our 5 folds and forgo examining the differences between the folds as they are irrelevant for our analysis.

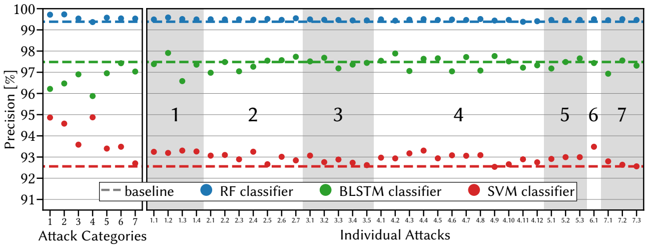


Figure 3: Precision over the test set changes only minimally when attack categories (left and shaded in on the right) or individual attacks (right) are omitted from training. Notably, BLSTM's precision falls below the baseline in many cases, while the precision for RF and SVM generally exceeds it.

5.2 Understanding the Impact of Novel Attacks

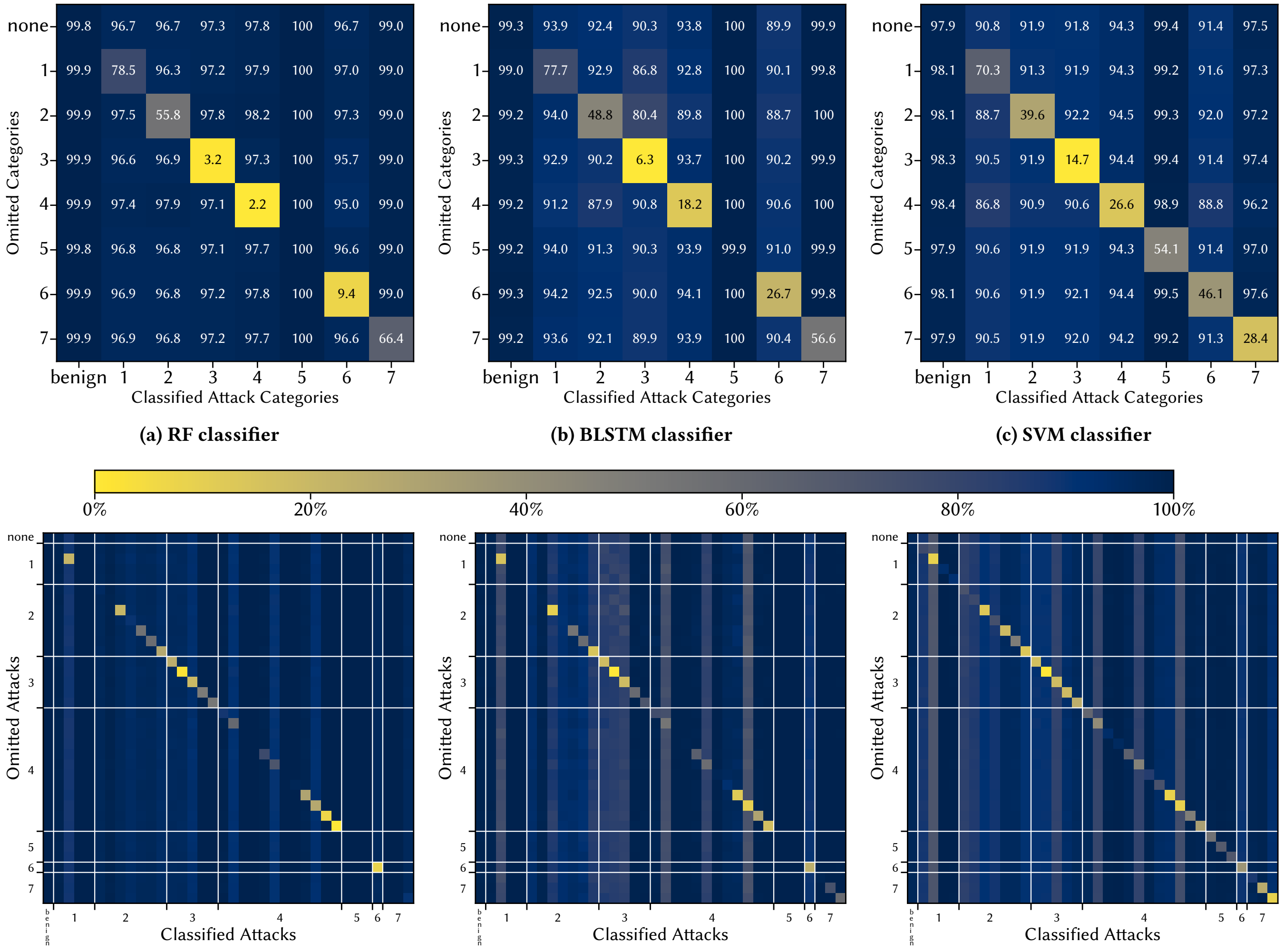
In our first analysis (*cf.* Section 4.2.2), we investigate how the three ML-based IIDSs perform when they are challenged to classify novel, previously unseen, attacks. To this end, we omit individual attacks or entire categories one by one during training. As established earlier (*cf.* Section 2.2), we have to consider both precision and recall to gain a complete understanding of a classifier's performance.

In Figure 3, we thus analyze the effects on precision first. For the RF- and SVM-based classifiers, we observe a slight improvement, indicating that the classifiers can separate benign and malicious traffic easier if the trained on attacks are less diverse. Contrary, the BLSTM performance decreases by up to 1.6 percentage points if certain attacks are not seen during training, especially if entire attack categories are not trained on, indicating difficulties identifying benign behavior. In general, the effects on precision when omitting certain attacks from the train sets are, however, rather marginal.

Besides precision as a performance metric, recall is the key metric for our evaluation (and real-world deployments) as it allows measuring how the detection of an attack type or category changes when it is omitted from training. Optimally, we would expect an anomaly detector to retain good recall for an attack even if it is omitted from training, indicating its ability to generalize beyond the known attacks. Next, we discuss how well ML-based IIDSs actually detect novel attacks, as well as variations of known attacks.

5.2.1 Omitting Entire Attack Categories. First, we analyze the recall when entire attack categories are omitted, which we illustrate as the recall value heatmaps for the three classifiers in Figures 4(a)–4(c). We further include the baseline of training on all attack categories as well as the changed recall when removing a single attack category during training. Each row corresponds to an experiment where the category, denoted on the y-axis by its ID (*cf.* Table 1), was omitted. Additionally, the first row ("none") denotes the baseline where the whole train set was used. Each column again corresponds to an attack category, but now with the first column ("benign") denoting benign traffic. Each field of the heatmap shows the recall averaged over 5 folds. The value of 93.7 % in the fourth row and the fifth column of, *e.g.*, Figure 4(b) thus indicates that 93.7 % of network packets in the test set belonging to Attack Category 4 were detected by the BLSTM classifier when it was not trained on Category 3.

Throughout Figures 4(a)–4(c), we observe drops in the recall values primarily on the main diagonals of the heatmaps falling from, *e.g.*, 90.3 % to just 6.3 % in Category 3 using the BLSTM classifier.



(a) RF classifier

(b) BLSTM classifier

(c) SVM classifier

(d) RF classifier

(e) BLSTM classifier

(f) SVM classifier

Figure 4: We applied our evaluation methodology (Section 4.2.2) to RF, BLSTM and SVM classifiers from literature on two aggregation levels: once by omitting attack categories ((a)–(c)) and once by omitting individual attacks ((d)–(f)). The achieved recall [%] is visualized as a heatmap where each row corresponds to one omitted attack or category, and each column matches the attack or category for which the recall is measured, respectively. The results illustrate a significant drop in recall for many cases, indicating that those attacks or categories are hardly detected by the used IIDS when not explicitly trained on.

These results are expected as those fields correspond to the recall of attacks that were omitted from training. However, the severity varies widely between categories and classifiers: Categories 3, 4, and 6 are most severely affected, with drops in recall between 45 and 94 percentage points, reaching a recall of as low as 2.2 % compared to the baseline of 97.8 % for Category 4 and the RF classifier. Except for the SVMs, all values in those categories fall below 30 % from a baseline of over 90 %. Categories 1, 2, and 7 undergo a smaller drop between 16 and 52 percentage points, reaching as low as 39.6 % for Category 2 in the SVM classifier. For the SVM classifier, Category 7 is an exception as its recall drops by almost 70 percentage points to just 28.4 %. Finally, Category 5 is an outlier as RF and BLSTM still recognize it with a recall of almost 100 %, even when not being trained on, showing virtually no performance degradation compared to the baseline. With the SVM, however, the performance decreases more significantly, achieving a recall of only 54.1 %.

The primary, but incomplete, explanation for the recall reduction is that (i) the classifiers have mostly learned signatures of attacks in contrast to the repetitive normal behavior of the ICS, and (ii) it depends on the amount of overlap between the omitted and other remaining categories. As such, Categories 1 and 2, both representing different forms of malicious response injection (*cf.* Section 5.1), contain multiple attacks that manipulate the reported pressure value in different ways, which is a plausible explanation for the classifiers to correlate both categories. However, this explanation would imply similar behavior across different classifiers.

While the observed drop in recall across the classifiers is similar in magnitude for most categories, there are some notable differences. In particular, a classifier's ability to retain a decent recall for Categories 1, 2, 5, and 7 seems to correlate with an inability to retain it for Categories 3, 4, and 6. While the SVM classifier fares worse on Categories 1, 2, 5, and 7, being consistently outperformed with at least 9 percentage points by the RF and BLSTM classifiers, it outperforms them in Categories 3, 4, and 6, where it reaches a higher recall by between 8 and 20 percentage points. Similarly, the BLSTM classifier is outperformed by RF on Categories 1, 2, 5, and 7 while outperforming it on Categories 3, 4, and 6. It seems that classifiers, which model known attacks accurately and thus achieve high recall on them (*e.g.*, RF-based classifiers), tend to specifically overfit those known attacks and are therefore less likely to detect different kinds of anomalies. This observation indicates the existence of a major difference between the classifiers' general or specific understandings of anomalous behavior. Most indicative of this phenomenon is the detection of attacks from Category 5 when it is not learned: While the RF- and BLSTM-based classifiers are perfectly capable of identifying this traffic as malicious, with recall values of 100 % and 99.9 %, respectively, the SVM classifier is only able to label about half (54.1 %) of the corresponding malicious traffic correctly.

Finally, we examine the recall observed aside from the main diagonal, which mostly details little variation. However, cases exist where the recall of a specific attack category increases as a different category is omitted, *e.g.*, the recall for Category 6 *increases* from 96.7 % to 97.3 % when Category 2 is omitted for the RF classifier. Contrary, there are cases where we notice the opposite observation, *e.g.*, as the recall for Category 6 *decreases* from 96.7 % to 95.0 % when Category 4 is omitted when training the same classifier. The size of those fluctuations seems to be dependent on the classifier: While the recall in Category 3 changes by at most 1.2 percentage points when omitting a different category during the training of the RF and SVM classifiers, it drops by nearly 10 percentage points when Category 2 is omitted for the BLSTM classifier. Overall, we can thus conclude that not training for a specific category in general only influences the detection rate of that specific category, with a mostly insignificant influence on the detection rate of related categories.

5.2.2 *Omitting Individual Attack Types.*

To investigate the classifiers' ability to generalize *within* categories, *i.e.*, across attacks that are allegedly much more similar, we also analyze the recall when specific attacks are omitted from training one by one. We include the corresponding heatmaps in Figures 4(d)–4(f). The labels on both axes correspond to the categories the individual attacks belong to.

Similar to omitting entire attack categories, we do not observe any significant adverse effects on other attack types when removing individual attack types from the train set, as the effects are again confined to the main diagonal. Merely for the BLSTM classifier, the attacks from Categories 2 and 3 exhibit a marginally visible effect outside the main diagonal. These effects indicate an existing, though minimal and insignificant, ability of the BLSTM classifier to abstract attack patterns across attack categories.

Focusing again on the main diagonal, we observe similar patterns across the different classifiers. For Category 1, the classifiers' ability to detect attacks is primarily unaffected. However, Attack 1.2, *i.e.*, the second attack from Category 1, is an exception, with recall values dropping significantly across all three classifiers. The dataset describes Attacks 1.1–1.3 as "Random Value Attacks" on the pressure measurement without further differentiation between those attacks [39]. In light of this description, our reported numbers raise the question to which extent Attack 1.2 differs from the others to warrant the observed recall drop. A manual investigation of the dataset revealed that Attack 1.2 sends pressure values mainly within the normal bounds of the system, *e.g.*, 7.52, while Attacks 1.1 and 1.3 send values clearly out of bounds, such as 0 or 6.9×10^{31}. Thus, we assume that the classifiers cannot detect those attacks within the normal operating bounds without explicit training.

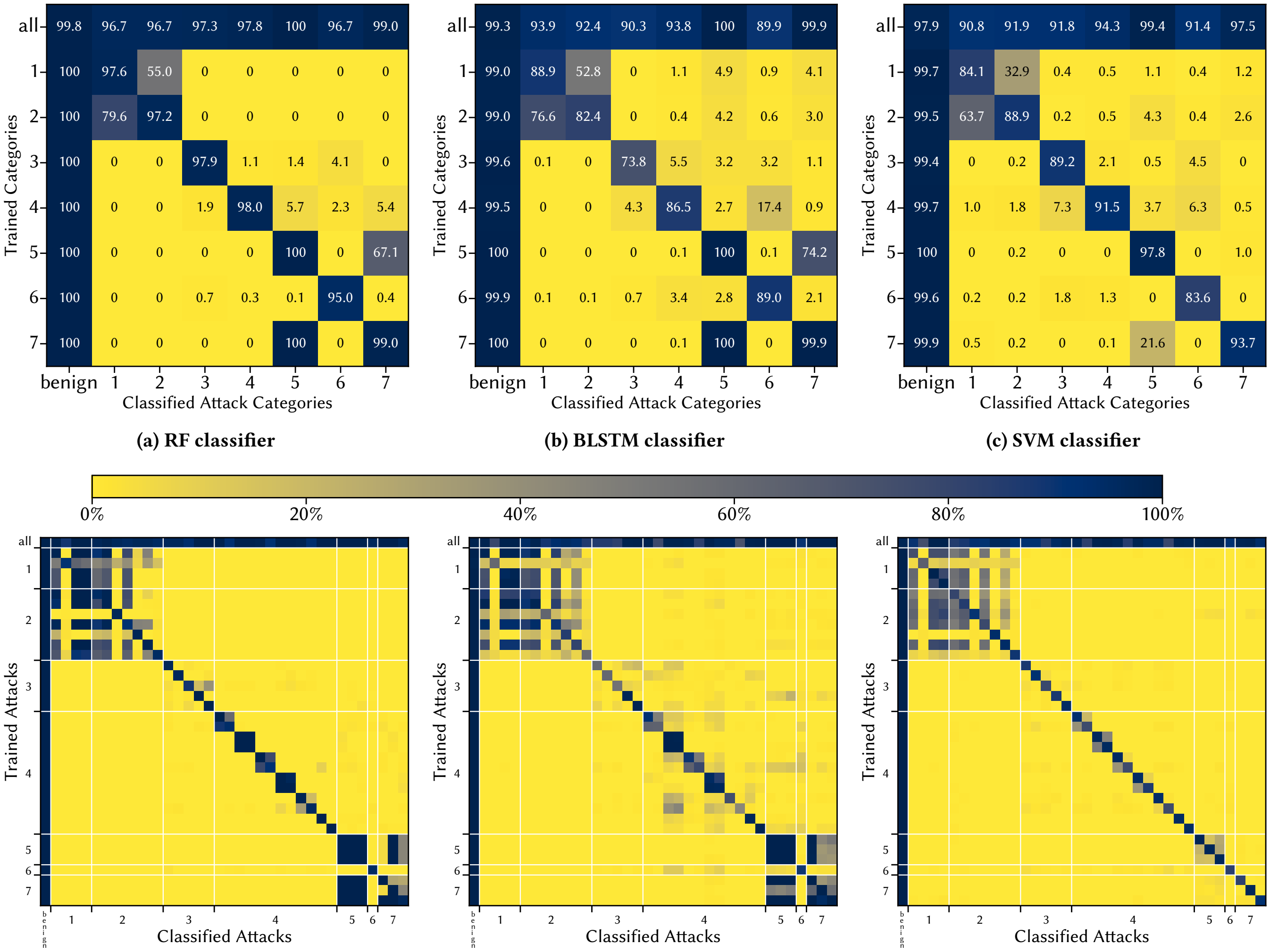
Similar patterns apply to Categories 2, 3, 4, and 6, *i.e.*, some unlearned attacks are detected as a variation of another attack independently of the classifier, while others are not identified as such despite originating from the same category. Yet, Category 7 shows major differences across our evaluated classifiers. While the RF classifier exhibits no drop in recall, the others (BLSTM and SVM) achieve a reduced recall for Attacks 7.2 and 7.3, motivating an in-depth analysis of the attacks in Category 7. Attack 7.1 (Device Scan Attack) generates packets whose "address" field deviates from the regular address 4, *e.g.*, setting it to 0 or 9, to scan for the presence of those addresses in the network. Attacks 7.2 and 7.3, in contrast, do not change the address field but introduce "novel" Modbus function codes instead. All classifiers are able to detect the abnormal address, while only the RF classifier is able to detect the malicious function codes without prior training. This detail shows that classifiers are able to learn the system's normal behavior to some extent and thus detect anomalous deviations from it. However, this ability is restricted to a specific scenario, and real-world deployments cannot generally trust ML-based IIDSs to detect previously unseen attacks.

Overall, within Categories 1, 2, 4, 5, and 7, most attacks show only minor drops in the recall value, contrary to Categories 3 and 6, with more significant drops across all attacks. These numbers match our observations when omitting entire categories except for Category 4, which resulted in a very low recall when omitted completely. It seems that removing some attacks from this category has no significant, immediate effect, while completely removing them drops the recall significantly. This aspect indicates some form of generalization within Category 4, but not beyond it and to other categories.

Takeaways. Our analysis highlights the limited generalization capabilities of the analyzed classifiers *within* and *across* attack categories, as many attacks cannot be detected reliably without explicit training. Further, we observe significant variances in recall rates of previously unseen attacks. These variances can be (partly) attributed to the similarities between related attacks, but what is interpreted as similar differs between classifiers and does not necessarily match human-created attack categorizations, *i.e.*, existing dataset labels. Consequently, despite the predictable nature of ICSs, the analyzed ML-based IIDSs were unable to detect truly novel attacks reliably and thus fail in serving as reliable anomaly detectors.

5.3 Understanding (Dis-)Similarities of Attacks

After investigating and validating that ML-based IIDSs hardly generalize to unseen attacks, we now intend to provide a better understanding of the relations between different attacks. To this end, we



(a) RF classifier

(b) BLSTM classifier

(c) SVM classifier

(d) RF classifier

(e) BLSTM classifier

(f) SVM classifier

Figure 5: We specifically trained the classifiers only for single attacks or single attack categories, respectively, using our evaluation methodology (Section 4.2.3). The resulting recall [%] is visualized as a heatmap similar to Figure 4 and underlines that the examined classifiers are hardly able to detect attacks apart from those attacks they were explicitly trained on.

explicitly train the classifiers on a single attack (category) only. By doing so, we can examine the classifiers' ability to detect different attacks beyond the one presented during training. For the same reasons as in Section 5.2, we again focus on recall as the primary performance metric and visualize our measured results in Figure 5.

5.3.1 Training Specific Attack Categories. We begin by training on individual attack categories. We provide the corresponding results in Figures 5(a)–5(c). Contrary to the previous experiment, we expect the recall values on the main diagonal to be especially high as those fields correspond to explicitly trained attacks. However, assuming that the classifiers embody anomaly detectors, we would expect noticeable recall apart from the main diagonal. This observation would indicate some form of general understanding of malicious activities as well as the ability to detect variations of known attacks.

Evidently, detecting novel attack patterns and anomalies between categories is limited to very few cases, with the plots turning out mostly yellow, constituting low recall. Nevertheless, we observe two main instances of substantial positive recall apart from the main diagonal. First, we notice an interrelation between Categories 1 and 2: If one of the two categories is trained, the classifiers also achieve a high recall in the other category, *e.g.*, when training the BLSTM classifier using attacks from Category 2, attacks from Category 1 are detected with a high recall of 76.6 %. As discussed in Section 5.2, those categories contain attacks that manipulate the pressure reading, which could be the cause of this interrelation.

In addition, Categories 5 and 7 are connected: If one of the two categories is trained for, the recall in the other category is generally high (up to 100 %). According to the dataset's description, one similarity between the six attacks in those categories is that they all employ Modbus function codes that are otherwise not used. The diagnostic function 0x08, for example, is only used in the attacks of Category 5 and the "Device Scan Attack" in Category 7 [39]. Thus, we assume that the classifiers learn that only specific function codes are used during normal operation when training on one of the two categories, and, therefore also detect attacks in the respective other category. Here, the behavior of the SVM classifier is in stark contrast to the others', as it does not derive the same strong relationship between both categories. Thus, the SVM seems to differentiate benign and malicious traffic on different properties.

Moreover, we observe further differences between the classifiers. For all but the RF classifier, the recall values of the trained attack categories drop in comparison to the baseline when all attacks are trained, indicating that these classifiers benefit from a general understanding of malicious activities. The recall values offside the diagonal further underline this observation as the classifiers are partly able to correctly detect attacks they have not been trained on (predominantly noticeable in the yellow regions of Figures 5(b) and 5(c)). In contrast, the RF classifier achieves lower recall values outside of the main points of interest but is, therefore, able to detect attacks more reliably when explicitly trained for them. These observations suggest that the RF classifier realizes a much more targeted training of the attacks it knows from the train set.

Finally, another notable observation is the increase in recall for benign traffic when training on single attacks. For the RF classifier, the recall reaches 100 % when single attack categories are trained compared to 99.8 % before. A similar improvement can be seen for the SVM classifier. As the amount of benign traffic in a real-world deployment is expected to be orders of magnitude larger than malicious traffic, this improvement implies a significant decrease in false-positive alarms. In contrast, both recall and precision, which we calculated additionally, drop when using the BLSTM classifier and training the IIDS exclusively on Categories 1 and 2.

5.3.2 Training Individual Attacks Exclusively. In the following, we take a look at the attack detection when the classifier is trained on individual attacks only (Figures 5(d)–5(f)). The main observation is that the classifiers generalize within and across categories to some extent, and the SVM classifier generalizes less than RF and BLSTM. The distinct patterns that we observe do, however, highlight the importance of digging deeper into the classifiers' results to get an accurate impression of their capabilities in a real deployment.

When focusing on Categories 1 and 2, we observe a generalization *within* and *across* the categories for all three classifiers, *e.g.*, most attacks are detected with a significant recall if the classifiers are trained on Attack 2.4. Thus, the classifiers are able to generalize well enough to detect the different attacks in those categories, even with such a reduced sample set. Noticeable exceptions are Attacks 1.2, 2.3, 2.5 for SVM only, and 2.7, which are hardly generalized when training on the other attacks from this attack category.

The pattern for Attack 1.2 is particularly interesting, as it is not correctly detected when training on other attacks in its category. In contrast, when training the RF classifier on this category, it also detects other attacks in both categories with a significant recall. As discussed in Section 5.2, this attack mainly generates malicious pressure readings within the normal operating bounds of the system, while Attacks 1.1 and 1.3 send out-of-bounds readings. A plausible explanation for this behavior is that abstracting to detect malicious readings *within* bounds enabled the classifier to also detect malicious readings *outside* the normal bounds, but not *vice versa*. We observe similar generalization patterns between and across attacks of Categories 5 and 7 for RF and BLSTM. These results again underline that the SVM classifier seems to base its model on different characteristics than the others, at least for this subset of attacks.

Finally, we discuss the pattern that emerges in Category 4 where pairs of two neighboring attacks display an interrelation, *e.g.*, Attacks 4.1 and 4.2, while interrelations to any other attacks in the category are mostly missing. Attacks 4.1 and 4.2 both cover "Setpoint Attacks" [39] and differently manipulate the same communicated value. Due to the focus on one value, the classifiers can more easily correlate both of these attacks, even if only one of the attacks has been known before. Similar patterns are also found when looking at the other interrelated attacks from Category 4, *e.g.*, Attacks 4.4 and 4.5. Thus, while attacks on the same parameter can be interrelated by the classifiers to some extent, manipulations of other parameters cannot be handled in the same way.

These results further strengthen the assumptions that ML-based IIDSs do not base their detections on abstracted process knowledge but rather on pre-trained signatures of attacks. Within Category 3, we virtually do not notice any generalization. Together with the limited generalization in Category 4, where only pairs of attacks are interrelated, we again conclude that human-made categorizations of the dataset, while sensible to humans, are of limited relevance.

Takeaways. Our second experiment shows that cross-learning *within* categories is limited to a few cases, particularly those patterns where very similar attacks manipulate the same process values. Thus, ML-based IIDSs, despite contrary promises, only achieve limited generalizability and act much more like signature-based IIDSs. We further observe major differences between the classifiers' abilities that warrant such in-depth analyses. Otherwise, we cannot truly understand which attacks an IIDS is able to detect reliably.

5.4 Implications for IIDS Generalizability

As the last part of our evaluation, we want to take a holistic look at what can be concluded from the detection capabilities of our examined ML-based IIDSs when combining both of our previous experiments. We observed a distinct connection between Attack Categories 1, 2, 5, and 7 that performed well when being omitted (*cf.* Section 5.2) and the interrelations between Categories 1 and 2 as well as 5 and 7 when being trained individually (*cf.* Section 5.3). In the case of RF, we observe a recall of 55.0 % in Category 2 when trained on Category 1, which matches closely with the 55.8 % when Category 2 is omitted. Furthermore, we can make identical observations for Categories 5 and 7 across all evaluated classifiers.

A similar picture emerges when looking at the fine-grained results, i.e., individual attacks: Those attacks that showed interrelations when classifiers were trained exclusively on them, *e.g.*, Attack 4.2, show decent recall even when being omitted. Meanwhile, those attacks that demonstrate hardly any interrelations, *e.g.*, Attack 4.11, cannot be detected without training for them explicitly. We also observe that the same attacks are outliers in both experiments, *i.e.*, Attacks 1.2, 2.3, 2.5 for SVM, and 2.7. When omitted, those attacks faced a significant drop in recall while other attacks in their category fared much better. Simultaneously, when training exclusively for those attacks, other attacks were rarely detected.

To conclude, it seems that the observed ability to generalize when omitting attacks can be mainly explained by the interrelations found when training on single attacks. This finding also puts the observations from Figures 4(d)–4(f), which hinted at slight generalizability, at least within categories, into a better perspective. Overall, we emphasize that the evaluated ML-based IIDSs do not actually learn normal system behavior but rather directly learn signatures of the attacks that they have been trained on. Nonetheless, we believe that ML-based IIDSs can constitute an effective and reliable defense mechanism for ICSs. However, they are in need of additional research and analyses to foster an understanding of their actual capabilities and to allow for accurate assessments and understandings of their benefits outside of (artificial) lab environments.

6 A FALSE SENSE OF SECURITY

Applying our evaluation methodology to quantify the generalizability of ML-based IIDSs to unseen attacks (*cf.* Section 4), we find that all three of our analyzed classifiers are largely unable to successfully detect unknown attacks (*cf.* Section 5), despite scoring high in widely-used performance metrics. Thus, the results reported by related work can lead to a false sense of security for practitioners.

While we observe cases in which unknown attacks are detected, they mostly result from an overlap of specific attack patterns with trained attacks, rather than the classifiers being able to abstract from known attacks. Thus, overall, our results support concerns and made claims formulated in literature that ML-based IDSs can only detect variations of known attacks (*cf.* Section 3). Thereby, we confirm that our considered classifiers only learn signatures of attacks instead of realizing proper anomaly detection. Consequently, they fail to reap the benefits anomaly-based detection can provide, highlighting that further attention concerning this issue is crucial.

We further observed that the human-made categorization of "similar" attacks (within ICS datasets) does not constitute a constructive approach as classifiers tend to source relevant information for the classification from other characteristics than humans. As a result, research must reconsider its understanding of similarities within the scope of IIDSs. Otherwise, further research on the generalizability of attack categories could be unnecessarily impaired.

Additionally, our analysis shows that the ability to detect unknown attacks varies between the considered classifiers, highlighting the need to compare multiple classifiers and approaches regarding their abilities in real-world deployments. The currently used (traditional) evaluation methodology, mainly focusing on recall, precision, or F_1-score metrics on a randomly chosen test set, is insufficient when targeting industrial settings with the need for anomaly detection. Thus, it is incapable of providing researchers and practitioners with a realistic understanding of the capabilities of an IIDS (particularly w.r.t. the protection against novel (unseen) attacks), which is technically the primary goal of any evaluation.

This situation limits the assessment and comparability of different approaches and hinders further advances in this field. Our proposed methodology addresses this issue by explicitly testing the system on selectively filtered datasets, enabling an in-depth analysis and explanation of the results. Still, conducting such detailed analyses requires access to a dataset with explicit labeling of (similar) attacks and enough repetitions or observations of each of them. Unfortunately, the availability of suitable ICS datasets for IDS use is limited [12, 16, 42].

Given that such analyses are lengthy and require detailed manual analyses of the results, the used dataset, and the aggregated, achieved performance, developing real-world-practical IIDSs is far from trivial. To at least improve the comparability of different approaches, we argue that developing a precise, comprehensible metric (*e.g.*, based on our evaluation methodology) should be a primary concern of future work. With such a metric at hand, research can then properly compare the IIDSs' abilities to detect unknown attacks, *i.e.*, to truly perform anomaly detection. Regardless, even with corresponding advances in the area of generalizability, the real-world feasibility of ML-based IIDSs is still in its infancy as significant deployment challenges, such as model tuning, sampling rates, and operational changes of the monitored ICSs, await [2].

7 CONCLUSION

The convergence of ICSs with the Internet leads to an increasing number of cyberattacks against such systems [43, 48]. To detect and prevent these attacks, anomaly-based intrusion detection is especially interesting as its detection rate benefits from repetitive communication patterns that frequently occur in industrial settings. While research focuses on the usage of machine learning enabling industrial IDSs to automatically determine what constitutes benign and malicious behavior, it still remains unclear whether these IDSs have any ability to detect novel attacks as they are typically trained not only on benign behavior but also on attacks [28]. Notably, promised detection rates of up to 99 % are reached by training the IDSs on attack types that are used for training *and* evaluation.

In this paper, we showed that these standard evaluation methods disguise the missing ability of ML-based IIDSs to detect formerly unseen attacks. More specifically, we proposed a methodology to analyze the ability of ML-based IIDSs to detect novel forms of attacks and applied it to three IIDSs [35]. We discovered that these IIDSs are widely unable to detect unseen attacks and find detection rates dropping to between 3.2 % and 14.7 % for specific unseen attack types. Furthermore, we proved that the ML-based IIDSs mainly learn specific attack signatures instead of process-specific properties. Hence, in scenarios where the training data does not cover all possible attacks, the IIDSs can only detect types of attacks that are known beforehand and fail to generalize to new attacks.

We suggest that our methodology should be performed on more IIDSs to ensure comparability w.r.t. the achieved level of generalization and to prevent the manifestation of a false sense of security based on the good performance numbers achieved with state-of-the-art evaluation methods.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Shingo Abe, Mariko Fujimoto, Shinichi Horata et al. 2016. Security threats of Internet-reachable ICS. In *SICE*.

[2] Chuadhry Mujeeb Ahmed, M. R. Gauthama Raman, and Aditya P. Mathur. 2020. Challenges in Machine Learning Based Approaches for Real-Time Anomaly Detection in Industrial Control Systems. In *ACM CPSS*.

[3] Johan Åkerberg, Mikael Gidlund, and Mats Björkman. 2011. Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation. In *IEEE INDIN*.

[4] Abdulrahman Al-Abassi, Hadis Karimipour, Ali Dehghantanha et al. 2020. An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System. *IEEE Access* 8.

[5] Magnus Almgren, Wissam Aoudi, Robert Gustafsson et al. 2018. The Nuts and Bolts of Deploying Process-Level IDS in Industrial Control Systems. In *ICSS*.

[6] Victor M. Alvarez. 2008. YARA - The pattern matching swiss knife for malware researchers. https://virustotal.github.io/yara/.

[7] Eirini Anthi, Lowri Williams, Pete Burnap et al. 2021. A three-tiered intrusion detection system for industrial control systems. *J. Cybersecur.* 7, 1.

[8] Simon D. Duque Anton, Sapna Sinha, and Hans Dieter Schotten. 2019. Anomaly-based Intrusion Detection in Industrial Data with SVM and Random Forests. In *SoftCOM*.

[9] Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. 2018. Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. In *ACM CCS*.

[10] Pallavi Arora, Baljeet Kaur, and Marcio Andrey Teixeira. 2021. Evaluation of Machine Learning Algorithms Used on Attacks Detection in Industrial Control Systems. *J. Inst. Eng. India Ser. B* 102, 3.

[11] Daniel Arp, Erwin Quiring, Feargus Pendlebury et al. 2022. Dos and Don'ts of Machine Learning in Computer Security. In *USENIX SEC*.

[12] Deval Bhamare, Maede Zolanvari, Aiman Erbad et al. 2020. Cybersecurity for industrial control systems: A survey. *Comput. Secur.* 89.

[13] Yuqi Chen, Christopher M. Poskitt, and Jun Sun. 2018. Learning from Mutants: Using Code Mutation to Learn and Monitor Invariants of a Cyber-Physical System. In *IEEE SP*.

[14] Ankang Chu, Yingxu Lai, and Jing Liu. 2019. Industrial Control Intrusion Detection Approach Based on Multiclassification GoogLeNet-LSTM Model. *Secur. Commun. Netw.* 2019.

[15] Riccardo Colelli, Filippo Magri, Stefano Panzieri et al. 2021. Anomaly-Based Intrusion Detection System for Cyber-Physical System Security. In *MED*.

[16] Gideon Creech and Jiankun Hu. 2013. Generation of a new IDS test dataset: Time to retire the KDD collection. In *IEEE WCNC*.

[17] Markus Dahlmanns, Johannes Lohmöller, Ina Berenice Fink et al. 2020. Easing the Conscience with OPC UA: An Internet-Wide Study on Insecure Deployments. In *ACM IMC*.

[18] Markus Dahlmanns, Johannes Lohmöller, Jan Pennekamp et al. 2022. Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things. In *ACM ASIACCS*.

[19] Alessandro Erba and Nils Ole Tippenhauer. 2020. No Need to Know Physics: Resilience of Process-based Model-free Anomaly Detection for Industrial Control Systems. arXiv:2012.03586.

[20] Cheng Feng, Tingting Li, and Deeph Chana. 2017. Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks. In *IEEE/IFIP DSN*.

[21] Cheng Feng, Tingting Li, Zhanxing Zhu et al. 2017. A Deep Learning-based Framework for Conducting Stealthy Attacks in Industrial Control Systems. arXiv:1709.06397.

[22] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur et al. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. *NDSS*.

[23] Jonathan Goh, Sridhar Adepu, Marcus Tan et al. 2017. Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In *IEEE HASE*.

[24] Haibo He and Jun Yan. 2016. Cyber-physical attacks and defences in the smart grid: a survey. *IET Cyber-Phys. Syst.: Theory Appl.* 1, 1.

[25] Jens Hiller, Martin Henze, Martin Serror et al. 2018. Secure Low Latency Communication for Constrained Industrial IoT Scenarios. In *IEEE LCN*.

[26] Hajar Homayouni, Sudipto Ghosh, Indrakshi Ray et al. 2020. An Autocorrelation-based LSTM-Autoencoder for Anomaly Detection on Time-Series Data. In *IEEE Big Data*.

[27] Yan Hu, An Yang, Hong Li et al. 2018. A survey of intrusion detection on industrial control systems. *Int. J. Distrib. Sens. Netw.* 14, 8.

[28] Khurum Nazir Junejo and Jonathan Goh. 2016. Behaviour-Based Attack Detection and Classification in Cyber Physical Systems Using Machine Learning. In *ACM CPSS*.

[29] Veeramreddy Jyothsna, V. V. Rama Prasad, and Koneti Munivara Prasad. 2011. A Review of Anomaly based Intrusion Detection Systems. *Int. J. Comput. Appl.* 28, 7.

[30] Aleksei Kharitonov and Axel Zimmermann. 2019. Intrusion Detection Using Growing Hierarchical Self-Organizing Maps and Comparison with Other Intrusion Detection Techniques. In *CPSS*.

[31] Ansam Khraisat, Iqbal Gondal, Peter Vamplew et al. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur.* 2, 1.

[32] Tim Krause, Raphael Ernst, Benedikt Klaer et al. 2021. Cybersecurity in Power Grids: Challenges and Opportunities. *Sensors* 21, 18.

[33] Ajit Kumar, Neetesh Saxena, and Bong Jun Choi. 2021. Machine Learning Algorithm for Detection of False Data Injection Attack in Power System. In *ICOIN*.

[34] Qin Lin, Sridha Adepu, Sicco Verwer et al. 2018. TABOR: A Graphical Model-based Approach for Anomaly Detection in Industrial Control Systems. In *ACM ASIACCS*.

[35] Rocio Lopez Perez, Florian Adamsky, Ridha Soua et al. 2018. Machine Learning for Reliable Network Attack Detection in SCADA Systems. In *IEEE TrustCom*.

[36] Yuan Luo, Ya Xiao, Long Cheng et al. 2022. Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities. *ACM Comput. Surv.* 54, 5.

[37] Stephen McLaughlin, Charalambos Konstantinou, Xueyang Wang et al. 2016. The Cybersecurity Landscape in Industrial Control Systems. *Proc. IEEE* 104, 5.

[38] Sohrab Mokhtari, Alireza Abbaspour, Kang K. Yen et al. 2021. A Machine Learning Approach for Anomaly Detection in Industrial Control Systems Based on Measurement Data. *Electronics* 10, 4.

[39] Thomas H. Morris, Zach Thornton, and Ian Turnipseed. 2015. Industrial control system simulation and data logging for intrusion detection system research. In *SCSS*.

[40] Open Information Security Foundation (OISF). 2021. Suricata. https://suricata.io/.

[41] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney et al. 2019. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. In *USENIX SEC*.

[42] Jan Pennekamp, Erik Buchholz, Markus Dahlmanns et al. 2021. Collaboration is not Evil: A Systematic Look at Security Research for Industrial Use. In *LASER*.

[43] Jan Pennekamp, René Glebke, Martin Henze et al. 2019. Towards an Infrastructure Enabling the Internet of Production. In *IEEE ICPS*.

[44] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, George Efstathopoulos et al. 2020. DIDEROT: An Intrusion Detection and Prevention System for DNP3-Based SCADA Systems. In *ARES*.

[45] M. R. Gauthama Raman, Chuadhry Mujeeb Ahmed, and Aditya Mathur. 2021. Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. *Cybersecur.* 4.

[46] M. R. Gauthama Raman, Nivethitha Somu, and Aditya P. Mathur. 2019. Anomaly Detection in Critical Infrastructure Using Probabilistic Neural Network. In *ATIS*.

[47] Payam Refaeilzadeh, Lei Tang, and Huan Liu. 2016. *Cross-Validation.* Springer.

[48] Martin Serror, Sacha Hack, Martin Henze et al. 2021. Challenges and Opportunities in Securing the Industrial Internet of Things. *IEEE Trans. Ind. Informat.* 17, 5.

[49] Robin Sommer. 2003. Bro: An open source network intrusion detection system. In *Security, E-learning, E-Services, 17. DFN-Arbeitstagung über Kommunikationsnetze.* Bonn.

[50] Robin Sommer and Vern Paxson. 2010. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *IEEE SP*.

[51] Mahbod Tavallaee, Natalia Stakhanova, and Ali Akbar Ghorbani. 2010. Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods. *IEEE Trans. Syst., Man, Cybern. C* 40, 5.

[52] The Zeek Project. 2019. The Zeek Network Security Monitor. https://zeek.org/.

[53] Rafael Uetz, Christian Hemminghaus, Louis Hackländer et al. 2021. Reproducible and Adaptable Log Data Generation for Sound Cybersecurity Experiments. In *ACSAC*.

[54] Muhammad Azmi Umer, Chuadhry Mujeeb Ahmed, Muhammad Taha Jilani et al. 2021. Attack Rules: An Adversarial Approach to Generate Attacks for Industrial Control Systems using Machine Learning. In *CPSIoTSec*.

[55] Muhammad Azmi Umer, Khurum Nazir Junejo, Muhammad Taha Jilani et al. 2022. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *Int. J. Crit. Infrastruct. Prot.*

[56] David I. Urbina, Jairo A. Giraldo, Alvaro A. Cardenas et al. 2016. Limiting the Impact of Stealthy Attacks on Industrial Control Systems. In *ACM CCS*.

[57] Jie Wang. 2009. *The Art of Intrusion Detection.* Springer.

[58] Herman Wijaya, Maurício Aniche, and Aditya Mathur. 2020. Domain-Based Fuzzing for Supervised Learning of Anomaly Detection in Cyber-Physical Systems. In *IEEE/ACM ICSEW*.

[59] Konrad Wolsing, Eric Wagner, and Martin Henze. 2020. Poster: Facilitating Protocol-independent Industrial Intrusion Detection Systems. In *ACM CCS*.

[60] Konrad Wolsing, Eric Wagner, Antoine Saillard et al. 2021. IPAL: Breaking up Silos of Protocol-dependent and Domain-specific Industrial Intrusion Detection Systems. arXiv:2111.03438.

[61] Chunjie Zhou, Shuang Huang, Naixue Xiong et al. 2015. Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation. *IEEE Trans. Syst., Man, Cybern., Syst.* 45, 10.