A Study on IIoT Datasets and Machine Learning Classifiers for Malicious Traffic Classification in IIoT Environments

GyuHyun Jeon[1], Ilhwan Ji[1], Seungho Jeon[1], and Jung Teak Seo[1]

[1] University of Gachon, Seongna-daero 1342, Republic of Korea
{pengchan88, ilhwan1013, shjeon90, seojt}@gachon.ac.kr

Abstract

Industrial Internet of Things (IIoT) devices are used in industrial environments for high operational efficiency, improved productivity, and remote management. IIoT can be controlled by connecting through heterogeneous wired and wireless networks. However, due to the low security of IIoT devices, the number of cyberattacks targeting IIoT devices is steadily increasing. In existing research on dataset-based machine learning (ML) models for classifying malicious traffic, various machine learning models were not applied to multiple datasets, so it is difficult to see the performance comparison results for each IIoT dataset and ML classifier at a glance. Therefore, in this paper, we researched IIoT datasets and machine learning classifiers for classifying malicious traffic in the IIoT environment. The data sets used in the experiment are UNSW-NB15, ToN-IoT (IoT_Modbus), WUSTL-IIoT-2021, X-IIoTID, and Edge-IIoT (ML-Edge-IIoT). The ML classifiers to be used for each dataset are LR, K-NN, RF, NB, DT, LightGBM, XGBoost, AdaBoost, CatBoost, and MLP. The evaluation metrics used to compare the performance of ML classifiers are ACC, REC, PRE, ROC-AUC Score, and F1-Score. After confirming and analyzing the experimental results, undersampling/oversampling, feature engineering techniques, and hyperparameter tuning was explained as methods to complement the data set and ML model to improve ML classifier performance.

Keywords: IIoT, Machine Learning, Traffic Classification

The 7th International Conference on Mobile Internet Security (MobiSec'23), Dec 19-21, 2023, Okinawa, Japan, Article No.20

1 Introduction

Industrial Internet of Things (IIoT) devices are IoT devices used in industrial environments [1]. They can be interconnected and controlled through heterogeneous wired and wireless networks such as sensors, Wi-Fi/mobile (3G/4G/5G/LTE) networks, and industrial buses. Through this, IIoT devices are used for high operational efficiency, improved productivity, and remote management. However, IIoT devices have relatively smaller storage capacity and lower computing power than IT devices, so they comprise lightweight protocol communication and limited resources [2]. For this reason, it is difficult to install high-performance security systems or security solutions into IIoT devices. As a result, it is difficult to mount high-performance security systems or security solutions on IIoT devices, and the number of cyberattacks targeting IIoT devices continues to increase as the usage rate of IIoT devices increases through the development of the smart industry environment [3]. Unlike IoT devices, IIoT devices are mainly used in industrial environments. Hence, they are likely to cause significant damage, such as direct factory outages, physical damage, and tampering and deodorization of sensitive data due to attacks. Therefore, because IIoT security is critical, research on various defense techniques, such as detecting and classifying malicious traffic in IIoT devices, is still being actively conducted to respond to cyber-attacks. To classify malicious traffic, the performance of various machine learning (ML) classification models is verified and evaluated using IIoT datasets containing malicious traffic information. However, there may be significant differences in model performance depending on factors such as attack type, characteristics, and balance of data volume values that comprise the dataset. In addition, since performance differences exist depending on the classifier, even when using the same dataset, it is important to use IIoT datasets and models suitable for classifying malicious traffic.

Several studies have been conducted to evaluate and compare performance using IIoT datasets and ML classifiers. However, it is difficult to grasp the performance comparison results for each IIoT dataset and ML classifier at a glance because each experimental environment implemented is different in previous studies, and multiple types of ML classifiers were not used in numerous IIoT datasets. Therefore, this paper aims to research IIoT data sets and ML classifiers to classify malicious traffic in the IIoT environment. For this purpose, several ML classifiers are used on different IIoT datasets. We then compare the performance of each ML classifier using several performance metrics. You can derive an ML classifier suitable for each dataset and check the performance of the ML classifier on various IIoT datasets at a glance.

This research is expected to contribute as follows:

- Comparison of ML classifier performance through various IIoT datasets and model learning in the same experimental environment
- Analyze the characteristics and limitations of IIoT data sets through IIoT data set analysis.
- Research and analysis of data set and model supplementation measures to improve ML classifier performance.

The structure of this paper is as follows. In Chapter 2, we review research on classifying malicious traffic in IIoT environments using IIoT datasets and ML classifiers. Chapter 3 describes the IIoT dataset, data preprocessing, and ML classifier used in the experiment. In Chapter 4, we look at ways to supplement the dataset and model to improve model performance based on confirmation and analysis of experimental results. Chapter 5 concludes this paper with conclusions and future research.

2 Related Works

[4] applied filter-based feature reduction techniques using the UNSW-NB15 dataset and the XGBoost algorithm. They then implemented ML approaches: Support Vector Machine (SVM), K-

Nearest Neighbor (K-NN), Logistic Regression (LR), Artificial Neural Network (ANN), and Decision Tree (DT). They showed that the XGBoost-based feature selection method could increase the test accuracy from 88.13% to 90%.

[5] trained an XGBoost ML model to perform class balancing using chi-squared (Chi 2) feature selection technique and Synthetic-Multiplicative oversampling (SMOTE) technique to establish a high level of security for Vehicular ad hoc networks (VANETs), a subsystem of Intelligent Transportation Systems (ITS) that allows vehicles to communicate over wireless communication infrastructure. ML approaches were implemented for performance comparisons, such as K-NN, Random Forest (RF), AdaBoost, and XGBoost.

[6] proposed an approach to offload ML model selection tasks to the cloud and real-time prediction tasks to fog nodes. The proposed method enables real-time detection of attacks on fog nodes after building ensemble machine-learning models in the cloud based on historical data. The proposed ML approach is tested using the NSL-KDD dataset. The results are presented using several performance metrics such as precision (PRE), recall (REC), accuracy (ACC), and ROC simulation experiments.

[7] proposed a hierarchical architecture that Integrates Blockchain Technology (BCT) and ML in the context of IIoT for smart manufacturing applications. Experiments based on IIoT datasets were conducted using ACC, PRE, sensitivity, and Matthews Correlation Coefficient (MCC) performance metrics. The ML classifiers used for training are ANN, DT, RF, Naïve Bayes (NB), AdaBoost, and SVM.

However, the NSL-KDD dataset used in existing studies needs to be more comprehensive, contain many duplicate records, and have been produced for a long time, which may reduce model performance. Additionally, in the case of studies that use only accuracy as an evaluation metrics, the model is trained biased to optimize for that indicator, or the model cannot be evaluated from various aspects. If the classes are unbalanced, it is difficult to properly evaluate the model's performance using only the accuracy evaluation metrics. If an ML model is trained on only one dataset rather than multiple datasets, it is difficult to check the ML model performance on various datasets.

3 Simulation Experiment

This session provides information about IIoT datasets, the data preprocessing process, and the ML classifier to be used.

3.1 IIoT Datasets

The UNSW-NB15 dataset was proposed by [8, 9] in 2015. Raw network packets were generated by the IXIA PerfectStorm tool at the Cyber Range Lab at UNSW Canberra. The UNSW-NB15 dataset consists of 9 attack types (DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms, Fuzzers, Analysis, and Backdoor), including abnormal packets. The dataset is saved in 4 CSV files and consists of training and test datasets. The total number of records in the training and test datasets are 175,341 and 82,332, respectively.

The ToN-IoT dataset is an IoT and IIoT dataset proposed by [10, 11] in 2020. It was created to evaluate the fidelity and efficiency of various cybersecurity applications based on ML/DL algorithms. The dataset was saved in multiple CSV files. The ‘Processed_IoT_dataset’ folder contains 7 datasets saved as CSV files by filtering the raw data sets. Among them, the ‘IoT_Modbus’ dataset consists of data on Modbus Function Code (FC) that performs discrete value, coil, and input/holding register reading functions. This paper uses the IoT_Modbus dataset included in the Ton-IoT dataset.

The WUSTL-IIoT-2021 dataset was created by [12] in 2021 and is a network-based dataset of IIoT applications that models and emulates real industrial systems for cybersecurity research. It includes a variety of IIoT sensors and actuators, HMI, and PLC devices to simulate real industrial applications [13,

14]. The WUSTL-IIoT-2021 dataset includes four attack types (Reconnaissance, Backdoor, Command Injection, and DoS).

The X-IIoTID dataset was created by [15] in 2021 and is a dataset for evaluating and training ML/DL-based IDS for IIoT systems. The architecture used consists of three levels: edge level, platform level, and enterprise level, and various industrial and IoT devices, protocols, cloud services, and attack machines are deployed at each level. The X-IIoTID dataset includes 9 attack types (Reconnaissance, Weaponization, Exploitation, Lateral Movement, Command and Control, Exfiltration, Tampering, Crypto-Ransomware, and Ransom DoS).

The Edge-IIoT dataset was created by [13] in 2022 and is used for the cybersecurity of IoT and IIoT applications. It consists of multiple datasets from various devices, including temperature/humidity digital sensors, water level sensors, and pH sensor meters. The Edge-IIoT dataset includes 5 attack types (DoS, DDoS, Information Gathering, Injection, AITM). The dataset contains 20,952,648 records (Normal: 11,223,940, Attack: 9,728,708). The 'ML-EdgeIIoT-dataset.csv' file in the Edge-IIoT dataset contains the dataset selected for evaluating ML-based intrusion detection systems. In this study, ML-EdgeIIoT-dataset is used.

[Table 1] overviews of feature and record information for 5 IIoT datasets.

Datasets	Year	Features	Normal Records	Malicious Records	Total Records
UNSW-NB15	2015	47	2,218,761	321,283	2,540,044
ToN-IoT (IoT_Modbus)	2020	8	405,904	72,489	478,393
WUSTL-IIoT-2021	2021	41	1,107,448	87,016	1,194,464
X-IIoTID	2021	59	421,417	399,417	820,834
Edge-IIoT (ML-EdgeIIoT)	2022	61	24,301	133,499	157,800

Table 1: OverView of the IIoT Datasets

3.2 Data preprocessing

- Remove features that are not necessary for the model to classify data. For example, the dataset contains features unrelated to learning, such as date or time. Therefore, the label corresponding to the feature was removed from the dataset.
- Remove features related to the host. For example, the dataset includes host-related features such as StartTime, LastTime, SrcAddr, DstAddr, ip.src_host, and ip.dst_host. These features attack the model and can have a negative impact on model learning. Because this may degrade model performance, labels corresponding to features related to the host were removed from the dataset.
- Process missing values present in the data set. After checking whether Null, None, or NAN data exists in the dataset, missing values must be removed or filled. Therefore, Null, None, and NAN data were replaced with the real number 0.
- Perform a normalization process to convert categorical, character, and boolean data into numeric variables. When learning a model, categorical and character data cannot be recognized, and an error occurs. Therefore, the categorical, character, and boolean data included in the dataset were converted into floats corresponding to numeric variables using the LabelEncoder class supported by the Scikit-Learn library.
- Separate the data set into the training and test data sets. Therefore, 80% of all 5 datasets were used as training datasets, and the remaining 20% were used as test datasets.

3.3 ML (Machine Learning) Classifier

Learning methods for machine learning models are divided into supervised, unsupervised, and reinforcement learning (RL). First, supervised learning uses a learned model to predict the output of test data [16]. Unsupervised learning is used to find hidden or useful patterns in a dataset without labels or categories of existing data [17]. Reinforcement learning is used to optimize decision-making according to the steps [18].

[Table 2] overviews of the machine learning classifier to be used in the experiment.

ML Classifier	Descriptions
Logistic Regression [19]	LR is a supervised learning model that uses a logistic function, and instead of predicting continuous data, it classifies data as true or false.
K-Nearest Neighbors [20]	K-NN is a model that uses a non-parametric algorithm in classification and regression tasks. To predict a class, the model assigns a test sample's class based on most of the k nearest neighbors of a given test sample.
Random Forest [21]	RF consists of decision trees and can be used for classification or regression problems. For classification, predictions are made based on a majority vote of the predictions using a decision tree, but for regression, the result is the average of the tree outputs.
Naive Bayes [22]	NB provides feature learning independent of a given class and uses and learns probabilistic knowledge. It is based on the Bayesian theorem and is suitable for high input dimension classification.
Decision Tree (DT) [20]	DT is a tree-type classification model. Each node in the tree specifies a test for a single feature, and each branch descending from that node corresponds to one of the possible values for that feature.
Light Gradient-Boosting Machine [23]	LightGBM is based on decision trees and is used for several ML techniques tasks such as classification, ranking, and regression. It is designed to create fast, distributed algorithms to process large data sets. Additionally, it features fast training speed and low memory usage.
eXtreme Gradient Boost [24]	XGBoost is a decision tree-based ensemble ML algorithm that uses the Gradient Boost Framework, sequentially adds predictors, and modifies previous models.
Adaptive Boost [25]	AdaBoost constructs multiple uncorrelated weak learners and then combines their predictions. The algorithm trains each weak learner sequentially and assigns weights to all instances. The next training set of samples is based on the instances' weights, and the entire process is repeated.
Categorical Boost [26]	CatBoost uses the Gradient Boost Framework. We use one-time encoding to implement a symmetric tree that handles categorical features and helps reduce prediction time.
Multi-Layer Perceptron [27]	MLP is a backpropagation model that uses delta learning rules to spread errors through the network. It consists of an input layer, a hidden layer, and an output layer.

Table 2: OverView of the ML Classifier

4 Results and Discussion

This session provides an overview of the hardware and software used in the implemented experimental environment and explains the evaluation metrics used to compare the performance of machine learning models. After confirming and analyzing the experimental results, we discuss ways to supplement the dataset and model to improve model performance based on the analyzed information.

4.1 Implementation Environment and Performance Evaluation

[Table 3, 4] overviews of the hardware and software specifications used for model training, testing, and preprocessing in the implemented experimental environment.

Hardware configuration	CPU	GPU	RAM
Device	Intel(R) Xeon(R) Silver 4216	Tesla V100S-PCIE-32GB	64GB DDR4
Core	16	640(Tensor)/5120(CUDA)	-
Clock Speed	2.10GHz	1.23GHz	-

Table 3: Hardware specifications of the implementation environment

Software configuration	OS	Python	Scikit-Learn
	Ubuntu 20.04.5 LTS	3.9.7	1.0.2

Table 4: Software specifications of the implementation environment

The results derived from the experiment were expressed as ACC, REC, PRE, ROC (Receiver Operating Characteristic)-AUC (Area Under Curve) Score, and F1-Score, which are evaluation metrics used quantitatively to compare performance. The evaluation metrics can be expressed using True Positive (TP), True Positive (TN), False Positive (FP), and False Negative (FN) values derived from the confusion matrix [28].

ACC can be used to identify the proportion of correctly classified predictions. REC can be used to identify the ratio of positive predictions to all positive cases. PRE can be used to identify the proportion of positive predictions that are correctly classified. F1-Score is the weighted average of PRE and REC and has a value between 0 and 1.

The evaluation metrics are expressed mathematically as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad REC = \frac{TP}{TP + TN}$$

$$PRE = \frac{TP}{TP + FP} \qquad F1\ Score = 2 \times \frac{REC \times PRE}{REC + PRE}$$

ROC (Receiver Operating Characteristic)-AUC (Area Under Curve) Score is a graph used to plot model results at various thresholds when making predictions. The graph uses TPR (True Positive Rate) and FPR (False Positive Rates).

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

4.2 Experimental Results

[Table 5-9] shows the testing results after learning ML classifiers using UNSW-NB15, ToN-IoT (IoT_Modbus), WUSTL-IIoT-2021, X-IIoTID, and Edge-IIoT (ML-Edge-IIoT) datasets. Subsequently, evaluation metrics were used to represent the performance of each ML classifier.

[Table 5] shows test results using evaluation metrics after training an ML classifier using the UNSW-NB15 data set.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
LR	83.17	98.94	80.72	74.18	88.90
K-NN	88.87	93.57	90.43	86.20	91.97
RF	95.77	97.98	95.89	94.50	96.93
NB	82.65	97.89	80.74	73.96	88.49
DT	94.44	95.96	95.89	93.58	95.92
LightGBM	95.96	98.00	96.14	94.79	97.06
XGBoost	95.95	97.96	96.16	94.80	97.05
AdaBoost	93.61	97.22	93.65	91.56	95.40
CatBoost	95.83	98.02	95.94	94.58	96.97
MLP	86.66	96.77	56.66	80.89	90.81

Table 5: Classification Results obtained using the UNSW-NB15 Dataset

[Table 6] shows test results using evaluation metrics after training an ML classifier using the ToN-IoT (IoT_Modbus) dataset.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
LR	55.31	41.11	22.60	50.25	29.17
K-NN	86.23	71.55	68.41	81.01	69.95
RF	96.80	87.92	97.55	93.64	92.48
NB	50.04	0.50	0.22	50.02	30.94
DT	94.95	90.14	87.66	93.24	88.88
LightGBM	79.36	9.13	87.35	54.37	16.53
XGBoost	82.19	21.44	95.63	60.58	35.04
AdaBoost	77.61	0.05	58.33	50.02	0.10
CatBoost	77.90	1.50	87.78	50.72	2.96
MLP	51.71	-	-	-	-

Table 6: Classification Results obtained using the ToN-IoT (IoT_Modbus) Dataset

[Table 7] shows test results using evaluation metrics after training an ML classifier using WUSTL-IIoT-2021 dataset.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
LR	94.75	33.13	86.16	66.36	47.95
K-NN	99.77	98.73	98.20	99.29	98.47
RF	99.88	99.86	98.60	99.87	99.23
NB	95.55	47.91	84.40	73.61	61.13
DT	99.87	99.75	98.56	99.81	99.15
LightGBM	99.85	99.74	98.30	99.80	99.01

XGBoost	99.89	99.85	98.53	99.86	99.19
AdaBoost	99.56	99.77	94.57	99.66	97.10
CatBoost	99.78	99.45	97.59	99.63	98.51
MLP	92.71	-	-	-	-

Table 7: Classification Results obtained using the WUSTL-IIoT-2021 Dataset

[Table 8] shows test results using evaluation metrics after training an ML classifier using X-IIoTID dataset.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
LR	66.86	92.13	60.46	67.53	73.01
K-NN	94.78	94.63	94.64	94.77	94.63
RF	99.49	99.19	99.76	99.48	99.48
NB	56.57	97.14	52.92	57.64	68.52
DT	99.35	99.40	99.26	99.35	99.33
LightGBM	99.71	99.65	99.76	99.71	99.70
XGBoost	96.66	99.52	99.77	99.65	99.65
AdaBoost	95.72	94.63	96.51	95.69	95.56
CatBoost	99.47	99.28	99.64	99.47	99.46
MLP	52.68	99.99	50.69	53.92	67.27

Table 8: Classification Results obtained using the X-IIoTID Dataset

[Table 9] shows test results using evaluation metrics after training an ML classifier using Edge-IIoT (ML-Edge-IIoT) dataset.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
LR	85.76	99.87	85.61	55.20	92.19
K-NN	87.32	95.29	90.20	70.07	92.68
RF	99.53	99.73	99.71	99.09	99.72
NB	30.07	18.30	97.16	57.72	30.80
DT	95.34	99.91	94.83	85.44	97.30
LightGBM	99.50	99.72	99.68	99.02	99.70
XGBoost	99.48	99.71	99.66	98.97	99.69
AdaBoost	99.21	99.79	99.27	97.96	99.53
CatBoost	99.54	99.68	99.77	99.23	99.73
MLP	88.22	99.30	88.20	64.24	93.42

Table 9: Classification Results obtained using the Edge-IIoT (ML-Edge-IIoT) Dataset

4.3 Analysis and Discussion

[Table 10] shows the ML classifier that provided the highest performance index for each dataset. In the classification results obtained using the UNSW-NB15 dataset, the LightGBM classifier and the XGBoost classifier achieved similar performance. The RF classifier achieved the highest performance in the classification results obtained using the ToN-IoT (IoT_Modbus) dataset and the WUSTL-IIoT-2021 dataset. In the classification results obtained using the X-IIoTID dataset, the LightGBM classifier achieved the highest performance. In the classification results obtained using the Edge-IIoT (ML-Edge-IIoT) dataset, the CatBoost classifier achieved the highest performance.

ML Classifier	Accuracy (ACC)	Recall (REC)	Precision (PRE)	ROC-AUC Score	F1-Score
UNSW-NB15	LightGBM	LR	XGBoost	XGBoost	LightGBM
ToN-IoT (IoT_Modbus)	RF	DT	RF	RF	RF
WUSTL-IIoT-2021	XGBoost	RF	RF	RF	RF
X-IIoTID	LightGBM	MLP	XGBoost	LightGBM	LightGBM
Edge-IIoT (ML-Edge-IIoT)	CatBoost	DT	RF	CatBoost	CatBoost

Table 10: Highest performance metrics ML classifier for each dataset

Through the feature and record information of the dataset, it can be seen that there is a dataset in which the ratio of normal and abnormal data is unbalanced. In this case, problems such as overfitting and increased model complexity occur, preventing the model from properly training. Ultimately, the accuracy of classification and performance evaluation of normal and abnormal data may be less likely to decrease. Therefore, before learning a model, the data must be balanced by augmenting or reducing specific classes through undersampling or oversampling.

The performance of ML classifiers using the ToN-IoT (IoT_Modbus) dataset is overall low. You can see that the number of features is small compared to other datasets. Suppose an ML model learns a pattern with an insufficient number of dataset features. In that case, the lack of learning data will result in a lack of features needed to identify the difference between malicious traffic and normal traffic or specific types. Because of this, the model's performance may decrease, and the attack may be incorrectly determined. Accordingly, the model's performance can be improved by performing the Feature Engineering process, which creates new features using existing data [29].

Among the methods for improving the performance of ML classifiers, optimal model performance can be derived through hyperparameter tuning, in which the user directly sets the model parameters. [30] applied the Grid Search technique for hyperparameter optimization. They improved the performance of a tree-based ML classifier by searching for optimal parameter values.

5 Conclusions and Future Work

This paper shows the importance of security for IIoT devices used in industrial environments and the increasing number of cyberattacks. We researched IIoT datasets and machine learning classifiers for classifying malicious traffic in the IIoT environment. We briefly explained the 5 IIoT datasets (UNSW-NB15, ToN-IoT (IoT_Modbus), WUSTL-IIoT-2021, X-IIoTID, Edge-IIoT (ML-Edge-IIoT)) and 10 ML classifiers (LR, K-NN, RF, NB, DT, LightGBM, XGBoost, AdaBoost, CatBoost, MLP) to be used in the experiment. The data preprocessing process of the dataset was explained. 5 evaluation metrics (ACC, REC, PRE, ROC-AUC Score, F1-Score) were used to evaluate the performance of ML classifiers. The ML classifiers that showed high performance in each IIoT dataset were LightGBM, CatBoost, and RF classifiers. As a result of the analysis observed deterioration in model performance due to several factors, such as normal and abnormal data imbalance in the dataset and insufficient number of features. As ways to supplement the dataset and model to improve model performance, undersampling or oversampling, Feature Engineering, and hyperparameter tuning using Grid Search techniques were explained.

In future work, we will research to create a new IIoT dataset that complements the shortcomings of the existing IIoT dataset.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00241376, Development of security monitoring technology-based network behavior against encrypted cyber threats in maritime environment)

References

[1] Mekala, S. H., Baig, Z., & Anwar, A. (2022, December). Industrial Internet of Things (IIoT): Testbed and Datasets for CyberSecurity and Digital Forensics. In 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) (pp. 1-10). IEEE.

[2] Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. Ieee Access, 8, 165130-165150.

[3] Ankita, Rani, S., Singh, A., Elkamchouchi, D. H., & Noya, I. D. (2022). Lightweight hybrid deep learning architecture and model for security in IIOT. Applied Sciences, 12(13), 6442.

[4] Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7, 1-20.

[5] Gad, A. R., Nashat, A. A., & Barkat, T. M. (2021). Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. IEEE Access, 9, 142206-142217.

[6] Tomer, V., & Sharma, S. (2022). Detecting iot attacks using an ensemble machine learning model. Future Internet, 14(4), 102.

[7] Mrabet, H., Alhomoud, A., Jemai, A., & Trentesaux, D. (2022). A secured industrial Internet-of-things architecture based on blockchain technology and machine learning for sensor access control systems in smart manufacturing. Applied Sciences, 12(9), 4641.

[8] Moustafa, N., & Slay, J. (2015, November). 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) (pp. 1-6). IEEE.

[9] UNSW SYDNEY, The UNSW-NB15 Dataset(2015), https://research.unsw.edu.au/projects/unsw-nb15-dataset

[10] Moustafa, N., Keshky, M., Debiez, E., & Janicke, H. (2020, December). Federated TON_IoT Windows datasets for evaluating AI-based security applications. In 2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom) (pp. 848-855). IEEE.

[11] UNSW SYDNEY, The TON_IoT Datasets Dataset(2020), https://research.unsw.edu.au/projects/toniot-datasets

[12] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain. "WUSTL-IIOT-2021 Dataset for IIoT Cybersecurity Research," Washington University in St. Louis, USA, October 2021, http://www.cse.wustl.edu/~jain/iiot2/index.html

[13] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281-40306.

[14] Al-Hawawreh, M., Sitnikova, E., & Aboutorab, N. (2021). X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things. IEEE

[15] Alani, M. M. (2023). An explainable efficient flow-based Industrial IoT intrusion detection system. Computers and Electrical Engineering, 108, 108732.

[16] Khalil, M., McGough, A. S., Pourmirza, Z., Pazhoohesh, M., & Walker, S. (2022). Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption—A systematic review. Engineering Applications of Artificial Intelligence, 115, 105287.

[17] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

[18] Martín-Guerrero, J. D., & Lamata, L. (2021). Reinforcement learning and physics. Applied Sciences, 11(18), 8589.

[19] Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., ... & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. Sensors, 21(2), 446.

[20] Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., & Aloul, F. (2020, November). Botnet attack detection using machine learning. In 2020 14th International Conference on Innovations in Information Technology (IIT) (pp. 203-208). IEEE.

[21] Alduailij, M., Khan, Q. W., Tahir, M., Sardaraz, M., Alduailij, M., & Malik, F. (2022). Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method. Symmetry, 14(6), 1095.

[22] Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., & Sakurai, K. (2020). Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors, 20(16), 4372.

[23] Ghourabi, A. (2022). A security model based on lightgbm and transformer to protect healthcare systems from cyberattacks. IEEE Access, 10, 48890-48903.

[24] Cherif, I. L., & Kortebi, A. (2019, April). On using extreme gradient boosting (XGBoost) machine learning algorithm for home network traffic classification. In 2019 Wireless Days (WD) (pp. 1-6). IEEE.

[25] Firdaus, G. M., & Suryani, V. (2023, August). DDoS Attack Detection Analysis Using Ensemble Learning with XGBoost and AdaBoost Algorithms. In 2023 11th International Conference on Information and Communication Technology (ICoICT) (pp. 17-22). IEEE.

[26] Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.

[27] Kilincer, I. F., Ertam, F., Sengur, A., Tan, R. S., & Acharya, U. R. (2023). Automated detection of cybersecurity attacks in healthcare systems with recursive feature elimination and multilayer perceptron optimization. Biocybernetics and Biomedical Engineering, 43(1), 30-41.

[28] Alkadi, S., Al-Ahmadi, S., & Ben Ismail, M. M. (2023). Toward Improved Machine Learning-Based Intrusion Detection for Internet of Things Traffic. Computers, 12(8), 148.

[29] Khan, A. S., Ahmad, Z., Abdullah, J., & Ahmad, F. (2021). A spectrogram image-based network anomaly detection system using deep convolutional neural network. IEEE Access, 9, 87079-87093.

[30] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv preprint arXiv:1912.06059.