

Safeguarding Industry 4.0: A Machine Learning Approach for Cyber-Physical Systems Security and Sustainability

Imad El Hassak^{1*}, Zahra Oughannou², Soufyane Mounir¹, and Yassin Maleh¹

¹National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco

²National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

Abstract. The proliferation of connected objects, with over 8 billion IoT devices currently and a projected increase to 41 billion by 2027, signifies the widespread integration of technology in sectors like Smart City, Industry 4.0, e-commerce, and e-health. This study focuses on the security assessment of Cyber-Physical Systems (CPS) in manufacturing processes, utilizing six supervised algorithms on a dataset with 61 features. The results not only offer valuable insights into security but also contribute to the optimization of machine learning models. This research implicitly addresses the sustainability aspect by acknowledging the broader impact of CPS technologies. Cyber-Physical Systems (CPS) optimization of machine learning models not only fits in with the industry 4.0 framework's overarching goal of promoting environmentally friendly practices, but it also creates a vital connection between sustainability and the security paradigm that these complex systems are built upon. This mutually beneficial relationship highlights how improving machine learning algorithms with the goal of reducing environmental impact also helps to strengthen the security infrastructure of CPS. Industry 4.0 prioritizes environmental responsibility by emphasizing the development and application of eco-conscious practices. It also acknowledges the interdependence of sustainability and security within the framework of dynamic cyber-physical ecosystems.

Keywords: Cyber-Physical-Systems (CPSs), Cyber Security, Deep Learning, IoT, CPSs Attacks, DL Models, Sustainable Technologies.

1 Introduction

In the 21st century, the escalating pace of technological evolution introduces a formidable security challenge across various sectors. For instance, the reported surge to 1.5 billion IoT cyber-attacks in the first half of 2021, an increase of 639 million compared to the entirety of 2020 by Kaspersky, underscores the imperative for secure technological advancements [1]. This phenomenon is particularly evident in the context of modern industrial infrastructure,

* Corresponding author: imad.elhassak@gmail.com

emblematic of Industry 4.0. Cyber-Physical Systems (CPS) within this paradigm seamlessly integrate the physical capabilities of the Internet of Things (IoT) with computational efficiency. Beyond addressing security concerns, this research examines the interaction of technology with environmental sustainability and energy efficiency. The intricate connection between CPS, IoT, and computational efficiency signifies a pivotal exploration at the intersection of technological progress and environmental consciousness, contributing to the ongoing discourse on sustainable technological advancements [2].

The structure of the CPS consists of three hierarchical layers [3], The application layer represents the computer system, with intelligent tasks for processing sensor data, this layer contains SCADA Supervisory Control and Data Acquisition which is a type of application that can make decisions and predict the state of the physical system (e.g., detect anomaly data injected by the sensors), the network layer is charge of facilitating communication between the physical and application layers, and the physical layer which represents the multiple sensors, actuators and control system of the physical system (see fig.1). The safety and security of CPS against cyber-attacks is one of the biggest challenges for cybersecurity researchers. One of these challenges is the complexity of the structure of CPS, how make the Cyber physical systems vulnerable to many types of attacks in different layers (Backdoor Attack, False Injection, Flooding Attack, DDoS Attack...).

The recent advances in information technologies offer many solutions to predict and create a prevention system. In order to secure the CPS against attacks we use the classification algorithms (Logistic Regression - LR, Decision Tree - DT, Random Forest - RF, K-Nearest Neighbors - KNN, Deep Neural Network - DNN) and dimensionality reduction algorithm (Principal Component Analysis - PCA) to detect attacks. In this paper, we study the performance of the above algorithms in detecting attacks by observing and comparing the evaluation metrics (Confusion Matrix, Accuracy, F1-Score, Precision, Recall). The dataset we use in our paper is Edge-IIoTset (Kaggle : Edge-IIoTset Cyber Security Dataset of IoT & IIoT) it free Dataset, this Dataset has been generated using a purpose-build IoT/IIoT tested with a large representative set of devices, sensors, protocols and cloud/Edge configurations[4].

To achieve this, our paper is structured as follows: In Section 2, we provide a comprehensive review of existing ML models for attack detection and the security challenges in CPS systems. Section 3 outlines the methodology, detailing the implementation steps. Section 4 presents the results of our implementation and discusses the implications of the results for attack detection. Finally, in Section 5, we conclude with a summary of key findings and outline our next steps in research.

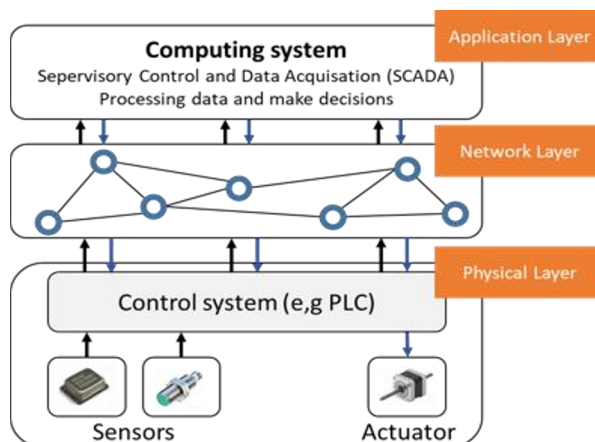


Fig1. The hierarchical CPS structure.

2 Cyber physical systems and Machine Learning models for attack detection

2.1 Cyber Physical Systems

The cyber-physical systems refer to the integration between the developments of computer science (the information and communication technologies) [5, 6] and the physical capability, which represent some mechanical and electrical system. Cyber-Physical Systems (CPS) are consequently combinations of networking, computation, and physical (mechanical). The CPS has been constructed on a large geographical scale with a large number of components. The structure of the CPS system consists of three hierarchical layers [7], Physical layer, Network layer and Application layer.

2.1.1 Application layer

This layer provides the systems, intelligent to process and monitor (SCADA) the data collected from the sensors/actuators, system control and data circulating in all layers (Network data and Communication data circulating between layers). This layer plays the biggest part of security in CPS, from this layer we can detect the network behavior, the normal traffic or detect unauthorized traffic (Attack DDoS) and returns the right decision, by installing an intrusion detection system (IDS) detection system, reinforced by trained models of Machine Learning (DNN) [3,7].

2.1.2 Network layer

The network layer is in charge of facilitating communication between different components of the system, such as sensors/actuators, controllers (SCADA), and other devices. In CPS, the network layer includes wired and wireless communication. One of the biggest challenges in this layer is to ensure real-time communication between the physical and application layer. Another challenge in this layer is the security challenge, which is centered on preventing unauthorized access to the network, denial of service attacks, and interception or manipulation of data [3,7].

2.1.3 Physical layer

In the context of a cyber-physical system (CPS), the physical layer refers to the layer that connects the system to the physical environment. This layer consists of the physical devices, including sensors, actuators, and other control elements (e.g PLC), that collect data from the environment (e.g Temperature, Road traffic situation, Traffic Light...etc) [8] and convert it into digital signals that can be processed by other layers of the system. The physical layer also translates digital signals from other layers into physical signals that can be used to control the system's actuators, which in turn affect the physical world. The security challenges in this layer are also very important. If the controller were to be attacked, it could result in significant damage. For example, an attacker could use the sensors to send the wrong command to the actuators, potentially compromising the entire system's workflow [3,7].

2.2 Security challenge and attack per layer

As we can see above, the cyber-physical system refers to the integration in physical processes and computing, the structure of CPS is complex, the nature of the physical devices in

heterogeneous, the different protocols and technologies used for communication and networking between layers and devices. this makes the system also vulnerable and Difficult to secure.

Poor security in the CPS system can result in malfunctions in data processing, communication and networking, or device functionality (such as IoT, sensors or actuators). If any of these layers are attacked, the entire system may become unstable and stop working.

The challenge of security in CPS system is different and variante from challenge in the physical attack, Interoperability, human factor to the cyber-attacks.

The challenges of cyber security attacks is:

- Interconnected
- Complexity
- Real-Time Processing
- Limited Resources
- Legacy Systems

Overall, the challenge of cybersecurity attacks on cyber-physical systems requires a comprehensive approach that takes into account the unique characteristics (standardisation: protocols, communication ...) of these systems. The (Table 1) displays the safety criteria and the type of attack that can be affected for each layer.

Table 1. CPS Layers & Attacks Details

Layer	Function	Types Attacks	safety criteria
Application	Processing & Analyzing Data	Code Injection	Application
Network	Transmission & Communication	DoS/DDoS	Network
Physique	Collection of Data & Produce Motion	DDoS	Physique

2.3 Dataset chosen

In this section, we will present the dataset used to evaluate the performance of each machine learning model chosen to test attack detection by observing and comparing the evaluation metrics (F1 score, recall...etc).

In this paper we use the Edge-IIoT Dataset, which was published in 2022 by Mohamed Amine Ferrag [04]. This Dataset includes 49 files, which are organized into three subdirectories.

Sub- directorie 01: this directory named "Normal Traffic", this directory contain 20 files (10 files CSV & 10 file PCAP). The data in this directory present normal traffic, the sensors data (water level, pHValue...etc).

Sub- directorie 02: this directory named "Attack Traffic", this directory contain 20 files (14 files CSV & 14 file PCAP). The data in this directory present attack traffic.

Sub-directorie 03: this directory contains two CSV files selected from the entire Dataset for used to evaluate Machine learning and Deep Learning.

- File 01 Named : ML-EdgeIIoT-dataset.csv
- File02 Named : DNN-EdgeIIoT-dataset.csv

The number of records in this dataset is 20,952,648, which is divided into two types of records:

- Normal traffic records : 11 223 940
- Attack traffic records : 9 728 70

The size of this dataset is 10.4 GB

3 Methodology and Implementation

In this section, we will discuss the steps taken to train the machine learning models and extract the results. The first step in this process is to prepare the data.

3.1 Data preparation

The first step in our process (see fig.2 Algorithm.1) is data preparation, as we can see above, the size of the selected dataset is more than 10.4 GB and more than 20 million records, with our computer resources we cannot process this dataset, for this reason we decide to extract 5% of records from this dataset, equivalent to 1 188 090 records.

In (see fig.3), we can observe that the data preparation process begins by selecting all the CSV files from two sub-directories, namely 'Normal Traffic' and 'Attack Traffic'. Next, we iterate through the selected files and extract 5% of their records. In each iteration, we concatenate the extracted records with the previously extracted ones.

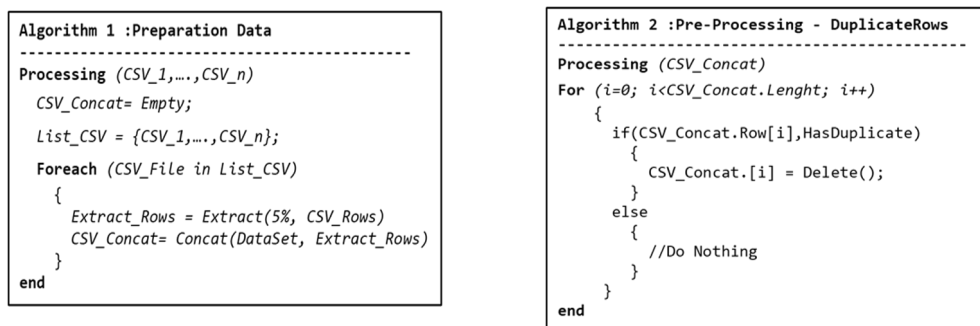


Fig 2. (Algorithm 1) Preparation Data (Algorithm 2) Delete Duplicate Rows from a Dataset

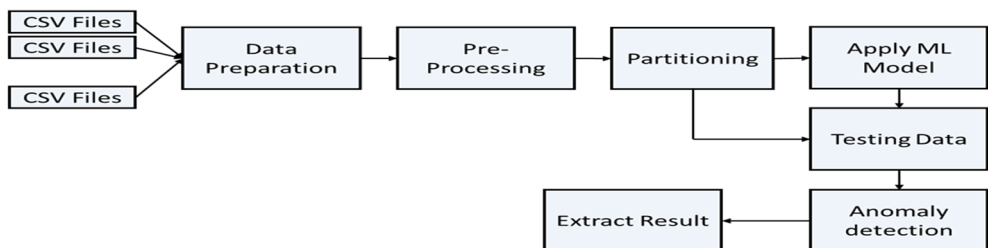


Fig 3. Methodology of training & Pre-processing

3.2 Pre-Processing

This step is very important in any data analysis or machine learning training and prediction, as it involves cleaning, transforming and formatting the series data before it is fed into a model for training. The pre-processing data helps to ensure that the data is of high quality, and allows the identification and selection of relevant features or variables that are useful in the training and prediction step.

In order to start the preprocessing by cleaning the data, the first method we used is to delete duplicate rows: in preprocessing (see fig.2 Algorithm.2), deleting duplicate rows means

removing rows from the dataset that have the same values in all columns. deleting duplicate rows helps to ensure the accuracy and reliability of the data used for training.

As we can see in (see fig.2 Algorithm.2), the process of deleting duplicate rows begins by selecting the CSV file (CSV_Concat) that was created in the previous step (Preparing data). Next, we loop through all the rows in the dataset and if the algorithm identifies a duplicate row, it is removed from the dataset.

```

Algorithm 3 :Pre-Processing - ConstantValue
-----
Processing (CSV_Concat)
For (i=0; i<CSV_Concat.Column.Length; i++)
{
  MyConstat = CSV_Concat[0][i];
  for(j=0; j<CSV_Concat.Length; i++)
  {
    if (MuConstat == CSV_Concat[i][j]);
    {
      }//Do Nothing
    else
    {
      //Stop Executing and go to the
      //Next Column
      CSV_Concat.Column[i] = Delete();
    }
  }
}
end

```

```

Algorithm 4 :Pre-Processing - ConvertToNumber
& Normalisation
-----
Processing (CSV_Concat)
For (i=0; i<CSV_Concat.Column.Length; i++)
{
  for(j=0; j<CSV_Concat.Length; i++)
  {
    if (CSV_Concat[j][i].Type == Number)
    {
      X = CSV_Concat[j][i];
      X' = (X-X_min)/(X_max-X_min);
      CSV_Concat[j][i] = X';
    }
    else
    {
      ConvertToNumber(CSV_Concat[j][i])
      X = CSV_Concat[j][i];
      X' = (X-X_min)/(X_max-X_min);
      CSV_Concat[j][i] = X';
    }
  }
}
end

```

Fig 4. (Algorithm 3) Identify and Delete Constant Column (Algorithm 4) Normalization & Formatting

The second step in the data cleaning process is to filter out constant columns. This technique helps to identify and remove columns (features) that do not contribute to the training and prediction steps, saving time and resources in training machine learning models.

As we can see in (see fig.4 Algorithm.3), the process of deleting duplicate rows begins by selecting the CSV file (CSV_Concat) that was created in the previous step (Preparing data). Next, we loop through all the rows in the dataset and if the algorithm identifies a duplicate row, it is removed from the dataset.

The second step in data pre-processing is data formatting, which is very important before training, as it prepares the data for analysis by ensuring that it is in a suitable format for the ML algorithms or models being used (e.g "For logistic regression models, it is necessary that the data is in a numeric format before training. Only the class labels used for prediction can be in a string format).

(see fig.4 Algorithm.4) describes the steps followed to convert the data type and columns of a dataset. The process begins by loading the dataset and looping through all the columns and rows in the dataset. The algorithm first selects a column and then loops through the rows in that column. It converts each category in the column to a numerical value (for example, converting "192.168.1.2" to 1). This process is repeated for all columns in the dataset.

Using formatting techniques in a dataset with a variety of categories within each feature can lead to a huge number in each column (feature). This can require more resources to process. To address this issue, we can perform normalization as the final step in our data pre-processing.

Normalisation is an important step in data pre-processing, the aim of normalisation is to make different variables or features in the dataset comparable and to eliminate the influence of the scale of the data on the analysis results. The normalisation can be done with many different settings, generally there are three types of normalisation, Z-score normalisation is a strategy of normalising the data that avoids this outlier problem, the second method is normalisation by decimal scaling , this method divides a feature's values by a power of ten to scale the

feature values, the last method is Max-Min normalisation, this is the simplest method that scales the data so that it is bounded between [Min, Max] (e.g. [0,1]). (see fig.4 Algorithm.4) shows the normalisation steps with the Max-Min normalisation of the data.

- Z-score Normalization:

$$X' = \frac{X - \mu}{\sigma}$$

X: Original Value

μ : Mean of data

X': New Value

σ : Standard deviation of data

- Normalization by Decimal Scaling:

$$X' = \frac{X_j}{10^j}$$

j: The number of digits present in the maximum value in the data

- Max-Min Normalization:

$$X' = \frac{X - Min}{Max - Min}$$

3.3 Training Models

After preparing and preprocessing the data, the next step is to train the selected machine learning models. The first step in this process is to partition the preprocessed dataset. In our training model, we tested two types of models. In the first type, we used classical machine learning models (LR, DT, RF and KNN), and we partitioned the data into two parts: 80% for training and 20% for testing. In the second type, we used an ensemble method by combining two models: PCA and DNN. For this type, we partitioned the data into three parts: 70% for training, 15% for validation, and 15% for testing the trained model.

The dataset used in this paper contains 61 features and 2 classes [9]. However, training DNN models with this many features requires significant computational resources. Therefore, we decided to reduce the number of features while keeping the same characteristics of the dataset. To achieve this, we used principal component analysis (PCA) to reduce the size of the dataset from 61 features to 5 features.

Principal Component Analysis is a machine learning dimensionality reduction model, this model operates by changing a group of variables that may be correlated, called features, into a new set of uncorrelated variables, called principal components. The principal components are obtained by finding the linear combinations of the original features that explain the maximum amount of variance in the dataset [10].

3.4 Performance Evaluation Metrics

Performance evaluation metrics are measurements used to assess the performance of tested models [11]. In machine learning, performance evaluation metrics are used to quantify how well a model is able to predict, classify, or estimate values on input data. The performance of trained machine learning models is computed using the performance evaluation metrics. these metrics help to find the better machine learning models.

Here are some metrics for evaluating the performance of machine learning models.

Table 2. Confusion matrix

True Class	Prediction Class	
	Attack	Normal
Attack	TP	FN
Normal	FP	TN

The confusion matrix (Table 2) is used to calculate various performance metrics of a binary classification model. The Confusion Matrix consists of four cells: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). TP represents the number of positive cases correctly predicted by the model, FP represents the number of negative cases incorrectly predicted as positive, FN represents the number of positive cases incorrectly predicted as negative, and TN represents the number of negative cases correctly predicted as negative.

3.4.1 Accuracy

In machine learning, accuracy is a widely used metric for evaluating the performance of a model. It quantifies the proportion of correct predictions made by the model out of the total number of predictions. The formula for calculating the Accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.4.2 Recall

The true positive rate (TPR), also known as Recall, is a statistical measure that represents the proportion of actual positive cases (in this context, attack cases) that are correctly identified as positive by the model.

The formula for calculating the true positive rate (TPR) is:

$$Recall = \frac{TP}{TP + FN}$$

3.4.3 Precision

Precision is a statistical measure used in machine learning to assess the performance of a binary classification model. It measures the proportion of predicted positive cases that are truly positive, or the percentage of correct positive predictions among all positive predictions. A high precision score indicates that the model is accurately predicting positive outcomes, while a low precision score indicates that the model is generating a significant number of false positive errors. The formula for calculating precision is:

$$Precision = \frac{TP}{TP + FP}$$

4 Results and Discussion

In this section, we present the results of our investigation into two categories of machine learning models: classical models and multi-method models, combining PCA (Principal Component Analysis) and DNN (Deep Neural Networks). The assessment of each model's performance is conducted through a diverse set of metrics. Beyond the immediate scope of security evaluation for Cyber-Physical Systems, our research implicitly addresses broader concerns related to environmental impact and energy efficiency. The utilization of PCA and DNN highlights a strategic consideration of the balance between computational efficiency and environmental implications. Through this nuanced analysis, our work contributes to the ongoing dialogue surrounding sustainable technological advancements, aligning with key themes in the realm of environmental and energy considerations.

4.1 Classical models of Machine Learning

As previously mentioned, we tested four classical models of machine learning: LR, DT, RF, and KNN. The results of these trained models are summarized in (Table 3), as we see in these results the performance evaluation metrics records a good result, but if we see the number of False Negatives - FN (this mean the attack traffic predict a normal traffic) (Table 4) we notice the number is higher.

It is important to note that the dataset used in this study did not include DoS attacks but only included DDoS attacks. The difference between DoS and DDoS attacks can create vulnerabilities for systems secured by these models.

Table 3. Classification results (Evaluation metrics)

Model ML	Evaluation metrics		
	Accuracy	Precision	Recall
LR	0,9953	0,9970	0,9942
DT	0,9997	0,9998	0,9996
RF	0,9984	0,9986	0,9983
KNN	0,9941	0,9956	0,9932
PCA & DNN	0,9873	0,9911	0,9848

Table 4. Confusion matrix for trained models

Model ML	TrueClass	Prediction Class	
		Attack	Normal
LR	Attack	74866	436
	Normal	224	67035
DT	Attack	75277	25
	Normal	13	67246
RF	Attack	75179	123
	Normal	99	67160
KNN	Attack	74790	512
	Normal	326	66933
PCA & DNN	Attack	7392	114
	Normal	66	6685

4.2 Multi-Method: PCA & DNN

The deep neural network requires significant computing resources, but in our case, we do not have sufficient resources to process the entire dataset. Therefore, to test these models, we chose to work with 20% of the pre-processed data.

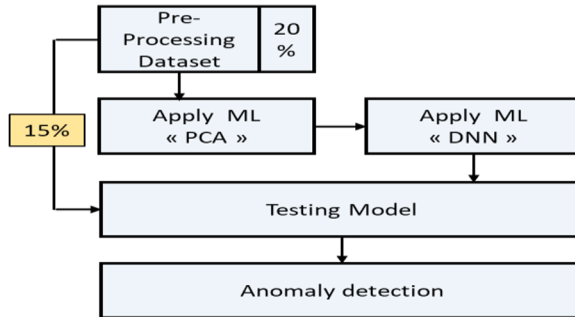


Fig 5. Methodology of Multi-Method: PCA & DNN

To initiate these models, the initial step involves the application of Principal Component Analysis (PCA) to decrease the dataset's dimensionality, from 61 features to 5 features (See fig.5). The next step involves transforming the class from 'One to Many,' which means converting 1 column to the number of targets in that column. In our case, we have two targets ('Attack' and 'Normal'). The final step before starting the training is partitioning the dataset into three parts (70% for training, 15% for validation, and 15% for testing). To define the architecture of our Deep Neural Network for training, we first specify the number of inputs (5, reduced from the original set of features). Next, we create a hidden layer with 8 units and we use the Rectified Linear Unit (ReLU) activation function. We then declare the number of outputs as 2, corresponding to "Attack" and "Normal" classes. Finally, we specify the number of epochs (100) and the optimizer (Adam). The confusion matrix presented in the (Table 4) provides a comprehensive portrayal of the classification model's performance in distinguishing between normal and attack traffic scenarios. For example, the Logistic Regression Model (LR) in the domain of normal traffic, the model demonstrated a discerning capability, accurately classifying 67, 035 instances as normal. This proficiency stands as an indicative testament to the model's efficacy in identifying non-malicious data. However, a salient limitation surfaced as the model misclassified 224 instances of normal traffic, erroneously attributing them to the category of attacks. Conversely, when confronted with attack traffic, the model exhibited commendable acuity, accurately identifying 74, 866 instances as attacks. Nevertheless, a noteworthy drawback manifested as the model erroneously classified 436 instances of attack traffic as normal. These findings underscore the nuanced performance of the model and emphasize the imperative of a nuanced evaluation strategy. Such a strategy must consider both true positives and false positives to holistically assess the model's efficacy across diverse traffic scenarios.

In this experiment, we found that these models have a high detection rate and efficiency in two types of training and validation techniques (classical models LR, DT, RF, KNN and multi-method: PCA & DNN). The model achieved a high level of accuracy, precision, and recall, surpassing 98% (Table 3) in each measure. This result suggests that implementing these models in another system (e.g CPS environment) could successfully meet everyday challenges. However, in the context of scientific computing, these results may be considered inadequate for system deployment. Therefore, further optimisation of these models, including optimisation of pre-processing techniques and model parameters, is required.

5 Conclusion

In this article, we introduce established machine learning models applied to the Edge-IIoT dataset. Successful pre-processing of the data yields favorable results, yet we anticipate the potential for further enhancements. To achieve this, we propose the development of new training models, leveraging deep learning techniques for optimization. Our forthcoming research is poised to address key themes related to environmental sustainability and energy efficiency. Specifically, our focus will center on optimizing systems through the integration of deep learning techniques during the training phase. Additionally, we aim to refine our practical approach to data extraction and generation in the pre-processing stage, acknowledging the critical role that improved data processing plays in the efficiency and environmental impact of Cyber-Physical Systems. This trajectory aligns with the broader discourse on sustainable technological advancements, reflecting our commitment to not only bolster the security and efficiency of systems but also to contribute to the ongoing evolution of environmentally conscious technological solutions.

References

1. S. Al-Sarawi, M. Anbar, R. Abdullah and A. B. Al Hawari, Internet of Things Market Analysis Forecasts, 2020–2030, Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, pp. 449–453 (2020)
2. C. K. Keerthi, M. A. Jabbar, B. Seetharamulu, Cyber Physical Systems (CPS): Security Issues, Challenges and Solutions, in IEEE International Conference on Computational Intelligence and Computing Research (ICIC), 2017, pp. 1–4 (2017)
3. S. Kim and K. J. Park, A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems, Applied Sciences 11 (12) (2021) 5458–5458 (2021)
4. M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning, IEEE Access 10 40281–40306 (2022)
5. L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn and K. Ueda, Cyber-physical systems in manufacturing, CIRP Annals 65 (2) 621–641 (2016)
6. J. Lee, B. Bagheri and Hung-An Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems,” Manufacturing Letters 3 18–23 (2015)
7. N. Boulila. Guidelines for Modeling Cyber-Physical Systems – A Three-Layered Architecture for Cyber Physical Systems. 10.13140/RG.2.2.21599.30881 (2017)
8. I. E. Hassak, A. Addaim, Proposed Solutions for Smart Traffic Lights using Machine Learning and Internet of Thing, in: 2019 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6 (2019)
9. O. M. A. Ferrag, D. Friha, L. Hamouda, H. Maglaras, Janicke, Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications: Centralized and Federated Learning, IEEE Dataport (2022).
10. I.T. Jolliffe and C. Jorge 2016, Principal component analysis: a review and recent developments Phil. Trans. R. Soc. A.374: 2015020220150202 (2016)
11. M. Steurer, R. J. Hill and N. Pfeifer , Metrics for evaluating the performance of machine learning based automated valuation models, Journal of Property Research, 38:2, 99-129 (2021)