



Enhancing Critical Infrastructure Security: Unsupervised Learning Approaches for Anomaly Detection

Andrea Pinto¹ · Luis-Carlos Herrera² · Yezid Donoso¹ · Jairo A. Gutierrez³

Received: 14 May 2024 / Accepted: 23 August 2024
© The Author(s) 2024

Abstract

Traditional security detection methods face challenges in identifying zero-day attacks in critical infrastructures (CIs) integrated with the industrial internet of things (IIoT). These attacks exploit unknown vulnerabilities and are difficult to detect due to their connection to physical systems. The integration of legacy ICS networks with modern computing and networking technologies has significantly expanded the attack surface, making these systems more susceptible to cyber-attacks. Despite existing security measures, attackers continually find ways to breach these operating networks. Anomaly detection systems are critical in protecting these CIs from current cyber threats. This study investigates the effectiveness of unsupervised anomaly detection models in detecting operational anomalies that could lead to cyber-attacks, thereby disrupting and negatively impacting quality of life. We preprocess the data with a focus on cybersecurity and chose the SWAT dataset because it accurately represents the types of attack vectors that critical infrastructures commonly encounter. We evaluated the performance of isolation forest (IF), local outlier factor (LOF), one-class SVM (OCSVM), and Autoencoder algorithms—trained exclusively on normal data—in enhancing cybersecurity within IIoT environments. Our comprehensive analysis includes an assessment of each model's detection capabilities. The findings highlight the VAE-LSTM model's potential to identify cyber-attacks within seconds in a high-frequency dataset, suggesting near real-time detection capability. The final model combines the reconstruction ability of the variational autoencoder (VAE) with regularization using the Kullback–Leibler divergence, reflecting the non-Gaussian nature of industrial system data. Our model successfully detected 23 out of 26 attack scenarios in the SWAT dataset, demonstrating its effectiveness in improving the security of IIoT-based CIs.

Keywords Unsupervised learning · IIoT · Cybersecurity · Critical infrastructures · Anomaly detection

✉ Andrea Pinto
ya.pinto10@uniandes.edu.co

Luis-Carlos Herrera
luis.herrera@fac.mil.co

Yezid Donoso
ydonoso@uniandes.edu.co

Jairo A. Gutierrez
jairo.gutierrez@aut.ac.nz

¹ Systems and Computer Engineering Department, School of Engineering, Universidad de los Andes, 111711 Bogotá, Colombia

² Colombian Defense Ministry's CSIRT, 11132 Bogotá, Colombia

³ Networking and Security Research Centre, Department of Computer Science and Software Engineering, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand

1 Introduction

The cybersecurity of critical infrastructures (CIs) is a national security concern [1]. The exponential growth of cyberattacks targeting these infrastructures, not only for political but also for economic motives [2], has highlighted the need for new and innovative measures to protect the adequate functioning of vital systems, such as communications, hydroelectric, or transportation systems [3]. Zero-day attacks are the most difficult to detect, manage, and respond to; these attacks can take advantage of weaknesses that are not yet known or can mimic normal operational functioning. Both scenarios constitute challenges for cybersecurity systems. Traditional intrusion detection systems (IDS) such as signature-based, and misuse-based systems rely on known attack patterns and predefined rules to detect threats; thus, are not effective against zero-day attacks [4]. Moreover, security measures learned from information technology

(IT) infrastructures may not fully meet the security needs of CIs. Contrary to the typical emphasis on data confidentiality in general IT, in industrial control systems (ICS), the paramount importance lies in ensuring real-time availability and integrity for the smooth operation of industrial plants [5].

Over the last few decades, CIs have become interconnected through new technologies that increase their effectiveness, reduce operational costs, and enhance management capabilities [6]. Thus, isolating operational technology is an obsolete security measure that does not work for current architectures [5]. The separation between IT systems and operational technology (OT) systems has become blurred, making the attack surface bigger [7]. Additionally, the intentions of attackers targeting CIs could have a physical impact [8]; for instance, changing the state of one of the many actuators could negatively affect the production of a hydroelectric plant. Therefore, there are specific kind of threats against CIs, which must be tackled using adequate measures [9].

Research in the field of ICS has intensified due to the growing concern surrounding the security of CIs [10]. This complex network of systems includes components such as Supervisory Control and Data Acquisition (SCADA) systems, Programmable Logic Controllers (PLCs), Remote I/O (RIO) units, sensors, and actuators [11]. Many successful attacks have occurred since the classical Stuxnet attack, which targeted Iran's nuclear program. Specifically, it targeted SCADA systems used to control and monitor industrial processes, such as those found in nuclear facilities and power plants [12]. The attack aimed to manipulate the programmable logic controllers (PLCs) within these systems to sabotage the machinery they controlled, all without raising suspicions or triggering alarms. These threats have evolved reaching the current state in which even national cybersecurity organizations such as: the UK's National Cyber Security Centre (NCSC), in their NCSC Annual Review 2023, warned about the significant cyber threats faced by the nation's CI [13].

An anomaly is a value or outcome that deviates from the expected or normal value [4]. From the cybersecurity perspective, anomalies serve as crucial early indicators of potential cyber-attacks. These deviations, whether evident as unusual patterns in network traffic, physical devices, irregularities in system logs, or atypical user behavior, often signify the presence of malicious activity within a system. Moreover, anomalies offer invaluable insights into the tactics and techniques employed by adversaries, empowering security teams to proactively enhance their defense strategies [14].

Given that normal operational behavior constitutes the baseline knowledge for CI operators, any departure from this norm warrants scrutiny as it may indicate a potential cyber-attack. Building upon this understanding, the present study poses the following research question: How effective are machine learning-based anomaly detection models, trained

solely on high-frequency normal data representing physical systems, in improving the capacity of detecting zero-day attacks targeting CIs?

The key contributions of this work to the field of cybersecurity for critical infrastructures are outlined as follows:

- A preprocessing method was devised to extract meaningful information from industrial devices (sensors and actuators) and address data science challenges such as different distributions in training and testing sets, as well as a high number of categorical variables. These characteristics are particularly prevalent in data from industrial physical devices.
- The modeling and evaluation of six different unsupervised algorithms (IF, LOF, One-Class SVM, and various autoencoders) were conducted to identify which model achieves better results in detecting anomalous data in operating systems, based on learning the normal operational behavior. This approach enhances the possibility of identifying zero-day attacks.
- A generative deep learning model has been proposed to identify anomalous behavior in operational infrastructures, with a particular focus on detecting anomalies linked to cyber-attacks targeting physical devices. The experimental evaluation demonstrates the viability of our model in detecting 23 out of 26 cyber-attacks selected from the SWAT dataset, with better generalization compared to existing methods and using smaller sequence windows of 10 s.

The rest of the work is organized into the following sections: Sect. 2, presents the methodology used to develop this research; Sect. 3, explains some of the most notable works that have been developed in the research area; Sect. 4, details the series of experiments conducted to test unsupervised algorithms for detecting anomalies; Sect. 5, presents the results of the tested algorithms and analyzes their detection capacity; finally, Sect. 6, concludes the work and the discusses future directions.

2 Methodology

The design science research methodology (DSRM) was selected to systematically resolve the research question of this work. This methodology offers a structured framework for researchers to construct and assess artifacts, with the goal of producing actionable insights that benefit both theoretical understanding and practical application in addressing real-world challenges [15]. Details of the methodology are in Fig. 1. The process begins with problem identification and motivation, followed by the analysis and comparison of the obtained results.

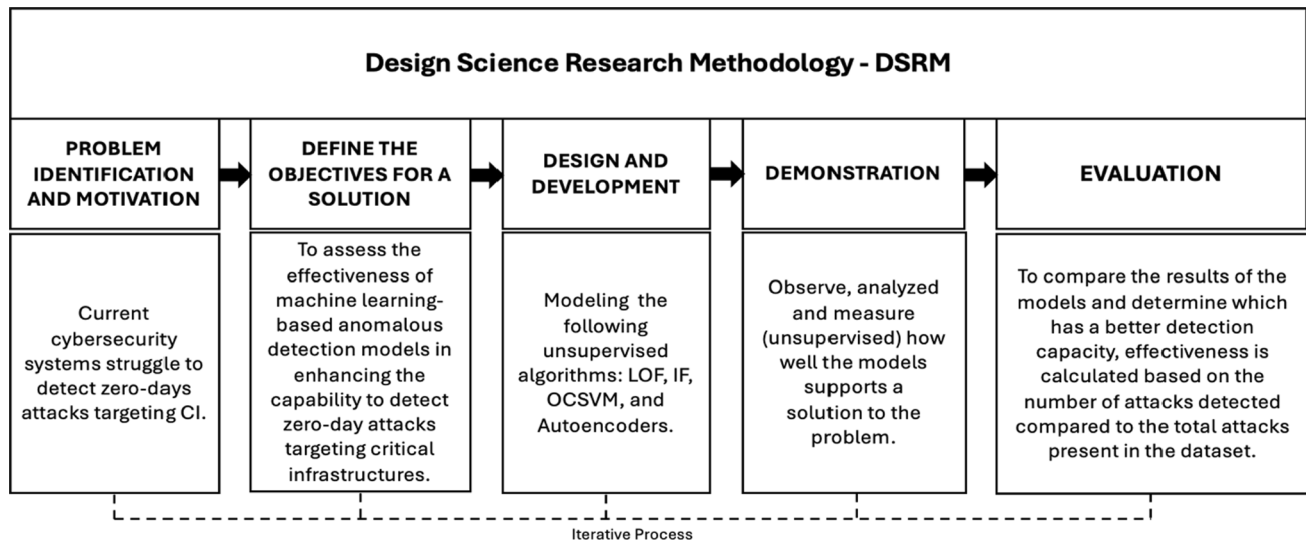


Fig. 1 DRSRM methodology

The DSRM is highly beneficial for cybersecurity research, primarily due to its structured yet flexible framework that emphasizes practical and theoretical contributions simultaneously. In the realm of cybersecurity, where the landscape of threats continuously evolves, DSRM's problem-centric approach ensures that research efforts are directly aligned with current and emerging security challenges. This methodology encourages the creation and iterative refinement of artifacts such as models, methods, and tools, which are crucial for detecting and responding to cyber threats. Moreover, DSRM mandates rigorous evaluation of these artifacts, a process vital for assessing their effectiveness in real-world scenarios. This not only enhances the practical applications of research outcomes but also contributes to the theoretical foundations of cybersecurity, ensuring that developments are both innovative and applicable.

3 Background and Related Work

In [5], the authors developed a CNN-based method for detecting cyberattacks in Industrial Control Systems (ICS). They solely utilized sensor data to train the model and excluded certain sensors, namely AIT201, AIT203, and PIT502, due to discrepancies between the training and test data. The elimination of features with different distributions in the training and test sets is a common preprocessing step in data preparation. Nevertheless, the observed behavior of these sensors suggests that some devices did not stabilize after experiencing a cyber-attack, indicating that the systems did not return to normal behavior post-attack. This type of behavior can be identified by anomaly detection systems and warrants a deeper analysis. Furthermore, using a CNN

for univariate auto-regression of signals has limitations, as it cannot capture the interdependencies between physical devices. Attacks targeting multiple processes or devices that were excluded from the training set may be more challenging to detect. Although the results are promising and indicate a positive detection of attacks, it should be noted that these detections did not always correlate with a physical impact. In [7], the authors eliminated 11 features to build their model. However, from a cybersecurity perspective, any feature representing a physical device—whether a sensor or actuator—can potentially serve as an attack surface and thus should not be excluded from the training set. Consequently, attacks targeting devices excluded from the modeling process, such as attack number 24, were not detected by the authors' method.

Another artificial neural network approach is explained in [3]: a One-Class Neural Network for Anomaly Detection in Water Treatment Systems. The authors demonstrated superior recall values in 15 out of the 36 attack scenarios. However, 16 attack scenarios had a recall value of zero, and six had values under 0.08. Consequently, the model failed to correctly identify many of the positive instances representing cyberattacks. Moreover, no noise reduction was applied to the dataset, and noise in real-world scenarios is inevitable and needs to be mitigated to achieve good anomaly detection performance. The authors also introduced a regularization term, but the range of hyperparameters was manually specified and tuned. This approach can introduce bias and limit the model's potential for optimization.

In [8], the authors propose a causality-inspired unsupervised learning approach for detecting cyberattacks in water treatment plants. The proposed method utilizes causal inference to create a resilient anomaly score in two stages. Initially, minimal domain knowledge through causal models

assists in pinpointing crucial interdependencies within the system, while univariate models aid in independently learning the typical behavior of the system's components. However, in the univariate scenario, it is impossible to capture any temporal connections among the physical devices, nor dependencies or relationships, let alone uncover any underlying causality within the data. Therefore, the model could have an incomplete understanding of system behavior. A different approach can be observed in [16], where the authors developed an intrusion detection technique for ICS that relies on an enhanced version of comparative learning, specifically SimCLR. Initially, an unlabeled dataset was utilized to train a feature extraction network within the SimCLR framework. Subsequently, a linear classification layer is appended to this trained network model. Finally, a small subset of labeled data is employed to supervise the training process and fine-tune the model parameters.

In scenarios where critical systems are distributed across various locations, federated learning (FL) emerges as a computational framework for decentralized machine learning. It enables disparate entities to collaborate on training a singular global model without the need to share raw data. This approach ensures the confidentiality of sensitive information within critical systems. As is shown in [17], FL models are trained locally on the devices, and only their parameters are sent to the central server. The data were split vertically by technological processes in the SWAT dataset. However, FL can indirectly uncover some level of interdependencies between data sources due to the nature of the collaborative training process.

A Lightweight Long Short-Term Memory Variational Auto-Encoder was implemented in [18], and the authors reported detecting 82% of anomalies present in the SWAT dataset. They applied the K-S test to verify the similarity between the probability distributions of the training and testing data. As a result, the authors eliminated 15 features—AIT201, AIT202, AIT203, P201, AIT401, AIT402, AIT501, AIT502, AIT503, AIT504, FIT504, PIT501, PIT502, PIT503—even those targeted in some attack scenarios, such as attack scenario number 38 with AIT401 and AIT502. Thus, if the model detects anomalies during the time when an attack scenario was launched, it is done through dependencies among other physical devices. However, to bridge the gap between academic research and models that can be implemented in real-world scenarios, it is necessary to demonstrate the generalization capabilities of machine learning models. This would entail retaining the features that represent the normal system behavior. A similar case is depicted in [11], where an LSTM neural network was modeled, and 22 features representing sensors or actuators were excluded from the process due to having zero variance, a K-S test result higher than 0.25, or mismatched distributions between training and testing data. In the latter case, the

authors detected 23 out of 28 cyberattacks after excluding those attacks that did not have a physical impact or were unsuccessful. Furthermore, they utilized a window of 120 s to construct new features using Fourier Transformation. As a result, the model would detect an attack at least two minutes after it began. Another approach that implemented an LSTM auto-encoder for detecting anomalies in industrial control systems (ICS) is proposed in [9]. In this work, the authors also introduced a statistical feature extraction method for both industrial network data and external networks, yielding promising results in attack detection through network traffic data. However, the performance of the deep auto-encoder-based long short-term memory (LSTM) model is heavily dependent on the quality of the input features derived from raw network traffic data. Our proposal differentiates itself by utilizing data typical of cyber-physical systems (CPS), such as sensor and actuator data. Consequently, our preprocessing includes considerations specific to these data types, such as signal frequency, to better capture the nuances and characteristics of the CPS environment. Additionally, comparisons are not straightforward, as a model may perform exceptionally well on one dataset and poorly on another. Therefore, enhancing the model's generalization capacity is essential for achieving robust performance across diverse datasets.

An unsupervised dual variational generative adversarial model named MST-DVGAN is proposed in [19]. Their model identifies anomalies that lie either close to normal samples in distribution or near the latent dimension manifold of the normal data cluster in the embedding space. They designed a contrastive module and incorporated two augmented losses to increase the reconstruction error gaps between abnormal and normal data by applying the contrastive constraint on the embedded space. Additionally, the authors utilized an LSTM-based variational inference network with a window of 30 s and applied PCA to extract the principal components resolutions. Particularly, in the SWAT dataset, the authors report achieving an F1 score of 79.87, a recall of 66.93, a precision of 99.0, and an accuracy of 94.84.

4 Experiments

The overview of the suggested approach is shown in Fig. 2. For the anomaly detection process, first, the dataset was selected and downloaded from the official source. Next, preprocessing steps were applied to the SWAT dataset. Then, LOF, IF, OCSVM, and Autoencoder algorithms were used to create anomaly detection models, with all training performed using normal data. Finally, the capacity to detect anomalies was analyzed using data that contains both normal and anomalous data. Each of these fundamental steps is presented in the following subsections.

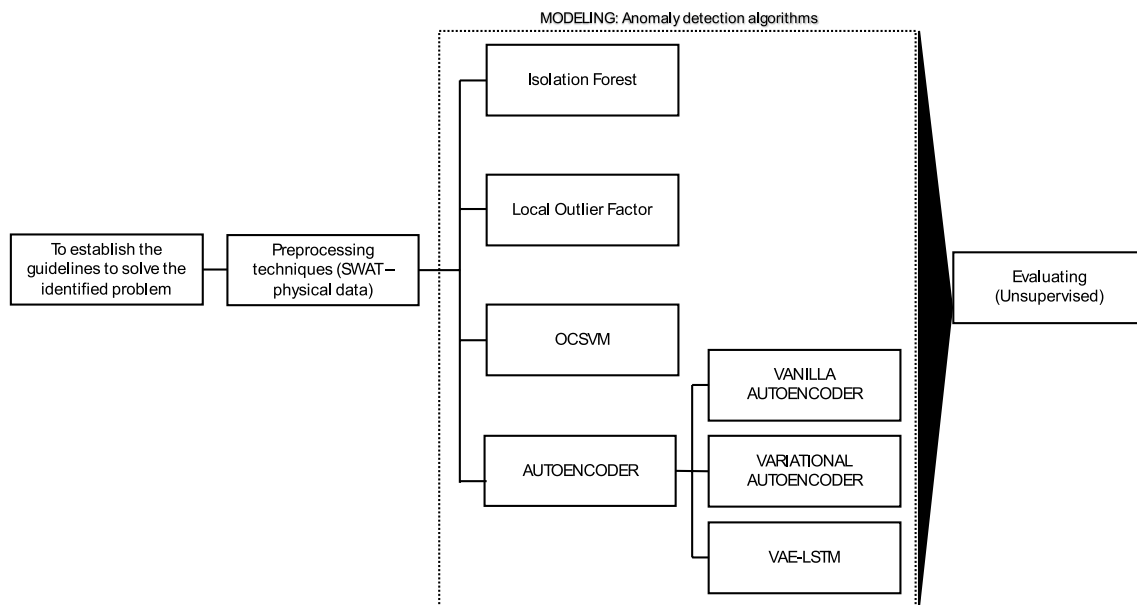


Fig. 2 Overview of ML modeling

4.1 Datasets

Analysis and evaluations were conducted on a total of ten (10) datasets. The evaluated datasets were as follows: NGIDS-DS [20], TON IoT [21], MQTT-IOT-IDS [22], X-IIoTID [23], Edge-IIoTset [24], NF-UQ-NIDS-v2 [25], MedBIoT [26], SWAT [27], WADI,¹ HAI Security Dataset [28]. The SWAT dataset was selected due to the following reasons: (1) it was collected from a scaled real CIs—a hydroelectric plant; (2) its infrastructure is industrial, including sensors and actuators; (3) it is well documented; (4) it contains both normal data and data from cyber-attacks; and (5) the tactics, techniques, and procedures (TTPs) used to attack the hydroelectric plant are a suitable representation of cyberattacks targeting CIs with zero-day vulnerabilities.

The SWAT is a dataset collected from a water treatment plant developed by the Singapore University of Technology and Design (SUTD). This represents a contemporary six-stage water treatment process, referred to as P1 through P6, mirroring an actual treatment facility. This setup encompasses stages such as raw water supply, chemical dosing, ultrafiltration, dichlorination, reverse osmosis, and backwash processes [9]. Generally, the attack surfaces are sensors and actuators, and the attack durations vary from a few minutes to hours. The stabilization time of the hydroelectric plant after a cyberattack also varies depending on the attack's intent, ranging from a shorter duration if the attack aims to alter flow rates to a longer duration if it intends to disrupt the

plant's production. Additionally, the processes are interconnected, and depending on the preceding steps; for instance, the attack on one sensor could impact the entire process or subsequent processes. Therefore, a multivariate approach is necessary to consider the correlations among the devices.

The SWAT dataset includes network traffic information and sensor data (51 sensors, actuators, and PLC control devices, among others). In its initial version, data was collected over an 11-day period, with 7 days of normal operation and 4 days of attack scenarios. These attack scenarios, totaling 41, were designed to simulate a variety of cyber-physical attacks on the water treatment plant. The list of attacks is in Table 1. However, five attacks (attacks number 5, 9, 12, 15, and 18) had no physical impact. Attack number 4, targeting a motorized valve named MV-504, could not be analyzed as information about this valve was not included in the official dataset. Moreover, specific attacks, such as attacks numbered 24 and 34, had either no or minimal impact on system performance. Additionally, attacks numbered 13, 14, and 29 were unsuccessful. Finally, cyber-attacks 6, 19, 20, and 38 targeted unused chemical sensors. Although these attacks were expected to impact other sensors, various malfunctions prevented this from occurring, thereby hindering the detection of these cyberattacks. Thus, there are a total of 26 attacks to be detected.

4.2 Pre-processing

The A2 version of SWAT is a high-frequency time-series dataset containing 51 features 25 sensors and 26 actuators. It comprises 449,919 samples collected during 4 days of

¹ https://itrust.sutd.edu.sg/itrust-labs_datasets/.

Table 1 Cyberattacks on SWAT dataset

Attack number	Attack point	Attack	Additional information
1	MV-101	Open MV-101	
2	P-102	Turn on P-102	
3	LIT-101	Increase LIT-101 by 1 mm every second	
4	MV-504	Open MV-504	Information about this valve was not included in the official dataset
5	No physical impact attack		
6	AIT-202	Set value of AIT-202 as 6	Malfunctions prevented sensors impact
7	LIT-301	Water level increased above HH	
8	DPIT-301	Set value of DPIT as > 40 kpa	
9	No Physical Impact Attack		
10	FIT-401	Set value of FIT-401 as < 0.7	
11	FIT-401	Set value of FIT-401 as 0	
12	No Physical Impact Attack		
13	MV-304	Close MV-304	Unsuccessful cyberattack
14	Mv-303	Do not let MV-303 open	Unsuccessful cyberattack
15	No physical impact attack		
16	LIT-301	Decrease water level by 1 mm each second	
17	MV-303	Do not let MV-303 open	
18	No physical impact attack		
19	AIT-504	Set value of AIT-504 to 16 uS/cm	Malfunctions prevented sensors impact
20	AIT-504	Set value of AIT-504 to 255 uS/cm	Malfunctions prevented sensors impact
21	MV-101, LIT-101	Keep MV-101 on continuously; Value of LIT-101 set as 700 mm	
22	UV-401, AIT-502, P-501	Stop UV-401; Value of AIT502 set as 150; Force P-501 to remain on	
23	P-602, DIT-301, MV-302	Value of DPIT-301 set to > 0.4 bar; Keep MV-302 open; Keep P-602 closed	
24	P-203, P-205	Turn of P-203 and P-205	No impact on system
25	LIT-401, P-401	Set value of LIT-401 as 1000; P402 is kept on	
26	P-101, LIT-301	P-101 is turned on continuously; Set value of LIT-301 as 801 mm	
27	P-302, LIT-401	Keep P-302 on continuously; Value of LIT401 set as 600 mm till 1:26:01	
28	P-302	Close P-302	
29	P-201, P-203, P-205	Turn on P-201; Turn on P-203; Turn on P-205	Unsuccessful cyberattack
30	LIT-101, P-101, MV-201	Turn P-101 on continuously; Turn MV-101 on continuously; Set value of LIT-101 as 700 mm; P-102 started itself because LIT301 level became low	
31	LIT-401	Set LIT-401 to less than L	
32	LIT-301	Set LIT-301 to above HH	
33	LIT-101	Set LIT-101 to above H	
34	P-101	Turn P-101 off	No impact on system
35	P-101; P-102	Turn P-101 off; Keep P-102 off	
36	LIT-101	Set LIT-101 to less than LL	
37	P-501, FIT-502	Close P-501; Set value of FIT-502 to 1.29 at 11:18:36	
38	AIT-402, AIT-502	Set value of AIT402 as 260; Set value of AIT502 to 260	Malfunctions prevented sensors impact
39	FIT-401, AIT-502	Set value of FIT-401 as 0.5; Set value of AIT-502 as 140 mV	
40	FIT-401	Set value of FIT-401 as 0	
41	LIT-301	Decrease value by 0.5 mm per second	

attacks and 495,000 samples representing normal plant operation. According to the official documentation, the first six hours of traffic were excluded because it is the time required for the hydroelectric plant to reach a stabilization point. According to the analysis, AIT201, AIT203, P301, AIT401, AIT402, AIT501, AIT502, AIT503, PIT502 need a longer period of stabilization, then additional 98,200 samples were eliminated.

Although a common practice is to eliminate features with zero or low variance in the training set, this work chooses to retain them. From a cybersecurity perspective, all features can potentially become attack surfaces. Therefore, the system should learn their normal behavior, even if it remains constant. Most of the sensor data exhibit non-Gaussian distributions, and there are differences in distribution between the train and test sets for some features. Particularly, sensor AIT201 shows the most pronounced disparities in distribution. Additionally, `StandardScaler()` was applied to the numerical variables.

In ICS datasets, features often exhibit correlations, reflecting the interconnected nature of the system's components. For instance, when a sensor indicates that a specific water tank has reached its capacity, it is common for the pump regulating the tank's input to cease operation or for the pump governing the tank's output to activate. Such correlations are not only typical but also beneficial in the context of anomaly detection, as they help establish the system's normal behavior patterns. Thus, no features were excluded from the modeling process because of correlation analysis.

Two different sets were created: first, the original features underwent preprocessing, and second, the numerical features -25 sensors- were analyzed, and seasonal features were identified using autocorrelation. These seasonal features served as the basis for creating: (1) Rolling Window Statistics Features (RWSF)—mean, median, standard deviation, maximum and minimum values—for smoothed representation of the data and noise reduction, and (2) Time-shifted Features (TF) to capture temporal dependencies and trends in time-series data. The second dataset comprises a total of 115 features, with 51 being the original features, 40 being RWSF extracted with windows of 120 s, and 15 being TF using lags of 10, 20, and 30 s.

4.3 Anomaly Detection Using Unsupervised Algorithms

The training procedures were developed on a server with the following configuration: Intel(R) Xeon(R) Silver 4310 CPU @ 2.10 GHz; operation system: Ubuntu 22.04.3 LTS and NVIDIA-SMI 525.147.05; Driver Version: 525.147.05 and CUDA Version: 12.0. The LOF, IF, and OCSVM models were trained using scikit-learn 1.3.0, with Optuna 3.6.0 employed for hyperparameter optimization. For deep neural

networks, TensorFlow 2.11.0 and Keras 2.11.0 were utilized, in conjunction with Python 3.10.12.

The following algorithms were used to model anomaly detectors: LOF, IF, OCSVM, and Autoencoders. All of them were trained only with normal data representing the operational behavior of the hydroelectric plant. They were then tested on the dataset containing 41 attacks, with the list of attacks shown in Table 1. In the case of the Autoencoders, different versions were modeled, including Vanilla Autoencoder, Variational Autoencoder (VAE) with dense layers, and Variational Autoencoder with Long Short-Term Memory (LSTM) layers. For all models, hyperparameter search was implemented. The details of the implementation are shown in Table 2.

LOF operates under the assumption that anomalies are likely to be situated in regions of lower density compared to their neighbors. Trained specifically for anomaly detection, the LOF algorithm computes a score for each data point based on its local density compared to that of its neighbors. Instances with substantially lower local densities relative to their neighbors are assigned higher anomaly scores, indicating a higher likelihood of being anomalous. Because, ground truth labels are not used, the contamination parameter and predict method cannot be applied. Therefore, involves analyzing the distribution of anomaly scores, which are shown in Fig. 3.

The IF algorithm diverges from traditional techniques, which typically rely on profiling normal instances based on distance or density metrics. Instead, IF focuses on the direct isolation of anomalies, exploiting the principle that anomalies are both few and exhibit distinct characteristics compared to normal instances. The algorithm constructs a series of isolation trees by randomly selecting a feature and a corresponding split value within the feature's range for each tree. This process results in anomalous instances being isolated at shorter path lengths from the root of the tree, facilitating a quicker and more efficient detection. Consequently, when an unseen instance is presented for anomaly detection, its path length to reach a leaf node within the isolation tree is measured and normalized. Instances with shorter path lengths are considered anomalies, indicative of being significantly different from most normal data points.

The Anomaly Score Distribution from IF model, shown in Fig. 4, enables the assessment of the range and pattern of abnormality in the data. Various thresholds were tested to gain insights into the algorithm's behavior and achieve a balance between False Positives and False Negatives. In this context, instances with the most negative anomaly scores indicate a higher likelihood of being anomalous.

OCSVM is specifically designed for scenarios where only normal data is available for training, this algorithm endeavors to identify a hyperplane that encapsulates the normal data instances within a high-dimensional feature space

Table 2 Hyperparameter search

No	Model	Hyperparameters search	Selected model
1	LOF (Novelty = True)	<pre> trial.suggest_int('n_neighbors', 5, 20) trial.suggest_float('contamination', 0.01, 0.1) trial.suggest_int('leaf_size', 10, 50) trial.suggest_categorical('metric', ['euclidean', 'manhattan']) trial.suggest_categorical('algorithm', ['auto', 'ball_ tree', 'kd_tree', 'brute']) trial.suggest_float('p', 1.0, 2.5) </pre>	<p>LocalOutlierFactor(algorithm = 'ball_tree', leaf_size = 11, metric = 'manhattan', n_neighbors = 378, novelty = True, p = 1.3931390899550997)</p> <p>Features: 51</p>
2	IF	<pre> n_estimators = trial.suggest_int('n_estimators', 200, 1000) max_samples = trial.suggest_float('max_samples', 0.1, 1.0) contamination = "auto" random_state = 42 </pre>	<p>Best Hyperparameters: {'n_estimators': 662, 'max_samples': 0.7345955924722449}</p> <p>Features: 51</p>
3	OCSVM (SVDD)	<pre> trial.suggest_float('nu', 0.01, 0.5) trial.suggest_categorical('kernel', ['linear', 'rbf', 'poly', 'sigmoid']) if kernel in ['rbf', 'poly', 'sigmoid']: gamma = trial.suggest_categorical('gamma', ['scale', 'auto']) else: gamma = 'auto' trial.suggest_int('degree', 2, 5) trial.suggest_float('coef0', 0.0, 1.0) </pre>	<p>Best Hyperparameters: {'nu': 0.3965959621077191, 'kernel': 'linear'}</p> <p>Features: 51</p>
4	VANILLA AUTOENCODER	<pre> Default search space size: 4 layer2 (Int) {'default': None, 'conditions': [], 'min_value': 640, 'max_value': 1024, 'step': 32, 'sampling': 'linear'} layer1 (Int) {'default': None, 'conditions': [], 'min_value': 256, 'max_value': 640, 'step': 16, 'sampling': 'linear'} learning_rate (Float) {'default': 1e-05, 'conditions': [], 'min_value': 1e-05, 'max_value': 0.001, 'step': None, 'sam- pling': 'log'} activation (Choice) {'default': 'tanh', 'conditions': [], 'values': ['tanh', 'selu', 'elu', 'relu', 'LeakyReLU'], 'ordered': False} </pre>	<p>Hyperparameters:</p> <p>layer2: 768</p> <p>layer1: 640</p> <p>learning_rate: 1e-05</p> <p>activation: elu</p> <p>Score: 0.004791472130206856</p> <p>Batch Size: 256</p> <p>Features: 51</p>
5	VAE	<pre> Search space summary Default search space size: 4 intermediate_dim (Int) {'default': None, 'conditions': [], 'min_value': 640, 'max_value': 1024, 'step': 32, 'sampling': 'linear'} latent_dim (Int) {'default': None, 'conditions': [], 'min_value': 256, 'max_value': 640, 'step': 16, 'sampling': 'linear'} learning_rate (Float) {'default': 1e-05, 'conditions': [], 'min_value': 1e-05, 'max_value': 0.001, 'step': None, 'sam- pling': 'log'} activation (Choice) {'default': 'tanh', 'conditions': [], 'values': ['tanh', 'selu', 'elu', 'relu', 'LeakyReLU'], 'ordered': False} </pre>	<p>Hyperparameters:</p> <p>intermediate_dim: 832</p> <p>latent_dim: 640</p> <p>learning_rate: 5.163978446960434e-05</p> <p>activation: elu</p> <p>Score: 0.02933853982703518</p> <p>Batch Size: 256</p> <p>Features: 51</p>

Table 2 (continued)

No	Model	Hyperparameters search	Selected model
6	VAE-LSTM	Search space summary Default search space size: 3 lstm_units (Int) {'default': None, 'conditions': [], 'min_value': 256, 'max_value': 512, 'step': 32, 'sampling': 'linear'} latent_dim (Int) {'default': None, 'conditions': [], 'min_value': 128, 'max_value': 256, 'step': 32, 'sampling': 'linear'} activation (Choice) {'default': 'tanh', 'conditions': [], 'values': ['tanh', 'selu', 'elu', 'relu', 'LeakyReLU'], 'ordered': False}	Hyperparameters: lstm_units: 512 latent_dim: 256 activation: tanh Score: 0.06326609241827592 Batch Size: 1024 Features: 115

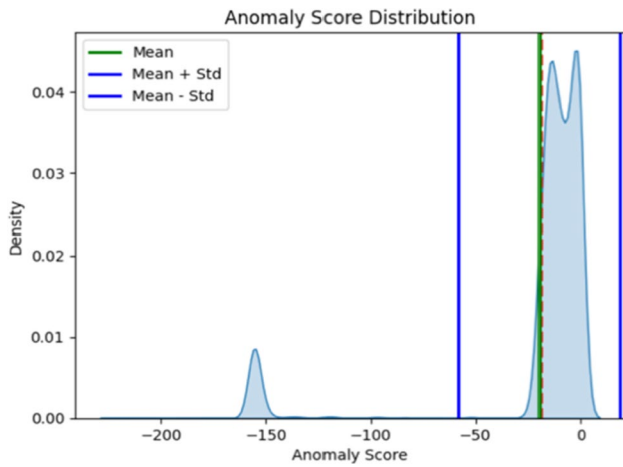


Fig. 3 Anomaly score distribution from LOF model

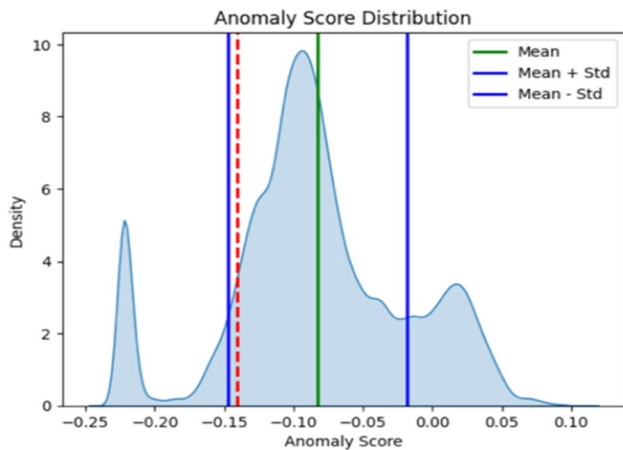


Fig. 4 Anomaly score distribution from IF model

while maximizing the margin and minimizing the risk of incorporating anomalies. Through this optimization process, OCSVM learns a decision boundary that effectively separates normal data from potential outliers. This decision

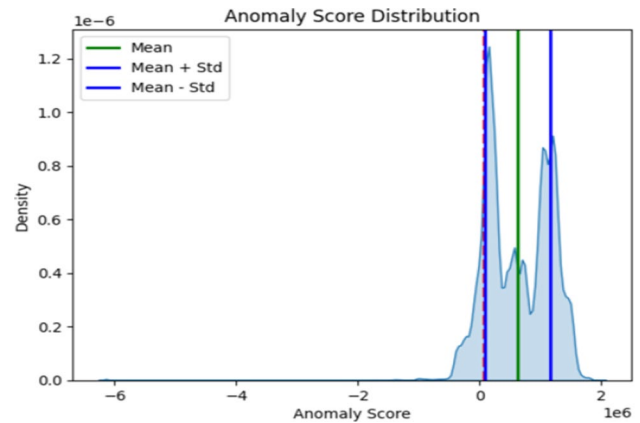


Fig. 5 Anomaly score distribution from OCSVM model

boundary, often referred to as the support or hyperplane. OCSVM accomplishes this by formulating a convex optimization problem that seeks to minimize the volume of the region containing the normal data points while simultaneously maximizing the margin. While the predict method can classify instances as either normal (inlier) or abnormal (outlier) based on the learned representation of normal data during training, anomalies may not be accurately classified since the model was not explicitly trained on them. Therefore, various thresholds were tested over the anomaly scores obtained from the decision function method—as shown in Fig. 5—to classify the data as normal or anomalous. In this case, instances with more negative anomaly scores indicate a higher likelihood of being classified as anomalies.

In the case of autoencoder algorithms, different versions were tested. First, the vanilla autoencoder, a basic neural network architecture, is utilized in anomaly detection for its ability to learn meaningful representations of input data and reconstruct it accurately. During training on normal data, the autoencoder learns to minimize reconstruction errors. Anomalies, deviating from the learned data distribution, often result in higher reconstruction errors, making them detectable. Second, Variational Autoencoder (VAE) stands

out for its distinctive probabilistic modeling approach and capability to capture intricate data distributions. Trained on a dataset primarily comprising normal instances, the VAE learns to encode input data into a lower-dimensional latent space, characterizing essential features of the data. Anomalies, which deviate markedly from the learned data distribution, often exhibit higher reconstruction errors and increased uncertainty in the latent space. Leveraging these characteristics, VAE effectively identifies anomalies by detecting instances with elevated reconstruction errors and uncertainty levels.

Variational autoencoders (VAEs) provide significant advantages for anomaly detection within the secure water treatment (SWAT) dataset, leveraging a sophisticated probabilistic framework. Unlike vanilla autoencoders that learn deterministic functions for reconstruction, VAEs model input data as distributions over an encoded latent space. This probabilistic approach is particularly beneficial in handling the inherent uncertainty and variability in the SWAT dataset, which comprises complex, high-dimensional time-series data capturing various operational and cyber-physical attack scenarios in a water treatment system. Crucially, VAEs implement an anomaly detection mechanism based on the total loss, which combines reconstruction loss and the Kullback–Leibler (KL) divergence, using the formula: $\text{total_loss} = 0.5 \times \text{reconstruction_loss} + 0.5 \times \text{KL_loss}$. This method not only aids in preventing overfitting to "normal" data but also enhances their capability to generalize across different types of data. By evaluating both the fidelity of the reconstruction and the statistical distance from the learned distribution of "normal" data (KL divergence), VAEs can more accurately flag anomalies as data points that deviate significantly from the model's expectations.

Finally, VAE with Long Short-Term Memory (LSTM) layers, blends the strengths of VAE's probabilistic modeling and LSTM's sequence modeling capabilities. This architecture is tailored to capture both temporal dependencies and complex data distributions. By integrating LSTM layers into the VAE framework, the model gains the ability to encode and decode sequential data while simultaneously learning a probabilistic representation of the data distribution in the latent space. During training, the VAE-LSTM model learns to reconstruct input sequences faithfully while also modeling the uncertainty inherent in the data. Anomalies, characterized by deviations from the learned sequential patterns and data distribution, manifest as sequences with elevated reconstruction errors and increased uncertainty.

The integration of Variational Autoencoders (VAEs) with Long Short-Term Memory (LSTM) recurrent neural networks, as exemplified by VAE-LSTM models, presents a compelling avenue for interpretability in time series analysis

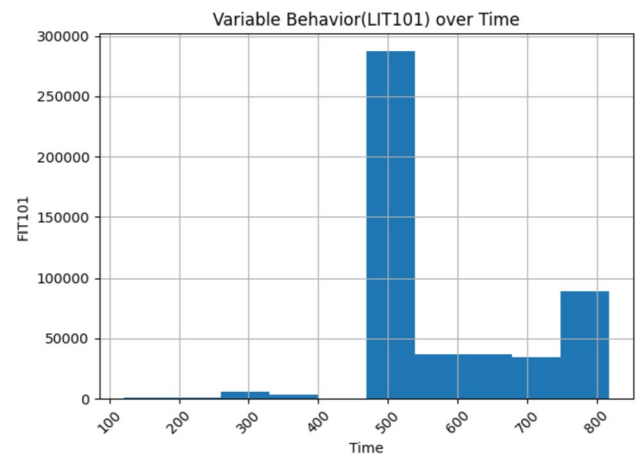


Fig. 6 Non-normal data distribution from LIT101

within the realm of cybersecurity research. By harnessing the latent space representation capabilities of VAEs and the sequential learning process of LSTMs, VAE-LSTM models offer a nuanced understanding of temporal patterns encoded within cybersecurity-related time series data. This interpretability stems from the ability to analyze the learned latent space, which encapsulates the underlying structure of the data, as well as the generative properties of the model, facilitating insights into the generation of new sequences based on learned patterns. Additionally, the inherent interpretability of LSTM states further elucidates the model's decision-making process, shedding light on the temporal dependencies captured during training. These attributes position VAE-LSTM models as valuable tools for cybersecurity researchers seeking to unravel the intricacies of time series data and glean meaningful insights to enhance threat detection and response strategies.

5 Results

In the context of unsupervised anomaly detection for cybersecurity in CIs, the evaluation process is challenging due to the absence of labeled data. The evaluation relies on model's performance based on its ability to capture anomalies within the data distribution. However, when the distribution of data in CI environments is not normal, it adds complexity to the task of anomaly detection. This is particularly evident in the case of data originating from IIoT systems, such as the SWAT dataset shown in Fig. 6. Specifically, it pertains to a sensor—LIT101—responsible for measuring the raw water tank level. In such cases, unsupervised anomaly detection models may struggle to accurately capture and distinguish between normal and anomalous behavior. Some unsupervised anomaly detection algorithms are more robust to

deviations from normality than others. For instance, IF and LOF can handle non-normal data distributions effectively.

In ICS, preserving the temporal order in the dataset is crucial. Consequently, the value of certain features should reflect the evolution over time of other features. For instance, at a given instant t , the state of the water tank output valve being open may depend on the fact that, during the preceding instant $t - 1$, the water tank input valve was opened, and the water tank was being filled [29]. While LOF, IF, and OCSVM have shown promise in capturing temporal dynamics for anomaly detection, Vanilla Autoencoder and Variational Autoencoder (VAE) may not offer the same level of fidelity in representing the evolving nature of features over time. Autoencoders, although proficient in dimensionality reduction and feature learning, inherently lack mechanisms to account for temporal dependencies. Conversely, the integration of VAE with long short-term memory (LSTM) recurrent neural networks, as in VAE-LSTM, holds promise for capturing both temporal dynamics and latent representations in time series data.

The approach of this work is to evaluate the model's ability to distinguish between normal and anomalous instances based on statistical properties or deviations from expected behavior. This involves analyzing the model's output scores or anomaly scores and setting appropriate thresholds to separate normal from anomalous instances. Following the separation, the multivariate time series is analyzed to assess the models' outcomes. Finally, to gain a better understanding of the results, a label was created based on the threshold and compared with the original label. However, the original labeling may not be entirely accurate, as it was based solely on the start and end times of attacks, without considering several important factors. First, it is possible for an attack to begin without immediately impacting the plant's operations; some attacks may take hours before their effects become apparent. Second, the impact of an attack may persist even after the attack itself has ceased. Third, the time required for the plant to return to normal operation behavior can vary depending on the severity of the attack's effects.

To assure that models detecting was accurate the time-series and attack times were analyzed based on what models identified as an attack. Finally, with the aim of have some comparison with other models the labels created from models results and original labels were used to calculate the following metrics Recall, Precision, AUC, False Positives, False Negatives, True Positives and True Negatives [3].

5.1 Isolation Forest Anomaly Detection Model

The isolation forest (IF) model was trained using the following hyperparameters obtained from the hyperparameter search: `'n_estimators': 662`, `'max_samples': 0.7345955924722449`. IF was the fastest model for

training (59.8 s) and testing (15.2 s). With this model, 19 out of 26 attacks were detected. However, 15 attack scenarios were partially detected, ranging from some points to almost none, as indicated in Table 3. For instance, in attack scenario number 8, the sensor DPIT301, responsible for controlling the backwash process, was altered. Consequently, the backwash process initiated multiple times, causing interruptions in normal operations and changes in tank levels 301 and 401. As shown in Fig. 7, the upper time series represents the original label, with red color indicating anomalies. In the lower time series, red color represents points marked as anomalies by the IF model.

5.2 Local Outlier Factor (LOF) Model

In this study, the LOF algorithm with novelty detection enabled (`novelty=True`) was employed. Its goal is to distinguish between 'normal' data points—those similar to the training data—and novel, potentially anomalous data points that deviate significantly from the training distribution. The best model got have the following parameters: `algorithm='ball_tree'`, `leaf_size=11`, `metric='manhattan'`, `n_neighbors=378`, `novelty=True`, `p=1.39`. This model identified 8 attack scenarios out of the 26 presents in the SWAT dataset, with only attack scenario number 41 being partially detected, as showed in Table 3.

5.3 OneClass SVM Model

The OCSVM modelling used the hyperparameter search as is showed in Table 2. The Best Hyperparameters were: `'nu': 0.3965959621077191`, `'kernel': 'linear'`. In this case, a linear kernel works by implicitly mapping the input data into a higher-dimensional feature space where a linear separation boundary is sought to encapsulate the normal data points while minimizing the influence of outliers or anomalies. In the higher-dimensional feature space, the data points may exhibit linear separability, even if they were not linearly separable in the original feature space. A total of 16 attacks out of 26 were detected using the OCSVM model; however, 12 of them were partially identified. This model was the only one to partially detect attack scenarios numbers 1, 16, and 21. As illustrated in Fig. 8, the number of points marked in red, representing points identified as anomalous from attack scenario number 16, is quite low.

5.4 Vanilla Autoencoder

The application of vanilla autoencoders for anomaly detection in the SWAT dataset presents significant challenges due to the dataset's specific characteristics. The SWAT dataset

Table 3 Attacks detected by unsupervised models

No	Scenary	IF	LOF	OCSVM	VAN AE	VAE	VAE LSTM
1	1			X**			
2	2			X**			X**
3	3			X			X**
4	7			X			X*
5	8	X*		X		X**	X*
6	10	X**	X	X*	X	X	X
7	11	X*	X	X	X	X	X
8	16			X**			
9	17	X**		X**			X**
10	21						
11	22	X	X	X	X	X*	X
12	23	X*		X*			X*
13	25	X*		X**			X
14	26	X*		X**			X**
15	27						X*
16	28	X	X	X*	X	X	X
17	30	X**					X*
18	31	X*					X*
19	32	X**			X*	X**	X
20	33	X*				X**	X
21	35	X*				X*	X*
22	36	X*			X*	X*	X
23	37	X	X		X	X	X
24	39	X*	X		X	X	X
25	40	X	X	X*	X	X	X
26	41	X**	X*		X*	X*	X*

*Attack partially detected

**Only some points of the scenary where identified as anomalies

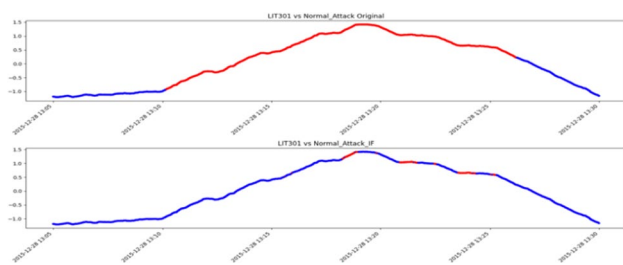


Fig. 7 Attack scenario number 8 with IF model (sensor LIT 301)

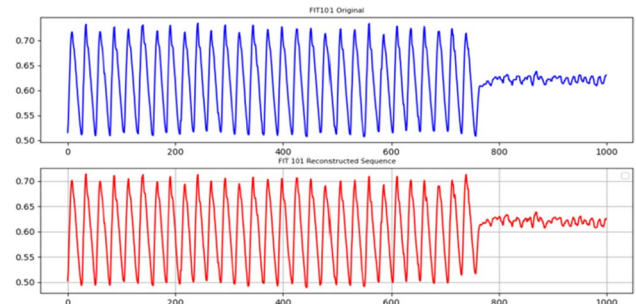


Fig. 9 Reconstruction capacity of Vanilla Autoencoder

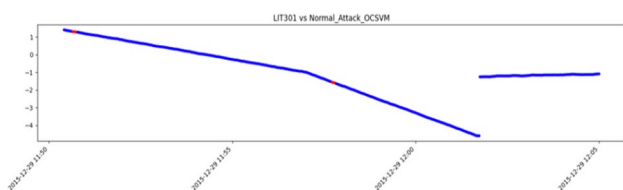


Fig. 8 Attack scenario number 16 with OCSVM model

consists of high-dimensional time-series data that captures the dynamic interactions of various components in a water treatment system; this includes both normal operations and cyber-physical attack scenarios. Vanilla autoencoders, which learn a deterministic function for reconstruction, are designed to minimize reconstruction error based on ‘normal’ operational data but tend to reconstruct anomalies

Table 4 Evaluation of unsupervised models

Model	AUC	True positives	True negatives	False positives	False negatives	Precision	Recall	F1 score
IF	0.82	36,287	369,406	21,244	15,606	0.63	0.69	0.66
LOF	0.83	35,401	381,796	8854	16,492	0.79	0.68	0.73
OCSVM	0.72	27,136	360,255	30,395	24,757	0.47	0.52	0.49
VAN-AUC	0.82	35,804	368,923	21,727	16,089	0.62	0.68	0.65
VAE	0.82	35,912	369,031	21,619	15,981	0.62	0.69	0.65
VAE-LSTM (115)	0.87	40,490	373,566	17,034	11,403	0.70	0.78	0.74

Test shape: 442,543

too accurately, as shown in Fig. 9. This model achieves a reconstruction loss of 0.0047, calculated using the Mean Absolute Error. However, this over-generalization leads to a critical failure in detecting anomalies, as the reconstruction error—a pivotal metric in flagging deviations—remains minimal for both normal and abnormal data inputs. Furthermore, vanilla autoencoders fail to capture the crucial temporal dynamics and interdependencies of the sensor and actuator readings inherent in the dataset. Consequently, this model detects 10 cyberattacks present in the SWAT dataset, with 3 of them detected partially.

5.5 Variational Autoencoder (VAE)

The VAE model without LSTM layers had better metrics compared to the Vanilla Autoencoder Model. VAE had higher numbers of True Positives and True Negatives, and lower results in False Negatives and False Positives. The Recall is also higher, which is the preferred scenario in cybersecurity for CI, as illustrated in Table 4. These improved results allowed for better detection of cyberattacks, with a total of 13 scenarios out of 26 being identified. However, for scenarios number 8, 32, and 33, only a few points were effectively identified.

VAE without the integration of LSTM does not inherently contemplate the temporal dynamics of time-series data, as they process each input independently of the others. Water treatment processes are inherently sequential and continuous, with the state of the system at any given time dependent on its previous states. Therefore, to improve the attack detection capability of the VAE model, LSTM layers were incorporated into the model. These layers can capture these temporal dependencies, essential for modeling and understanding process behaviors over time. As explained in the following section.

5.5.1 VAE-LSTM Model

Previous experiments demonstrated the need for incorporate temporal dependencies over time to enhance detection capacity of the model. Therefore, various feature engineering techniques were implemented to reinforce the context of timeseries. The best results got when applied these techniques over the features that demonstrated to have seasonality in their behavior like FIT101, LIT101, FIT201, DPIT301, FIT301, LIT301, AIT401, FIT601. Particularly, lagged features, which represent values from previous time steps can enable the model to capture temporal dependencies more effectively. This enhancement is crucial for understanding sequential data, where current states are often directly influenced by previous events. Additionally, windowed statistics such as moving averages provide aggregated information over specified intervals, offering insights into trends and variability within the data stream. These statistics serve to smooth out noise—frequently present in sensor data—and reduce data variability, which helps the model in focusing on significant changes rather than normal fluctuations. Consequently, the VAE-LSTM model shown in Fig. 10—was trained over 115 features, they also improve the quality of the latent representations learned by the VAE, thereby facilitating a more precise reconstruction of normal states and a more distinct delineation of anomalies. The high-level procedure is explained in the following pseudo-code:

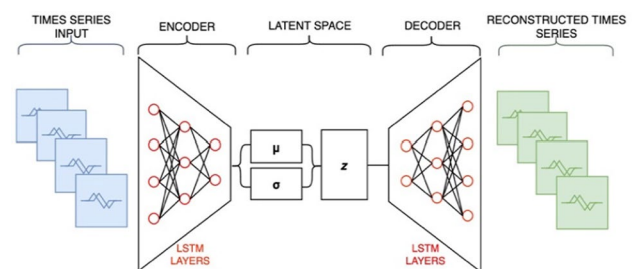


Fig. 10 VAE-LSTM Architecture

Step 1: Preprocessing

- Create two different sets from the original data:
 1. Original Features Preprocessing:
 - Standardize or normalize the original features as required
 2. Numerical Features Analysis:
 - Identify seasonal features using autocorrelation for the 25 sensor numerical features
 - Create Rolling Window Statistics Features (RWSF) to smooth data and reduce noise by calculating mean, median, standard deviation, maximum, and minimum values with a window of 120 s
 - Create Time-shifted Features (TF) to capture temporal dependencies by applying lags of 10, 20, and 30 s
- Combine the preprocessed features:
 - 51 original features
 - 40 RWSF features
 - 15 TF features
- The resulting dataset comprises a total of 115 features

Step 2: Define the Encoder

- Create an input layer for the time series data
- Add multiple LSTM layers to learn temporal dependencies
- Create dense layers to represent the latent space
- Define a sampling function to generate the latent variable from the mean and log variance

Step 3: Define the Decoder

- Create an input layer for the latent variable
- Use a repeat vector to match the number of timesteps
- Add multiple LSTM layers to reconstruct the sequences
- Create an output layer to produce the reconstructed sequences

Step 4: Define the VAE Loss

- Calculate the reconstruction loss between the input and the output
- Calculate the KL divergence loss using the mean and log variance of the latent space
- Combine the reconstruction loss and the KL divergence loss to get the total VAE loss

Step 5: Compile the VAE-LSTM Model

- Define the VAE model using the encoder and decoder
- Add the VAE loss to the model
- Compile the model with an appropriate optimizer

Step 6: Train the VAE-LSTM Model

- Train the model using the training data
- Validate the model using the validation data

Step 7: Perform Anomaly Detection

- Use the trained model to get reconstructed data from the validation data
 - Calculate the reconstruction error by comparing the original and reconstructed data
 - Set a threshold for anomaly detection based on the reconstruction error
 - Detect anomalies by comparing the reconstruction error to the threshold
-

A hyperparameter search using Bayesian Optimization allows us to select the best parameters for the feature engineering dataset to train the model. See Table 2. The distribution of Mean Absolute Error (MAE) values obtained during the training of a VAE-LSTM model indicates that the model was able to reconstruct most of the training data with a small error. Moreover, the peak of the histogram, illustrated in Fig. 11 and slightly above 0. This suggests that the most frequent reconstruction error across the dataset is low, implying that the model performs well on most of the training data. Understanding that anomalies are present only in the test set, the histogram for the MAE, shown in Fig. 12, displays a bimodal distribution with peaks near zero and around 8. This pattern suggests that the model consistently yields low MAE values for normal data but also identifies anomalies as instances with significantly higher MAE values. The distinct peak around 8 is indicative of the model encountering anomalies that it has not seen during the training phase, which it fails to reconstruct accurately, resulting in higher MAE scores. The spread of errors toward the higher end of the scale confirms the presence of anomalies within the test data.

This model significantly enhances the capability to detect cyberattacks, successfully identifying 23 out of 26 attack scenarios. It also improves the consistency of detection, with only 4 scenarios being partially detected—specifically, attacks number 2, 3, 17, and 26. Additionally, the VAE-LSTM model has identified the post-attack effects in numerous scenarios. This reflects the actual behavior of cyber-physical plants, which require a stabilization period after experiencing a cyberattack. The amount of time needed for stabilization varies with the impact of the attack, as observed in scenarios 11, 23, 25, 28, 32, and 39. It can also be concluded that some anomalies may be related to the destabilization of the water-treatment plant due to preceding cyberattacks, rather than being attributable to the current scenario. This conclusion is supported by scenarios such as number 6, which, according to SWAT developers, had an impact that became apparent two hours post-attack—a period during which additional attacks were launched.

The reconstruction capacity of the VAE-LSTM model, as demonstrated by the comparison of the original and reconstructed sequences in Fig. 13 (original in blue and reconstructed in red), elucidates both its strengths and limitations.

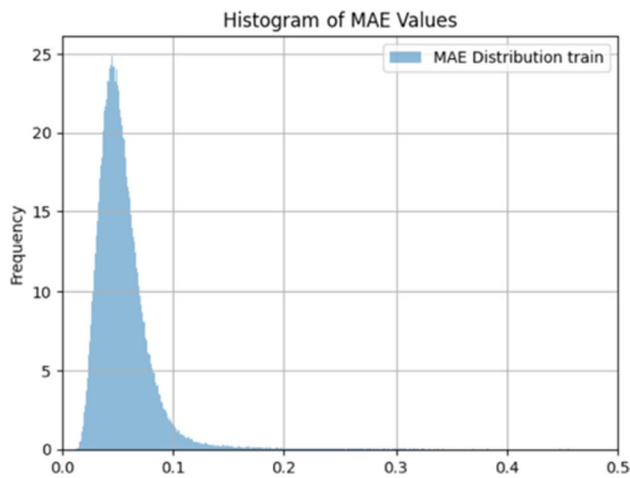


Fig. 11 MAE on training set

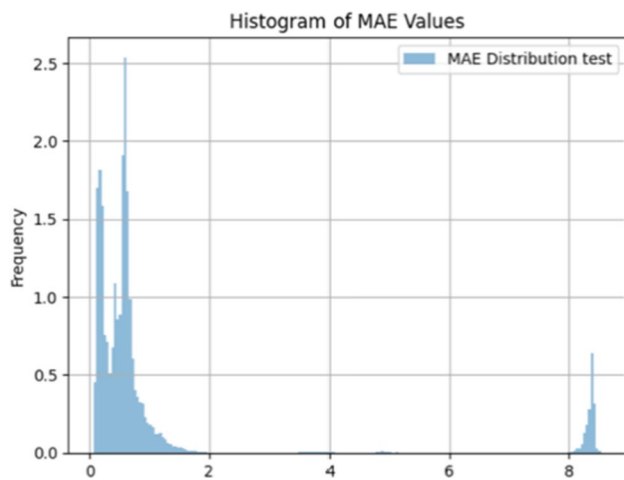


Fig. 12 MAE on test set

The model proficiently captures the overall cyclic patterns and temporal dependencies inherent in the sensor readings, signifying its robust capability to learn and replicate normal operational behavior. This proficiency is particularly advantageous for detecting significant deviations or anomalies, as the model encounters difficulty in accurately reconstructing data that diverges from the learned patterns. However, the observed discrepancies in the amplitude and sharpness of peaks and troughs between the original and reconstructed sequences underscore its limitations in capturing precise details and managing high variability. Notably, while an excessively precise reconstruction might appear beneficial, it could lead to overfitting to the noise and minor fluctuations present in the training data, thereby diminishing its generalizability and effectiveness in identifying genuine anomalies. Moreover, the model may struggle to detect subtle anomalies that closely resemble normal patterns, which are inherently

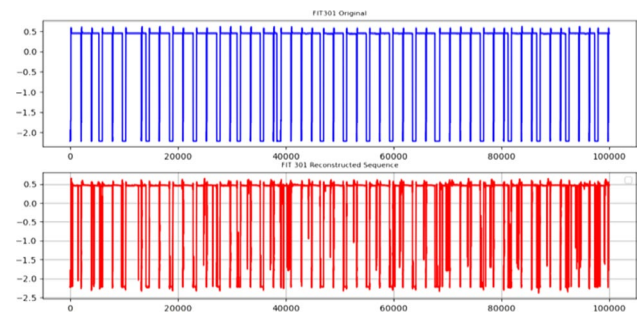


Fig. 13 Reconstruction capacity of VAE-LSTM

more challenging to distinguish. Consequently, the model's current balance between capturing general patterns and avoiding overfitting to noise is crucial for effective anomaly detection.

Table 5 provides a comprehensive comparative analysis of the computational efficiency and effectiveness of the tested models. Among these, the VAE-LSTM-115 model demonstrates a remarkable balance between training and testing times, with a training duration of 21 min 51 s and a testing duration of 4 min 14 s. This model's performance underscores its suitability for applications requiring moderate computational resources while maintaining high efficacy in anomaly detection. Although models such as Isolation Forest (IF) exhibit faster training (1 min) and testing (15.4 s) times, their detection accuracy is insufficient for the complexities of industrial applications. Conversely, the OCSVM model, despite its extensive computational costs—training for 1 h 41 min 25 s and testing for 41 min 21 s—fails to offer practical utility for near real-time deployment. The autoencoder models, particularly the VAE-LSTM-115, provide a compelling case for their adoption in industrial settings due to their efficient training and testing times combined with robust anomaly detection capabilities. Therefore, the VAE-LSTM-115 model emerges as the optimal choice among the evaluated models, balancing computational efficiency with high detection accuracy, which is essential for industrial applications.

Finally, making comparisons with other studies is challenging because, to the best of our knowledge, there are no other works using the SWAT dataset that prioritize the cybersecurity perspective and real-world requirements for developing their models. Comparing models trained on different datasets is not accurate because a model can perform significantly well with one dataset but may not yield similar results with another. Additionally, our unsupervised anomaly detection approach does not allow the use of labels for training, and the label information obtained from SWAT developers is inaccurate; the labeling was based on the time in which developers started the attack until the time they stopped the attack. Nevertheless, the time when the physical sensor was

Table 5 Computational efficiency of tested models

Model	Training time	Testing time
LOF (Novelty = True)	CPU times: user 20 min 15 s, sys: 4.64 s, total: 20 min 19 s Wall time: 20 min 19 s	CPU times: user 33 min 24 s, sys: 4.2 s, total: 33 min 28 s Wall time: 33 min 28 s
IF	CPU times: user 59.8 s, sys: 384 ms, total: 1 min Wall time: 1 min	CPU times: user 15.2 s, sys: 141 ms, total: 15.4 s Wall time: 15.4 s
OCSVM (SVDD)	CPU times: user 1 h 41 min 12 s, sys: 13.3 s, total: 1 h 41 min 25 s Wall time: 1 h 41 min 25 s	CPU times: user 41 min 21 s, sys: 212 ms, total: 41 min 21 s Wall time: 41 min 21 s
VANILLA AUTOENCODER	CPU times: user 21 min 4 s, sys: 44.3 s, total: 21 min 48 s Wall time: 19 min 37 s	CPU times: user 24.5 s, sys: 1.2 s, total: 25.7 s Wall time: 21.6 s
VAE	CPU times: user 31 min 40 s, sys: 1 min 22 s, total: 33 min 3 s Wall time: 32 min 33 s	CPU times: user 43.3 s, sys: 2.42 s, total: 45.7 s Wall time: 45.6 s
VAE-LSTM-115	CPU times: user 21 min 21 s, sys: 30.3 s, total: 21 min 51 s Wall time: 16 min 36 s	CPU times: user 4 min 10 s, sys: 3.8 s, total: 4 min 14 s Wall time: 3 min 52 s

affected is different. For instance, in attack scenario number 6, the real impact on physical systems was two hours after the official start time, causing an additional cascading effect over other attack scenarios that were launched at that time. Therefore, studies like [5] report a positive detection of an attack in a window of time in which no physical device was disrupted. Additionally, most of the studies that used the SWAT dataset, such as [16, 17], and [18] do not consider the post-attack effects that could be longer depending on the impact of the cyber-attack. In the SWAT dataset, there are different timeframes among the scenarios, ranging from less than a minute between scenarios 19 and 20 to several hours. Thus, evaluating the results based on measures like F1-score, recall, or precision may not be entirely accurate because these metrics depend on the accuracy of the labels used. It is important to highlight that this work utilized only physical data (sensors and actuators) to build and test our model, aiming to detect attacks that have a physical impact on CIs.

Thus, we decided to analyze the results based on a deep interpretation of what our model considered an anomaly, allowing us to identify 23 out of 26 scenarios. However, to get a glimpse of the model's behavior, we developed our own labels based on the results and compared them to the original ones. This process allows us to calculate the following metrics, as illustrated in Table 4.

6 Conclusions and Future Work

This study highlights the effectiveness of machine learning-based anomaly detection models, trained exclusively on high-frequency normal data, in enhancing the detection of

zero-day attacks targeting Critical Infrastructures (CIs). Various algorithms were tested on the same preprocessed dataset. However, LOF, IF, and OCSVM struggled to capture the complex, non-linear relationships and interdependencies of data extracted from Cyber-Physical Systems (CPS). These models were not effective in handling the high-dimensional, temporal data and the interactions between various sensors and actuators. In contrast, the VAE-LSTM model demonstrated the ability to capture temporal dependencies and adapt to non-stationary data, yielding better results. The VAE-LSTM model was trained using 10-step sequences, leveraging data from the preceding 120 s to identify anomalies. While this approach does not provide real-time detection, it offers a feasible timeframe for identifying attacks that impact physical systems.

The research prioritizes cybersecurity considerations over traditional data science practices, aiming to develop a model with strong generalization capabilities and practical applicability. To ensure comprehensive coverage of potential attack surfaces in physical systems, all available features were used for training. This included categorical features, those with low variance, differing distributions between training and test datasets, and highly correlated features like P501 and UV401, which have a 99.99% correlation. Such correlations are logical in Industrial Control Systems (ICS), where sensor and actuator functionalities are often interdependent. Eliminating one of these features is not feasible because both can become attack surfaces. This study also addressed challenges such as the post-attack effect, which anomaly detection systems are likely to identify. However, this is less concerning in the context of CI cybersecurity, where human intervention is essential.

While the results demonstrate the promise of this approach, relying solely on anomaly detection based on physical system data is not sufficient. To gain a more holistic view of security, it is crucial to incorporate network information. This will be a focus of future research. Upcoming work will involve integrating network traffic information with physical data to develop multimodal approaches that can detect anomalies from different sources. Combining data from different sources allows for the extraction of complementary features and the capture of intricate dependencies between modalities. For instance, sensor data can highlight anomalies in physical operations, such as unauthorized access or tampering, whereas network traffic data can reveal digital threats like distributed denial-of-service (DDoS) attacks, malware communications, or data exfiltration attempts. This integration is critical as it provides a comprehensive view of both physical and digital activities, greatly improving the detection of complex cyber threats that might not be identified if each type of data were examined separately.

Additionally, it is essential to develop models that enhance the sensitivity for detecting subtle anomalies, as current attack vectors are increasingly designed to closely mimic normal behavior to evade detection.

Author contributions Conceptualization, A.P., Y.D., H.L.-C., and G.J.A.; methodology, A.P.; validation, Y.D., H.L.-C., and G.J.A.; formal analysis, A.P. and Y.D.; investigation, A.P., Y.D., H.L.-C., and G.J.A.; resources, Y.D. and ; data curation, A.P.; writing—original draft preparation, A.P.; writing—review and editing, Y.D., H.L.-C., and G.J.A.; visualization, A.P. and H.L.-C.; supervision, Y.D. and G.J.A.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding Andrea Pinto has received research support from Universidad de los Andes. Other authors have no relevant financial or non-financial interests to disclose.

Data availability The datasets generated during the current study, as well as the models developed, are available from the corresponding author on reasonable request. Interested researchers may contact the corresponding author to inquire about access to the data and models.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If

material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Herrera, L.C., Maennel, O.: A comprehensive instrument for identifying critical information infrastructure services. *Int. J. Crit. Infrastruct. Protect.* (2019). <https://doi.org/10.1016/j.ijcip.2019.02.001>
- Nazir, S., Patel, S., Patel, D.: Autoencoder based anomaly detection for SCADA networks. *Int. J. Artif. Intell. Mach. Learning* **11**(2), 83–99 (2021). <https://doi.org/10.4018/ijaiml.20210701.oa6>
- Boateng, E.A., Bruce, J.W., Talbert, D.A.: Anomaly detection for a water treatment system based on one-class neural network. *IEEE Access* **10**, 115179–115191 (2022). <https://doi.org/10.1109/ACCESS.2022.3218624>
- Raman, M.R.G., Somu, N., Mathur, A.P.: A multilayer perceptron model for anomaly detection in water treatment plants. *Int. J. Crit. Infrastruct. Protect.* (2020). <https://doi.org/10.1016/j.ijcip.2020.100393>
- Nedeljkovic, D., Jakovljevic, Z.: CNN based method for the development of cyber-attacks detection algorithms in industrial control systems. *Comput. Secur.* (2022). <https://doi.org/10.1016/j.cose.2021.102585>
- Xie, X., Wang, B., Wan, T., Tang, W.: Multivariate abnormal detection for industrial control systems using 1D CNN and GRU. *IEEE Access* **8**, 88348–88359 (2020). <https://doi.org/10.1109/ACCESS.2020.2993335>
- Kim, B., Alawami, M.A., Kim, E., Oh, S., Park, J., Kim, H.: A comparative study of time series anomaly detection models for industrial control systems. *Sensors* **23**(3), 1310 (2023). <https://doi.org/10.3390/s23031310>
- Koutroulis, G., Mutlu, B., Kern, R.: A causality-inspired approach for anomaly detection in a water treatment testbed. *Sensors* **23**(1), 257 (2023). <https://doi.org/10.3390/s23010257>
- Khan, I.A., Keshk, M., Pi, D., Khan, N., Hussain, Y., Soliman, H.: Enhancing IIoT networks protection: A robust security model for attack detection in Internet Industrial Control Systems. *Ad Hoc Netw.* **134**, 102930 (2022). <https://doi.org/10.1016/j.adhoc.2022.102930>
- Khan, I.A., Pi, D., Abbas, M.Z., Zia, U., Hussain, Y., Soliman, H.: Federated-SRUs: A federated-simple-recurrent-units-based IDS for accurate detection of cyber attacks against IoT-augmented industrial control systems. *IEEE Internet Things J.* **10**(10), 8467–8476 (2023). <https://doi.org/10.1109/JIOT.2022.3200048>
- Umer, M.A., Junejo, K.N., Jilani, M.T., Mathur, A.P.: Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *Int. J. Crit. Infrastruct. Protect.* **38**, 100516 (2022). <https://doi.org/10.1016/j.ijcip.2022.100516>
- Kim, S.J., Jo, W.Y., Shon, T.: APAD: Autoencoder-based payload anomaly detection for industrial IoE. *Appl. Soft Comput. J.* **88**, 106017 (2020). <https://doi.org/10.1016/j.asoc.2019.106017>
- UK's National Cyber Security Centre: NCSC Annual Review 2023, London (2023).
- Pinto, A., Herrera, L.C., Donoso, Y., Gutierrez, J.A.: Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. *MDPI* (2023). <https://doi.org/10.3390/s23052415>

15. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007). <https://doi.org/10.2753/MIS0742-1222240302>
16. Li, C., Li, F., Zhang, L., Yang, A., Hu, Z., He, M.: Intrusion detection for industrial control systems based on improved contrastive learning SimCLR. *Appl. Sci.* **13**(16), 9227 (2023). <https://doi.org/10.3390/app13169227>
17. Novikova, E., Doynikova, E., Golubev, S.: Federated learning for intrusion detection in the critical infrastructures: Vertically partitioned data use case. *Algorithms* **15**(4), 104 (2022). <https://doi.org/10.3390/a15040104>
18. Fährmann, D., Damer, N., Kirchbuchner, F., Kuijper, A.: Light-weight long short-term memory variational auto-encoder for multivariate time series anomaly detection in industrial control systems. *Sensors* **22**(8), 2886 (2022). <https://doi.org/10.3390/s22082886>
19. Sun, H., Huang, Y., Han, L., Fu, C., Liu, H., Long, X.: MTS-DVGAN: Anomaly detection in cyber-physical systems using a dual variational generative adversarial network. *Comput. Secur.* (2024). <https://doi.org/10.1016/j.cose.2023.103570>
20. Haider, W., Hu, J., Slay, J., Turnbull, B.P., Xie, Y.: Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *J. Netw. Comput. Appl.* **87**, 185–192 (2017). <https://doi.org/10.1016/j.jnca.2017.03.018>
21. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.N.: TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **8**, 165130–165150 (2020). <https://doi.org/10.1109/ACCESS.2020.3022862>
22. Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., Bellekens, X.: Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 Dataset). Preprint at <http://arxiv.org/abs/2006.15340> (2020)
23. Al-Hawawreh, M., Sitnikova, E., Aboutorab, N.: X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. *IEEE Internet Things J.* **9**(5), 3962–3977 (2022). <https://doi.org/10.1109/JIOT.2021.3102056>
24. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access.* **10**, 40281–40306 (2022). <https://doi.org/10.1109/ACCESS.2022.3165809>
25. Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. *Mobile Netw. Appl.* **27**(1), 357–370 (2022). <https://doi.org/10.1007/s11036-021-01843-0>
26. Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., Nömm, S.: MedBioT: Generation of an IoT botnet dataset in a medium-sized IoT network. In: *ICISSP 2020 - Proceedings of the 6th International Conference on Information Systems Security and Privacy*, pp. 207–218 (2020).
27. Adepu, S., Junejo, K. N., Mathur, A., Goh, J.: A Dataset to Support Research in the Design of Secure Water Treatment Systems. <https://www.researchgate.net/publication/305809559>
28. Shin, H.-K., Lee, W., Yun, J.-H., Kim, H.: HAI 1.0: HIL-based Augmented ICS Security Dataset.
29. Gómez, Á.L.P., Maimó, L.F., Celdrán, A.H., Clemente, F.J.G.: MADICS: A methodology for anomaly detection in industrial control systems. *Symmetry (Basel)* **12**(10), 1583 (2020). <https://doi.org/10.3390/SYM12101583>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.