Introducing Multi-Layer Concatenation as a Scheme to Combine Information in Water Distribution Cyber-Physical Systems

Côme Frappé-Vialatoux, Pierre Parrend

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

Introducing Multi-Layer Concatenation as a Scheme to Combine Information in Water Distribution Cyber-Physical Systems

Côme Frappé - - Vialatoux[a,b,*], Pierre Parrend[a,b]

[a]*ICube, UMR 7357, Université de Strasbourg, CNRS, Strasbourg 67000, France*
[b]*Laboratoire de Recherche de l'EPITA, 14-16 Rue Voltaire, Le Kremlin-Bicêtre 94270, France*

Abstract

As Water distribution infrastructures are ageing, their modernization process is leading to an increased incorporation of connected devices into these physical systems. This transition is changing the nature of water distribution control systems from physical systems to cyber-physical systems (CPS). However, this evolution is associated with an increased vulnerability to cyber-attacks. Detecting such attacks in CPS is gaining traction in the scientific community with the recent release of cyber-physical datasets that capture simultaneously the network traffic and the physical state of a water distribution testbed. This novel paradigm of conjoint availability of these two types of data from a common source infrastructure opens a new question on how to combine their information when training machine learning models for attack detection. As an alternative approach to previous models that rely on model aggregation, this paper introduces *Multi-Layer Concatenation*, a combination scheme to merge the information from the physical and network parts of a CPS from a data perspective, through a time-based join operation coupled with a propagation process to keep the coherence of the global system. The evaluation of its impact assesses its benefits for machine learning-based detection on three cyber-physical datasets, by measuring machine learning models' performances on physical and network data separately, and then on data combined through the proposed scheme.

Keywords: Water Distribution; Cyber-Physical Systems; Machine Learning; Cyber-Security

1. Introduction

Because water distribution systems are critical infrastructures, the attacks targeting them often stay undisclosed to the public. Since the year 2000, the diversity of attackers' profiles, motives and means are converging towards an archetype of attacks conducted by outsiders using cyberattacks such as ransomware or remote-access, with the intent of disrupting or paralyzing the infrastructure [15] [33]. The development of proactive approaches that avoid such attacks through early detection is a task that Machine Learning (ML) algorithms have proved to be proficient

* Corresponding author.
E-mail address: come.frappe-vialatoux@etu.unistra.fr

at [35]. The rapid development of Internet of Thing (IoT) technologies causes an increase in the connectedness of water distribution systems, shifting these physical systems into CPS, meaning that they are now composed of two interacting subsystems, a physical one for water handling and a cyber one supporting the communication between the devices and third party supervision systems. With cyber-attacks becoming prominent in the landscape of threats against water distribution infrastructures, there has been an effort in the scientific community to better represent these CPS in publicly available datasets, through physical datasets incorporating cyber-attacks [3] [31], and more recently via cyber-physical datasets, with simultaneous captures from physical sensors and network traffic. Although the use of either of these two types of data to develop ML models for cyber-attack detection is well represented in the literature, the release of cyber-physical datasets opens novel research questions on how to combine the information present in the whole system through those two data types, and what impact it has on detection performance.

To address these challenges, this paper introduces *Multi-Layer Concatenation* as a way to combine the information from both subsystems from a data perspective, which complements the approach proposed in [12] that combines the information from a model perspective, for which numerous models must run in parallel, implying heavy computational cost, in addition to requiring a very specific implementation for the underlying infrastructure. Proposed *Multi-Layer Concatenation* solves these issues by not being tied to the infrastructure and requiring less computational power. Experiments consisting of applying *Multi-Layer Concatenation* on 3 cyber-physical datasets led to better Machine-Learning model performances than when using physical or network data separately for multiple machine-learning models, further establishing the combination of information from physical and network data as a promising research field.

The remainder of this paper is organized as follows: Section 2 presents the state of the art, Section 3 explicits the requirements for ML base cyber-attack detection, Section 4 presents the different cyber-physical datasets used and a taxonomy of the cyber-attacks they incorporate, Section 5 describes the physical and network characteristics of the datasets, Section 6 defines the proposed *Multi-Layer Concatenation*, Section 7 describes the experiments conducted to assess its effect on ML models performances, followed by the presentation of the results in Section 8 and their discussion in Section 9. Section 10 concludes this work.

2. State of the Art

This section highlights the traditional algorithms used for cyber-attack detection by comparing their performances on network data and physical data respectively. Recent works on approaches to combine the information from physical and network data is presented, with benefits on detection that motivate our work on combining the information from a data perspective.

2.1. Detection of attacks against water distribution infrastructures

As described in the precedent section, the CPS nature of water distribution infrastructures makes cyber-attack detection possible both on the network part and on the physical part. This section reviews the literature on attack detection first on network data, then on physical data.

In a review comparing published ML algorithms performances for attack detection on cybersecurity datasets [25], SVM, RF and DT achieve the best performances on intrusion detection tasks in terms of accuracy and precision, but not for the recall metric, which represents a trade-off in prioritizing a high true positive over a low false negative rate. In addition to the global performance evaluation of models, this study highlights how the performances vary depending on the datasets and model, for instance, RF achieved 98.10% precision and 98.10% recall on KDD CUP99 dataset [24], but only 81.40% precision and 75.30% recall on the NSL-KDD dataset (2009) [32]. A benchmark study comparing tree-based approaches to deep learning approaches for tabular data, which network traces are part of, showed that tree-based approaches outperform on both classification and regression tasks, with XGBoost [7] standing out as the overall best performer. An empirical explanation of this observation points towards ANN being more affected by uninformative features and less efficient at fitting irregular functions.

The use of physical data is already well established in the field of water distribution to monitor the physical processes, with the first Supervisory Control And Data Acquisition (SCADA) systems being developed in the 1970s [13], democratising the acquisition and storage of data. This availability of data benefits ML algorithms, allowing them to be trained on physical monitoring tasks such as pipe failures[34] or leaks detection and localization [17]. However, in addition to purely physical events, cyber attacks effects also reflect on the physical processes and thus ML algorithms

can be used to detect them [22]. The *Battle of the Attack Detection Algorithms* (BATADAL) [31], which is a competition to develop cyber-attack detection algorithms on physical data of a water distribution network, had participants that used ML algorithms in their final solution, such as Recurrent Neural networks (RNN) [5], Convolutional Variational Auto-encoder (CVAE) [6], RF [2] and MLP [1]. The ranking of these solutions is MLP, RNN, CVAE, and RF. It is to be noted that these algorithms were not used as standalone solutions and were only part of bigger algorithmic pipelines and that the best solution of the competition [16], with an accuracy of 97.5%, is not based on ML but used an error threshold between a simulation of the water network and the SCADA data. In addition to the original contestants, recent work on the BATADAL dataset used Temporal Graph Convolutional Network and High Confidence Auto-Encoder (HCAE) [28], respectively ranking eighth and third when put in the context of the competition, with the latter being pointed out by the authors as more robust against previously unseen attacks.

These performance results show that the information present in network datasets varies between datasets and can be insufficient to achieve good cyber-attack detection performances, while physical datasets allow good cyber-attack detection performances. This points towards the potential benefits of combining the information present in these two distinct types of data, allowing a complete representation of the system for the models to be trained on.

2.2. Combining physical and network information

CPS are multi-layer systems, with each layer in interaction with the others. The security of these systems needs to consider these multiple layers together with their interaction, which cannot be done when considered individually. ML models applied to CPS security thus need to integrate both the cyber and physical parts. However, the recent identification of this problem is reflected in the literature by a lack of methods for performing the combination of information of the different layers and is identified as a research gap in the field of CPS security [4].

One method based on ”hybrid multi-formalism” described in [12] consists of a combination of the prediction of unsupervised anomaly detection models run separately on the physical and network data, that are then combined through Bayesian networks, leveraging the interpretability of probabilistic modeling. Their results on specific attack scenarios of the HITL dataset [11] achieve a precision of 99,75% and recall of 95,74% for the first scenario, and a precision of 99,25% and a recall of 97,79% for the second scenario. These performances are an improvement over the performances of the Naive Bayes algorithm, also a probabilistic model, from the HITL dataset associated paper [11], which upon being trained separately on physical and network data achieved a precision of 66% and a recall of 92% on physical data, and a precision of 90% and a recall of 15% on network data. It is to be noted that these last performances were obtained with all attack scenarios put together rather than training on them individually. The results obtained by this combination method focusing on the consolidation of the output of different models prove the progression margin that can be obtained when shifting from treating physical and network data separately, to combining their information during the learning process.

3. Requirements for detecting Attacks

This section explains the requirements for the efficient use of ML models for cyber-attack detection, and identify the factors to take into account for evaluating our proposed *Multi-Layer Concatenation*.

Cyber-attack detection is well supported in the literature by the regular release of labelled datasets from the network communication field on which to train models [27] [23]. The metrics used by the community to evaluate the detection capability of developed models are mainly accuracy, precision and recall [26] (formulas given in appendix Appendix A) but biases in these metrics notably in the case of unbalanced classes leads to an increased reliance on metrics that take class imbalance into account such as balanced accuracy and Matthews Correlation Coefficient (MCC) [8] which produces values that better represents the overall model performance in the case of negatively and positively unbalanced datasets. The trustworthiness of the model and its prediction are challenges for ML applied to cybersecurity, especially to critical infrastructures such as water distribution networks. This trustworthiness is measured by the proportion of false alarms, which translates to the False Positive Rate (FPR) metric, which needs to be minimized to avoid triggering ”alarm fatigue”, which effect on the personnel includes distrust in the alarm, boredom and apathy [9]. Trustworthiness is also tied to the explainability of the model used [19]. Models deemed ”Black-Boxes” such as MLP cannot provide an interpretability of the output results, contrary to tree-based models. In the context of cybersecurity, explainability is crucial to allow for quick identification of the attack entry-point, which then leads to faster isolation of the compromised systems and quickens the resolution of the attack against the system.

In the context of *Multi-Layer Concatenation*, its impact on the aforementioned metrics and models is a key evaluation of its viability for cyber-attack detection.

4. Detecting Attacks against Water Distribution Networks

The multiplication of cyber-attacks against water treatment facilities [**?**] [33] calls for better attack detection models. These infrastructures categorized as CPS [30] use computational means to operate physical operations. This duality comes with an enlarged attack perimeter, making them vulnerable to attackers operating in the cyberspace.

The release by the scientific community of cyber-physical datasets as well as tools to generate cyber-physical data [21] [20] represents a notable step towards more representative data for these types of infrastructures. This section first presents the testbeds used to generate the cyber-physical datasets used in our study, then places the cyber-attacks from the datasets in the context of MITRE ATT&CK framework.

4.1. Testbeds and Datasets

A Hardware In the Loop . "A Hardware-in-the-Loop Water Distribution Testbed" [11] is a cyber-physical testbed composed of two subsystems, one real and one simulated. It served for the acquisition of the Hardware In The Loop dataset (HITL) published in 2021. The addition of the simulated subsystem enables for a more complex architecture, in an intent to overcome the limitations posed by too simplistic testbeds. The dataset comprises the data from multiple scenarios, with attacks both of physical and network nature, with an emphasis made on the diversity of cyberattacks. This dataset contains the physical readings of the Programmable Logic Controllers (PLC) and a capture of the network traffic generated by the running testbed.

SWaT dataset. Secure Water Treatment (SWaT) is a water distribution testbed [**?**] used to generate and publish physical and cyber-physical datasets since 2015. It has been used to produce 7 different datasets between 2015 and 2020 with different attack scenarios. The dataset used in this study is the SWaT.A6 dataset from 2019, which has been chosen for its variety of attacks and the presence of documentation explaining the conduct of the attack steps.

ICS Flow. The ICS-Flow dataset [10] published in 2023 contains data from a fully simulated bottle-filling pipeline. Although it is not the same water distribution task as the other datasets, it uses the same range of components, including valves, tanks, pipes, flow sensors and tank level sensors as well as comparable risks with regard to water consumption. The difference resides in the additional components that are not found in the water distribution sector, such as conveyor belts, and sensors for bottle positions.

4.2. A taxonomy of cyber attacks against Water Treatment Facilities

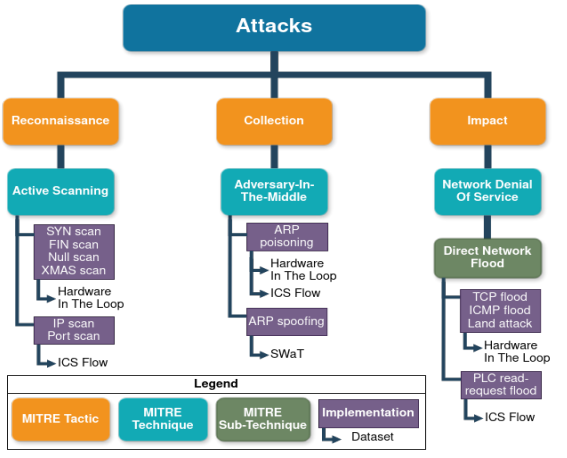

Fig. 1: Taxonomy of datasets attacks according to the MITRE ATT&CK framework

This section compiles the different cyber-attacks found in the three datasets introduced and places them in a taxonomy following the Mitre ATT&CK framework composed of the reconnaissance, collection and impact tactics.

The cyber-attacks present on the HITL dataset are composed of 4 different types of active scanning from the reconnaissance tactic, ARP-poisoning as an Adversary-in-the-middle technique, part of the collection tactic, and 3 different Direct Network Flood, a sub-technique part of Network Denial of service technique in the impact tactic. The ICS-flow dataset also only uses ARP-poisoning in the collection tactic, but differs in the Active Scanning technique by using IP and port scans, as well as in the Network Denial Of Service technique by using PLC Read-request flood.

Figure 1 presents a taxonomy of the cyber-attacks present in the datasets based on MITRE ATT&CK framework [29]. It can be observed that HITL and ICS-Flow datasets both include attacks from the Reconnaissance, Collection and Impact tactics, covering multiple steps of the cyber-kill chain. The

SWaT.A6 dataset attacks consisted of multiple SCADA data exfiltration via a first malware inserted from a USB thumb drive and disruption of sensors and actuators from a second malware inserted in the system via download. The documentation does not provide further details on the implementation of the different steps.

5. Characterization of Water Treatment Datasets Complexity

The physical and network topologies of the testbeds of each dataset are detailed here to provide an assessment of their complexity in term of the quantity and repartition of their physical components, as well an assessment of their network complexity from a graph theory perspective. The physical complexity of each testbed summarized in Table 1 shows the diversity of configurations between the studied datasets.

Table 1: Characteristics of physical testbeds

Physical Testbed	**Rows**	**Columns**	**Physical PLCs**	**Simulated PLCs**	**Percentage of attacks in data**
HITL	10923	43	1	3	18.47%
SWAT	13201	84	6	0	18.88%
ICS_FLOW	39302	24	0	2	20.45%

The chosen datasets show a diversity in their physical characteristics. The main aspect of divergence is the number and nature of the Programmable Logic Controllers (PLCs) that compose them, with SWAT being a purely physical testbed in addition to being the most elaborated in terms of number of PLC, ICS-Flow being purely simulated and the most simple one, while HITL is a mixed testbed that incorporates a physical and a simulated part. All of the physical data from the 3 datasets have a similar proportion of attack.

This diversity is impacting the information contained in the physical data between the datasets, with smaller infrastructures containing less information. However, oppositely, smaller infrastructures also have a reduced attack perimeter, with potential attack effects having only a few components to reflect on. To the best of our knowledge, no study has yet measured the impact of testbed complexity on attack detection performances.

Fig. 2: Network topologies of MAC and IP communication for each dataset

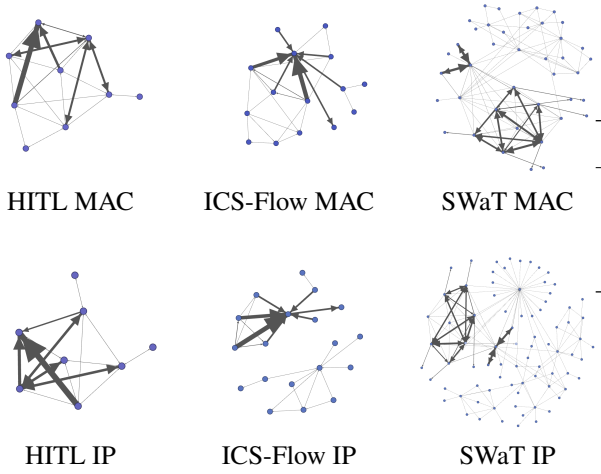

Table 2: Characteristics of the network graphs

Dataset	**Graph**	**Nodes**	**Edges**	**Weights**				
				min	max	mean	median	sum
HITL	mac	10	38	11	$6.20{\cdot}10^{8}$	$6.44{\cdot}10^{7}$	$1.01{\cdot}10^{6}$	$2.45{\cdot}10^{9}$
HITL	ip	8	28	30	$5.67{\cdot}10^{8}$	$8.74{\cdot}10^{7}$	$2.04{\cdot}10^{6}$	$2.45{\cdot}10^{9}$
ICS-Flow	mac	15	29	3	$7.02{\cdot}10^{6}$	$8.68{\cdot}10^{5}$	1530	$2.52{\cdot}10^{7}$
ICS-Flow	ip	19	25	3	$7.02{\cdot}10^{6}$	$1.01{\cdot}10^{6}$	895	$2.52{\cdot}10^{7}$
SWaT	mac	41	150	3	$1.86{\cdot}10^{7}$	$2.14{\cdot}10^{6}$	$6.13{\cdot}10^{4}$	$3.21{\cdot}10^{8}$
SWaT	ip	80	219	2	$1.86{\cdot}10^{7}$	$1.47{\cdot}10^{6}$	2149	$3.21{\cdot}10^{8}$

As for the physical characteristics, the network topologies also greatly differ between datasets. Figure 2 shows the network graphs of each dataset, both for communications between IP addresses and MAC addresses. In these graphs, the nodes represent unique MAC addresses for the first row and unique IP addresses for the second row. The edges

represent a communication between two nodes, weighted by the total number of packets exchanged during the whole dataset acquisition. The graphs properties computed in table 2 show a disparity between the datasets on the number of nodes and edges, with ICS-flow having two times more IP nodes than HITL, and SWaT having ten times the number of IP nodes of HITL dataset. The difference between the number of IP and MAC nodes can have multiple explanations for each dataset such as multiple network interfaces on single machines, or in the case of the HITL dataset, the settings of the virtualisation that ran the simulated part of the testbed.

A common property of all the datasets is the difference between the minimum and maximum weight of the graph being from six to seven orders of magnitude, regardless of the number of nodes. Moreover, on HITL and ICS-Flow datasets the maximum weight represents around a quarter of the sum of the weights, which means that in these datasets, only two nodes are responsible for 25% of the total traffic. This ratio is 5% for the SWaT dataset, which can be interpreted as it having more spread traffic between the nodes.

6. Multi-Layer Concatenation

To leverage both physical and network information, and thus to ease the investigation and improve incident response, we propose a novel scheme for combining the information of both sources for the learning process from a data perspective. First, the overall process is described, and the treatment pipeline is specified. We then provide an algorithmic formalisation of the process with complexity analysis to assess its applicability.

6.1. The Multi-Layer Concatenation process

This paragraph explains the context and choices made for developing **Multi-Layer Concatenation** as well as a process comparison with model-based information combination method from the literature.

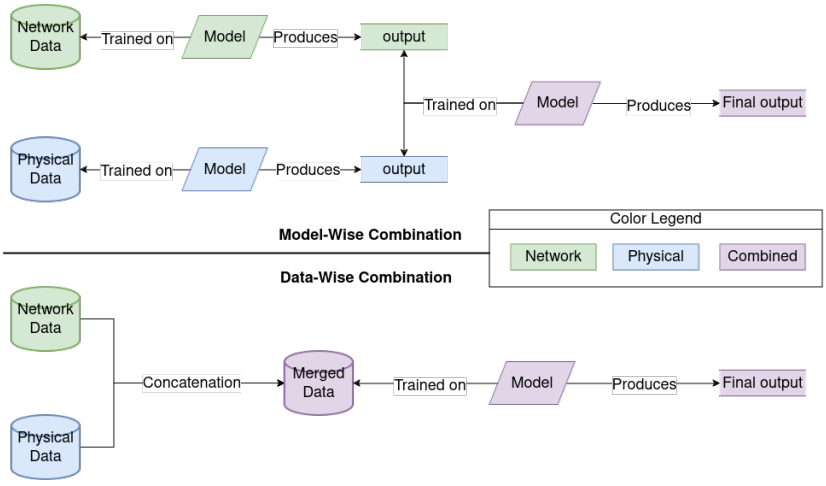

Fig. 3: Comparison of model-wise and data-wise combination processes

As the network and physical data are obtained from independent sources but contain information from common events, a challenge resides in preserving their coherence when combining them. The proposed scheme consists of concatenating each network data row with the physical data row having the closest inferior time. This choice is motivated by the different acquisition frequencies of these two types of data due to the timescale of the different processes that they measure. The physical data time granularity is tied to the physical processes that are occurring and thus limited by the laws of physics, specifically in the water distribution sector where it is bound to the speed at which water displacement can occur. This implies that the acquisition frequency has a maximum, past which no physical changes would have been measured between two consecutive acquisitions. This limitation is not present for network data, where the limiting factors such as bandwidth or processing speed of the involved devices are of much higher frequency. This results in the time granularity of network data reaching four orders of magnitude higher than that of

the physical data, the latter being typically acquired each second. Once the data are merged, they can be directly used to train ML models that will then use the information of both the physical and network layers at once.

This data-wise approach to combination differs from the model-wise combination process presented in [12]. Figure 3 explains the different processes: as stated by the authors, in a model-wise combination process we train models on the physical and network data "separately and simultaneously", which means that the same timestep is treated in parallel by models on physical data and models on network data. The outputs of these models are then used as input for another model whose predictions leverage information from both physical and network data as provided by its inputs. In a data-wise combination process such as the proposed **Multi-Layer Concatenation**, the physical and network data are merged, then used as input for a model, whose prediction will also leverage information from both network and physical data.

6.2. Data Treatment Pipeline

The steps of the data treatment pipeline that constitutes *Multi-Layer Concatenation* are as follows: The physical and network data are typically segmented into multiple files. The files of interest are selected, cleaned and treated to be in a consistent format. This step depends on the dataset and can include columns name uniformisation, time format changes, and erroneous data corrections such as typos. The treated data are then concatenated and sorted by time. The physical data are then checked for any row that would contain only missing values, which are dropped if any. In our implementation, the data are then saved in a *parquet* format, which is a compressed format that allows more memory-efficient loading than *csv* files, as their size increased due to concatenation. The next step consists of creating a common time column on which to join the data. For both physical and network data we use the time column that we duplicate in a new column named "Time_join". For the network data we change this "Time_join" column's time granularity to match that of the physical time column. For instance, with physical data acquired each second and network data acquired at the millisecond scale, the "Time_join" of the network data is truncated at the second, resulting in all network data acquired during the same second having an identical "Time_join" value. We now proceed to a left join of the physical data into the network data, on the "Time_join" column. This results in all network data being acquired during the same second as physical data to have this physical data concatenated to them. An edge case then occurs if network data was acquired during a second where there were no physical data. This can be caused by uneven physical data acquisitions that sometimes have more than a one-second interval, or at the end of the acquisition if the physical data acquisition is stopped before the network data acquisition. This edge case is characterized in the data by network rows with missing physical data concatenated to them. We handle this edge case by replacing these missing correspondences with the values of the last row that had a correspondence. The resulting data are then saved in a parquet format for them to be used for the training of ML models.

6.3. Algorithmic formalisation

Algorithm 1 is a formalisation of *Multi-Layer Concatenation* where the three main operations described in the precedent section are performed: first to create the columns for the left join, second to perform the join, and lastly to handle missing correspondences. The time complexity on network data of size n is $O(n \cdot log(n))$ for the sort operation, $O(n)$ to duplicate the time column, $O(n)$ to change its format, which adds up to $O(n \cdot log(n)) + 2 \cdot O(n) = O(n + n \cdot log(n))$. The time complexity on physical data of size m is $O(m \cdot log(m))$ for the sort operation, $O(m)$ to drop the rows and $O(m)$ to duplicate the time column, which also adds up to $O(m \cdot log(m)) + 2 \cdot O(m) = O(m + m \cdot log(m))$. The left join operation is $O(n)$ and $O(m)$ for respective datasets as the data are already sorted. The missing correspondences search is $O(\frac{n(m-1)}{2})$, characterised by the worst case when we have to look at all precedent data for each row, which simplifies to $O(nm)$. The final time complexity for algorithm 1 is $O(n + n \cdot log(n) + m + m \cdot log(m) + nm)$

The memory complexity of this algorithm with network data of size $n \times N$ and physical data of size $m \times M$ is $O(n \cdot (N + M))$, as it has for upper bound the memory space needed to contain the merged data because other operations only apply to either single rows, values or columns at a time.

7. Experiments

Experimental Setup. The experiments were run on a laptop with 32Gb of RAM, 13th Gen Intel® Core™ i7-13700H 20 cores CPU, NVIDIA RTX A500 GPU. The operating system is Ubuntu 22.04.3 LTS, and evaluations were run using Python 3.11.4 and the libraries pandas (2.0.2), numpy (1.25.1), scikit-learn (1.2.2), xgboost (1.7.6) and keras (2.13.1).

Algorithm 1: Efficient Algorithm for *Multi-Layer Concatenation*

Require: *ConcatenatedNetwork_data*, *ConcatenatedPhysical_data*
Ensure: *merged_data*
Sort *Network_data* by time
Sort *Physical_data* by time
drop *Physical_data* rows with only missing values
Physical_data[*time_join*] ← *Physical_data*[*time*]
Network_data[*time_join*] ← *Network_data*[*time*] with *Physical_data* time granularity
merged_data ← left_join(*Network_data*, *Physical_data*) on *time_join*
for *row* in *Merged_data* **do**
  if all physical values are missing **then**
    fill *row* with latest physical_values
  end if{ensures closest anterior physical data on missing correspondances}
end for
return *merged_data*

Implementation. *Multi-Layer Concatenation* is implemented using a Python environment and the pandas and numpy libraries following Algorithm 1. The missing correspondences have been handled directly during the left join with the use of the *merge_asof* function of pandas with the argument *direction* set to "backward". This implementation choice has the particularity to fill the missing correspondences with the value of the closest anterior valid row, *including its missing values*, that are kept as missing values when copied in the missing correspondence, whereas other possible implementations such as the *fillna* function of the pandas library do not keep it as missing value and instead replace them with the last non-missing value of the column. Our implementation of algorithm 1 took less than 8 seconds to complete the join and missing correspondence operations together on the HITL dataset with network data of size 29829204×16 and physical data of size 10923×43. The models evaluated are XGBoost, Random Forest, Multi-Layer Perceptron and Random Forest using cross-validation.

8. Evaluation

This section provides the analysis of the results of the experiments described above. Discussed results are shown in Table 3 which provides the detection performances of the XGBoost algorithm on all the datasets considered in this study. A complete listing of the performances of all the algorithms on physical and network data separately, as well as on data combined through *Multi-Layer Concatenation* can be found in Appendix C. This section is organized as follows: first we compare the performances obtained when using only network and physical data for training, then we compare it to the results obtained when applying *Multi-Layer Concatenation*.

Table 3: Detection Performances of XGBoost on SWAT, ICS_FLOW and HITL datasets

Testbed	Dataset	Precision	Recall (TPR)	TNR	Accuracy	F1_score	Balanced_accuracy	MCC
SWAT	Physical	0.9980	0.9936	0.9997	0.9980	0.9980	0.9949	0.9956
SWAT	Network	0.7416	0.0533	0.9976	0.7372	0.6383	0.2277	0.1910
SWAT	Combined	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ICS_FLOW	Physical	0.9934	0.9764	0.9975	0.9935	0.9934	0.9675	0.9805
ICS_FLOW	Network	0.9979	**0.9977**	0.9979	0.9979	0.9979	0.9816	0.9938
ICS_FLOW	Combined	**0.9988**	0.9966	**0.9994**	**0.9988**	**0.9988**	**0.9950**	**0.9966**
HITL	Physical	0.9769	0.9185	0.9921	0.9780	0.9773	0.7602	0.9327
HITL	Network	0.8355	0.5944	0.9987	0.8718	0.8152	0.4978	0.7252
HITL	Combined	**0.9993**	**0.9991**	**0.9994**	**0.9993**	**0.9993**	**0.9503**	**0.9985**

8.1. Detecting Attacks on physical and network layers individually

The results obtained without using *Multi-Layer Concatenation* correspond to the first and second rows for each testbed of Table 3. Physical data alone allows for high detection performances in all three datasets, which are superior by a significant margin to the performances on network data alone except for the ICS-Flow dataset. The detection performances obtained on this dataset are extremely high regardless of the data, for which a possible explanation is the much lower number of components that composes its testbed as discussed in section 5, which results in a simpler learning task for the models. For all datasets, the physical data have a lower acquisition frequency, leading to a lower number of physical data points. However, this difference is negligible for the ICS-Flow dataset with only a 15% increase in the number of rows compared to a difference of 3 orders of magnitudes for the SWAT and HITL datasets. The training time of XGBoost algorithm with regards to the number of data points is represented in Figure 4. The time values are represented in a barplot, the dataset sizes are represented with linked points and expressed with the total number of point in the dataset ($\#row \times \#columns$) with values bound to the secondary y axis on the right. This figure shows that the low number of physical data is associated with low training time of XGBoost of less than 3.5 seconds. This lower size has the downside of making the least frequent attacks too rare for the model to be able to generalize on them. This is particularly observed for the HITL physical data which only contains a total of 14 rows labeled with scan attacks which goes down to 4 occurrences for the testing set after splitting the data. This effect has an impact on the performances of XGBoost on the HITL physical data, where all 4 occurrences of scan attack were misclassified as normal thus lowering the metrics of balanced accuracy and MCC.

These performances greatly vary from one dataset to another even though the data are of the same nature, which illustrates that the classification task difficulty is dependent on the inner characteristics of the data itself. Combining the information from the physical and network data is a way to overcome this limitation.

8.2. Detecting Attacks after Multi-Layer Concatenation

The use of the proposed *Multi-Layer Concatenation* results in higher detection performance of XGBoost on all datasets and across all metrics. The highest benefit is measured on the HITL dataset with a balanced accuracy going from 76.02% on physical data to 95.03% with *Multi-Layer Concatenation*. This increase on the HITL dataset is explained by better performances on all classes, with a notable increase for the scan attacks. As the *Multi-Layer Concatenation* uses network data as the base for the concatenation, the attacks have more occurrences for the model to learn, which led to the model accurately classifying 16 out of the 21 occurrences on the test set after *Multi-Layer Concatenation* which resolves this limitation of physical data.

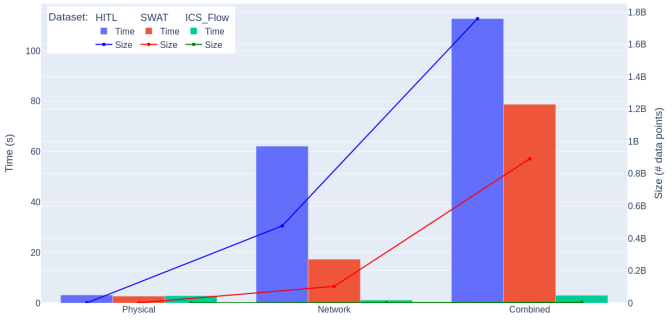

Fig. 4: Training time of XGBoost algorithm in relation to datasets sizes

The training time of XGBoost after *Multi-Layer Concatenation* in Figure 4 shows that the training time is correlated to the dataset size, with HITL having the highest training times of 3 seconds, 62 seconds, 112 seconds respectively for physical, network, and combined data. SWAT comes second with 3 seconds, 17 seconds and 78 seconds. ICS-FLOW has the lowest training times with 3 seconds on physical data, 1 seconds on Network data and 3 seconds on combined data. These extremely low values for ICS-Flow training time are correlated with its very low size, being at most 3 Million individual data points for the combined dataset, compared to 892 Million for SWAT and 1.75 Billion for HITL after concatenation. The overall impact of the *Multi-Layer Concatenation* on the training time of algorithms is dependent on the original data sizes, namely the more physical features there are, the more the combined data size will be expanded in comparison to the network data. This increase is of 1.3 billion data point for HITL, 2 Million for ICS flow and 890 Million for SWAT.

These results show that *Multi-Layer Concatenation* led to improvement of detection performances on all datasets, even when extremely high performances are attained with network or physical data, establishing it as a viable and efficient method for leveraging information combination for attack detection in CPS.

9. Discussion

In this section, we evaluate the contribution of proposed MLC scheme with regards to requirements given in section 3, identify its limits, and place it in the context of ML-based attack detection.

Results. The use of *Multi-Layer Concatenation* when compared to the best performance obtained on separate data leads to respectively 0.51%, 1.34% and 16.01% improvements in balanced accuracy for the XGBoost model on SWAT, ICS-Flow and HITL datasets. The magnitudes of the improvements vary depending on the dataset and the detection algorithm used, with an increase in performances observed even when extremely high performances are achieved without using *Multi-Layer Concatenation*, as illustrated in ICS-Flow dataset. These results reinforce the combination of information from physical and network data as being beneficial to ML predictions, as the model-based approach proposed in [12] suggested. The performances of MLP on ICS-flow is the only instance from this study where *Multi-Layer Concatenation* showed no significant improvement, the model performances being rather constant and overall bad regardless of the data. Among potential explanations for this performance is the model being too small or with unsuited layer architecture. The *Multi-Layer Concatenation* showed beneficial effects on tree-based models Decision Tree, Random Forest and XGBoost (Appendix C.6), that are proven to perform better on tabular data than Deep-Learning [14].

Regarding the requirements of Section 3, the increase of performance after applying *Multi-Layer Concatenation* does include a reduction of FPR which aligns with the need for a reduced false alarm rate to reduce alarm fatigue and associated effects. Moreover, it does not impose prerequisites on the model to be used on the merged data, resulting in a neutral impact on explainability, which this approach neither impeded nor addressed. This places *Multi-Layer Concatenation* as a step towards addressing the challenges faced by ML-based detection methods.

Limits. One limit of this approach resides in the augmented size of the merged data, which for network data with billions of rows requires a consequent memory space to store the data after *Multi-Layer Concatenation*. This limit can be overcome with the use of aggregation strategies on network data, which can also be coupled with using the physical data as the base for the merge. A side effect of *Multi-Layer Concatenation* in the case of labelled data is the creation of inconsistencies in the labels from the physical and network data after merging. These inconsistencies have two causes, the first is the delay between the update of network and physical data, which causes the most frequently acquired data to have its label updated before the other, and causes an inconsistency until the least frequently acquired data gets updated. This effect occurs at the start and end of an attack. The second cause is when an attack starts and ends without the least frequently acquired data having received an update. In that case, it will only be labelled in the most frequently updated data. Using the labels from the data with the highest acquisition rate is thus the best solution as they are the ones that are not affected by the described delays.

Taking these limitations into account allows for a pertinent and most beneficial use of *Multi-Layer Concatenation*, for which the accessible mitigation means renders it an accessible solution for improving ML models detection performances on CPS.

10. Conclusions and Perspectives

We propose *Multi-Layer Concatenation* as a novel method for combining the information of physical and network data in the context of the CPS of water distribution networks. By assessing its effect on the performance of reference ML algorithms for cyber-attack detection, compared to using physical and network data separately, we observe a general improvement in the quality of detection on 3 different cyber-physical datasets from the literature. These findings confirm the benefit of combining the information of physical and network data demonstrated in [12]. It opens the way to more research into alternative combining methods and to a more widespread use of these methods to allow for a better understanding of their effects.

Acknowledgements

This work is funded by French ANR under grant ANR-22-CE39-0010 for Correau Project.

Appendix A. Performance Metrics

A.1. Precision:

Precision measures the proportion of true positive predictions among all positive predictions made by the model.

$$Precision = \frac{TP}{TP + FP} \tag{A.1}$$

A.2. Recall or True Positive Rate (TPR):

The recall is the percentage of data positively labelled by the model that are effectively positive.

$$Recall = \frac{TP}{TP + FN} \tag{A.2}$$

A.3. F1-Score:

The F1-Score is the harmonic mean of precision and recall.

$$F1Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{A.3}$$

A.4. True Negative Rate (TNR):

The TNR is the percentage of data negatively labelled by the model that are effectively negative.

$$TNR = \frac{TN}{TN + FP} \tag{A.4}$$

A.5. Accuracy:

The accuracy is the percentage of data correctly labelled by the model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{A.5}$$

A.6. Balanced Accuracy:

The balanced accuracy is the arithmetic mean between TPR and TNR.

$$BalancedAccuracy = \frac{TPR + TNR}{2} \tag{A.6}$$

A.7. Mathews Correlation Coefficient (MCC):

The MCC [18] measures the correlations between the predicted and the true labels. It is regarded as a better single-value representation of all four of the confusion matrix categories [8].

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{A.7}$$

Appendix B. Models Hyperparameters

Model	Specified Hyperparameters	Details
Decision Tree	None	Uses default hyperparameters (SKlearn V1.2.2)
Random Forest	max_depth=3	
XGBoost	tree_method="gpu_hist"	Enables use of GPU
MLP	*t*alpha=0.1, max_iter=500, early_stopping=True, n_iter_no_change=10	

Appendix C. Detailed Results

Table C.4: Detection Performances on Physical Data

Testbed	Classifier	Precision	Recall (TPR)	TNR	Accuracy	F1_score	Balanced Accuracy	MCC	fit_time
SWAT	XGB	**0.9980**	**0.9936**	**0.9997**	**0.9980**	**0.9980**	**0.9949**	**0.9956**	2s
SWAT	RandomForest	0.8035	0.4549	0.9983	0.8478	0.7960	0.6202	0.6421	0s
SWAT	MLP	0.8680	0.5544	0.9854	0.8664	0.8417	0.6532	0.6849	2s
SWAT	DecisionTree	0.9947	0.9909	0.9962	0.9947	0.9947	0.9921	0.9884	0s
ICS_FLOW Physical-all	XGB	**0.9934**	**0.9764**	0.9975	**0.9935**	**0.9934**	**0.9675**	**0.9805**	2s
ICS_FLOW Physical-all	RandomForest	0.6821	0.0516	0.9995	0.8189	0.7412	0.2431	0.2159	0s
ICS_FLOW Physical-all	MLP	0.6548	0.0000	**1.0000**	0.8092	0.7238	0.1667	0.0000	1s
ICS_FLOW Physical-all	DecisionTree	0.9767	0.9414	0.9852	0.9769	0.9767	0.9091	0.9311	0s
ICS_FLOW PLC_bottle	XGB	**0.9410**	0.8261	0.9753	**0.9434**	**0.9411**	0.7865	**0.8433**	0s
ICS_FLOW PLC_bottle	RandomForest	0.6785	0.1470	0.9980	0.8166	0.7393	0.2884	0.3617	0s
ICS_FLOW PLC_bottle	MLP	0.6167	0.0000	**1.0000**	0.7853	0.6909	0.1667	0.0000	0s
ICS_FLOW PLC_bottle	DecisionTree	0.9283	**0.8418**	0.9501	0.9272	0.9277	**0.8076**	0.8057	0s
ICS_FLOW PLC_Water	XGB	**0.9896**	0.9680	0.9948	**0.9897**	**0.9896**	0.9572	**0.9691**	0s
ICS_FLOW PLC_Water	RandomForest	0.7277	0.0504	0.9991	0.8187	0.7409	0.2407	0.2127	0s
ICS_FLOW PLC_Water	MLP	0.6548	0.0000	**1.0000**	0.8092	0.7238	0.1667	0.0000	0s
ICS_FLOW PLC_Water	DecisionTree	0.9879	**0.9710**	0.9918	0.9879	0.9879	**0.9602**	0.9640	0s
HITL	XGB	**0.9769**	**0.9185**	0.9921	**0.9780**	**0.9773**	**0.7602**	**0.9327**	3s
HITL	RandomForest	0.8343	0.1709	**0.9992**	0.8410	0.7906	0.2806	0.3833	0s
HITL	MLP	0.9032	0.6467	0.9677	0.9072	0.9019	0.5616	0.6975	1s
HITL	DecisionTree	0.9647	0.9183	0.9736	0.9631	0.9639	0.7532	0.8900	0s

Table C.5: Detection Performances on Network Data

Testbed	Classifier	Precision	Recall (TPR)	TNR	Accuracy	F1_score	Balanced_accuracy	MCC	fit_time
SWAT	XGB	**0.7416**	0.0533	0.9976	**0.7372**	**0.6383**	**0.2277**	**0.1910**	17s
SWAT	RandomForest	0.6294	0.0109	**0.9999**	0.7272	0.6153	0.2057	0.0923	2m2s
SWAT	MLP	0.5911	**0.0568**	0.9835	0.7298	0.6344	0.2269	0.1452	4m0s
SWAT	DecisionTree	0.6513	0.0525	0.9929	0.7344	0.6367	0.2271	0.1695	12s
ICS_FLOW	XGB	**0.9979**	**0.9977**	**0.9979**	**0.9979**	**0.9979**	**0.9816**	**0.9938**	1s
ICS_FLOW	RandomForest	0.9323	0.7295	0.9760	0.9321	0.9126	0.6303	0.7896	0s
ICS_FLOW	MLP	0.7087	0.0325	0.9899	0.8116	0.7314	0.3071	0.1714	3s
ICS_FLOW	DecisionTree	0.9934	0.9935	0.9934	0.9934	0.9934	0.9478	0.9808	0s
HITL	XGB	**0.8355**	0.5944	0.9987	**0.8718**	0.8152	0.4978	**0.7252**	1m2s
HITL	RandomForest	0.7616	0.5644	**0.9999**	0.8631	0.8070	0.3222	0.7045	6m52s
HITL	MLP	0.7558	0.5371	**0.9999**	0.8545	0.7988	0.3147	0.6838	24m7s
HITL	DecisionTree	0.8077	**0.6013**	0.9933	0.8704	**0.8180**	**0.5044**	0.7199	27s

Table C.6: Detection Performances on Combined Data

Testbed	Classifier	Precision	Recall (TPR)	TNR	Accuracy	F1_score	Balanced_accuracy	MCC	fit_time
SWAT	XGB	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	1m18s
SWAT	RandomForest	0.8045	0.4363	**1.0000**	0.8447	0.7926	0.6172	0.6306	9m15s
SWAT	MLP	0.9695	0.9471	0.9780	0.9695	0.9693	0.9495	0.9329	2h11m52s
SWAT	DecisionTree	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	2m5s
ICS_FLOW Combined-all	XGB	**0.9988**	**0.9966**	0.9994	**0.9988**	**0.9988**	**0.9950**	**0.9966**	3s
ICS_FLOW Combined-all	RandomForest	0.9329	0.7238	0.9768	0.9318	0.9116	0.6209	0.7884	1s
ICS_FLOW Combined-all	MLP	0.6999	0.0124	**0.9998**	0.8084	0.7251	0.1749	0.1007	1s
ICS_FLOW Combined-all	DecisionTree	0.9944	0.9935	0.9946	0.9944	0.9944	0.9621	0.9835	0s
ICS_FLOW Combined-PLC_bottle	XGB	**0.9988**	**0.9966**	**0.9993**	**0.9988**	**0.9988**	**0.9923**	**0.9964**	0s
ICS_FLOW Combined-PLC_bottle	RandomForest	0.9329	0.7357	0.9751	0.9325	0.9134	0.6334	0.7911	0s
ICS_FLOW Combined-PLC_bottle	MLP	0.6747	0.0122	0.9950	0.8076	0.7256	0.1732	0.0946	1s
ICS_FLOW Combined-PLC_bottle	DecisionTree	0.9955	0.9939	0.9959	0.9955	0.9955	0.9676	0.9867	0s
ICS_FLOW Combined-PLC_Water	XGB	**0.9988**	**0.9966**	0.9993	**0.9988**	**0.9988**	**0.9923**	**0.9964**	1s
ICS_FLOW Combined-PLC_Water	RandomForest	0.9329	0.7357	0.9751	0.9325	0.9134	0.6334	0.7911	0s
ICS_FLOW Combined-PLC_Water	MLP	0.6499	0.0000	**1.0000**	0.8061	0.7196	0.1667	0.0000	1s
ICS_FLOW Combined-PLC_Water	DecisionTree	0.9950	0.9916	0.9958	0.9950	0.9950	0.9617	0.9852	0s
HITL	XGB	0.9993	**0.9991**	0.9994	0.9993	0.9993	0.9503	0.9985	1m52s
HITL	RandomForest	0.8318	0.5545	1.0000	0.8600	0.8040	0.3195	0.6971	11m41s
HITL	MLP	0.9401	0.8477	0.9832	0.9406	0.9378	0.6259	0.8747	48m0s
HITL	DecisionTree	**0.9994**	**0.9991**	**0.9995**	**0.9994**	**0.9994**	**0.9836**	**0.9987**	2m11s

References

[1] Abokifa, A.A., Haddad, K., Lo, C.S., Biswas, P., 2017. Detection of Cyber Physical Attacks on Water Distribution Systems via Principal Component Analysis and Artificial Neural Networks , 676–691URL: https://ascelibrary.org/doi/10.1061/9780784480625.063, doi:10.1061/9780784480625.063. publisher: American Society of Civil Engineers.

[2] Aghashahi, M., Sundararajan, R., Pourahmadi, M., Banks, M.K., 2017. Water Distribution Systems Analysis Symposium–Battle of the Attack Detection Algorithms (BATADAL) , 101–108URL: https://ascelibrary.org/doi/10.1061/9780784480595.010, doi:10.1061/9780784480595.010. publisher: American Society of Civil Engineers.

[3] Ahmed, C., Palleti, V., Mathur, A., 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems, pp. 25–28. doi:10.1145/3055366.3055375.

[4] Ahmed Jamal, A., Mustafa Majid, A.A., Konev, A., Kosachenko, T., Shelupanov, A., 2023. A review on security analysis of cyber physical systems using Machine learning. Materials Today: Proceedings 80, 2302–2306. URL: https://www.sciencedirect.com/science/article/pii/S2214785321047118, doi:10.1016/j.matpr.2021.06.320.

[5] Brentan, B.M., Campbell, E., Lima, G., Manzi, D., Ayala-Cabrera, D., Herrera, M., Montalvo, I., Izquierdo, J., Luvizotto, E., 2017. On-Line Cyber Attack Detection in Water Networks through State Forecasting and Control by Pattern Recognition , 583–592URL: https://ascelibrary.org/doi/10.1061/9780784480625.054, doi:10.1061/9780784480625.054. publisher: American Society of Civil Engineers.

[6] Chandy, S.E., Rasekh, A., Barker, Z.A., Campbell, B., Shafiee, M.E., 2017. Detection of Cyber-Attacks to Water Systems through Machine-Learning-Based Anomaly Detection in SCADA Data , 611–616URL: https://ascelibrary.org/doi/10.1061/9780784480625.057, doi:10.1061/9780784480625.057. publisher: American Society of Civil Engineers.

[7] Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. pp. 785–794. URL: https://dl.acm.org/doi/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

[8] Chicco, D., Jurman, G., 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21, 6. URL: https://doi.org/10.1186/s12864-019-6413-7, doi:10.1186/s12864-019-6413-7.

[9] Deb, S., Claudio, D., 2015. Alarm fatigue and its influence on staff performance. IIE Transactions on Healthcare Systems Engineering 5, 183–196. URL: https://doi.org/10.1080/19488300.2015.1062065, doi:10.1080/19488300.2015.1062065. publisher: Taylor & Francis _eprint: https://doi.org/10.1080/19488300.2015.1062065.

[10] Dehlaghi-Ghadim, A., Helali Moghadam, M., Balador, A., Hansson, H., 2023. Anomaly Detection Dataset for Industrial Control Systems. IEEE Access PP, 1–1. doi:10.1109/ACCESS.2023.3320928.

[11] Faramondi, L., Flammini, F., Guarino, S., Setola, R., 2021. A Hardware-in-the-Loop Water Distribution Testbed Dataset for Cyber-Physical Security Testing. IEEE Access 9, 122385–122396. doi:10.1109/ACCESS.2021.3109465. conference Name: IEEE Access.

[12] Faramondi, L., Flammini, F., Guarino, S., Setola, R., 2023. A hybrid behavior- and Bayesian network-based framework for cyber–physical anomaly detection. Computers and Electrical Engineering 112, 108988. URL: https://www.sciencedirect.com/science/article/pii/S0045790623004123, doi:10.1016/j.compeleceng.2023.108988.

[13] Fradkov, A.L., 2020. Early History of Machine Learning. IFAC-PapersOnLine 53, 1385–1390. URL: https://www.sciencedirect.com/science/article/pii/S2405896320325027, doi:10.1016/j.ifacol.2020.12.1888.

[14] Grinsztajn, L., Oyallon, E., Varoquaux, G., 2022. Why do tree-based models still outperform deep learning on tabular data? URL: http://arxiv.org/abs/2207.08815, doi:10.48550/arXiv.2207.08815. arXiv:2207.08815 [cs, stat].

[15] Hassanzadeh, A., Rasekh, A., Galelli, S., Aghashahi, M., Taormina, R., Ostfeld, A., Banks, M.K., 2020. A Review of Cybersecurity Incidents in the Water Sector. Journal of Environmental Engineering 146, 03120003. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29EE.1943-7870.0001686, doi:10.1061/(ASCE)EE.1943-7870.0001686. publisher: American Society of Civil Engineers.

[16] Housh, M., Ohar, Z., 2017. Model Based Approach for Cyber-Physical Attacks Detection in Water Distribution Systems , 727–736URL: https://ascelibrary.org/doi/10.1061/9780784480625.067, doi:10.1061/9780784480625.067. publisher: American Society of Civil Engineers.

[17] Mashhadi, N., Shahrour, I., Attoue, N., El Khattabi, J., Aljer, A., 2021. Use of Machine Learning for Leak Detection and Localization in Water Distribution Systems. Smart Cities 4, 1293–1315. URL: https://www.mdpi.com/2624-6511/4/4/69, doi:10.3390/smartcities4040069. number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[18] Matthews, B.W., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica Et Biophysica Acta 405, 442–451. doi:10.1016/0005-2795(75)90109-9.

[19] Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A.Y., Tari, Z., 2023. Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. IEEE Communications Surveys & Tutorials 25, 1775–1807. URL: https://ieeexplore.ieee.org/abstract/document/10136827, doi:10.1109/COMST.2023.3280465. conference Name: IEEE Communications Surveys & Tutorials.

[20] Murillo, A., Taormina, R., Tippenhauer, N.O., Galelli, S., 2023a. High-Fidelity Cyber and Physical Simulation of Water Distribution Systems. II: Enabling Cyber-Physical Attack Localization. Journal of Water Resources Planning and Management 149, 04023010. URL: https://ascelibrary.org/doi/10.1061/JWRMD5.WRENG-5854, doi:10.1061/JWRMD5.WRENG-5854. publisher: American Society of Civil Engineers.

[21] Murillo, A., Taormina, R., Tippenhauer, N.O., Salaorni, D., van Dijk, R., Jonker, L., Vos, S., Weyns, M., Galelli, S., 2023b. High-Fidelity Cyber and Physical Simulation of Water Distribution Systems. I: Models and Data. Journal of Water Resources Planning and Management 149, 04023009. URL: https://ascelibrary.org/doi/10.1061/JWRMD5.WRENG-5853, doi:10.1061/JWRMD5.WRENG-5853. publisher: American Society of Civil Engineers.

[22] Nader, P., Honeine, P., Beauseroy, P., 2016. Detection of cyberattacks in a water distribution system using machine learning techniques, in: 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC),

pp. 25–30. URL: https://ieeexplore.ieee.org/abstract/document/7470786?casa_token=KpyKLaQg75cAAAAA:yDCpIDpYCjI-AyxALDB4wSR4ytC6sllJbzOCvAE2JQnt8v_W_pt8RbepdHDB3CfJulzJHNA_1xR9, doi:10.1109/ICDIPC.2016.7470786.

[23] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A., 2019. A survey of network-based intrusion detection data sets. Computers & Security 86, 147–167. URL: https://www.sciencedirect.com/science/article/pii/S016740481930118X, doi:10.1016/j.cose.2019.06.005.

[24] Salvatore Stolfo, W.F., 1999. KDD Cup 1999 Data. URL: https://archive.ics.uci.edu/dataset/130, doi:10.24432/C51C7N.

[25] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A., Chen, S., Liu, D., Li, J., 2020a. Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity. Energies 13, 2509. URL: https://www.mdpi.com/1996-1073/13/10/2509, doi:10.3390/en13102509. number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[26] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A., Xu, M., 2020b. A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. IEEE Access 8, 222310–222354. URL: https://ieeexplore.ieee.org/document/9277523/, doi:10.1109/ACCESS.2020.3041951.

[27] Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security 31, 357–374. URL: https://www.sciencedirect.com/science/article/pii/S0167404811001672, doi:10.1016/j.cose.2011.12.012.

[28] Sikder, M.N.K., Nguyen, M.B.T., Elliott, E.D., Batarseh, F.A., 2023. Deep H2O: Cyber attacks detection in water distribution systems using deep learning. Journal of Water Process Engineering 52, 103568. URL: https://www.sciencedirect.com/science/article/pii/S2214714423000855, doi:10.1016/j.jwpe.2023.103568.

[29] Strom, B., Applebaum, A., Miller, D., Nickels, K., Pennington, A., Thomas, C., 2020. MITRE ATT&CK: Design and Philosophy URL: https://www.mitre.org/news-insights/publication/mitre-attck-design-and-philosophy.

[30] Sun, C., Cembrano, G., Puig, V., Meseguer, J., 2018. Cyber-Physical Systems for Real-Time Management in the Urban Water Cycle, in: 2018 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), pp. 5–8. URL: https://ieeexplore.ieee.org/abstract/document/8434710, doi:10.1109/CySWater.2018.00008.

[31] Taormina, R., Galelli, S., Tippenhauer, N.O., Salomons, E., Ostfeld, A., Eliades, D.G., Aghashahi, M., Sundararajan, R., Pourahmadi, M., Banks, M.K., Brentan, B.M., Campbell, E., Lima, G., Manzi, D., Ayala-Cabrera, D., Herrera, M., Montalvo, I., Izquierdo, J., Luvizotto, E., Chandy, S.E., Rasekh, A., Barker, Z.A., Campbell, B., Shafiee, M.E., Giacomoni, M., Gatsis, N., Taha, A., Abokifa, A.A., Haddad, K., Lo, C.S., Biswas, P., Pasha, M.F.K., Kc, B., Somasundaram, S.L., Housh, M., Ohar, Z., 2018. Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks. Journal of Water Resources Planning and Management 144, 04018048. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000969, doi:10.1061/(ASCE)WR.1943-5452.0000969. publisher: American Society of Civil Engineers.

[32] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/5356528, doi:10.1109/CISDA.2009.5356528. iSSN: 2329-6275.

[33] Tuptuk, N., Hazell, P., Watson, J., Hailes, S., 2021. A Systematic Review of the State of Cyber-Security in Water Systems. Water 13, 81. URL: https://www.mdpi.com/2073-4441/13/1/81, doi:10.3390/w13010081. number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[34] Winkler, D., Haltmeier, M., Kleidorfer, M., Rauch, W., Tscheikner-Gratl, F., 2018. Pipe failure modelling for water distribution networks using boosted decision trees. Structure and Infrastructure Engineering 14, 1402–1411. URL: https://doi.org/10.1080/15732479.2018.1443145, doi:10.1080/15732479.2018.1443145. publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15732479.2018.1443145.

[35] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C., 2018. Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access 6, 35365–35381. URL: https://ieeexplore.ieee.org/abstract/document/8359287, doi:10.1109/ACCESS.2018.2836950. conference Name: IEEE Access.