One IDS is not Enough! Exploring Ensemble Learning for Industrial Intrusion Detection

✉ Konrad Wolsing[1,2], Dominik Kus[2], Eric Wagner[1,2],
Jan Pennekamp[2], Klaus Wehrle[2], and Martin Henze[3,1]

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

Abstract. Industrial Intrusion Detection Systems (IIDSs) play a critical role in safeguarding Industrial Control Systems (ICSs) against targeted cyberattacks. Unsupervised anomaly detectors, capable of learning the expected behavior of physical processes, have proven effective in detecting even novel cyberattacks. While offering decent attack detection, these systems, however, still suffer from too many False-Positive Alarms (FPAs) that operators need to investigate, eventually leading to alarm fatigue. To address this issue, in this paper, we challenge the notion of relying on a single IIDS and explore the benefits of combining multiple IIDSs. To this end, we examine the concept of ensemble learning, where a collection of classifiers (IIDSs in our case) are combined to optimize attack detection and reduce FPAs. While training ensembles for supervised classifiers is relatively straightforward, retaining the unsupervised nature of IIDSs proves challenging. In that regard, novel time-aware ensemble methods that incorporate temporal correlations between alerts and transfer-learning to best utilize the scarce training data constitute viable solutions. By combining diverse IIDSs, the detection performance can be improved beyond the individual approaches with close to no FPAs, resulting in a promising path for strengthening ICS cybersecurity.

Keywords: Intrusion Detection · Ensemble Learning · ICS

1 Introduction

Industrial Intrusion Detection Systems (IIDSs) represent a fundamental building block to defend Industrial Control Systems (ICSs) against constantly emerging and highly targeted cyberattacks [4]. Besides offering a cost-effective security upgrade and serving as a second line of defense once preventive measures have been breached, IIDSs are also suitable for deployment in legacy ICSs with poor security where preventive measures, e.g., encryption and authentication, are hard to retrofit. To this end, research proposed a plethora of algorithms that automatically raise alerts for suspected malicious activities in ICSs [8,11,18,33,47,50].

Since ICS deployments are rather unique and adversaries in that domain are particularly sophisticated, ICSs often face zero-day attacks [4,33]. This threat significantly hampers the effectiveness of *supervised* detection approaches since their reliance on samples of (known) attacks for training runs the risk of detecting only these trained attacks or slight variants at best [2,24]. In contrast, *unsupervised* anomaly detectors [8], which can learn the expected behavior of repetitive physical processes by means of, e.g., machine learning, have proven successful across many scientific evaluations [5,14,22,29,46]. Their main benefit lies in training on benign-only data which can easily be recorded during regular operation of an ICS as well as their ability to detect novel cyberattacks.

Despite unsupervised IIDSs being capable of detecting even zero-day attacks, they come at the notorious risk of emitting False-Positive Alarms (FPAs) [2,41]. In practice, every alert has to undergo analysis by an operator to decide whether it is reasonable to interrupt operation in case of a suspected cyberattack. Consequently, as stated by Etalle et al. [13], FPAs significantly contribute to an IIDS's total cost of ownership. Moreover, since cyberattacks against ICSs' physical processes are still rare [4], minimizing the number of FPAs is equally as important as detecting attacks because they increase the risk of *alarm fatigue* where operators start ignoring the IIDS over time, such that actual attacks can slip through.

To meet these high detection standards, most scientific proposals aim to find yet another single-best, *monolithic* IIDS that outperforms existing work. But, studies of unsupervised IIDSs reveal that no approach currently detects all attacks contained in prominent datasets [46] or documented in literature [12]. Fortunately, real-world deployments are not restricted to integrating just the best-performing IIDS and may instead choose from the several available approaches to play off their advantages and disadvantages against each other. These observations raise the question whether one IIDS is enough to provide strong ICS security or if a combination of multiple IIDSs can join forces to optimize detection performance beyond what each individual approach can achieve.

The problem of combining a collection of classifiers (e.g., IIDSs) into a single system is generally known as ensemble learning [38,49,51]. However, while the training of an ensemble is relatively straightforward for supervised IIDSs, where simply another round of training suffices [16,25,27,34], retaining the unsupervised nature, i.e., requiring no attack knowledge during training, as is desirable for usage in ICSs, proves challenging. Consequently, we explore how *ensembles of unsupervised IIDSs* can be built and to which extent they are superior to monolithic deployments, e.g., by detecting more attacks in total or reducing FPA.

Contributions. To better understand the potential of ensemble learning for industrial intrusion detection, we make the following contributions:

- We uncover weaknesses in monolithic IIDS deployments and identify three challenges on the path to realize unsupervised IIDS ensembles (Sec. 3).

- Exploring the potential of IIDS ensembles, we reveal an enormous theoretical potential that, however, proves hard to transfer into practice (Sec. 4).
- Digging deeper, we find one reason inhibiting this potential in the ensembles' insufficiency to regard temporal information and prove that a novel class of time-aware ensembles can drastically improve this situation (Sec. 5).
- Lastly, we consider transfer-learning, i.e., training an ensemble in a different ICS under attack and transferring the model to the target ICS, as a promising method to tackle the lack of attack data during ensemble training (Sec. 6).

Availability Statement. We open-source the ensembles' implementations [15] and publish the base-IIDSs' alerts and experiments as public artifacts [45].

2 Industrial Intrusion Detection and Ensemble Learning

Before assessing whether the fusion of unsupervised IIDSs into an ensemble is feasible, we provide an introduction to intrusion detection methods in the ICS domain (Sec. 2.1), datasets, and metrics (Sec. 2.2) that we leverage in our evaluations. We then lay out the basic principles of ensemble learning (Sec. 2.3).

2.1 Unsupervised Industrial Intrusion Detection

The transition to Internet-connected ICSs establishes new paths for cyberattacks [4], where attackers can, e.g., manipulate sensors or actuators to cause harm [42,50]. Since ICSs are usually operated for decades without major downtime, retrofittable security measures are desirable [47]. To this end, IIDSs closely monitoring the physical behavior of an ICS can report even subtle deviations to an operator. As ICSs feature regular and repetitive patterns, IIDSs make great use of temporal coherences in this data for attack detection [5,22,29,46].

The methodologies followed in research to implement intrusion detection can broadly be classified into supervised and unsupervised IIDSs [2]. Supervised approaches rely on benign *and* malicious training data, which in return yields high detection rates [2]. However, not only is obtaining attack samples for the target ICS difficult [6], but supervised IIDSs are also prone to merely detecting those attacks presented to them during training [24] and thus fail to succeed on zero-day attacks. Unsupervised IIDSs, in contrast, require only benign training data and can thus indicate any deviation from the learned normality model [8]. Despite being prone to emit more FPAs, their promise to detect any attack violating normal behavior regardless of attack samples resembles a key feature for ICS.

Looking closer at how unsupervised IIDSs function, we present eight approaches from related work. The first approach (*Invariant*) [14] proposes a method to derive invariants that must be fulfilled by the ICS at all times, e.g., if the water level of a tank is rising, its inlet valve must be open. In contrast, *PASAD* [5] leverages a singular spectrum analysis to distinguish deterministic ICS behavior from non-determinism induced by attacks. *Seq2SeqNN* [22], which trains a neural network, and *TABOR* [29], based on timed automata, both alert deviations from the model's predicted ICS behavior. Lastly, Wolsing et al. [46] presented four lightweight approaches, i.e., minimum and maximum range checks (*MinMax*), the detection of steep in- and declines (*Gradient*) as well as frozen physical values (*Steadytime*), and unnatural distributions of process data (*Histogram*). These eight IIDSs build the foundation for our subsequent analyses.

2.2 Evaluating Industrial Intrusion Detection Systems

In the ICS domain, researchers leverage specialized datasets [10,21] to prove their IIDSs' capabilities. The eight IIDSs considered in this publication have originally been designed for and evaluated on the SWaT [19] or WADI [1] datasets, which are predominantly used in the IIDS domain [10]. Those datasets are specifically suited to evaluate unsupervised IIDSs as they ship with a large training part of benign-only data and another testing part containing several cyberattacks. In total, SWaT, which models a water treatment plant, contains 36 cyberattacks, and WADI, which represents a water distribution testbed, contains 14 attacks.

Besides datasets, metrics play an important role in judging an IIDS's performance. However, since attacks against ICSs and their effects can last for a certain time, point-based metrics (e.g., accuracy, precision, recall, or F1 score) are heavily skewed in this case as they score more points to longer attacks than (several) shorter ones [17,20]. To cope with these issues, newer metrics implementing time-aware variants of precision, recall, and F1 score were recently proposed, with the enhanced time-aware (eTa) metrics [20] emerging as a promising idea.

For these reasons, we refer to the time-aware metrics *eTaP* (precision), *eTaR* (recall), and *eTaF1* (F1) in our evaluations and, in addition, rely on the following two metrics from related work [29,46]: First, *FPAs* counts the number of continuous alerts that do not overlap with any attack. Second, detected scenarios (*Scen.*) enumerates how many continuous attacks from the dataset are detected. Together, these metrics promise accurate insights into the IIDSs' performance.

2.3 Ensemble Learning

Instead of proposing and evaluating new IIDSs, leveraging multiple IIDSs side-by-side can be another path to more secure ICSs. In this regard, ensemble learning represents a subfield of machine learning concerned with the process of training multiple classifiers, i.e., IIDSs, and fusing them into a single model to increase predictive performance [38,49,51] as proven successful in, e.g., the fields of computer vision [26] or biometric recognition [39]. The methods followed in the literature can be roughly divided into homogeneous and heterogeneous ensembles. Since homogeneous ensembles combine the same type of classifier, i.e., only Support Vector Machines (SVMs), they are rather suited to create monolithic IIDSs, as demonstrated before [9,23,30,31,35,48]. In contrast, heterogeneous ensembles combine diverging approaches. Considering the diversity of IIDSs proposed in the literature (cf. Sec. 2.1), our paper focuses on heterogeneous ensemble learning.

The general methodology for heterogeneous ensemble learning is visualized in Fig. 1. First, a set of *base-classifiers* is pre-trained on a training dataset. Then,

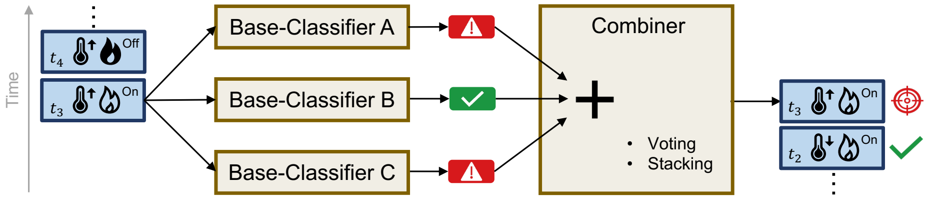

Fig. 1. With ensemble learning, a combiner joins the output of several base-classifiers into a single decision. In the case of learning-based ensembles, an additional training step on the outputs of the base-classifiers and the expected outcome may be necessary.

their outputs are fed into a *combiner* which fuses them into a single classification. More precisely, the input of the combiner is a judgment of each IIDS, whether it has identified an alert (1) or classifies the situation as benign (0) for the current data point. For a set of n IIDSs, the algorithm receives an alert vector $v \in \{0,1\}^n$. Hereby, the combiner can either be learning-based, i.e., *stacking* machine-learning classifiers trained over a dataset, or rule-based, e.g., weighted *voting* to combine binary outputs [32,51]. According to best practices [37,52], a learning-based combiner must be trained on out-of-sample data for the base-classifiers, which can be achieved, e.g., by splitting the dataset into separate train sets for base-classifiers and combiner. When measuring the combiner performance, the evaluation must be conducted on a set of previously unseen data.

Lastly, although ensembles promise to improve detection performance, they can hinder interpretability [40] or accountability [13], i.e., they complicate reconstructing why an alarm is emitted, which is important for attack mitigation.

3 From Monolithic IIDSs to IIDS Ensembles

Given the tremendous efforts invested in designing IIDSs (cf. Sec. 2.1), ICS operators must decide which conceptual approach they adopt by weighing all proposals based on their capabilities. However, this decision can be difficult as prior work raised the suspicion that no optimal IIDS exists that detects sufficiently many attacks [12,46]. To confirm this claimed insufficiency, we analyze eight IIDSs in Sec. 3.1. We then examine to which extent related work on ensemble learning can provide a solution (Sec. 3.2) and identify three unique challenges for *unsupervised* ensemble learning that inhibit its immediate adoption (Sec. 3.3).

3.1 Insufficiency of Monolithic Detectors

To study the capabilities of the eight IIDSs (cf. Sec. 2.1), we leverage open-source implementations of them [47] and evaluate them on the SWaT dataset[4]. Given the results from Tab. 1, we can confirm the IIDSs' insufficiency, as no single approach detects all 36 cyberattacks (cf. Scen.), yet 33 would be detectable

[4] The results for a second dataset (WADI) are compiled in Appx. A.

Table 1. These eight approaches from relevant literature to detect cyberattacks highlight that relying on a monolithic IIDS introduces risks, as none of them detects all attacks. Moreover, determining the "best" IIDS heavily depends on the chosen metric.

IIDS (Baseline)	eTaP	eTaR	eTaF1	Scen.	FPAs
Invariant [14]	54.7	29.8	38.6	30	217
PASAD [5]	16.0	4.9	7.5	16	14
Seq2SeqNN [22]	42.8	47.2	44.9	26	37
MinMax [46]	67.8	47.1	55.6	23	9
Gradient [46]	20.5	6.0	9.2	25	64
Steadytime [46]	81.6	30.1	44.0	14	4
Histogram [46]	70.9	23.2	34.9	13	0
TABOR [29]	49.1	18.9	27.3	19	28

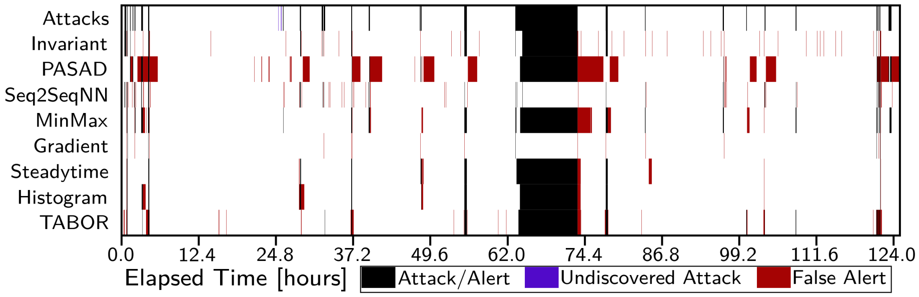

Fig. 2. The alerts emitted by IIDSs on the SWaT dataset exemplify the challenges operators face during investigations, as they each exhibit distinct alerting behavior.

in combination. Moreover, across the five metrics introduced in Sec. 2.2, no monolithic IIDS performs best in more than one of the metrics (cf. grey cells in Tab. 1), indicating that certain compromises towards one or another metric have to be made by operators selecting an IIDS for their ICS. More precisely, while the Invariant IIDS excels in detected scenarios, thus likely unveiling most cyberattacks, it fares badly concerning FPAs (217). On the contrary, Histogram detects the fewest cyberattacks but also yields no FPAs, resulting in reliable indications that can counteract the risk of alarm fatigue. Both extremes seem undesirable for deployment, and a trivial combination, e.g., with a logical OR, would enable great detection yet add up all the FPAs from both approaches.

Aside from detection performance, the alerts should be accountable to ICS operators to initiate appropriate countermeasures [13,40]. Yet, when visually analyzing the alerts (cf. Fig. 2), we observe how differently these IIDSs indicate attacks. E.g., MinMax produces nine FPAs occurring near the actual attacks, which is in stark contrast to Invariant spreading alarms even across broad regions of benign behavior. This qualitative difference is not expressable with current metrics. Contrarily, PASAD, and Histogram exhibit a different phenomenon of "overhanging" alerts (red) after the attacks, which complicates determining the actual range of an attack. Given these distinct behaviors, truly understanding an IIDS's alerts requires expert knowledge of the underlying detection mechanism.

Takeaway. We observed that IIDSs have complementary strengths and opposing weaknesses. Thus, relying on a single IIDS can yields suboptimal ICS security. Furthermore, it increases the burden for operators, who must select an IIDS that best fits their deployment, as a trivial ensemble, where all alerts are simply ORed, would better detect attacks yet likely suffer from an excessive multitude of FPAs. This disappointing situation motivates looking for ensembles that cleverly combine the advantages of multiple IIDSs (cf. Sec. 2.3) to (i) improve the detection capabilities and (ii) make alarms generally more comprehensible.

3.2 IIDS Ensemble Learning in Related Work

In the past, the idea of ensemble learning has already been used to fuse (smaller) detection concepts into a final IIDS [3,22,29,36]. As an example, TABOR [29] fuses three detection methods together. Likewise, Seq2SeqNN [22] trains one neural network for each ICS process stage (six for SWaT), and only a single model has to emit an alert. Al-Abassi et al. [3] use decision trees to stack the results of multiple neural networks. Lastly, Radoglou-Grammatikis et al. [36] combine two IIDSs, tasked to detect known and unknown attacks respectively, using a logical OR. While such approaches fall into the category of rule-based ensembles, they aim to establish a monolithic IIDS rather than complex ensembles.

Still, more complex ensemble learning has also been examined for IIDS applications: Kus et al. [25] analyzed the impact of several methods covering voting and stacking to join seven *supervised* IIDSs, such as Random Forests (RFs) or SVMs, but only achieved marginal improvements below 1 % in the F1 score. Likewise, Upadhyay et al. [44] combined six *supervised* IIDSs with majority voting achieving similar results. Gao et al. [16] combined two deep learning models using a stacked Multi-Layer Perceptron (MLP) with improvements in F1 score of only up to 0.41 %. Nguyen et al. [34] combined three classifiers using stacking with an MLP. They achieved an increase of 0.14 % in F1 score but had little room for improvement in the first place as the best base-classifier achieved an F1 score of 99.58 %. Li et al. [27] combined three classifiers using majority voting to increase accuracy by up to 1.62 %. Lastly, Balaji et al. [7], who experimented with Logistic Regression (LR) as ensemble method, conclude that they rather learned attack signatures and leave unsupervised learning to future work. However, to date, comparable research on unsupervised IIDS ensembles is still missing.

3.3 Challenges for Unsupervised IIDS Ensembles

In related work, ensemble learning managed to achieve only slight improvements (around +1 % in F1 score) in the IIDS context, likely due to small margins within the used base-IIDSs. In our baseline results from Tab. 1, however, we observed grossly different behaviors and IIDSs distinctly detecting attacks, indicating much greater potential for ensembles to work with.

Notably, most related work from the IIDS domain considers *supervised* ensembles [16,25,27,34]. This severe limitation introduces the risk of only detecting the trained attacks [24], effectively transforming unsupervised base-IIDSs back into a supervised IIDS. Moreover, given their reliance on attack data during training which is scarce in the ICS domain [6,10], the standard methodology used to train (and evaluate) such supervised ensembles is incompatible with *unsupervised* training, which is imperative in the ICS domain. To this end, we explain the three challenges (**C1**–**C3**) pertaining to the current methodology.

C1—Benign training only. Unsupervised IIDSs train exclusively on benign data, eliminating the need for attacks in the training set. This property enables them to detect zero-day attacks [41] and simplifies their training, as benign data can be recorded during normal ICS operation and is, therefore, abundantly available, while attack data is rare and difficult to obtain [6]. While, in the ICS context, the ensemble's base-IIDSs could still be trained on benign data, ensemble learning usually remains dependent on observing the IIDSs under benign *and* attack conditions (cf. Sec. 3.2). This limitation effectively rules out the types of learning-based ensembles discussed in related work.

C2—Sequential data series. One property of IIDSs critical for their success is the ability to analyze dependencies in sequential data series (cf. Sec. 2.1). E.g., Seq2SeqNN accumulates the drift between predicted and observed behavior, detecting even subtle deviations [22]. Thus, the input data must be ordered chronologically, making the standard methodology from related work to randomly shuffle and split datasets into training and evaluation parts [16,25,27,34] infeasible for unsupervised IIDS ensembles as it alters a data series' order.

C3—Temporal dependencies. Another temporal effect is that cyberattacks persist over a longer period, e.g., to thwart detection by inflicting the damage slowly [5]. Since the effects may not happen instantaneously, an IIDS may detect attacks with some delay. For example, one base-IIDS might detect the attack at an early stage while a second IIDS requires more time, such that their alerts do not overlap. This issue demands time-aware ensembles which are capable of correlating both alerts to one attack. Yet, current ensembles base their decisions on a single data point without considering recent history, and no methods that take advantage of such temporal dependencies are known to us.

Takeaway. Due to their individual inefficiencies, a single IIDS is not enough for strong ICS cybersecurity. However, ensemble learning promises to transfer precisely this diversity into an advantage, albeit existing ensemble methodologies turn out infeasible because of unsupervised IIDSs' unique requirements.

4 The Potential of Unsupervised Ensembles

As established learning-based ensemble methods are inapplicable to the *un*supervised IIDSs predominantly used in ICSs (cf. Sec. 3.3), novel ways to fuse IIDSs into ensembles are required. First, we examine weight-based voting schemes and measure their maximal potential in Sec. 4.1 to establish a theoretical baseline. We then study to which extent this potential can actually be leveraged in Sec. 4.2.

4.1 Potential Analysis of Weight-based Ensembles

Ensemble methods for unsupervised IIDSs have to fulfill a new set of criteria compared to related work relying on supervised training (cf. Sec. 3.3). For our first potential analysis, we consider voting mechanisms, an approach that is not dependent on attack data for training and retains the temporal order of the data.

Design. As the first step, we train every base-IIDS on benign data for the target ICS such that, when presented with new data, we obtain an alert vector v encoding the judgment of each IIDS. Next, the ensemble assigns an individual weight $w \in \mathbb{R}^n$ to each IIDS. Then, a combined alert is emitted if the weighted sum of IIDS outputs meets a threshold $t \in \mathbb{R}$, i.e., $v_1 \cdot w_1 + \ldots + v_n \cdot w_n \geq t$.

The crucial part is finding suitable weights w and a threshold t which can either be achieved manually to implement strategies such as majority votes, or derived from knowledge of previous deployments. Consequently, this ensemble method does not necessarily require training (**C1**). Moreover, it leaves the data series intact (**C2**) yet does not consider temporal dependencies (**C3**).

Selecting a Baseline. Before exploring this ensemble method's potential, we must define a baseline for comparison. However, as apparent from our previous study, an IIDS can be optimized for certain objectives, e.g., to perform well in a particular metric (cf. Sec. 3.1). This decision is also heavily influenced by how FPAs and the risk of missing an attack are weighted by researchers (or operators), which is why finding one suitable metric is difficult in general [20,22,29,46].

Consequently, we decided to compare our ensembles against two promising IIDSs from the baseline. Given the results from Tab. 1, we select the MinMax IIDS as our primary baseline since it features the best eTaF1 score, which is a tradeoff between good precision and recall. In addition, we compare the results against the Invariant IIDS as it exhibits the most detected scenarios but also the most FPAs to investigate how the ensembles cope with this input condition.

Results. To examine the *theoretical* potential achievable with this methodology in the first step, we leveraged an optimization algorithm to systematically search for optimized weights and thresholds (Opt. Weights). To this end, we leveraged Ray Tune [28] to optimize these parameters for the eTaF1 metric. Note that weights and the threshold were searched within the interval $[-1, +1]$ instead of the entire $\mathbb{R}$ w.l.o.g. (any weighted vote can simply be scaled such that all the parameters are within any arbitrary non-empty interval in $\mathbb{R}$). We now examine results on the SWaT dataset and detail the results on WADI in Appx. A.

A weight-based ensemble with optimized parameters manages to outperform any base-IIDS in eTaP, eTaR, and eTaF1 (cf. upper part of Tab. 2). Surprisingly, MinMax's eTaF1 score is exceeded by over +11% points, which is a major improvement, especially considering related work yielded improvements of only around +1 % point (cf. Sec. 3.2). Aside from Invariant, this ensemble detects the most scenarios, namely 29 of the 33 attacks alerted by any base-IIDS. While

Table 2. IIDS ensembles have the potential to yield better results than any monolithic approaches (cf. Opt. Weights). But, to find these weights via means of trivial strategies (lower part), a tradeoff between detected scenarios and FPAs has to be made.

SWaT	**eTaP**	**eTaR**	**eTaF1**	**Scen.**	**FPAs**
MinMax [46]	67.8	47.1	55.6	23	9
Invariant [14]	54.7	29.8	38.6	30	217
Opt. Weights (eTaF1)	82.3	56.1	66.7	29	11
Any	21.1	35.3	26.4	33	160
≥2-Alerts	48.3	36.5	41.6	30	68
≥3-Alerts	61.6	35.3	44.9	26	60
Majority	82.4	34.4	48.5	17	34
≥5-Alerts	87.1	23.0	36.4	14	16
All	85.5	2.9	5.6	4	0

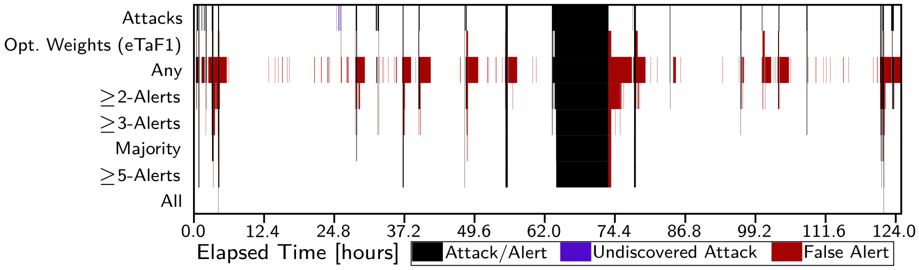

Fig. 3. Simple ensemble strategies drastically improve the understandability of alerts compared to the patterns shown in Fig. 2. Even though strategies like ≥2-Alerts measure 68 FPAs, these reside within the vicinity of attacks.

this ensemble still exhibits 11 FPAs, they are confined to the vicinity of actual attacks (cf. upper part of Fig. 3). More importantly, the ensemble's alerts do not show any phenomena discussed in Sec. 3.1, i.e., overhanging or randomly distributed alerts are eliminated, drastically improving understandability.

When analyzing the optimized weights, it is unsurprising that MinMax, with the highest eTaF1 score among the base-IIDSs, received the highest weight with 0.99 and could already yield an alert on its own since $t = 0.99$. The ensemble can even make use of Invariant with a weight of 0.88 despite its FPAs. Interestingly, some IIDSs received negative weights (Gradient with -0.33 and PASAD with even -0.91). Since PASAD, in some cases, does not cover any attack at all (cf. Fig. 2), the ensemble seems to leverage this phenomenon to filter out FPAs.

4.2 Finding Parameters for Weight-based Ensembles

Our theoretical results prove that weight-based ensembles are of practical use given a set of suitable weights and a threshold, but they do not provide ways to find such a configuration in a practical manner. Since performing such an optimization is infeasible in practice due to a lack of training data (**C1**), we now investigate six straightforward strategies to choose those parameters *manually*.

Manual Strategies. The most basic strategies we can implement with weights are *All* and *Any*, which emit an alert if all/any IIDSs emit an alert at the same time (corresponding to a logical AND or a logical OR). To realize them, setting all weights to 1 and the threshold to $t = 1$ (Any) or $t = n$ (All) is sufficient. Applications of these strategies can already be found in literature (cf. Sec. 3.2).

In between these extremes, further combinations are imaginable. For our evaluation, we consider just a subset of these combinations for the sake of simplicity: We define four more strategies that raise an alert when the *Majority* of IIDSs emit an alert or at least two, three, or five alerts are present. Note that these manual strategies can all be implemented with the proposed method, e.g., again setting all weights to 1 and $t = n/2$ corresponds to the Majority strategy.

Results. Overall, the results from such simple configurations cannot keep up with the optimized performance (cf. lower part of Tab. 2). When considering the trend of detected scenarios and FPAs from *Any* to *All*, these strategies suffer from being too conservative in either eTaP (precision) or eTaR (recall). While *Any* indicates all 33 detectable attacks and *All* has no FPAs, they perform worse in the respective opposite metrics. Moreover, none of the trivial combiners manage to improve the eTaF1 score compared to the single-best base-IIDS MinMax.

Fortunately, when digging deeper into the ensemble results by visualizing their alert patterns in Fig. 3 (lower part), all approaches, besides All and Any, score reasonably well on SWaT when evaluated qualitatively. Their alerts visually coincide with the attacks, and the FPAs are in close proximity to the actual attacks, notably eliminating Invariant's many randomly distributed FPAs. Compared to the original base-IIDSs' alerts from Fig. 2, even these straightforward ensemble strategies yield a usable result that can improve the perception for the ICS operator. However, this observation is not backed by current metrics.

Takeaway. Weight-based ensembles successfully increase the performance by up to +11% points in eTaF1 on the SWaT dataset. For the WADI dataset, we yield similar results (cf. Appx. A). Here, an ensemble with optimized weights can improve the eTaF1 score by nearly +12% points and even detects one more scenario than any base-IIDS. However, finding suitable parameters is non-trivial in the absence of training data because simple strategies, such as a Majority voting, leave a significant gap toward the optimum. Still, the advantages of weight-based ensembles lie in their simplicity, e.g., that parameters can be tuned manually throughout the operation, and their ability to reduce FPAs effectively.

5 Time-aware Ensemble Learning

One property of IIDSs that could impair the success of actual unsupervised ensembles may be temporal effects (cf. **C3**). While recent metrics, such as eTaF1,

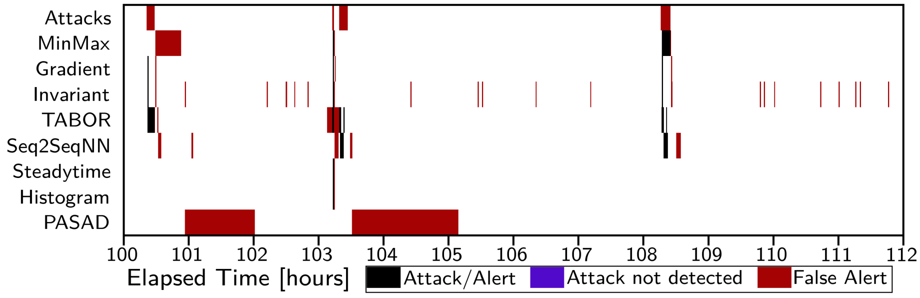

Fig. 4. This close-up reveals the individual alerting behavior of different IIDSs. However, by not considering temporal correlations, an ensemble cannot differentiate between, e.g., Invariant's short alarms that usually indicate FPAs and longer true alarms.

are already time-aware, i.e., they can deal with IIDSs that alert at different times during a longer attack, to the best of our knowledge, ensemble learning currently focuses solely on single instances without temporal dependence. To shed light on this issue, we first examine the alerting behavior of the IIDSs (cf. Sec. 5.1) and identify the potential for optimizing their alerts by taking temporal correlations into account (cf. Sec. 5.2). We then assess to which extent novel concepts of time-aware ensembles could improve the current situation (cf. Sec. 5.3).

5.1 Individual Alerting Behaviors Complicate Ensembling

Calculating optimal weights resulted in rather unexpected behavior (cf. Sec. 4.1): Gradient was given a negative score (-0.33), even though it visually performs nearly optimal (cf. Fig. 2). On the contrary, Invariant received a high score ($+0.88$), despite exhibiting 217 FPAs. To understand the roots of these outcomes, Fig. 4 provides a close-up of a few attacks and corresponding alerts.

Starting with MinMax, whose alerts are occasionally flagged as FPAs, these simply seem delayed such that a trained ensemble or an operator knowing this phenomenon could still use these indications for attack detection, especially if other IIDSs indicate attacks around the same time. Likewise, Gradient precisely indicates discontinuities at an attack's beginning and end. But, its second alert is often counted as FPA as the attack has ended at that time instance and is no longer labeled as malicious in the dataset [46]. Next, although hardly recognizable in Fig. 4, Invariant's FPAs, which occur randomly during benign regions, are usually short, i.e., lasting just a few seconds, compared to true alarms. Thus, Invariant can achieve a score of 0.81 in the point-based F1 score as these short alerts do not carry significant weight in that metric. The same effect can be observed for TABOR where FPAs are often just one second short.

Such effects pose issues to ensembles without temporal knowledge, as they can hardly distinguish different types of alarms. Yet, knowing such (often technically conditioned) effects provides unique opportunities for stronger ensembles.

Table 3. Incorporating temporal knowledge about the individual IIDSs' alerting behaviors not only improves their respective detection performance (upper part) but also significantly reduces the FPAs in an ensemble (lower part).

	IIDS/Combiner	eTaP	eTaR	eTaF1	Scen.	FPAs
New Baseline	Invariant [14]	$90.3^{+35.6}$	$28.0^{-1.8}$	$42.8^{+4.2}$	13^{-17}	10^{-207}
	MinMax [46]	$64.2^{-3.7}$	$43.8^{-3.3}$	$52.0^{-3.5}$	24^{+1}	1^{-8}
	Gradient [46]	$45.7^{+25.2}$	$37.4^{+31.5}$	$41.1^{+31.9}$	26^{+1}	2^{-62}
	TABOR [29]	$50.5^{+1.4}$	$18.9^{+0.0}$	$27.5^{+0.2}$	19^{+0}	9^{-19}
	The other base-IIDSs remained unaffected.					
New Results	Opt. Weights (eTaF1)	$87.2^{+4.9}$	$57.0^{+0.9}$	$69.0^{+2.2}$	28^{-1}	6^{-5}
	Manually selected weights[5]	$72.7^{-9.6}$	$60.3^{+4.2}$	$65.9^{-0.8}$	29^{+0}	3^{-8}
	Any	$22.0^{+0.9}$	$33.6^{-1.7}$	$26.6^{+0.2}$	32^{-1}	40^{-120}
	≥2-Alerts	$50.9^{+2.6}$	$42.3^{+5.8}$	$46.2^{+4.6}$	29^{-1}	15^{-53}
	≥3-Alerts	$65.4^{+3.8}$	$41.9^{+6.5}$	$51.1^{+6.1}$	25^{-1}	7^{-53}
	Majority	$83.2^{+0.8}$	$34.7^{+0.3}$	$48.9^{+0.4}$	16^{-1}	6^{-28}
	≥5-Alerts	$88.6^{+1.5}$	$28.2^{+5.2}$	$42.8^{+6.4}$	13^{-1}	5^{-11}
	All	$67.0^{-18.5}$	$3.9^{+1.0}$	$7.4^{+1.7}$	4^{+0}	0^{+0}

Superscript numbers show the difference between the prior baseline (Tab. 1) and results (Tab. 2).

5.2 Time-aware Ensemble Learning on Normalized Alarms

Our initial attempt to leverage this time-aware information in ensembles is to introduce a postprocessing step per IIDS right after each IIDS has emitted its alerts. Thereby, we can implement simple strategies that "normalize" the alerts prior to forwarding them to the ensemble method for decision-making.

In our case study in Sec. 5.1, we identified four strategies for MinMax, Gradient, Invariant, and TABOR, which help to clean up their alerts. For MinMax and Gradient, we simply extend their alerts artificially by one minute such that scenarios are detected if the alert is just slightly off. For Invariant, which suffers from randomly placed short alerts, we only consider those alerts where an alarm is emitted for more than ten consecutive seconds. We chose a similar approach for TABOR, where we filter out every alert that lasts just one time instance.

Incorporating temporal information into ensembles may help obtain better models and ultimately reduce the number of FPAs due to "misinterpretation". In practice, IIDS authors can provide guidance to aid in developing such strategies.

5.3 Potential of Time-aware Ensemble Learning

To assess the potential of time-aware ensembles, we applied these strategies in isolation for each of the four IIDSs: Invariant, MinMax, Gradient, and TABOR.

As shown in the upper part of Tab. 3 (New Baseline), this approach drastically improves their performance. Compared to the previous baseline, MinMax

[5] During experiments, we found an ensemble exceeding the eTaF1-optimized solution in Scen. and FPAs, indicating that eTaF1 does not fully coincide with subjective intuition.

now correctly identifies one additional scenario and reduces the total number of FPAs by eight. The same observation holds for Gradient, which even reduces the FPAs by 62 and can increase its eTaF1 score by 31.9 % points. These results indicate that most FPAs of MinMax and Gradient were in close proximity to actual attacks or triggered during the recovery of the ICS right after the attack. The strategies for Invariant and TABOR likewise yield a reduction in FPAs but also in the detected scenarios (in the case of Invariant). We assume that the reduction of detected scenarios for Invariant results from the filtered random alerts, which may have (falsely) contributed to the high number in the first place.

This improvement is not restricted to the base-IIDSs as it carries over to the ensembles, which now fare better using these "normalized" alerts (cf. New Results in Tab. 3). The optimized weight-based ensemble can slightly increase the eTaF1 score (+2.2) and reduce the FPAs by five. More importantly, the simple strategies with manually chosen weights now become usable as they feature a similar amount of FPAs compared to the optimized results, yet with slightly fewer detected scenarios. In that regard, the strategy ≥*3-Alerts* lacks behind the optimum, with just three fewer detected scenarios and one more FPA, significantly closing the gap between practically achieved and theoretical performance.

Takeaway. Temporal knowledge inside ensembles substantially improves their performances to a point where manual strategies become actually usable. Besides incorporating IIDS-specific information, future time-aware ensembles considering inter-IIDS alert dependencies may even exceed our initial results, but likely require sophisticated methods of finding (or training) an adequate model.

6 A Chance for Learning-based Ensembles

Until now, we evaluated the potential of IIDS ensembles on a single dataset. But one advantage of unsupervised IIDSs is that they can operate in different environments, i.e., generalize to new industrial domains after another training phase [47]. E.g., MinMax, Gradient, Steadytime, and Histogram have been designed, trained on, and evaluated for three datasets originating from different domains [46]. Given that the set of base-IIDSs remains fixed, training a learning-based ensemble on one dataset (or in the lab) under attack conditions and then transferring the *ensemble's model* to a different deployment/dataset without known attacks might be feasible. While it remains necessary to retrain the base-IIDSs on the new scenario, it hopefully suffices to keep the ensemble's model, i.e., the way in which the detection results are aggregated. This transfer-learning [43] is still in line with **C1** as only the unsupervised IIDSs would require retraining on benign-only data, while the pre-trained ensemble model is simply reused.

Transfer-learning promises a new level of flexibility for ensembles and may circumvent the issue of finding appropriate weights identified in Sec. 4.2. Also, it enables leveraging methods from related work such as ensemble stacking (cf. Sec. 3.2). To examine the feasibility of ensemble transfer-learning, we explain our new methodology in Sec. 6.1 and subsequently present the results in Sec. 6.2.

Table 4. Transfer-learning proves helpful in training unsupervised IIDS ensembles. E.g., if trained on the dataset SWaT and applied to WADI, most ensembles outperform the manual strategies (cf. Majority) in eTaF1 and come close to the Opt. Weights.

Transfer-learning		**eTaP**	**eTaR**	**eTaF1**	**Scen.**	**FPAs**
WADI → SWaT	Opt. Weights (Tab. 3)	87.2	57.0	69.0	28	6
	≥3-Alerts (Tab. 3)	65.4	41.9	51.1	25	7
	WADI's Opt. Weights (eTaF1)	75.1	33.8	46.6	18	11
	SVM [25]	76.6	43.1	55.2	21	12
	MLP [16,34]	75.9	55.5	64.1	27	18
	LR [25]	80.5	49.2	61.0	25	13
	Heuristic [25]	68.3	38.5	49.2	20	17
SWaT → WADI	Opt. Weights (Tab. 7)	86.9	62.2	72.5	11	2
	Majority (Tab. 7)	79.8	42.1	55.1	7	2
	SWaT's Opt. Weights (eTaF1)	76.6	50.4	60.8	10	2
	SVM [25]	71.3	63.5	67.2	11	9
	MLP [16,34]	80.2	63.5	70.9	11	3
	LR [25]	88.0	42.6	57.4	9	1
	Heuristic [25]	76.7	60.2	67.4	11	6

6.1 A New Methodology for Learning-based Ensembles

Learning-based ensembles require training on a dataset of exemplary attacks and expected labels of the outcome. Therefore, the methodology leveraged across related work usually bases on a *single* dataset artificially split into training and evaluation parts. Since we target an unsupervised scenario, we cannot assume to observe the ensemble's IIDSs under attack conditions in the target ICS (**C1**).

However, assuming that the same set of base-IIDSs behaves similarly in a different ICS, operators may have a strong interest in reusing a well-performing, pre-trained ensemble model. On the one hand, this approach enables assessing how well weights optimized for one scenario transfer to another ICS and, thus, helps to find these weights. On the other hand, supervised learning-based ensembles, which we neglected so far, may generalize too, such that approaches from related work can be leveraged even in the context of unsupervised IIDSs.

6.2 Putting Transfer-learning to the Test

To test the idea of transfer-learning, we leverage the SWaT and WADI datasets to which all eight base-IIDSs are applicable [47] and use one of the two datasets exclusively for training and evaluate the obtained ensemble model on the respective other dataset. Note that we use the "normalized" alerts according to Sec. 5 in this experiment to provide cleaner input data. As the first approach, we consider transferring the parameters of our weight-based ensembles. Furthermore, we analyze four learning-based variants from previous work. First, as leveraged in two publications [16,34], an MLP resembles a neural network classifier. Further classifiers include SVM [25] and LR [7,25]. Besides these, Kus et al. [25] proposed a heuristic that maps each possible alert vector to its most frequent output (benign or malicious), thereby optimizing the number of correct classifications.

Under the right conditions, transfer-learning can be an alternative to manual strategies (cf. Tab. 4). Yet, starting with ensembles trained on WADI and applied to SWaT (WADI → SWaT), all ensembles emit more FPAs than $\geq$ 3-Alerts. Nonetheless, MLP and LR outperform every base-IIDS and all manual strategies in eTaF1 (cf. Sec. 4.2), falling behind Opt. Weights by just 4.9 % points. MLP even detects two more scenarios than $\geq$3-Alerts. Unfortunately, transferring the weights optimized on WADI to SWaT yields the worst ensemble.

The reverse direction (SWaT → WADI) is more promising. Here, the weight-based ensemble detects ten scenarios with two FPAs, close to the optimum (cf. Opt. Weights). Also, learning-based ensembles perform better. Here, LR features just one FPA and nine detected scenarios, a great improvement over Majority. Again, MLP optimizes eTaF1 among all ensembles and exceeds the best base-IIDS by 12.8 % points, with only three FPAs. To validate whether this success stems from the larger training data (SWaT contains 36 attacks and WADI 14), we repeated this experiment by restricting the training data to the first 14 attacks of SWaT, making it similar to WADI. But, we could not verify this assumption as the outcomes were similar and differed on average by only 1.05 % points.

Takeaway. Given suitable training data, transfer-learning can be successful. Yet there is still room for improvement, i.e., by leveraging multiple datasets for training in the future. Most importantly, transfer-learning can outperform simple strategies (SWaT → WADI), thus providing an alternative to manually finding performant ensembles for actual ICS deployments with nearly no FPAs.

7 Conclusion

Enhancing ICSs' cybersecurity by augmenting them with an unsupervised IIDS promises to detect even zero-day attacks for which no training data exists. Despite a large research community inventing strong algorithms for IIDSs, we have outlined that relying on a single detector is insufficient given their individual weaknesses. Thus, we propose to leverage ensemble learning, as, e.g., used in computer vision, to combine a set of IIDSs and their strengths into one detector.

Surprisingly, we identify significant unused potential to improve not only the *combined* detection performance beyond what individual approaches can achieve but also their alerts' understandability for ICS operators. Incorporating temporal correlations into the ensemble's decisions reduces the number of FPAs further. Lastly, to ease the process of finding effective ensembles, we consider *transfer-learning*, i.e., training ensembles on attacks from a different ICS, to circumvent the difficulty of accessing training data for a target ICS under attack conditions.

In conclusion, ensemble learning poses an exciting direction for future work to strengthen ICSs' cybersecurity by tightly integrating strong methods for intrusion detection into an effective solution. To bootstrap ICS-specific research, we publish the ensembles' implementations [15], as well as the base-IIDSs' alerts and configuration files underlying our evaluations [45] as a dataset.

Acknowledgments. This work is part of the project MUM2 and was funded by the German Federal Ministry of Economic Affairs and Climate Action (BMWK) with contract number 03SX543B managed by the Project Management Jülich (PTJ). Funded by the German Federal Office for Information Security (BSI) under project funding reference number 01MO23016D (5G-Sierra) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. The responsibility for the content of this publication lies with the authors.

References

1. Ahmed, C., Palleti, V.R., Mathur, A.P.: WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In: CySWATER (2017)
2. Ahmed, C.M., Raman, M.R.G., Mathur, A.P.: Challenges in Machine Learning Based Approaches for Real-Time Anomaly Detection in Industrial Control Systems. In: ACM CPSS (2020)
3. Al-Abassi, A., et al.: An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System. IEEE Access **8** (2020)
4. Alladi, T., Chamola, V., Zeadally, S.: Industrial control systems: Cyberattack trends and countermeasures. Comput. Commun. **155** (2020)
5. Aoudi, W., Iturbe, M., Almgren, M.: Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. In: ACM CCS (2018)
6. Bader, L., et al.: Comprehensively Analyzing the Impact of Cyberattacks on Power Grids. In: IEEE EuroS&P (2023)
7. Balaji, M., et al.: Super Detector: An Ensemble Approach for Anomaly Detection in Industrial Control Systems. In: CRITIS (2021)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Comput. Surv. **41**(3) (2009)
9. Chen, X., et al.: Ensemble Learning Methods for Power System Cyber-Attack Detection. In: IEEE ICCCBDA (2018)
10. Conti, M., Donadel, D., Turrin, F.: A Survey on Industrial Control System Testbeds and Datasets for Security Research. IEEE Commun. Surv. Tutor. **23**(4) (2021)
11. Ding, D., et al.: A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing **275** (2018)
12. Erba, A., Tippenhauer, N.O.: Assessing Model-free Anomaly Detection in Industrial Control Systems Against Generic Concealment Attacks. In: ACSAC (2022)
13. Etalle, S.: From Intrusion Detection to Software Design. ESORICS **10492** (2017)
14. Feng, C., et al.: A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In: NDSS (2019)
15. Fraunhofer FKIE-CAD: IPAL - Industrial Intrusion Detection Framework. `https://github.com/fkie-cad/ipal_ids_framework` (2021)
16. Gao, J., et al.: Omni SCADA Intrusion Detection Using Deep Learning Algorithms. IEEE Internet Things J. **8**(2) (2021)
17. Gensler, A., Sick, B.: Novel Criteria to Measure Performance of Time Series Segmentation Techniques. In: KDML (2014)
18. Giraldo, J., et al.: A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. ACM Comput. Surv. **51**(4) (2018)
19. Goh, J., et al.: A Dataset to Support Research in the Design of Secure Water Treatment Systems. In: CRITIS (2016)

20. Hwang, W.S., et al.: “Do You Know Existing Accuracy Metrics Overrate Time-Series Anomaly Detections?”. In: ACM SAC (2022)
21. Kavallieratos, G., Katsikas, S.K., Gkioulos, V.: Towards a Cyber-Physical Range. In: CPSS (2019)
22. Kim, J., Yun, J.H., Kim, H.C.: Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks. In: CyberICPS (2020)
23. Kumar, A., Saxena, N., Choi, B.J.: Machine Learning Algorithm for Detection of False Data Injection Attack in Power System. In: ICOIN (2021)
24. Kus, D., et al.: A False Sense of Security? Revisiting the State of Machine Learning-Based Industrial Intrusion Detection. In: ACM CPSS (2022)
25. Kus, D., et al.: Poster: Ensemble Learning for Industrial Intrusion Detection. Tech. Rep. RWTH-2022-10809, RWTH Aachen University (2022)
26. Lee, J.J., et al.: AdaBoost for Text Detection in Natural Scene. In: ICDAR (2011)
27. Li, Y., et al.: Intrusion detection of cyber physical energy system based on multivariate ensemble classification. Energy **218** (2021)
28. Liaw, R., et al.: Tune: A Research Platform for Distributed Model Selection and Training. arXiv:1807.05118 (2018)
29. Lin, Q., et al.: TABOR: A Graphical Model-based Approach for Anomaly Detection in Industrial Control Systems. In: ACM ASIACCS (2018)
30. Louk, M.H.L., Tama, B.A.: Exploring Ensemble-Based Class Imbalance Learners for Intrusion Detection in Industrial Control Networks. Big Data Cogn. Comput. **5**(4) (2021)
31. Maglaras, L.A., Jiang, J., Cruz, T.J.: Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. J. Inf. Secur. **30** (2016)
32. Mendes-Moreira, J., et al.: Ensemble approaches for regression: A survey. ACM Comput. Surv. **45**(1) (2012)
33. Mitchell, R., Chen, I.R.: A Survey of Intrusion Detection Techniques for Cyber-Physical Systems. ACM Comput. Surv. **46**(4) (2014)
34. Nguyen, D.D., Le, M.T., Cung, T.L.: Improving intrusion detection in SCADA systems using stacking ensemble of tree-based models. Bull. Electr. Eng. Inform. **11**(1) (2022)
35. Ponomarev, S., Atkison, T.: Industrial Control System Network Intrusion Detection by Telemetry Analysis. IEEE Trans. Dependable Secure Comput. **13**(2) (2015)
36. Radoglou-Grammatikis, P., et al.: DIDEROT: An Intrusion Detection and Prevention System for DNP3-Based SCADA Systems. In: ARES (2020)
37. Rokach, L.: Ensemble-based Classifiers. Artif. Intell. Rev. **33**(1–2) (2010)
38. Sagi, O., Rokach, L.: Ensemble learning: A survey. WIREs Data Min. Knowl. Discov. **8**(4) (2018)
39. Singh, M., Singh, R., Ross, A.: A comprehensive overview of biometric fusion. Inf. Fusion **52** (2019)
40. Sommer, R., Paxson, V.: Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In: IEEE SP (2010)
41. Stallings, W., Brown, L.: Computer Security: Principles and Practice. Pearson, 4th edn. (2021)
42. Teixeira, A., et al.: Attack Models and Scenarios for Networked Control Systems. In: HiCoNS (2012)
43. Torrey, L., Shavlik, J.: Transfer Learning. chap. 11. IGI Global (2010)
44. Upadhyay, D., et al.: Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model With Majority Vote Ensemble Algorithm. IEEE Trans. Netw. Sci. Eng. **8**(3) (2021)

45. Wolsing, K., et al.: Artifact: One IDS is not Enough! Exploring Ensemble Learning for Industrial Intrusion Detection. Zenodo (2023)
46. Wolsing, K., et al.: Can Industrial Intrusion Detection Be SIMPLE? In: ESORICS (2022)
47. Wolsing, K., et al.: IPAL: Breaking up Silos of Protocol-dependent and Domain-specific Industrial Intrusion Detection Systems. In: RAID (2022)
48. Yazdinejad, A., et al.: An ensemble deep learning model for cyber threat hunting in industrial internet of things. Digit. Commun. Netw. **9**(1) (2023)
49. Zhang, C., Ma, Y.: Ensemble Machine Learning: Methods and Applications. Springer, 1st edn. (2012)
50. Zhang, D., et al.: A survey on attack detection, estimation and control of industrial cyber–physical systems. ISA Trans. **116** (2021)
51. Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. CRC Press, 1st edn. (2012)
52. Zhou, Z.H.: Machine Learning. Springer, 1st edn. (2021)

A WADI Results

In addition to the results of our experiments from Sec. 3, and Sec. 4, based on the SWaT dataset, we repeated the same analyses for the WADI dataset.

Table 5. No IIDS detects all attacks on WADI, and there is no single best detector that excels in all metrics. Also, the best IIDS for each metric differs from SWaT.

IIDS (Baseline)	eTaP	eTaR	eTaF1	Scen.	FPAs
Invariant [14]	92.3	32.6	48.1	6	3
PASAD [5]	5.4	4.3	4.8	5	3
Seq2SeqNN [22]	45.4	31.3	37.1	9	7
MinMax [46]	74.8	47.4	58.1	7	4
Gradient [46]	69.6	18.1	28.8	7	12
Steadytime [46]	87.0	38.7	53.5	6	2
Histogram [46]	63.7	43.2	51.5	7	6
TABOR [29]	14.9	13.0	13.9	8	4

In the baseline results, we observe a similar insufficiency for WADI (cf. Tab. 5) as previously discussed for SWaT in Sec. 3.1. No single IIDS is capable of detecting all 14 cyberattacks, and there is no single best IIDS for all metrics. MinMax achieves the highest score in two metrics, but Seq2SeqNN detects two more scenarios. In total 13 of WADI's 14 attacks would be detectable in combination.

We again assess the theoretical and practical potential for an ensemble on WADI as described in Sec. 4.1. Hereby, the eTaF1-optimized ensemble outperforms the best base-IIDS by +11.9% points in the eTaF1 score, detects more cyberattacks than any IIDS, and keeps the FPAs comparatively low (cf. upper part of Tab. 6). The manual voting strategies, however, fall short of this theoretical potential (cf. lower part of Tab. 6) with the best strategy lacking −25%

Table 6. Weight-based ensembles yield similar results on WADI as on SWaT (cf. Tab. 2) and can outperform each base-IIDS in eTaF1. While they have the potential to improve upon the base-IIDS, finding suitable weights is again non-trivial.

WADI	**eTaP**	**eTaR**	**eTaF1**	**Scen.**	**FPAs**
MinMax [46]	74.8	47.4	58.1	7	4
Seq2SeqNN [22]	45.4	31.3	37.1	9	7
Opt. Weights (eTaF1)	88.9	57.7	70.0	10	4
Any	25.5	47.0	33.1	13	14
≥2-Alerts	36.2	42.6	39.2	13	4
≥3-Alerts	61.2	50.0	55.0	8	10
Majority	85.8	36.4	51.1	7	2
≥5-Alerts	89.0	21.4	34.5	5	4
All	0.0	0.0	0.0	1	0

Table 7. As for SWaT (cf. Tab. 3), temporal knowledge improves the individual IIDSs' alerting behavior (upper part) and the ensembles' performance (lower part).

	Transfer-learning	**eTaP**	**eTaR**	**eTaF1**	**Scen.**	**FPAs**
New Baseline	Invariant [14]	$96.6^{+4.3}$	$32.2^{-0.4}$	$48.3^{+0.1}$	6^{+0}	0^{-3}
	MinMax [46]	$65.9^{-8.9}$	$41.4^{-6.0}$	$50.9^{-7.2}$	8^{+1}	1^{-3}
	Gradient [46]	$57.6^{-12.0}$	$31.6^{+13.5}$	$40.8^{+12.1}$	7^{+0}	3^{-9}
	TABOR [29]	$14.1^{-0.8}$	$13.0^{+0.0}$	$13.5^{-0.4}$	8^{+0}	1^{-3}
	The other base-IIDSs remained unaffected.					
New Results	Opt. Weights (eTaF1)	$86.9^{-2.0}$	$62.2^{+4.6}$	$72.5^{+2.5}$	11^{+1}	2^{-2}
	Any	$23.0^{-2.5}$	$43.4^{-3.6}$	$30.1^{-3.0}$	13^{+0}	8^{-6}
	≥2-Alerts	$34.6^{-1.6}$	$40.3^{-2.4}$	$37.2^{-1.9}$	13^{+0}	2^{-2}
	≥3-Alerts	$56.7^{-4.4}$	$43.7^{-6.3}$	$49.4^{-5.6}$	8^{+0}	8^{-2}
	Majority	$79.8^{-6.0}$	$42.1^{+5.7}$	$55.1^{+4.0}$	7^{+0}	2^{+0}
	≥5-Alerts	$79.0^{-10.0}$	$27.1^{+5.7}$	$40.3^{+5.8}$	5^{+0}	2^{-2}
	All	$100.0^{+100.0}$	$4.1^{+4.1}$	$8.0^{+8.0}$	1^{+0}	0^{+0}

Superscript numbers show the difference between the prior baseline (Tab. 5) and results (Tab. 6).

points behind in the eTaF1 score. Nonetheless, the ≥2-Alerts ensemble indicates *all* 13 cyberattacks detected by any base-IIDS while maintaining only four FPAs.

Incorporating temporal knowledge (cf. Sec. 5) enhances the optimum by +2.5% points in eTaF1 and improves the manual strategies, especially in FPAs (cf. Tab. 7). We see a substantial improvement in the eTaF1 score of the best manual vote (+5.8% points for ≥5-Alerts), and the ≥2-Alerts strategy still detects 13 cyberattacks, now with just 2 FPAs, matching the number of FPAs achieved by Opt. Weights. Unfortunately, this great result is not expressed by eTaF1.

These results support the previous conclusion that weight-based ensembles are useful given suitable weights and thresholds, yet finding them remains non-trivial. Lastly, adding time-awareness yielded a significant performance boost.