

# Anomaly detection system based on deep learning for cyber physical systems on sensory and network datasets

Muhammed Almendli, Jamshid Bagherzadeh Mohasefi

Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran

## Article Info

### Article history:

Received May 15, 2024

Revised Jul 19, 2024

Accepted Aug 6, 2024

### Keywords:

Anomaly detection  
Cyber physical systems  
Deep learning  
Intrusion detection  
Machine learning  
Network security

## ABSTRACT

Cyber-physical systems (CPSs), a type of computing system integrated with physical devices, are widely used in many areas such as manufacturing, traffic control, and energy. The integration of CPS and networks has expanded the range of cyber threats. Intrusion detection systems (IDSs), use signature based and machine learning based techniques to protect networks, against threats in CPSs. Water purifying plants are among the important CPSs. In this context some research uses a dataset obtained from secure water treatment (SWaT) an operational water treatment testbed. These works usually focus solely on sensory dataset and omit the analysis of network dataset, or they focus on network information and omit sensory data. In this paper we work on both datasets. We have created IDSs using five traditional machine learning techniques, decision tree, support vector machine (SVM), random forest, naïve Bayes, and artificial neural network along with two deep methods, deep neural network, and convolutional neural network. We experimented with IDSs, on three different datasets obtained from SWaT, including network data, sensory data, and Modbus data. The accuracies of proposed methods show higher values on all datasets especially on sensory (99.9%) and Modbus data (95%) and superiority of random forest and deep learning methods compared to others.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Jamshid Bagherzadeh Mohasefi

Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Urmia University  
11 km Serow Road, Urmia, West Azerbaijan, Iran

Email: j.bagherzadeh@urmia.ac.ir

## 1. INTRODUCTION

Cyber physical systems (CPSs) are different from the usual systems in the sense that they are specifically designed for controlling industrial systems that work at critical environments [1]. These systems are usually comprised of industrial equipment monitored by many sensors and controlled using many actuators. These sensors and actuators are communicated through programmable logic controller (PLC) and supervisory control and data acquisition (SCADA) systems. Deploying sensor networks in plants' interiors allows for the collection of crucial information to ensure the safe and proper operation of the physical plants. This information enables plant staff to react in real-time to any changes in the plants [2]. To name some, we have public infrastructures such as water purifying and treatment factories, power grids, transportation systems, manufacturing factories, traffic control systems, energy, and safety management [3]. Usually, such systems are geographically distributed and large, and for remote control and monitoring, they are connected through a network. Such connectivity open the possibility of cyber-attacks. This makes it necessary to apply techniques to protect CPSs against various threats: cyber threats or physical threats. Cyber-attacks are attacks that are transmitted through communication links. Physical attacks are making damages on physical devices such as motors or pumps to disrupt system functionality.

Intrusion detection systems (IDSs) are among the most important defense mechanisms against the evolving and complex network attacks [4]. An IDS is a technological tool that observes incoming and outgoing network traffic for signs of unauthorized activity and violations of rules. The primary goal of an IDS is to identify and stop unauthorized access within an information technology (IT) framework and notify the appropriate individuals. These tools can come in the form of hardware devices or software programs. Generally, an IDS serves as the initial layer of protection in a more extensive security information and event management (SIEM) system.

IDSs use various detection methods, including signature-based intrusion detection systems (SIDS), which aim to identify patterns and match them with known signs of intrusions. Anomaly-based intrusion detection systems (AIDS), on the other hand, leverage machine learning (ML) and statistical data analysis to create a model of “normal” behavior, flagging any traffic that deviates from the norm as suspicious. A hybrid system combines features from both methods to maximize effectiveness. All intrusion detection systems analyze some sort of data. A SIDS needs data of previous attacks to keep in database and an AIDS needs normal and attack data to create a ML model. After all they have to detect attacks from current data gathered online from various sources. To detect intrusion, they need to analyze current data against previous information (stored in a database or a model).

Intrusion detection in cyber physical systems (CPSs) includes many challenges [5]. These challenges include data gathering, keeping data privacy, adaptability to specific CPS, performance, coping with new zero-day attacks, and many other issues. Although it is difficult to implement a system which can provide all of these capabilities with high efficiency but there are many research done on subsets of these aspects with acceptable performance. There are a variety of AI-based intrusion detection methods which have been proposed to provide CPS security.

In [6], a multi-feature data clustering model was used to propose an intrusion detection algorithm for industrial networks. The algorithm involves calculating weighted distances of data and classifying security coefficients based on the priority of data features of network nodes. By doing so, the proposed algorithm aims to enhance the detection rate and real-time performance of identifying abnormal behavior in industrial networks with multi-feature data. The algorithm introduces two innovative aspects: quickly selecting a node with a high-security coefficient as the cluster center and grouping the multi-feature data around the center into a cluster.

In the field of cyber-physical systems, machine learning algorithms face criticism due to the difficulty in detecting novel attacks caused by the lack of labeled data. To address this challenge, De Araujo-Filho *et al.* [7] suggest using a generative adversarial network (GAN) as an unsupervised approach to detect cyber-attacks by implicitly modeling the system. Their article presents Fréchet inception distance-generative adversarial network (FID-GAN), a novel fog-based, unsupervised IDS designed for cyber-physical systems using GANs. The IDS is intended for a fog architecture, which aims to bring computation resources closer to the end nodes to meet low-latency requirements.

In study [8], a deep learning approach was utilized to develop a solution for detecting industrial traffic anomalies and classifying attacks. The detection model employed a representation model based on convolutional neural networks. Through the use of a feature mapping method, the original one-dimensional data was transformed to be compatible with the model processing. Leveraging the deep learning method enabled automatic extraction of crucial features, leading to precise attack classification. The model underwent evaluation using network attack data from a supervisory control and data acquisition (SCADA) system.

In [9], a system is introduced for protecting SCADA systems within power network infrastructures. This new system examines several attributes to offer a comprehensive solution that can address various cyber threats. The multi-attribute IDS incorporates a diverse whitelist and behavior-based approach to enhance the security of SCADA cyber-systems. Furthermore, a multilayer cybersecurity framework based on IDS is suggested to safeguard SCADA systems in smart grids without affecting the availability of normal data. Lastly, the paper describes a specific cybersecurity testbed for SCADA to study simulated attacks, which has been utilized to validate the proposed method.

There are many other works on CPS intrusion detection ([10]–[12]) which use general datasets like NSL-KDD [13], UNSW-NB15 [14] or other general purpose intrusion detection testbed datasets. The latest research indicates that existing datasets do not accurately capture modern cyber-physical systems (CPSs) network traffic and footprint attacks due to the current threat landscape. Additionally, CPS users are reluctant to share private samples of various attacks due to security and privacy concerns. Moreover, the ever-changing nature of unknown cyberattacks makes it challenging to obtain samples [15]. To alleviate this issue, fortunately some attempts has been done to generate specific datasets for intrusion detection in CPSs [16] such as secure water treatment (SWaT) [17] and water distribution testbed (WDT) [18].

In [19], a system for detecting anomalies in a CPS for water treatment plants is introduced to prevent damage and disruption in service. The system features an anomaly detector based on a multi-layer

perceptron (MLP) that utilizes an unsupervised approach to protect the system from the harmful effects of cyber-attacks. To train the proposed detector, data collected during the normal operation of the plant was used. The effectiveness of their approach was validated through experiments using the SWaT dataset. Fortunately, the SWaT testbed provides two types of data: the Sensory dataset, which contains information from sensors or actuators of the CPS system, and the network dataset, which depicts the packets transferred in http/https protocol between network nodes. To design an accurate anomaly detection system in CPS systems, many machine learning techniques have been proposed that experiment on SWaT dataset [17]–[19]. Although acceptable performance is achieved by most of the algorithms, but they only work on some subset of attacks or some part of dataset. Most of the work done on CPS on SWaT dataset concentrate only on Sensory dataset. Table 1 represents the IDS systems experimented on SWaT dataset with their features.

Table 1. Advantages and limitations of previous works on SWaT dataset

Ref.	Technique	Dataset	Advantages	Limitations
[19]	Probabilistic neural network (PNN)	SWaT dataset, (sensory data only)	Detecting rate is good	Training time high, No zero-day detection, only sensory dataset
[20]	FR, BR Tree, NN, J48, SVM	SWaT dataset, (sensory data only)	Noise-robust, detection rate is good, false alarms is low	Not detecting zero-day attacks, only sensory data
[21]	MLP	SWaT dataset, (sensory data)	Precision is good	Recall is low, No zero-day detection, only sensory data
[22]	Recursive principal component analysis (RPCA)	SWaT dataset (sensory data)	Precision is good	Recall is low, No zero-day detection, only sensory data
[23]	Support vector machine (SVM)+learning using privileged information (LUPI)	SWaT dataset (network data)	Using network dataset, using privilege information	Accuracy is low, only network dataset, no zero-day detection
[24]	Autoencoders	SWaT dataset (sensory data)	Detection rate and false alarm rate are good	High delay in attack detection, no zero-day detection, only sensory dataset

In this paper we contribute the following novelties: i) We implemented five machine learning algorithms including random forest, support vector machine, naïve Bayes, decision tree, and artificial neural network on sensory dataset as well as network dataset; ii) We extracted Modbus protocol data from the network data and implemented the above machine learning algorithms on Modbus data; iii) Deep neural networks (DNNs) and convolutional neural networks (CNNs) have been applied on network dataset, sensory dataset, and Modbus dataset as well; and iv) We have got very high accuracy owing to a good pre-processing on dataset. This paper is organized as follows. In section two, we explain the data preparation and preprocessing operations. Then in section three, we explain the proposed methods including deep and machine learning techniques; and we show the results of the implemented methods in section four. Finally, section five presents the discussion and concludes the paper.

## 2. DATA PREPARATION AND PREPROCESSING

An invaluable resource for developing and testing new anomaly and intrusion detection methods, as well as for training and validating machine learning algorithms, would be a comprehensive and representative dataset of intrusions in CPSs. Currently, there are only a limited number of datasets available to assess ML-based anomaly detection in CPSs [25]. However, some of these datasets are founded on impractical implementations. Certain CPS testbeds lack essential details or have implemented them incorrectly, which could affect the effectiveness and accuracy of anomaly detection methods. Usually, datasets for CPSs consist of a lot of information including Network traffic gathered using devices installed in the network to catch packets transferred in the network. There could be another type of data which represents values of sensors and actuators gathered in time shots. There might be another form of data depending on the types of CPSs.

A small-scale, but operational water treatment dataset (SWaT) is created by Mathur and Tippenhauer [17] used for cybersecurity research. Many attack scenarios are designed for this testbed. The final dataset was released including state variables, packet features, and logs of attacks. SWaT has six main processes to purify water [17], and includes a SCADA station connected to programmable logic controllers (PLCs) thorough communication network. Sensors' data is transmitted to the SCADA station and recorded for future use. In the subsequent sections we call this sensory dataset (SD). The SCADA system communicates with the PLCs via a communication network. The data gathered from sensors and actuators are sent to PLCs through Fieldbus protocol. Then this data is transferred through TCP/IP protocol to SCADA system. The TCP packets gathered from communication of PLCs and SCADA system is called network

dataset (ND) in the subsequent sections. Inside the network data there is a value known as *Modbus\_value* which contains the encoded information of sensors and actuators communicated between SCADA and PLCs. This invaluable feature of every packet is called Modbus dataset (MD) meanwhile. There are a few papers which have analyzed the ND and MD datasets exclusively.

During the data collection process, 36 different attacks were launched in the system. Sensory data is recorded once a second. In SD dataset, a total of 946,722 sample records were collected, belonging to 11 days, which comprises of 51 attributes. In Table 2 some of the features (10 out of 51) are mentioned. The central tendency of all features including min, max, mean, median, and mode are represented in the table as well.

Network traffic (ND dataset) collected using available equipment for network analysis. This data is bigger in size (number of rows) as it is collected in time intervals of milliseconds. Corresponding to each row in the SD there exists hundreds of rows (maximum 1,000) in ND. There are 36 attacks available in datasets (three of them are presented in Table 3). This table shows the start time, end time, attack point, the start state before attack, the attack, and its consequences.

Table 2. Some feature of sensory dataset with central tendency (for first two processes of SWaT)

No	Name	Type	Min	Max	Median	Mode	Mean
1	FIT 101	Sensor	0	2.76	1.714346	2.47702	0
2	LIT 101	Sensor	189.83	925.03	607.02	530.4225	700
3	MV 101	Actuator	0	2	1.66	2	2
4	P 101	Actuator	1	2	1.69	2	2
5	P 102 (backup)	Actuator	1	2	1.01	1	1
6	AIT 201	Sensor	168.03	267.72	210.30	193.51	176.59
7	AIT 202	Sensor	6	8.73	8.53	8.55	8.61
8	AIT 203	Sensor	285.34	384.46	320.30	321.66	333.63
9	P 205	Actuator	1	2	1.69	2	2
10	P 206 (backup)	Actuator	1	2	1.01	1	1

Table 3. Some sample attacks on the dataset

No	Start Time	End Time	Attacked device	Start State	Attack description	Change?	Attacker's intent
1	28-12-2015 10:29:14	10:44:5 3	MV_101	MV_101 is closed	Open MV_101	Yes	Overflow the tank
2	28-12-2015 10:51:08	10:58:3 0	P_102	P_101 is on P_102 is off	Turn on P_102	Yes	Burst pipes
3	28/12/2015 11:22:00	11:28:2 2	LIT_101	Water level between L and H	Increased 1 mm every second	No	Underflow tank damage P_101

### 2.1. Data preprocessing for sensory dataset (SD)

Data preprocessing includes different actions such as, data cleaning (missing value processing, noisy values, and outliers), data normalization, data integration, and data reduction. Proposed by IBM data analytics, we may spend up to 80% of our time cleaning data. After removing outliers, and smoothing missing values, we normalized the features using *MinMaxScaler* from sklearn, which converts all the values to the range of [0, 1]. Then we performed a feature selection using various methods such as correlation analysis. After careful analysis we selected features with the most direct or inverse relation with the target label.

### 2.2. Data statistics and preprocessing of network dataset (ND)

Various data cleaning operations are done on network dataset which includes features mentioned in the Table 4. The types and instances for features are represented in the table. We have different types of features which need transformations. First, we remove the features like *Date*, *Time*, *Modbud\_transaction\_id* which are not important in attack detection. Next, we remove instances with missing values. These preprocessing steps have been done on data of day Dec. 28, 2015 (1,581,399 instances) as the whole network dataset is huge.

For further processing we consider only the features with more than one possible value. These include features: *'if\_dir,' 'src,' 'dst,' 'proto,' 'appi\_name,' 'proxy\_src\_ip,' 'Modbus\_Function\_Description,' 'SCADA\_Tag,' 'Modbus\_Value,' 'service,' 's\_port,'* and *'Tag.'* Next, we convert categorical features (like *IP addresses, if\_dir, ...*) to numeric values for further processing. This is done using *ce.BinaryEncoder()* function from *category\_encoders* package. There exists an important attribute, *Modbus\_Value*, which

contains a list of 38 hexadecimal numbers, showing the values read or written to sensors and actuators using the Modbus protocol. Modbus protocol is usually used for communication in PLC devices. We can convert this value to a list of decimal numbers. Finally, we get a dataset with 20 numeric features for all except *Modbus\_Value*, and 38 for *Modbus\_Value*.

Table 4. Features of network dataset

Feature name	Tag	i/f_dir	appi_name
Number of unique values	2	2	3
Feature name	Orig	Src	proxy_src_ip
Number of unique values	1	7	7
Feature name	type	dst	Modbus_Function_Code
Number of unique values	1	8	1
Feature name	i/f_name	proto	Modbus_Function_Description
Number of unique values	1	2	2
Feature name	SCADA_Tag	Modbus_Value	Services_port
Number of unique values	5	21349	312

### 3. METHOD

Five classical ML methods and two DL methods are applied in this paper. There are three datasets prepared after careful preprocessing and removing non-important features (SD, ND, and MD datasets). Each ML or DL method is applied on all three datasets yielding 21 experiments. In the following subsections we illustrate each scenario with details.

#### 3.1. ML and DL methods using SD

Decision tree (DT), a well-known machine learning algorithm, is utilized for regression and classification tasks. Its interpretability and implementation are user-friendly, and it produces good results in many applications. We have used DT to train many models for SWaT dataset with various parameters.

The concept of an artificial neural network (ANN) is based on the neural structure of the human brain, with layers of interconnected nodes (neurons) forming the computational model. ANN is applied using one day data (with 4000 records) and full sensory dataset. The structure of the network is as follows:

```
model = tf.keras.Sequential()
model.add(layers.Dense(36, input_shape = (51, ), activation = 'relu'))
model.add(layers.Dense(1, activation = 'sigmoid'))
```

Support vector machine (SVM), a robust supervised algorithm, is most effective when applied to smaller datasets. SVM is suitable for both regression and classification tasks, but it typically performs better in classification problems. We applied SVM method to classify SWaT dataset. The first experiment is done using one day data (with 4000 records) and the second experiment of SVM is done using full SD.

The random forest (RF) algorithm is extensively used in machine learning. It aggregates the results of numerous decision trees to produce a single output. Its versatility and user-friendly nature have contributed to its widespread acceptance, as it is capable of addressing both classification and regression issues. One of its key strengths is its capacity to manage intricate datasets and reduce overfitting, rendering it an invaluable resource for a variety of predictive tasks in machine learning. In this paper we have applied RF to classify SWaT dataset using one day data (with 4,000 records) and the full SD dataset.

The naive Bayes classifier is a machine learning model that relies on Bayes' theorem and operates based on probabilities. It assumes that features are independent of each other and determines the likelihood of a specific input belonging to a certain class. This model is extensively employed in tasks such as text classification, spam detection, and recommendation systems. We have applied the naive Bayes (NB) method to classify SWaT dataset using one day data (with 4,000 records) and full sensory dataset.

Deep neural network (DNN) is a kind of ANN which has more than one hidden layer. In recent years DNNs have been applied extensively in many applications including intrusion detection. Variants of these networks have shown high accuracies in applications like speech recognition, image processing, and natural language processing. We applied DNN with structure as Figure 1(a) for intrusion detection in SWaT dataset.

Convolutional neural network (CNN) is a kind of DNN network which has special properties called receptive fields, shared weights, and pooling. CNNs have got very good results in many applications, especially on image processing. We applied CNN with structure as Figure 1(b) for intrusion detection in SWaT dataset.

Layer (type)	Output Shape	Param #
=====		
dense (Dense)	(None, 51)	2652
dense_1 (Dense)	(None, 36)	1872
dense_2 (Dense)	(None, 10)	370
dense_3 (Dense)	(None, 1)	11
=====		
Total params: 4905 (19.16 KB)		
Trainable params: 4905 (19.16 KB)		
Non-trainable params: 0 (0.00 Byte)		

(a)

Layer (type)	Output Shape	Param #
=====		
conv1d_1 (Conv1D)	(None, 46, 32)	224
max_pooling1d_1 (MaxPooling1D)	(None, 23, 32)	0
conv1d_2 (Conv1D)	(None, 21, 64)	6208
max_pooling1d_2 (MaxPooling1D)	(None, 10, 64)	0
flatten_1 (Flatten)	(None, 640)	0
dense_1 (Dense)	(None, 10)	6410
dense_2 (Dense)	(None, 1)	11
=====		
Total params: 12853 (50.21 KB)		
Trainable params: 12853 (50.21 KB)		
Non-trainable params: 0 (0.00 Byte)		

(b)

Figure 1. Structure of DNN and CNN on sensory dataset (a) Structure of DNN on SD dataset and (b) structure of CNN on SD dataset

### 3.2. ML and DL methods using MD and ND datasets

To achieve a better result, we performed some sort of pre-processing on ND dataset. Network dataset is bigger than the sensory dataset. It includes one record for each millisecond approximately, i.e., it includes around one billion records for 11 days. Processing this data is very difficult in commodity computers so we concentrate on some part of data (one day: 28-Dec-2015). In the preprocessing step, we dropped the records with Null value in all fields. We removed the features which were not descriptive according to statistical analysis of features such as correlation analysis, unique value analysis, and bar chart analysis. At the next step the features are converted to numerical values. There is a feature named *Modbus\_Value* and it plays an important role in every network packet. We removed the records which include meaningless values in *Modbus\_Value* feature. Then we converted the *Modbus\_Value* from string to a list of integers (the list includes 38 numbers). After performing statistical analysis such as correlation analysis we extracted the important values from the list of 38 values and only 23 of them were most important in detecting attacks.

The classical ML methods such as decision tree (DT), random forest (RF), naïve Bayes (NB), support vector machine (SVM), and artificial neural networks (ANN) are applied to both ND and MD datasets. Different DNN and CNN models are also used for both datasets as well. For the sake of space, we just represent the structure of one of these methods in the paper. Figure 2 shows the structure of the best CNN applied for the ND.

Layer (type)	Output Shape	Param #
=====		
conv1d_20 (Conv1D)	(None, 40, 32)	128
max_pooling1d_20 (MaxPooling1D)	(None, 20, 32)	0
conv1d_21 (Conv1D)	(None, 18, 64)	6208
max_pooling1d_21 (MaxPooling1D)	(None, 9, 64)	0
flatten_10 (Flatten)	(None, 576)	0
dense_23 (Dense)	(None, 36)	20772
dense_24 (Dense)	(None, 10)	370
dense_25 (Dense)	(None, 1)	11

Figure 2. Structure of CNN method, applied on ND dataset

## 4. RESULTS AND DISCUSSION

In all the experiments we have used the holdout set method for evaluation. In this regard 70 percent of dataset is used for training and 30 percent for testing. Different criteria are used for comparing the methods: Accuracy, precision, recall, and F1-score. The confusion matrix is used to show the results of classifications. For simplicity we have assumed a binary classification including normal and attack records. Thus, the confusion matrix includes 4 cells: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

#### 4.1. Results of experiments on sensory dataset

To evaluate a model, various criteria might be used. We first represent the confusion matrix of a model. Then based on the confusion matrix, we find the accuracy, precision, and recall of the models. For the sake of conciseness, we just show the confusion matrix for a few experiments. Figures 3(a) to 3(d) represent the confusion matrices of DT, RF, DNN, and CNN methods on SD dataset, respectively.

Table 5 shows the result of evaluation metrics of all methods on full SD dataset. The results for single day SD dataset are approximately same (a little less) and we omitted it here. Among all methods RF has the highest accuracy and F1-score, then DT and DNN are in the second and third places. The last two rows of the table show the evaluation of the method when some sort of extra feature selection is done using important feature selection.

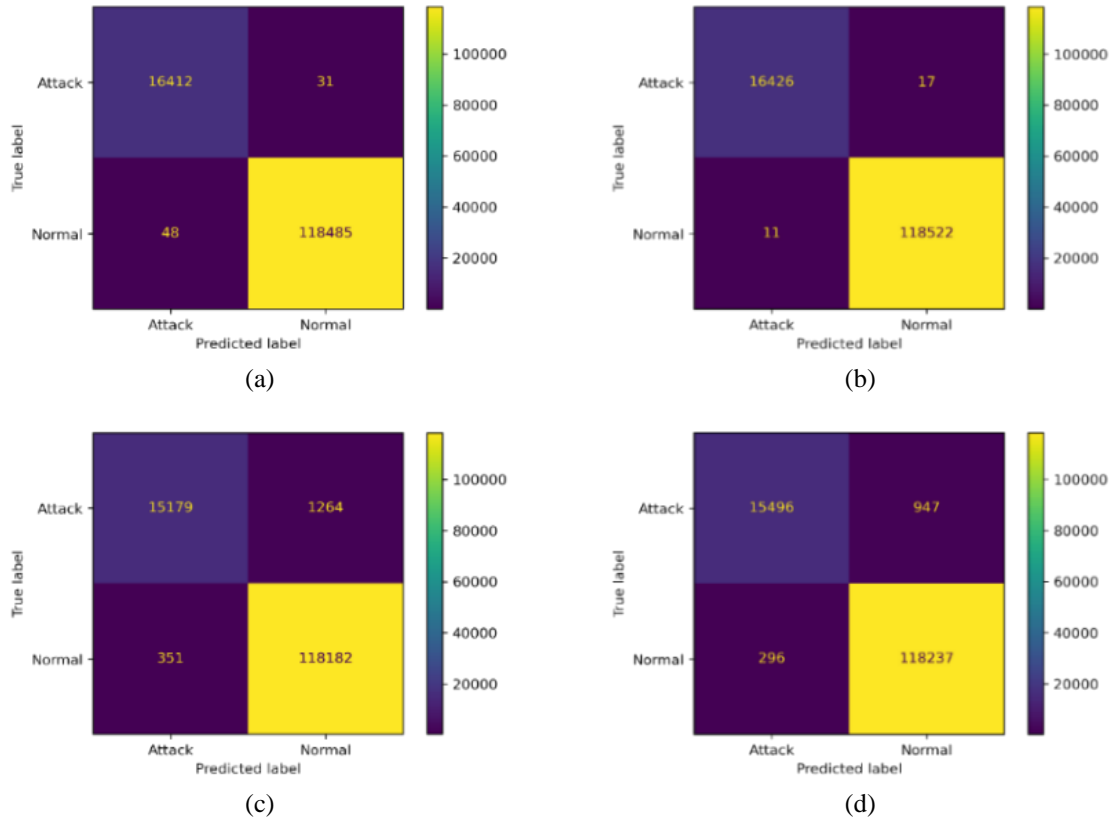


Figure 3. Confusion matrices of some experiments on SD (a) confusion matrix of DT method on SD dataset for all days, (b) confusion matrix of RF method on SD dataset for all days, (c) confusion matrix of DNN method on SD dataset for all days, and (d) confusion matrix of CNN method on SD dataset for all days

Table 5. Evaluation of proposed methods on sensory dataset

Method (SD data)	Accuracy	Precision	Recall	F1-score
Decision Tree	0.99941471	0.99941504	0.999414710	0.999414840
Neural Network	0.98284139	0.98295696	0.982841394	0.982352413
SVM	0.96873518	0.96915934	0.968735182	0.966952516
RF	0.99979255	0.99979253	0.999792555	0.999792539
NB	0.96153390	0.96210862	0.961533902	0.958740824
DNN	0.99258386	0.99254995	0.992583866	0.992532911
CNN	0.99079095	0.99073887	0.990790955	0.990710574
DT (with feature selection)	0.99937766	0.99937760	0.999377667	0.999377634
NB (with feature selection)	0.95985212	0.96036491	0.959852121	0.956826591

#### 4.2. Results of experiments on network and Modbus dataset

Confusion matrices of DT, RF, and CNN are represented in Figures 4(a) to 4(c), respectively, for ND dataset. The confusion matrices of DT, RF, and CNN are represented in Figures 4(d) to 4(f), respectively, for MD dataset. Accuracy, precision, recall, and F1-score are represented in Table 6 for both datasets. Also,

for informative purposes, we have mentioned some experiments on ND dataset minus MD (network dataset except *Modbus\_value*). The best results obtained for RF method once again. However, the results for both datasets are significantly less than the accuracy for SD dataset. Meanwhile the results obtained here are meaningfully better than literature [23]. The comparison of our best method with the results from a good literature paper is mentioned in Table 7. The reason we have got better results lies in the fact that we have done thorough preprocessing before doing any kind of machine learning. We have also carefully manipulated Modbus data which plays a crucial role in detecting attacks.

It is seen from the table that, here also the random forest leads based on all evaluation metrics. Decision tree is the second with just less than 0.5% difference and CNN is in the third position with another 0.5% difference. Another important point is that if we remove the Modbus data from network data (as seen in the rows ND-MD) the accuracy falls more by around 20%. This shows that the significant part of intrusion detection in this dataset is the *Modbus\_value* feature which is used for generating MD dataset.

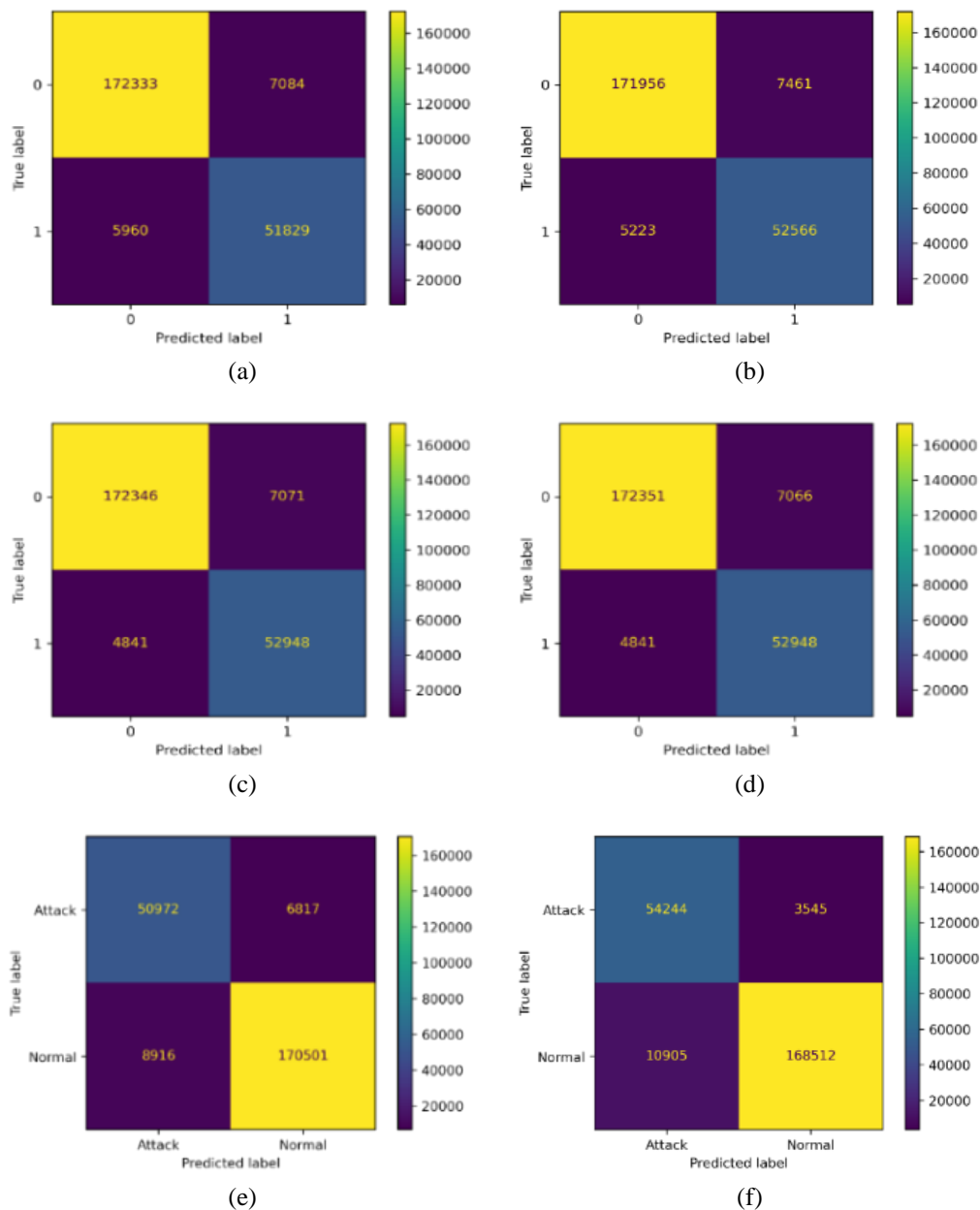


Figure 4. Confusion matrices of best methods in ND and MD datasets (a) DT on network dataset, (b) DT on Modbus dataset, (c) RF on network dataset, (d) RF on Modbus dataset, (e) CNN on network dataset, and (f) CNN on Modbus dataset



Table 6. Evaluation of various methods on ND, MD, and ND except MD

Method	Accuracy	Precision	Recall	F1-score
DT (ND)	0.94500982	0.94542120	0.94500982	0.94518824
DT (MD)	0.94652749	0.94742205	0.94652749	0.94686756
DT (ND-MD)	0.75637631	0.57210512	0.75637631	0.65146076
ANN (ND)	0.88719088	0.88481815	0.88719088	0.88558150
ANN (MD)	0.87616249	0.87373739	0.87616249	0.86829853
SVM (ND)	0.84303516	0.84151074	0.84303516	0.82483101
SVM (MD)	0.84782425	0.84881085	0.84782425	0.82970849
RF (ND)	0.94978204	0.95063276	0.94978204	0.95010031
RF (MD)	0.94980312	0.95065125	0.94980312	0.95012057
RF (ND-MD)	0.75637631	0.57210512	0.75637631	0.65146076
NB (ND)	0.59801607	0.65595754	0.59801607	0.62004492
NB (MD)	0.37139026	0.72863080	0.37139026	0.34388769
DNN (ND)	0.91879632	0.92029259	0.91879632	0.91940298
DNN (MD)	0.91937809	0.91882239	0.91937809	0.91906454
DNN (ND-MD)	0.75637631	0.57210512	0.75637631	0.65146076
CNN (ND)	0.93367368	0.93465088	0.93367368	0.93407008
CNN (MD)	0.93908248	0.94363681	0.93908248	0.94026815
CNN (ND-MD)	0.75637631	0.57210512	0.75637631	0.65146076

### 4.3. Discussion

As explained, three datasets were extracted from SWaT testbed: SD, ND, and MD. Several machine learning and deep learning techniques were implemented to effectively detect anomalies in this CPS system. As seen in Table 5: Evaluation of proposed methods, all methods have accuracy higher than 95% on SD dataset. Random forest, and decision tree have more than 99.9% accuracies while DNN, and CNN have more than 99% accuracies. This implies that, having access to the information of sensors and actuators in a CPS, we can find anomalies with very high accuracies. On the other hand, if we have only access to network traffic, we can detect the anomalies with acceptable accuracy, if we perform thorough preprocessing. This is seen from Table 6. To achieve best accuracies, we could extract the information of sensors and actuators from network data (like MD dataset) and use it separately. However, network data itself has approximately the same accuracy.

To avoid overfitting problem in our methods, we applied different mechanisms including data balancing, dropout in DNN and CNN (20%), limiting depth in DT and RF methods, and limiting the number of epochs. Fortunately using these mechanisms, there is no overfitting in our methods as the accuracy on test sets keep increasing in all the models. To find the best hyperparameters, we have used the GridCV method in our models. Feature selection is done in various ways to increase the accuracy of models. These include removing useless features manually, removing single valued features, and removing lowly related features using correlation analysis. Nonetheless there are some limitations in the work, such as using the entire network dataset and assuming multiclass classification (there are 36 attacks).

Table 7 represents the comparison of our best method with the results from literature. We have got very high accuracies owing to good pre-processing on datasets, removing noise, outliers, unnecessary features, and converting the values to normal ranges. The comparisons with other work showed the improvement of approximately 20% on different metrics using the ND and MD datasets, and 0.20% using sensory dataset compared to the best method of the literature [20].

Table 7. Comparison of proposed method with previous methods

	Method	Accuracy	Precision	Recall	F1-score
SD	BFTree [20]	99.72%	99.70%	99.70%	99.70%
	MLP_CUSUM [21]	97.77% (averaged)	99.87% (averaged)	90.19% (averaged)	94.78
	RPCA [22]	----	100%	86.10%	92.50%
	RF on SD (our best)	99.98%	99.98%	99.98%	99.98%
ND	SVM+LUPI [23]	74.2534% ( $\pm 1:022\%$ )	77.251% ( $\pm 0:849\%$ )	74.1692% ( $\pm 1:173\%$ )	73.4782% ( $\pm 1:375\%$ )
	RF on ND (our best)	94.9782%	95.0632%	94.9782%	95.0100%

## 5. CONCLUSION




In this paper we implemented some deep and classical machine learning techniques for anomaly detection in CPS. To experiment in a semi-real dataset, we used a dataset obtained from a small-scale water treatment plant called SWaT. The dataset included 2 databases: first the data collected from communication of SCADA system with PLCs which produced TCP protocol-based network dataset. The second was data extracted from these packets which contain information of the sensors and actuators which produced Sensory dataset. Many steps of preprocessing are done in the datasets. The unimportant features were removed from

both datasets. There was an important feature in the network dataset called *Modbus\_value* which is treated separately. Finally, the experiments mentioned that random forest, decision tree, and DNN performed very well on Sensory dataset with hundred percent of accuracy, approximately. On the Network dataset our method performed significantly better than previous methods. In this dataset again the random forest, decision tree and CNN got the highest accuracies compared to others. In the future we plan to perform other advanced methods like continual learning and federated learning to completely simulate the real environment of an IoT and cloud-based CPS anomaly detection system.




## REFERENCES

- [1] A. K. Tyagi and N. Sreenath, "Cyber physical systems: analyses, challenges and possible solutions," *Internet of Things and Cyber-Physical Systems*, vol. 1, pp. 22–33, 2021, doi: 10.1016/j.iotcps.2021.12.002.
- [2] X. Jiang and S. Li, "Plume front tracking in unknown environments by estimation and control," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 911–921, Feb. 2019, doi: 10.1109/TII.2018.2831225.
- [3] S. Adepun, N. K. Kandasamy, and A. Mathur, "EPIC: an electric power testbed for research and training in cyber physical systems security," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11387, Springer International Publishing, 2019, pp. 37–52.
- [4] S. S. S. Sindhu and S. G. S. Selvakumar, *Network intrusion detection system using machine learning techniques: a quick reference*. LAP LAMBERT Academic Publishing, 2013.
- [5] T. Liu, K. Zhou, and L. Liang, "Security in cyber-physical systems: challenges and solutions," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 10, no. 4, 2017, doi: 10.1504/ijaacs.2017.10009673.
- [6] W. Liang, K. C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, Mar. 2020, doi: 10.1109/TII.2019.2946791.
- [7] P. Freitas De Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macedo, and C. Zanchettin, "Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247–6256, Apr. 2021, doi: 10.1109/JIOT.2020.3024800.
- [8] Y. Lai, J. Zhang, and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *Security and Communication Networks*, vol. 2019, pp. 1–11, Sep. 2019, doi: 10.1155/2019/8124254.
- [9] Y. Yang *et al.*, "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092–1102, Jun. 2014, doi: 10.1109/TPWRD.2014.2300099.
- [10] Y. Xiao, J. Liu, and L. Zhang, "Cyber-physical system intrusion detection model based on software-defined network," in *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, Aug. 2021, pp. 170–173, doi: 10.1109/ICSESS52187.2021.9522345.
- [11] B. Tang, Y. Lu, Q. Li, Y. Bai, J. Yu, and X. Yu, "A diffusion model based on network intrusion detection method for industrial cyber-physical systems," *Sensors*, vol. 23, no. 3, Jan. 2023, doi: 10.3390/s23031141.
- [12] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021, doi: 10.1109/TII.2020.3023430.
- [13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
- [14] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
- [15] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *Eurasip Journal on Information Security*, vol. 2019, no. 1, Feb. 2019, doi: 10.1186/s13635-019-0084-4.
- [16] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020, doi: 10.1016/j.procs.2020.03.330.
- [17] A. P. Mathur and N. O. Tippenhauer, "SWaT: a water treatment testbed for research and training on ICS security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, Apr. 2016, pp. 31–36, doi: 10.1109/CySWater.2016.7469060.
- [18] L. Faramondi, F. Flammini, S. Guarino, and R. Setola, "A hardware-in-the-loop water distribution testbed dataset for cyber-physical security testing," *IEEE Access*, vol. 9, pp. 122385–122396, 2021, doi: 10.1109/ACCESS.2021.3109465.
- [19] M. R. Gauthama Raman, N. Somu, and A. P. Mathur, "Anomaly detection in critical infrastructure using probabilistic neural network," in *Communications in Computer and Information Science*, vol. 1116, Springer Singapore, 2019, pp. 129–141.
- [20] K. N. Junejo and J. Goh, "Behaviour-based attack detection and classification in cyber physical systems using machine learning," in *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, May 2016, pp. 34–43, doi: 10.1145/2899015.2899016.
- [21] G. Raman MR, N. Somu, and A. P. Mathur, "A multilayer perceptron model for anomaly detection in water treatment plants," *International Journal of Critical Infrastructure Protection*, vol. 31, Dec. 2020, doi: 10.1016/j.ijcip.2020.100393.
- [22] M. Al-Dhaheri, P. Zhang, and D. Mikhaylenko, "Detection of cyber attacks on a water treatment process," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 667–672, 2022, doi: 10.1016/j.ifacol.2022.07.204.
- [23] M. Pordelkhaki, S. Fouad, and M. Josephs, "Intrusion detection for industrial control systems by machine learning using privileged information," in *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Nov. 2021, pp. 1–6, doi: 10.1109/ISI53945.2021.9624757.
- [24] S. Kim, W. Jo, and T. Shon, "APAD: autoencoder-based payload anomaly detection for industrial IoE," *Applied Soft Computing*, vol. 88, Mar. 2020, doi: 10.1016/j.asoc.2019.106017.
- [25] Á. L. P. Gómez *et al.*, "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019, doi: 10.1109/ACCESS.2019.2958284.

**BIOGRAPHIES OF AUTHORS**

**Muhammed Almendli**    received the B.Sc. Eng. degree in technical computer engineering from Al-Rafidain University College, Iraq-Baghdad, in 2011. The M.S. information technology engineering-information systems management from Imam Reza International University and researcher PhD degrees in computer engineering-computer networks, he accepted for the 2021 academic year at Urmia University, Iran. In 2012, he worked in Al-Rafidain University College as a lecturer, and in 2022 he worked in Al-Mustafa University college as a teach assistant. He teaches the following material (computer organization, AutoCAD, electronic, computer application, AI, real time system, and computer architecture). He can be contacted at email: muhammedalmendli@gmail.com.



**Jamshid Bagherzadeh Mohasefi**    holds a Ph.D. in computer science from Indian Institute of Technology, India. He is currently a professor of the Department of Computer Science at Urmia University. His research topics include artificial intelligence, machine learning, and network security. His research has been funded by the Urmia University, Iranian Telecom Ministry, and Iranian Ministry of Science, Research, and Technology. He can be contacted at email: j.bagherzadeh@urmia.ac.ir.