algorithms

MDPI

Article

Detection of Cyberattacks and Anomalies in Cyber-Physical Systems: Approaches, Data Sources, Evaluation

Olga Tushkanova [1,†], Diana Levshun [1,†], Alexander Branitskiy [1,†], Elena Fedorchenko [1,2,*,†], Evgenia Novikova [1,†] and Igor Kotenko [1,2,†]

[1] Computer Security Problems Laboratory, St. Petersburg Federal Research Center of the Russian Academy of Sciences, 199178 Saint-Petersburg, Russia
[2] Department of Computer Science and Engineering, Saint-Petersburg Electrotechnical University ETU "LETI", 197022 Saint-Petersburg, Russia
* Correspondence: doynikova@comsec.spb.ru
† These authors contributed equally to this work.

Abstract: Cyberattacks on cyber-physical systems (CPS) can lead to severe consequences, and therefore it is extremely important to detect them at early stages. However, there are several challenges to be solved in this area; they include an ability of the security system to detect previously unknown attacks. This problem could be solved with the system behaviour analysis methods and unsupervised or semi-supervised machine learning techniques. The efficiency of the attack detection system strongly depends on the datasets used to train the machine learning models. As real-world data from CPS systems are mostly not available due to the security requirements of cyber-physical objects, there are several attempts to create such datasets; however, their completeness and validity are questionable. This paper reviews existing approaches to attack and anomaly detection in CPS, with a particular focus on datasets and evaluation metrics used to assess the efficiency of the proposed solutions. The analysis revealed that only two of the three selected datasets are suitable for solving intrusion detection tasks as soon as they are generated using real test beds; in addition, only one of the selected datasets contains both network and sensor data, making it preferable for intrusion detection. Moreover, there are different approaches to evaluate the efficiency of the machine learning techniques, that require more analysis and research. Thus, in future research, the authors aim to develop an approach to anomaly detection for CPS using the selected datasets and to conduct experiments to select the performance metrics.

Keywords: anomaly detection; attack detection; cyber-physical system; machine learning; datasets; evaluation metrics

Citation: Tushkanova, O.; Levshun, D.; Branitskiy, A.; Fedorchenko, E.; Novikova, E.; Kotenko, I. Detection of Cyberattacks and Anomalies in Cyber-Physical Systems: Approaches, Data Sources, Evaluation. *Algorithms* **2023**, *16*, 85. https://doi.org/10.3390/a16020085

Academic Editors: Francesco Bergadano and Giorgio Giacinto

Received: 21 December 2022
Revised: 28 January 2023
Accepted: 30 January 2023
Published: 3 February 2023

1. Introduction

Cybersecurity risks are highly relevant nowadays. It is almost impossible to completely exclude security risks for modern information systems, including cyber-physical systems (CPS) and Internet of Things (IoT). Thus, it is essential to continuously detect cyberattacks and anomalies to monitor security risks and provide security awareness.

Cyberattacks against cyber-physical systems can lead to severe impacts on physical, environmental, as well as economical safety of the population [1]. For example, the attack on the Colonial Pipeline disrupted fuel supply on the US East Coast in 2021 [2], and the attack on the Venezuelan hydroelectric power plant led to a nationwide blackout in 2019 [3]. In 2022, Germany's internal fuel distribution system was disrupted by a cyberattack [4]. Thus, it is extremely important to detect such attacks at early stages.

There are several challenges in this area, and one of the most critical challenges is the detection of the previously unknown attacks. Another challenge relates to the availability of the datasets used to train analytical models, as the performance of the attack detection strongly depends on the quality of the training datasets. The first challenge relates to

the fact that machine learning models are usually trained on datasets with known attack patterns, and as a result, they are unable to detect previously unseen attacks. One of the possible solutions is to use anomaly detection techniques based on the analysis of the cyber-physical entities' behaviour [5–7]. However, such approaches require high-quality datasets to model normal behaviour or apply unsupervised or semi-supervised machine learning techniques. The lack of datasets close to the real world is explained by the fact that organizations do not want to share data, as they can include confidential data. There are attempts to generate such datasets using cyber-physical or software test beds, but the completeness and validity of such generated datasets are questionable. The last challenge relates to the validation of the attack and anomaly detection models. The analysis of the research papers has shown that different researchers use different approaches to calculate performance metrics that complicate the comparison of the models.

In this paper, the authors review existing approaches to attack and anomaly detection, outline the most commonly used datasets, and evaluate the applicability of the selected datasets in the anomaly detection task. We also revealed that researchers use different approaches to calculate performance metrics to evaluate machine learning models. These metrics consider the fact that the anomalies in CPS have a certain duration, and malicious activity may result in a delayed response of the system process; however, the variety of used metrics makes the comparison of the obtained experimental results complicated.

Thus, the *contribution* of the research is as follows:

- analysis of the approaches to anomaly detection for the cyber-physical systems;
- analysis of the selected datasets, namely, ToN_IoT [8], SWAT [9], and HAI [10] containing normal and anomaly related data for the cyber-physical systems, and selection of the dataset for the experiments;
- overview of the metrics used to evaluate the anomaly and attack detection models.

The paper is organized as follows. Section 2 provides the results of the review of the approaches to anomaly and attack detection for cyber-physical systems. Section 3 analyzes the datasets used for the attack and anomaly detection that contain the data from the cyber-physical systems. Section 4 researches the metrics for the evaluation of the attack and anomaly detection models. The paper ends with a conclusion.

2. Approaches to the Anomaly and Attack Detection for the Cyber-Physical Systems

Anomaly detection is the process of identifying anomalous events that do not match the expected behaviour of the system. This allows the detection of new and hidden attacks. Currently, anomaly detection approaches are often implemented using machine learning, such as shallow (or traditional) learning and deep learning [7,11,12]. In this case, the profile of normal behaviour can be built using many data sources.

Anomaly and attack detection in CPS based on shallow learning methods uses algorithms such as support vector machine (SVM) [13], Bayesian classification [14], k-nearest neighbor (kNN) [15], Random Forest (RF) [16,17], Isolation Forest [18], XGBoost [19], and artificial neural networks (ANN) [20,21]. They are based on training intelligent models to profile the normal behaviour of a cyber-physical system, and then inconsistent observations are identified as anomalies. For example, Elnour et al. [18] propose an attack detection framework based on dual isolation forest (DIF). Two isolated forest models are trained independently using normalized raw data and a preprocessed version of the data using principal component analysis (PCA). The principle of the approach is to detect and separate anomalies using the concept of isolation after analyzing the data in the original and PCA-transformed representations. Mokhtari et al. [16] and Park and Lee [17] explore such supervised learning algorithms for anomaly detection as k-nearest neighbours, decision tree classifier, and random forest. In both studies, the random forest shows the best detection result.

The analysis of related works has shown that the research focus has now shifted towards the use of deep neural networks to detect anomalies in technological processes. A number of authors compare classical and deep learning approaches to anomaly detection.

So Inoue et al. [22] compare one-class SVM with radial basis function kernel deep and dense neural network with a layer of long short-term memory (LSTM), and the experiments have shown that the deep learning model is characterized by a lower rate of false positive alarms. Gaifulina and Kotenko [23] experimentally compare several models of deep neural networks for anomaly detection. Shalyga et al. [24] propose several methods to improve the quality of anomaly detection, including exponentially weighted smoothing to reduce the false positive rate, individual error weight for each feature, non-overlapping prediction windows, etc. The authors also propose their own anomaly detection model based on a multilayer perceptron (MLP).

Traditional machine learning methods tend to be inefficient when processing large-scale data and unevenly distributed samples. Deep learning models are more productive when analyzing such data. Researchers often use autoencoders (AE) [5,6], recurrent neural networks [25], convolutional neural networks (CNN) [26–28], and generative adversarial networks (GAN) [29,30] as deep neural networks for anomaly detection in CPS. Often, the approaches propose a hybrid use of neural network data. For example, Xie et al. [25] and Wu et al. [31] use CNN for data dimensionality reduction and gated recurrent units (GRU) for data prediction. GRU is one of the types of recurrent networks, as well as LSTM. Bian X. [32] also uses GRU for anomaly detection. The main idea of the anomaly detection method is to predict the value of the next moment and determine if an anomaly occurs due to a deviation between the predicted value and the actual value.

The autoencoder is trained on normal data, and then the incoming events are reconstructed based on the normal model. Exceeding the reconstruction error threshold indicates an anomaly. Such an approach is used in the APAD (Autoencoder-based Payload Anomaly Detection) model by Kim et al. [5]. Wang et al. [6] propose an approach to anomaly detection using a composite model. The proposed model consists of three components: the encoder and decoder used to reconstruct the error, and the LSTM classifier, which takes the encoder output as input and makes predictions. To detect an anomaly, both model outputs, i.e., reconstruction error and prediction value, are considered together to calculate the anomaly score. The authors also compare the change ratio of each attribute during the current period and the previous one, and those attributes that have changed more are considered anomalous.

Generative adversarial networks can be used to investigate the distribution of normal data for recognizing anomalies from unknown data. The generator creates new data instances, and the discriminator evaluates them for authenticity. In the MAD-GAN (Multivariate Anomaly Detection with GAN) approach by Li et al. [29], both generator and discriminator components are represented by LSTM. The discriminator is trained to distinguish anomalies from normal data, and the anomaly score is computed as a combination of the discrimination output and reconstruction error produced by the generator component. A similar approach is proposed by Neshenko et al. [30]. The building blocks for the proposed GAN are the recurrent neural network and convolutional neural network. The authors also extended the anomaly detection approach by incorporating a module that attributes potentially attacked sensors. This task is solved by the application of various techniques starting with feature importance evaluation and finishing with KernelShap [33] and LIME [34] techniques that are model agnostic methods.

We should also mention approaches to anomaly detection using graph probabilistic models, such as Bayesian networks (BN) and Markov models. For example, Lin et al. [35] propose TABOR (Time Automata and Bayesian netwORk). Time Automata simulate the operation of the sensors and actuators, and the Bayesian network (BN) models the dependencies among random variables from the sensors and the actuators. This approach allows for the detection of timing anomalies, anomalies of sensor, and actuator value range, as well as a violation in their dependencies. Another popular way to represent normal behaviour is the hidden Markov model (HMM). Sukhostat L. [36] uses hierarchical HMM to detect anomalies in sensor values.

Application of the proposed techniques requires high-quality datasets that allow proper modelling of the CPS system functioning. Depending on the technique, it is required to have only normal data; some techniques require having both samples with normal and abnormal behaviour.

The first group of datasets is the data containing the indicators of the sensors of the cyber-physical system in the form of logs. The analysis of the research papers showed that currently, the most commonly used CPS dataset is SWAT dataset [9]. It is used in [5,6,18,24,25,29,30,35,36]. This dataset contains records from sensors, actuators, control programmable logic controllers (PLCs), and network traffic. Another new dataset for anomaly detection is HAI [10], which is used in research [16,17,32]. The dataset contains the parameters of sensors for an industrial power generation system using steam turbines and pumped storage power plants. To detect anomalies in IoT devices, the authors in the papers [19,20,31] use the TON_IoT dataset [8]. The ToN_IoT dataset includes telemetry from heterogeneous IoT and Industrial Internet of Things (IIoT) sensors.

Another group of datasets that are often used to detect anomalies and attacks in CPS are represented by network traffic datasets. They include such datasets such as NSL-KDD [37], CICIDS2017 [38] and UNSW-NB15 [39], and are used in the following research papers [23,27,31,40]. However, these datasets are represented mainly by network data that could be given in form of the PCAP (Packet Capture) files or labelled network flows. Section 3 discusses datasets in detail.

We should note that differences in the experimental conditions affect the possibility of comparing the results of anomaly detection. For example, Elnour et al. [18] exclude the stabilization time from the SWaT dataset. The way metrics are calculated can also vary, and research papers do not always provide a way to calculate these metrics. In general, the above machine learning methods show high anomaly detection results and can be used in further developments. A promising area of research and development is the creation of hybrid machine learning models for anomaly detection. In particular, combined networks with RNNs are used to capture temporal relationships [6,29], and combined networks with CNNs are applicable for context analysis (e.g., packet order and content) [25,30].

3. Datasets for the Attack and Anomaly Detection

An essential challenge of anomaly detection research is generating or finding a suitable dataset for the experiments. The authors analyzed existing datasets to select the dataset for further research.

The authors specified the following requirements of the dataset based on the research goal of anomaly detection in cyber-physical systems:

R1: the dataset should be gathered from the cyber-physical system;

R2: the dataset should contain event logs;

R3: the dataset should contain anomalies;

R4: the dataset should be labelled (what is normal and what is abnormal);

R5: the dataset should be close to real data (i.e., data from the real or semi-real system).

Currently, there are a lot of datasets available for various purposes and systems; they represent the functioning of the computer networks and cyber-physical systems, including the Internet of Things, Industrial Internet of Things, and Industrial Control Systems (ICS), such as SCADA (Supervisory Control And Data Acquisition) system [41].

Alsaedi et al. [8] present the comparative analysis of the available datasets for security purposes. Thus, there are datasets containing computer network traffic that was generated for attack detection purposes: KDDCUP99, NSL-KDD [37], UNSW-NB15 [39], and CICIDS2017 [38]. Such datasets do not contain sensors' data that is specific to CPS. Moreover, they do not include CPS network traffic, both normal and abnormal.

There are also datasets generated for cyber-physical systems security research purposes. Choi et al. [42] provide a comparison of the existing datasets generated for ICSs security research based on attack paths. Lemay and Fernandez [43] generate the SCADA network datasets (Modbus dataset) for intrusion detection research. The SCADA network

datasets by Rodofile et al. [44] contain attacks on the S7 protocol. These datasets are SCADA specific and contain a limited set of protocol specific attacks.

There are also multiple datasets for IoT and IIoT. Suthaharan et al. [45] propose the labelled wireless sensor network dataset (LWSNDR). It contains homogeneous data collected from a humidity-temperature sensor. The sensor is deployed in single-hop and multi-hop wireless sensor networks (WSNs). The dataset does not contain attack scenarios, but does contain anomalies introduced by the author using a hot water kettle. Sivanathan et al. [46] propose the datasets gathered from a smart home testbed. It contains network traffic characteristics of IoT devices. The dataset is generated for the IoT devices classification. The dataset does not contain attack scenarios.

There are also multiple network-based IoT datasets [37–39,46–48]. These datasets do not consider sensor data; thus, they do not allow for the detection of the attacks that manipulate sensors' data.

The datasets that are suitable considering the set requirements, i.e., that contain labelled sensors and network data, are as follows: TON_IoT [8], SWaT [9], and HAI [10]. The authors conducted a more detailed analysis of these datasets.

3.1. TON_IoT Dataset Analysis

The TON_IoT dataset is created by the Intelligent Security Group of the UNSW Canberra, Australia, and positioned by its authors as realistic telemetry datasets of IoT and IIoT sensors. It contains data from seven IoT devices, namely, a smart fridge, GPS tracker, smart sense motion light, remotely activated garage door, Modbus device, smart thermostat, and weather monitoring system. All the data were generated using a testbed of Industry 4.0/Industrial IoT networks developed by the authors. The data include several normal and cyber-attack events, namely, scanning, DoS, DDoS, ransomware, backdoor, data injection, cross-site scripting, password cracking attacks, and man-in-the-middle. The TON_IoT dataset incorporates the ground truth indicating normal and attack classes for binary classification, and the feature indicating the classes of attacks for multi-classification problems. Statistics on class balance for device samples from the TON_IoT dataset are presented in Table 1.

Table 1. The statistics on the TON_IoT dataset class balance by devices.

IoT Device	Normal	Attack	Total	Class Balance, %
Fridge	35,000	24,944	59,944	58/42
Garage Door	35,000	24,587	59,587	59/41
GPS Tracker	35,000	23,960	58,960	59/41
Modbus	35,000	16,106	51,106	68/32
Motion Light	35,000	24,488	59,488	59/41
Thermostat	35,000	17,774	52,774	66/34
Weather	35,000	24,260	59,260	59/41

Alsaedi et al. and Moustafa [8] also tried several popular machine learning methods to show that the TON_IoT dataset may be used to train classifiers for intrusion detection purposes. To justify the results and ensure that attacks are indeed identifiable, we have tried to follow the course of the authors' experiment with binary classification. It should be mentioned that the authors reported very high accuracy for the majority of the investigated methods (more than 0.8 for the F-measure in most cases). As we tried to follow the authors, at first we applied the same preprocessing procedures, namely, transformed categorical features with two unique values into binary ones, applied the min-max scaling technique to numeric features, and randomly split data into train and test subsamples in 80% to 20% stratified proportion.

It should be noted that during data preprocessing, we found several artefacts in the data. For example, '*temp_condition*' feature for the fridge contains values 'high', 'low', 'low',

'high', 'low', 'high' values, and '*sphone_signal*' for fridge contains 'true', 'false', '0', '1' values. As there are no special notes about that in the paper or the dataset description, we supposed that those were inaccuracies in the data and fixed them.

Figure 1 shows the correlation between features for different devices, both with each other and with the anomaly behaviour label. We can note a high correlation between the features of the dataset for a fridge, garage door, GPS tracker, and motion light. At the same time, the correlation value between these features and the label is low. The correlation of features for Modbus, thermostat, and weather is close to zero.

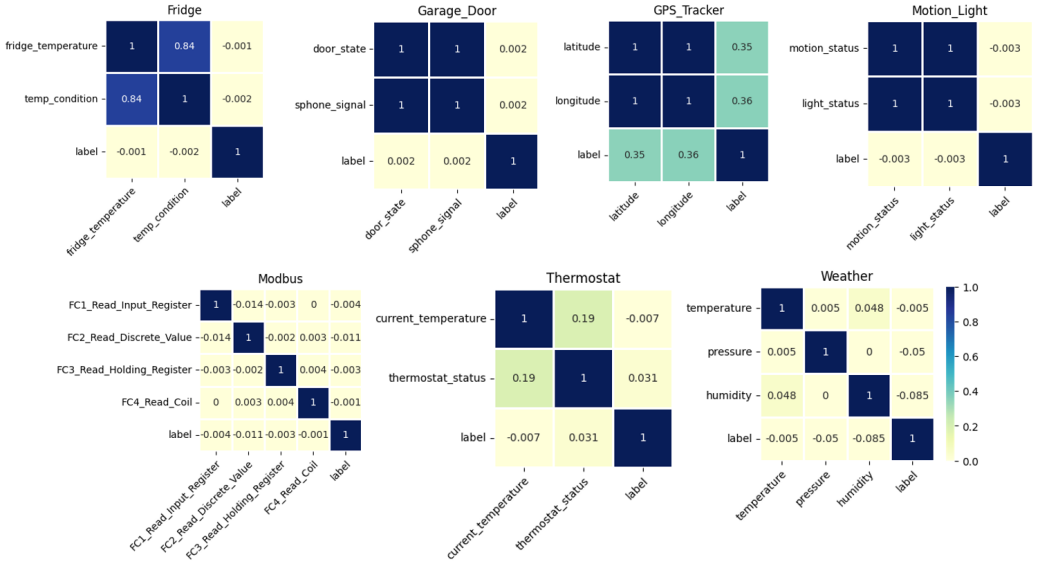


Figure 1. IoT device feature and label correlation.

We applied the same machine learning models to those mentioned in the original paper, namely, Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbour (kNN), Classification and Regression Trees (CART), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM), with the hyperparameters that authors specified, and also tried to tune those hyperparameters using 4-fold cross-validation.

We did not manage to reach the reported accuracy for most of the datasets, in either case. Table 2 shows the best values for F-measure that we received for classifiers trained on 80% of the data for each device calculated on the remaining 20% of the data.

Table 2. The F-measure values for the best hyperparameters of the model trained on the TON_IoT dataset calculated for the test subsample.

IoT Device	LR	LDA	kNN	RF	CART	NB	SVM	LSTM
Fridge	0	0	0.37	0.02	0	0	0	0
Garage Door	0.58	0	0.56	0	0	0	0	0
GPS Tracker	0.51	0.43	0.95	0.95	0.93	0.43	0.81	0.85
Modbus	0	0	0.87	0.97	0.97	0	0	0
Motion Light	0	0	0.50	0	0	0	0	0
Thermostat	0	0	0.26	0.31	0.33	0	0	0
Weather	0.10	0.10	0.95	0.98	0.97	0.53	0.58	0.61

The best F-measure values were obtained for the GPS Tracker dataset. We assume that this is due to the strongest correlation between features and anomaly class labels in this device dataset in comparison to the other device datasets. For other datasets, correlations are close to zero, that is, very weak. The strong correlations between features and the weak

correlations between features and anomaly class labels for fridge, garage door, and motion light may explain the low F-measure values for these datasets.

Further investigation of the data showed that anomaly class labels relate only to data and time of device events; although, according to the authors' experiment design, date and time are not taken into account. Figure 2 shows an example distribution of anomaly class labels in time for the temperature feature of the smart fridge.

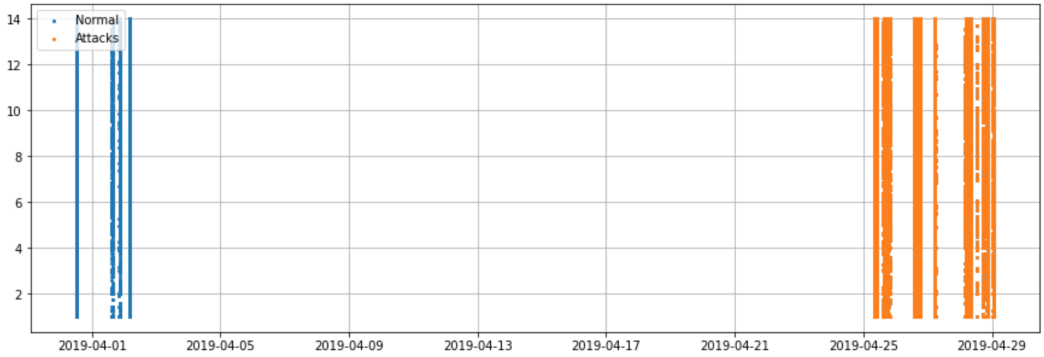


Figure 2. Normal and attack events for temperature feature of smart fridge.

Conclusions. We analyzed the obtained results considering the set requirements. Requirements R1, R2, and R4 are satisfied; requirement R3 is partially satisfied, as soon as the dataset contains attack scenarios. However, the performed experiments showed that these attacks do not affect IoT telemetry. The requirement R5 is not satisfied. The analysis demonstrated that there is no connection between the data in the network dataset and the data in the sensor's dataset. Moreover, the sensors do not follow any normal behaviour scenario and the obtained accuracy results are rather low. Thus, this data set is not suitable for the goals of further research.

3.2. SWaT Dataset Analysis

The Secure Water Treatment (SWaT) dataset [9] is generated by the Singapore University of Technology and Design (SUTD). The researchers deployed a six-stage SWaT testbed simulating a real-world industrial water treatment plant. The collected dataset contains both normal and attack traffic. It should be noticed that the deployed plant was run non-stop for eleven days: during the first seven it operated without any attacks, while during the remaining days, cyber and physical attacks were conducted against the plant. The collected dataset contains both the data from sensors and actuators of the plant (25 sensors and 26 actuators) and network traffic. Currently, there are several versions of this dataset; the researchers regularly update it by organizing cybersecurity events using it, thus, generating new data with different attack types.

We conducted a series of experiments with different machine learning models for anomaly detection using the SWaT dataset 2015 to evaluate this dataset and check its compliance with the criteria proposed above. The dataset incorporates three CSV files with anomaly (or attack) and normal data: "Attack_v0.csv", "Normal_ v0.csv", and "Normal_v1.csv". The attacks were performed on different technological processes, and Table 3 shows the number of abnormal records for different technological processes. It should be also noted that some network attacks do not impact the readings from physical sensors.

Table 3. Distribution of the attacks per processes in SWaT dataset

Record Type	Number of Impacted Processes	Impacted Processes	Number of Samples
Normal	0	0	399,157
Attack	1	P1	4053
	1	P2	1809
	1	P3	37,860
	1	P4	1700
	1	P5	1044
	2	P3, P4	1691
	2	P1, P3	1445
	2	P3, P6	697
	2	P4, P5	463

Experiment 1. For this experiment series, we tried both time and random train-test splits on the "Attack_v0.csv" dataset containing 449,919 rows in total, including 395,298 normal records and 54,621 anomaly records that correspond to attacks, meaning that the contamination rate is 0.138 for this subsample. For the time train-test split mode, the training sample was incorporated all rows before 2 January 2016 , while the testing sample contained rows after 2 January 2016 (inclusively). Due to uneven distribution of anomalies across time, the class balance for train and test subsamples was different: the train subsample included 344,436 normal instances and 51,483 attack instances meaning that the contamination rate was equal to 0.149; the test subsample included 50,862 normal instances and 3138 attack instances with a contamination rate of 0.062. The results of the experiment for the train-test split mode and different anomaly detection machine learning models are provided in Table 4. The best results were obtained for the K Nearest Neighbors method (KNN) with *F*1-measure 0.784, AUC-ROC 0.935, and AUC-PRC 0.739 on the test subsample.

Table 4. The results of Experiment 1 for the time split mode for the SWaT dataset.

Optimal Threshold	Train Data						Test Data					
	P	R	FPR	F1	AUC-ROC	AUC-PRC	P	R	FPR	F1	AUC-ROC	AUC-PRC
Sklearn												
ocSVM	0.300	0.795	0.205	0.436	0.723	0.087	0.355	0.193	0.807	0.250	0.654	0.051
isoF	0.045	0.240	0.760	0.076	0.868	0.072	0.065	0.839	0.161	0.120	0.567	0.051
PYOD												
ECOD	0.806	0.668	0.331	0.731	0.879	0.772	0.310	0.270	0.730	0.289	0.791	0.240
COPOD	0.879	0.662	0.338	0.755	0.878	0.791	0.497	0.268	0.732	0.348	0.796	0.236
KNN	0.252	0.008	0.993	0.015	0.204	0.087	0.819	0.752	0.248	0.784	0.935	0.739
Deep-SVDD	0.803	0.011	0.989	0.022	0.633	0.187	0.965	0.079	0.921	0.147	0.566	0.143
VAE	0.729	0.745	0.255	0.737	0.892	0.666	0.364	0.493	0.507	0.419	0.785	0.201
AutoEnc	0.721	0.753	0.247	0.737	0.894	0.672	0.305	0.460	0.540	0.367	0.793	0.205
AnoGAN	0.896	0.653	0.347	0.756	0.875	0.777	0.422	0.212	0.788	0.282	0.695	0.182

For the random train-test split mode, we used 80% to 20% ratio so the train subsample contained 316,238 normal instances and 43,697 attack instances, while the test subsample contained 79,060 normal instances and 10,924 attack instances with a contamination rate of 0.138 for both. The results of experiment 1 for the random train-test split mode and different anomaly detection machine learning models are provided in Table 5. It can be seen that rather close results were obtained for the ECOD (F1-measure 0.743, AUC-ROC 0.878, and AUC-PRC 0.758 on the testing sample), COPOD (F1-measure 0.744, AUC-ROC 0.874, and AUC-PRC 0.768 on the testing sample), VAE (F1-measure 0.766, AUC-ROC 0.892, and AUC-PRC 0.661 on the testing sample), AutoEnc (F1-measure 0.767, AUC-ROC 0.892, and AUC-PRC 0.660 on the testing sample), and AnoGAN (F1-measure 0.750, AUC-ROC 0.864, and AUC-PRC 0.753 on the testing sample).

Table 5. The results of Experiment 1 for the random split mode for the SWaT dataset.

Optimal Threshold	Train Data						Test Data					
	P	R	FPR	F1	AUC-ROC	AUC-PRC	P	R	FPR	F1	AUC-ROC	AUC-PRC
Sklearn												
ocSVM	0.211	0.017	0.983	0.031	0.813	0.072	0.237	0.019	0.981	0.036	0.811	0.073
isoF	0.209	0.861	0.139	0.336	0.859	0.07	0.210	0.862	0.138	0.338	0.86	0.069
PYOD												
ECOD	0.928	0.615	0.385	0.740	0.876	0.757	0.934	0.617	0.383	0.743	0.878	0.758
COPOD	0.942	0.610	0.390	0.741	0.873	0.769	0.946	0.613	0.387	0.744	0.874	0.768
KNN	0.121	1.000	0.000	0.217	0.227	0.085	0.121	0.999	0.000	0.217	0.232	0.085
Deep-SVDD	0.191	0.675	0.325	0.298	0.583	0.150	0.191	0.672	0.329	0.297	0.585	0.153
VAE	0.853	0.689	0.311	0.763	0.89	0.653	0.861	0.690	0.310	0.766	0.892	0.661
AutoEnc	0.853	0.690	0.310	0.763	0.89	0.652	0.860	0.691	0.309	0.767	0.892	0.660
AnoGAN	0.989	0.604	0.396	0.750	0.862	0.750	0.989	0.605	0.395	0.750	0.864	0.753

Experiment 2. For this experiment series, we used the data from "Attack_v0.csv" and "Normal_v0.csv" files to form train, test, and validation subsamples. The train and test subsamples incorporated all instances before 2 January 2016, with 672,989 normal instances and 41,186 attack instances for train and 168,247 normal instances and 10,297 attack instances for test (contamination is equal to 0.061 for both) after 80% to 20% stratified train test split. Meanwhile, the validation sample consisted of all instances after 2 January 2016 (inclusively), with 50,862 normal instances and 3138 attack instances and contamination of 0.062. The results of experiment 2 for different anomaly detection machine learning models are provided in Tables 6 and 7. It can be seen that rather close results are obtained for the ECOD (F1-measure 0.718, AUC-ROC 0.864, and AUC-PRC 0.530 on the testing sample), COPOD (F1-measure 0.729, AUC-ROC 0.867, and AUC-PRC 0.563 on the testing sample), VAE (F1-measure 0.732, AUC-ROC 0.896, and AUC-PRC 0.505 on the testing sample), AutoEnc (F1-measure 0.732, AUC-ROC 0.896, and AUC-PRC 0.505 on the testing sample), and AnoGAN (F1-measure 0.746, AUC-ROC 0.851, and AUC-PRC 0.555 on the testing sample).

Table 6. The results of Experiment 2 for the SWaT dataset (for train and test data).

Optimal threshold	Train data						Test data					
	P	R	FPR	F1	AUC-ROC	AUC-PRC	P	R	FPR	F1	AUC-ROC	AUC-PRC
Sklearn												
ocSVM	0.00	0.00	0.00	0.0	0.00	0.00	0.891	0.617	0.383	0.729	0.211	0.180
isoF	0.00	0.00	0.00	0.00	0.00	0.00	0.805	0.623	0.377	0.702	0.862	0.032
PYOD												
ECOD	0.862	0.623	0.377	0.724	0.865	0.540	0.856	0.619	0.381	0.718	0.864	0.530
COPOD	0.897	0.621	0.379	0.734	0.868	0.575	0.890	0.617	0.383	0.729	0.867	0.563
KNN	0.058	1.000	0.000	0.109	0.209	0.040	0.058	0.999	0.000	0.109	0.213	0.041
DeepSVDD	0.067	0.832	0.168	0.124	0.490	0.054	0.067	0.826	0.174	0.124	0.489	0.055
VAE	0.772	0.696	0.304	0.732	0.896	0.509	0.770	0.696	0.304	0.732	0.896	0.505
AutoEnc	0.772	0.696	0.304	0.732	0.896	0.509	0.770	0.696	0.304	0.732	0.896	0.505
AnoGAN	0.899	0.644	0.356	0.751	0.854	0.568	0.893	0.641	0.359	0.746	0.851	0.555

Experiment 3. The data from "Attack_v0.csv", "Normal_v0.csv", and "Normal_ v1.csv" files together were used to train algorithms in novelty detection or unsupervised mode in this experiment series. The data contain 1,441,719 instances in total, including 1,387,098 normal instances and 54,621 attack instances with contamination of 0.039. To train algorithms, all instances from "Normal_v0.csv" and "Normal_v1.csv" files (except stabilization period of 3 hours) were used, while all instances from "Attack_v0.csv" file were used for testing. The train sample included 972,000 normal instances and no attack instances. The test sample included 395,298 normal instances and 54,621 attack instances, that is, contamination is

equal to 0.139. The results of experiment 3 for the novelty detection mode and different anomaly detection machine learning models are provided in Table 8. It can be seen that the results are rather close for different models with a rather low false positive rate on the testing sample.

Table 7. The results of Experiment 2 for the SWaT dataset (for validation data).

	Validation Data				
	ACC	**P**	**R**	**FPR**	**F1**
			Sklearn		
ocSVM	0.942	0.000	0.000	1.000	0.000
isoF	0.935	0.022	0.003	0.997	0.005
			PYOD		
ECOD	0.942	0.466	0.011	0.989	0.021
COPOD	0.942	0.000	0.000	1.000	0.000
kNN	0.058	0.058	1.000	0.000	0.110
DeepSVDD	0.601	0.045	0.293	0.707	0.079
VAE	0.933	0.000	0.000	1.000	0.000
AutoEncoder	0.933	0.000	0.000	1.000	0.000
AnoGan	0.943	0.672	0.043	0.957	0.081

Table 8. The results of Experiment 3 for the SWaT dataset.

Optimal Threshold	**Train Data**	**Test Data**						
	ACC	**ACC**	**P**	**R**	**FPR**	**F1**	**AUC-ROC**	**AUC-ROC**
				Sklearn				
ocSVM	0.990	0.936	0.998	0.585	0.415	0.738	0.808	0.082
isoF	0.960	0.777	0.124	0.932	0.068	0.219	0.833	0.072
				PYOD				
ECOD	0.900	0.833	0.981	0.598	0.402	0.743	0.858	0.758
COPOD	0.960	0.919	0.948	0.619	0.381	0.749	0.855	0.756
KNN	0.960	0.127	0.987	0.636	0.364	0.774	0.816	0.727
DeepSVDD	0.960	0.766	0.991	0.646	0.354	0.783	0.838	0.732
VAE	0.960	0.410	0.991	0.633	0.368	0.772	0.820	0.732
AutoEnc	0.960	0.410	0.991	0.633	0.368	0.772	0.820	0.732

Conclusions. We analyzed the obtained results considering the dataset requirements listed above. All specified requirements are satisfied for this dataset. It is generated using physical devices and components, and this impacts the efficiency of the network attacks; not all network attacks result in changes in the readings of the sensors. Thus, we consider that this dataset is a realistic one. The preliminary results of the analysis of the sensors data are in conformance with the results obtained by other researchers [6,29,30,35,36]. Interestingly, all considered papers do not analyze network and sensor data together, and we believe that joint analysis of such data could significantly enhance the performance of the analysis models targeted to detect anomalies and network attacks.

3.3. HAI Dataset Analysis

The dataset describes the parameters of an industrial control system testbed with an embedded simulator. The testbed comprises four elements: a boiler, turbine, water-treatment component, and a hardware-in-the-loop (HIL) simulator. The HIL simulation

implements a simulation of the thermal power and pumped-storage hydropower generation.

When forming the dataset, several different attack scenarios were used, aimed at three types of devices: the Emerson Ovation, GE Mark-VIe, and Siemens S7-1500.

During the attack, the attacker operates with four types of variables: set points, process variables, control variables, and control parameters. The set of certain values of these variables in a given period of time determines one of two behaviours of the system: anomalous or normal. When the system is operating normally, the values of the process variables change within a predefined range. To this end, the operator adjusts the set point values, which allows for achieving stable and predictable results in the behaviour of the sensors, and the entire system as a whole.

This dataset has three versions: HAI 20.07, HAI 21.03, and HAI 22.04. Statistical information about each of them is given in Table 9.

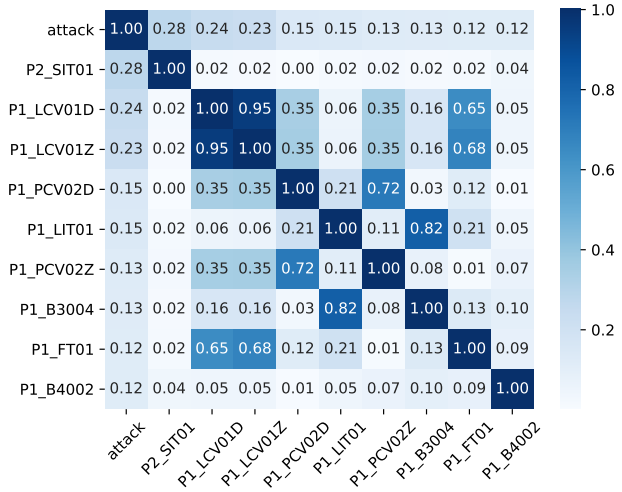
Figure 3 shows 10 features which keep the highest correlation value with the class label for files test1.csv within HAI 20.07, HAI 21.03, and HAI 22.04.

Table 10 contains the values of F-measure ($F1$) and accuracy (ACC) in percentages for 5 classifiers: decision tree (DT), KNN, random forest, logistic regression, and neural network (NN).

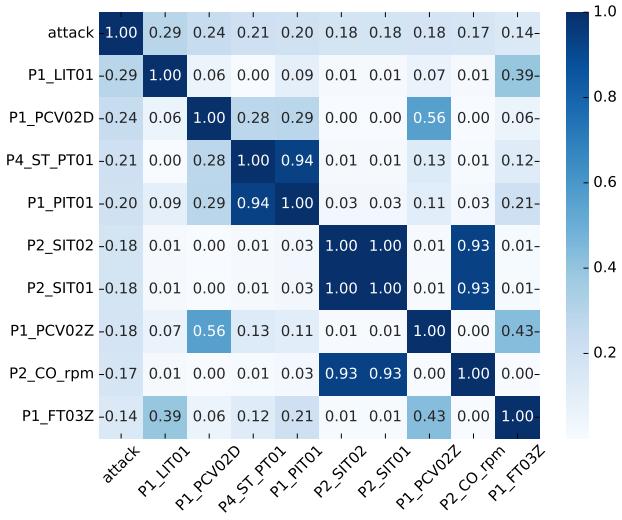
Conclusions. We analyzed the obtained results considering the set requirements. The requirements R1, R2, R3, and R4 are satisfied. The requirement R5 is also satisfied; however, considering the existence of the simulated part of the test bed, the quality of the dataset depends on the quality of the simulated part of the test bed. The preliminary experimental results are in line with the results obtained in other research papers. Thus, this dataset is consistent and suitable for the intrusion detection task.

Table 9. Statistical data on the HAI dataset class balance by version.

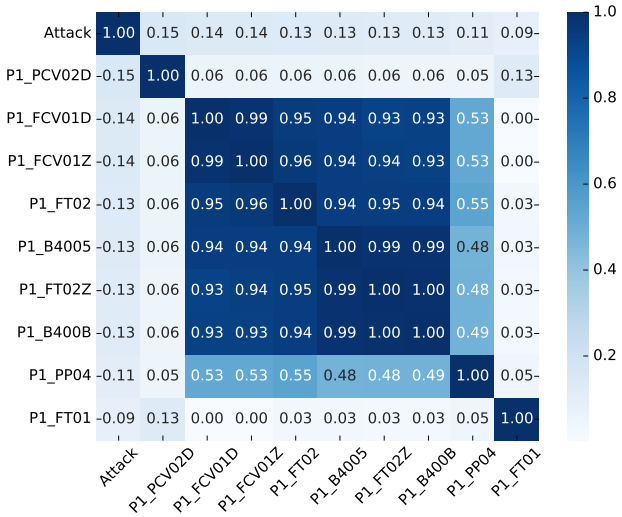
File	Normal	Attack	Total	Class Balance, %	Features
hai-20.07/train1.csv.gz	309,600	0	309,600	100/0	59
hai-20.07/train2.csv.gz	240,424	776	241,200	99.7/0.3	59
hai-20.07/test1.csv.gz	280,062	11,538	291,600	96/4	59
hai-20.07/test2.csv.gz	147,011	5989	51,106	96.1/3.9	59
hai-21.03/train1.csv.gz	216,001	0	216,001	100/0	79
hai-21.03/train2.csv.gz	226,801	0	226,801	100/0	79
hai-21.03/train3.csv.gz	478,801	0	478,801	100/0	79
hai-21.03/test1.csv.gz	42,572	629	43,201	98.5/1.5	79
hai-21.03/test2.csv.gz	115,352	3449	118,801	97.1/2.9	79
hai-21.03/test3.csv.gz	106,466	1535	108,001	98.6/1.4	79
hai-21.03/test4.csv.gz	38,444	1157	39,601	97.1/2.9	79
hai-21.03/test5.csv.gz	90,224	2177	92,401	97.6/2.4	79
hai-22.04/train1.csv	93,601	0	93,601	100/0	86
hai-22.04/train2.csv	201,600	0	201,600	100/0	86
hai-22.04/train3.csv	126,000	0	126,000	100/0	86
hai-22.04/train4.csv	86,401	0	86,401	100/0	86
hai-22.04/train5.csv	237,600	0	237,600	100/0	86
hai-22.04/train6.csv	259,200	0	259,200	100/0	86
hai-22.04/test1.csv	85,515	885	86,400	99/1	86
hai-22.04/test2.csv	79,919	2881	82,800	96.5/3.5	86
hai-22.04/test3.csv	58,559	3841	62,400	93.8/6.2	86
hai-22.04/test4.csv	125,177	4423	129,600	96.6/3.4	86



(a) HAI 20.07 dataset.



(b) HAI 21.03 dataset.



(c) HAI 22.04 dataset.

Figure 3. Features with the highest correlation value with class label.

Table 10. Result of evaluating classifiers on HAI dataset.

File	DT		KNN		RF		LR		NN	
	F1, %	ACC, %	F1, %	ACC, %	F1, %	ACC, %	F1, %	ACC, %	F1, %	ACC, %
hai-20.07/test1.csv.gz	99.00	99.85	86.42	98.28	99.67	99.95	80.88	97.73	96.70	99.50
hai-20.07/test2.csv.gz	99.48	99.92	94.93	99.30	99.76	99.96	97.29	99.60	99.30	99.90
hai-21.03/test1.csv.gz	98.61	99.91	93.39	99.59	99.31	99.95	90.10	99.43	49.57	98.29
hai-21.03/test2.csv.gz	97.60	99.73	89.89	98.99	99.52	99.95	74.99	98.03	88.52	98.81
hai-21.03/test3.csv.gz	99.38	99.96	99.61	99.61	99.38	99.96	90.96	99.53	72.23	98.76
hai-21.03/test4.csv.gz	99.45	99.94	95.65	99.52	99.67	99.96	99.22	99.91	49.25	97.05
hai-21.03/test5.csv.gz	98.26	99.84	92.72	99.38	99.30	99.94	81.30	98.63	49.39	97.59
hai-22.04/test1.csv	98.66	99.95	90.87	99.67	99.11	99.97	78.76	99.39	49.75	98.99
hai-22.04/test2.csv	97.85	99.70	89.80	98.73	99.39	99.92	72.80	97.46	49.07	96.34
hai-22.04/test3.csv	98.79	99.73	94.45	98.83	99.64	99.92	90.00	98.03	94.30	98.65
hai-22.04/test4.csv	98.38	99.78	88.26	98.65	99.45	99.93	62.88	97.02	49.12	96.53

4. Performance Metrics for Anomaly and Attack Detection

Finally, in this section, we describe performance metrics used for anomaly and attack detection. Precision, recall, and F-measure are the most used evaluation metrics. There is no specialized metric to measure the performance of anomaly detection methods. The listed metrics are classic for machine learning methods, on which most anomaly detection methods are based. However, we discovered that there are different approaches to calculating them [28,49,50]. This section reviews proposed approaches.

Let us denote the time series signal observed from K sensors during time T as

$$X = \{x_1, \ldots, x_T\}, x_t \in \mathbb{R}^N.$$

The normalized signal is divided into a number of time windows:

$$W = \{w_1, \ldots, w_{T-h+\tau}\},$$
$$w_t = \{x_t, \ldots, x_{t+h-\tau}\},$$

where h—window size, τ—step length.

The purpose of the time series anomaly detection method is to predict the binary label of the presence of an anomaly ($\hat{y}_t$), either for individual X instances or for time windows W. The labels are obtained by comparing the anomaly estimates A with a threshold δ. For the specific instances:

$$\hat{y}_t = \begin{cases} 1, & \text{if } A(x_t) > \delta, \\ 0, & \text{otherwise.} \end{cases}$$

For all windows in the test dataset:

$$\hat{y}_t = \begin{cases} 1, & \text{if } A(w_t) > \delta, \\ 0, & \text{otherwise.} \end{cases}$$

A set of test data may contain several sequences (segments) of anomalies within a certain period of time. Let us denote S as a set of M segments of anomalies:

$$S = \{S_1, \ldots, S_M\},$$
$$S_m = \{x_{t^{ms}}, \ldots, x_{t^{me}}\},$$

where t^{ms} and t^{me} are the S_m starting and ending time, accordingly.

Below, several approaches to calculate the performance metrics of anomaly detection are described.

Point-wise calculation approach. The calculation of the performance metrics is implemented using separate records within the dataset [28,49]. The calculation of precision (P), recall (R), and F-measure ($F1$) is implemented using all points within the dataset:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F1 = 2 \times \frac{P \times R}{P + R},$$

where

- TP—correctly detected anomaly ($y_t = 1, \hat{y}_t = 1$);
- FP—false detected anomaly ($y_t = 0, \hat{y}_t = 1$);
- TN—correctly assigned norm ($y_t = 0, \hat{y}_t = 0$);
- FN—false assigned norm ($y_t = 1, \hat{y}_t = 0$).

Point-adjusted (PA) calculation approach . The calculation of the performance metrics is implemented using the corrected labels. If at least one observation of an anomalous segment is detected correctly, all other observations of the segment are also considered to be correctly detected, even if they were not detected [28,49]. Observations outside the true anomaly segment are processed as usual. It can be specified as follows:

$$\hat{y}_t^{pa} = \begin{cases} 1, & \text{if } A(x_t) > \delta \text{ or } \exists A(x_{t'} > \delta), x_t, x'_t \in S_m, \\ 0, & \text{otherwise.} \end{cases}$$

The metrics are calculated considering the corrected labels in the dataset:

$$P_{pa} = \frac{TP_{pa}}{TP_{pa} + FP_{pa}}, \qquad R_{pa} = \frac{TP_{pa}}{TP_{pa} + FN_{pa}}, \qquad F1_{pa} = 2 \times \frac{P_{pa} \times R_{pa}}{P_{pa} + R_{pa}},$$

This idea is represented in Figure 4.

Figure 4. True, corrected, and predicted labels in case of the PA approach to metrics calculation.

Revised point-adjusted (RPA, event-wise) calculation approach. The calculation of metrics is implemented using time windows of records [50]. If any point at the anomaly window is labelled as anomalous, then one true positive result is fixed. If the anomalies were not labelled, then one false negative result is fixed. Any predicted anomalies outside the anomaly windows are considered false positives. This can be specified as follows:

$$P_{rpa} = \frac{TP_{rpa}}{TP_{rpa} + FP_{rpa}}, \qquad R_{pa} = \frac{TP_{rpa}}{TP_{rpa} + FN_{rpa}}, \qquad F1_{pa} = 2 \times \frac{P_{rpa} \times R_{rpa}}{P_{rpa} + R_{rpa}},$$

where

- TP_{rpa}—any part of the predicted anomaly sequence intersects with a sequence that actually has an anomaly;
- FN_{rpa}—if no sequence that is predicted to be anomalous intersects with a real anomalous sequence;
- FP_{rpa}—all predicted anomalous sequences that do not intersect with any really anomalous sequence.

This idea is represented in Figure 5.

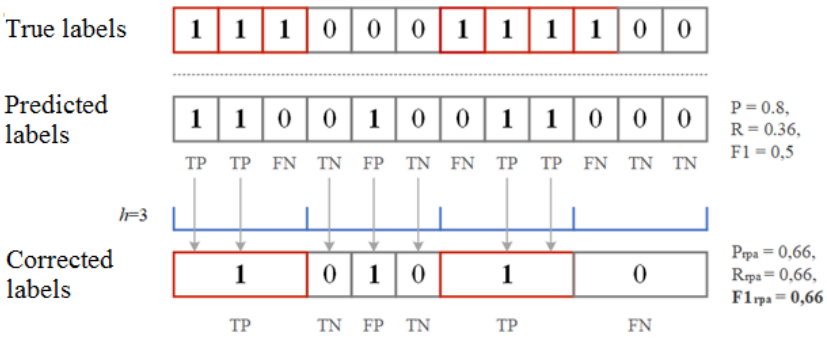


Figure 5. True, predicted, and corrected labels in case of the RPA approach to the metrics calculation.

Another metric is the composite $F1$ score [50]. For this metric, precision is considered as P (by the number of points), and recall is calculated as R_{rpa} (by the number of segments):

$$F1_c = 2 \times \frac{P \times R_{rpa}}{P + R_{rpa}}.$$

This idea is represented in Figure 6.

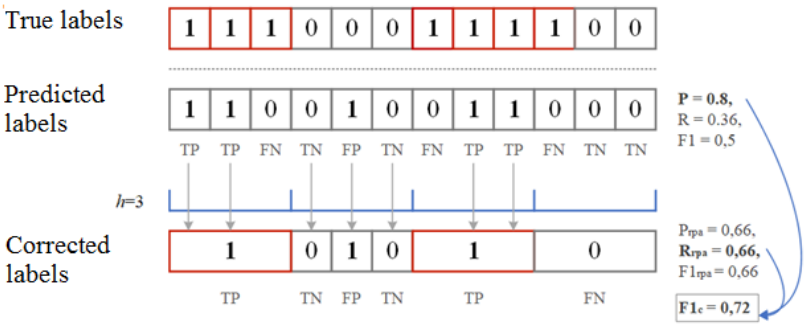


Figure 6. True, predicted, and corrected labels in case of the composite $F1$ score approach to the metrics calculation.

Conclusions. There are various approaches to the calculation of metrics for performance evaluation of the machine learning models. In addition to the classical way of calculating through TP, TN, FN, and FP, researchers present options with adjusted indicators. This is aimed at improving the quality of anomaly detection in a large amount of data, or at reducing the number of false positives. In this case, the choice of metrics strongly depends on the detection problem being solved. To select the appropriate approach to calculation, additional experiments are required: the authors plan to implement and compare all the described metrics in future experiments with anomaly detection methods.

5. Conclusions

In the paper, the authors considered existing approaches in the anomaly detection area, existing datasets that can be used for the experiments, and existing performance metrics. The analysis of the related works showed that the research focus has shifted to the application of deep neural networks to anomaly detection in technological processes; however, there are still solutions based on classical anomaly detection techniques. The application of machine learning techniques requires high-quality datasets. High-quality datasets are datasets that are relevant to the subject domain, meaningful, and reliable. We formulated five requirements for the datasets that consider these properties and evaluated three different datasets that are currently proposed for testing and evaluation of cybersecurity applications. The selected datasets are SWaT, HAI, and TON_IoT. Our experiments revealed that TON_IoT is not suitable for the intrusion detection task, as we did not dis-

cover any relations between sensor data and network data. We consider that SWaT and HAI datasets are more relevant for cybersecurity tasks, primarily due to the fact that they were generated using real physical test beds. The SWaT dataset contains both network and sensor data; this makes it preferable for intrusion detection, as authors believe that joint analysis of the network and sensor data could benefit the early detection of the attacks a lot, including multi-step attacks.

Another interesting finding relates to the performance evaluation of the machine learning techniques proposed to detect anomalies. These techniques consider the specificity of the anomalies of the CPS systems—their duration and the delayed response of the system. Although these features could significantly enhance the evaluation process of the proposed cybersecurity solutions, they require more analysis and research.

Finally, in future research, the authors plan to develop an approach to anomaly detection in cyber-physical systems that will provide accurate and explainable results, and will conduct experiments to select the performance metrics.

Author Contributions: Conceptualization, E.N., E.F. and I.K.; methodology, E.N. and E.F.; software, O.T., D.L. and A.B.; validation, E.N., E.F., O.T. and I.K.; investigation, E.F., E.N., O.T., D.L. and A.B.; writing—original draft preparation, E.F.; writing—review and editing, E.N., O.T., D.L. and A.B.; visualization, D.L. and A.B.; supervision, I.K., E.N. and E.F.; funding acquisition, I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research is being supported by the grant of RSF #21-71-20078 in SPC RAS.

Data Availability Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Levshun, D.; Chechulin, A.; Kotenko, I. Design of Secure Microcontroller-Based Systems: Application to Mobile Robots for Perimeter Monitoring. *Sensors* **2021**, *21*, 8451. [CrossRef]
2. Turton, W.; Mehrotra, K. Hackers Breached Colonial Pipeline Using Compromised Password. 4 June 2021. Available online: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password (accessed on 20 December 2022). .
3. Jones, S. Venezuela Blackout: What Caused It and What Happens Next. *The Guardian* 13 March 2019. Available online: https://www.theguardian.com/world/2019/mar/13/venezuela-blackout-what-caused-it-and-what-happens-next (accessed on 20 December 2022).
4. Graham, R. Cyberattack Hits Germany's Domestic Fuel Distribution System. 1 February, 2022. Available online: https://www.bloomberg.com/news/articles/2022-02-01/mabanaft-hit-by-cyberattack-that-disrupts-german-fuel-deliveries (accessed on 20 December 2022).
5. Kim, S.; Jo, W.; Shon, T. APAD: Autoencoder-based payload anomaly detection for industrial IoE. *Appl. Soft Comput.* **2020**, *88*, 106017. [CrossRef]
6. Wang, C.; Wang, B.; Liu, H.; Qu, H. Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8897926:1–8897926:10. [CrossRef]
7. Kotenko, I.; Gaifulina, D.; Zelichenok, I. Systematic Literature Review of Security Event Correlation Methods. *IEEE Access* **2022**, *10*, 43387–43420. [CrossRef]
8. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
9. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A dataset to support research in the design of secure water treatment systems. In Proceedings of the Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, 10–12 October 2016; Revised Selected Papers 11; Springer: New York, NY, USA, 2017; pp. 88–99.
10. Shin, H.K.; Lee, W.; Yun, J.H.; Kim, H. HAI 1.0: HIL-based augmented ICS security dataset. In Proceedings of the 13th USENIX Conference on Cyber Security Experimentation and Test, Boston, MA, USA, 10 August 2020; p. 1.
11. Meleshko, A.; Shulepov, A.; Desnitsky, V.; Novikova, E.; Kotenko, I. Visualization Assisted Approach to Anomaly and Attack Detection in Water Treatment Systems. *Water* **2022**, *14*, 2342. [CrossRef]
12. Shulepov, A.; Novikova, E.; Murenin, I. Approach to Anomaly Detection in Cyber-Physical Object Behavior. In *Intelligent Distributed Computing XIV*; Camacho, D., Rosaci, D., Sarné, G.M.L., Versaci, M., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 417–426.
13. Khan, A.A.; Beg, O.A.; Alamaniotis, M.; Ahmed, S. Intelligent anomaly identification in cyber-physical inverter-based systems. *Electr. Power Syst. Res.* **2021**, *193*, 107024. [CrossRef]

14. Parto, M.; Saldana, C.; Kurfess, T. Real-time outlier detection and Bayesian classification using incremental computations for efficient and scalable stream analytics for IoT for manufacturing. *Procedia Manuf.* **2020**, *48*, 968–979. [CrossRef]
15. Mohammadi Rouzbahani, H.; Karimipour, H.; Rahimnejad, A.; Dehghantanha, A.; Srivastava, G. Anomaly detection in cyber-physical systems using machine learning. In *Handbook of Big Data Privacy*; Springer: New York, NY, USA, 2020; pp. 219–235.
16. Mokhtari, S.; Abbaspour, A.; Yen, K.K.; Sargolzaei, A. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics* **2021**, *10*, 407. [CrossRef]
17. Park, S.; Lee, K. Improved Mitigation of Cyber Threats in IIoT for Smart Cities: A New-Era Approach and Scheme. *Sensors* **2021**, *21*, 1976. [CrossRef] [PubMed]
18. Elnour, M.; Meskin, N.; Khan, K.; Jain, R. A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems. *IEEE Access* **2020**, *8*, 36639–36651. [CrossRef]
19. Gad, A.R.; Haggag, M.; Nashat, A.A.; Barakat, T.M. A Distributed Intrusion Detection System using Machine Learning for IoT based on ToN-IoT Dataset. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 548–563. [CrossRef]
20. Kumar, P.; Tripathi, R.; Gupta, G.P. P2IDF: A privacy-preserving based intrusion detection framework for software defined Internet of Things-fog (SDIoT-Fog). In Proceedings of the Adjunct 2021 International Conference on Distributed Computing and Networking, Nara, Japan, 5–8 January 2021; pp. 37–42.
21. Huč, A.; Šalej, J.; Trebar, M. Analysis of machine learning algorithms for anomaly detection on edge devices. *Sensors* **2021**, *21*, 4946. [CrossRef] [PubMed]
22. Inoue, J.; Yamagata, Y.; Chen, Y.; Poskitt, C.M.; Sun, J. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), Orleans, LA, USA, 18–21 November 2017; pp. 1058–1065. [CrossRef]
23. Gaifulina, D.; Kotenko, I. Selection of deep neural network models for IoT anomaly detection experiments. In Proceedings of the 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Valladolid, Spain, 10–21 March 2021; IEEE: Hoboken, NJ, USA, 2021; pp. 260–265.
24. Shalyga, D.; Filonov, P.; Lavrentyev, A. Anomaly Detection for Water Treatment System based on Neural Network with Automatic Architecture Optimization. *arXiv* **2018**, arXiv:1807.07282.
25. Xie, X.; Wang, B.; Wan, T.; Tang, W. Multivariate abnormal detection for industrial control systems using 1D CNN and GRU. *IEEE Access* **2020**, *8*, 88348–88359. [CrossRef]
26. Nagarajan, S.M.; Deverajan, G.G.; Bashir, A.K.; Mahapatra, R.P.; Al-Numay, M.S. IADF-CPS: Intelligent Anomaly Detection Framework towards Cyber Physical Systems. *Comput. Commun.* **2022**, *188*, 81–89. [CrossRef]
27. Fan, Y.; Li, Y.; Zhan, M.; Cui, H.; Zhang, Y. IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT. In Proceedings of the 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), Guangzhou, China, 29 December 2020–1 January 2021; pp. 88–95. [CrossRef]
28. Audibert, J.; Michiardi, P.; Guyard, F.; Marti, S.; Zuluaga, M.A. USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In Proceedings of the KDD'20, 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 3395–3404. [CrossRef]
29. Li, D.; Chen, D.; Shi, L.; Jin, B.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019.
30. Neshenko, N.; Bou-Harb, E.; Furht, B. A behavioral-based forensic investigation approach for analyzing attacks on water plants using GANs. *Forensic Sci. Int. Digit. Investig.* **2021**, *37*, 301198. [CrossRef]
31. Wu, P.; Moustafa, N.; Yang, S.; Guo, H. Densely connected residual network for attack recognition. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; IEEE: Hoboken, NJ, USA, 2020, pp. 233–242.
32. Bian, X. Detecting Anomalies in Time-Series Data using Unsupervised Learning and Analysis on Infrequent Signatures. *J. IKEEE* **2020**, *24*, 1011–1016.
33. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the NIPS'17, 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 4768–4777.
34. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the KDD'16, 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1135–1144. [CrossRef]
35. Lin, Q.; Adepu, S.; Verwer, S.; Mathur, A. TABOR: A Graphical Model-Based Approach for Anomaly Detection in Industrial Control Systems. In Proceedings of the ASIACCS'18, 2018 on ACM Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 4–8 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 525–536. [CrossRef]
36. Sukhostat, L. Anomaly Detection in Industrial Control System Based on the Hierarchical Hidden Markov Model. In *Cybersecurity for Critical Infrastructure Protection via Reflection of Industrial Control Systems*; IOS Press: Amsterdam, The Netherlands, 2022; pp. 48–55.

37. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
38. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Funchal, Portugal, 22–24 January 2018; pp. 108–116. [CrossRef]
39. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; IEEE: Hoboken, NJ, USA, 2015; pp. 1–6.
40. Qin, Y.; Kondo, M. Federated Learning-Based Network Intrusion Detection with a Feature Selection Approach. In Proceedings of the 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, 12–13 June 2021; pp. 1–6. [CrossRef]
41. Murenin, I.; Doynikova, E.; Kotenko, I. Towards Security Decision Support for large-scale Heterogeneous Distributed Information Systems. In Proceedings of the 2021 14th International Conference on Security of Information and Networks (SIN), Edinburgh, UK, 15–17 December 2021; Volume 1, pp. 1–8. [CrossRef]
42. Choi, S.; Yun, J.H.; Kim, S.K. A Comparison of ICS Datasets for Security Research Based on Attack Paths. In Proceedings of the CRITIS, Kaunas, Lithuania, 24–26 September 2018.
43. Lemay, A.; Fernandez, J.M. Providing SCADA Network Data Sets for Intrusion Detection Research. In Proceedings of the 9th Workshop on Cyber Security Experimentation and Test (CSET 16), Austin, TX, USA, 8 August 2016; USENIX Association: Austin, TX, USA, 2016.
44. Rodofile, N.R.; Schmidt, T.; Sherry, S.T.; Djamaludin, C.; Radke, K.; Foo, E. Process Control Cyber-Attacks and Labelled Datasets on S7Comm Critical Infrastructure. In *Information Security and Privacy*; Pieprzyk, J., Suriadi, S., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 452–459.
45. Suthaharan, S.; Alzahrani, M.; Rajasegarar, S.; Leckie, C.; Palaniswami, M. Labelled data collection for anomaly detection in wireless sensor networks. In Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, Australia, 7–10 December 2010; pp. 269–274. [CrossRef]
46. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C.; Vishwanath, A.; Sivaraman, V. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Trans. Mob. Comput.* **2019**, *18*, 1745–1759. [CrossRef]
47. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.P. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
48. Hamza, A.; Gharakheili, H.H.; Benson, T.A.; Sivaraman, V. Detecting Volumetric Attacks on loT Devices via SDN-Based Monitoring of MUD Activity. In Proceedings of the 2019 ACM Symposium on SDN Research, San Jose, CA, USA, 3–4 April 2019.
49. Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; et al. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In Proceedings of the WWW'18, 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2018; pp. 187–196. [CrossRef]
50. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In Proceedings of the KDD'18, 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 387–395. [CrossRef]