National College of Ireland

Anomaly Detection Method for OT/ICS Environment Using Ensemble Learning

MSc Research Project
Master of Science in Cyber Security

Shifan Anwar Sayyed
Student ID: 22193162

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shifan Anwar Sayyed
Student ID:	22193162
Programme:	Master of Science in Cyber Security **Year:** 2023-2024
Module:	MSc Research Practicum
Supervisor:	Jawad Salahuddin
Submission Due Date:	12th August 2024 14:00
Project Title:	Anomaly Detection Method for OT/ICS Environment Using Ensemble Learning
Word Count:	7305 **Page Count:** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shifan Anwar Sayyed

Date: 12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Anomaly Detection Method for OT/ICS Environment Using Ensemble Learning

Shifan Anwar Sayyed
22193162

Abstract

In this research work we have looked into the current state of OT/ICS security, how important OT/ICS infrastructures are, what is their operational disruption impact and why it is important to secure them. Going forward we have delved into the literature review of relevant research papers, discussing challenges of OT/ICS space and approaches used for securing OT/ICS environment using different types of machine learning algorithms and techniques. This paper proposes an Anomaly Detection method based on ensemble learning model for securing OT/ICS environment. At the end we have discussed the implementation part carried out to develop an anomaly detection system, data transformation and discussion of the result and finally critically analysing the research and discussing the limitation.

Keywords – Machine Learning, Supervised Algorithm, Ensemble Learning, Electra Dataset, Operational Technology (OT), Industrial Control Systems (ICS), Cybersecurity.

1 Introduction

1.1 Background

OT (Operational Technology) / ICS (Industrial Control Systems) exist to monitor, regulate and control critical industrial processes like water supply management, energy power grids, transportation, manufacturing, oil and gas etc. Back in the days, the OT/ICS environments used to be air-gapped and isolated from the internet and didn't have a major focus on security of their network as the networks were isolated from the outside world (Jeffrey, Tan and Villar, 2024a).

In today's scenario OT/ICS networks are being integrated with IT (Information Technology) to make the OT/ICS environment more efficient in terms of monitoring, controlling and remote connectivity. But on the flip side, this merger of OT and IT has brought significant cybersecurity risks along with it. Which means that cyber-attacks which were previously occurring in Information Technology (IT) environments were now being targeted on OT/ICS networks as well.

The area of securing OT/ICS networks has not been researched heavily because providing security for OT networks is very difficult as small downtime of these networks leads to huge disruption which causes financial and reputation loss for the industry, government as well as the public (Jeffrey, Tan and Villar, 2024a). For example, Down-time of critical OT networks/environments can cause equipment damage, load shedding, power grid failure which can cause blackouts which does not only affects and damages the OT network/environment but also the government and public infrastructure as well. Also, OT/ICS networks many times use legacy technology which have many factors that prevent

them from being upgraded, like age of the equipment, mission criticality of the equipment, location of the equipment, highly sensitive to the jitter/delay etc. Due to which they by default become an easy target for cyber attackers and difficult for OT network owners to secure.

1.2 Importance

SANS Institute 2023 ICS/OT Cybersecurity Survey (Parsons, 2023) states that 25% of the survey responders considered cybersecurity threats against ICS/OT environment to be of Critical severity and 43.9% considered them to be of High severity. It also states that disruption to OT/ICS environments could negatively impact the safety of people.

Below are a few real-world events/cyber-attacks which occurred on OT/ICS networks causing huge damage:

- The most famous one is the Stuxnet malware (Kushner, 2013) which was designed for Iranian nuclear facilities where more than 200,000 machines of 14 Iranian nuclear facilities were destroyed leading to delayed Iranian uranium enrichment program.
- In 2015 (Cybersecurity & Infrastructure Security Agency, 2021), Ukraine's power grid got disrupted due to a malware named BLACKENERGY3. Approximately 225,000 plus customers were left without power for 1– 6 hours.
- In 2016 (Storm et al., 2023), a malware named CRASHOVERIDE was utilized to disrupt Kyiv's power grid. Due to the cyber-attack the transmission level substation was disrupted causing an outage for public for 1 hour.

As emphasized and elaborated with the examples provided above, the downtime or operational disruption of OT/ICS environments can cause a huge impact not only on the OT/ICS environment but also on people, hence research for securing or improving the current state of security of OT/ICS networks is very important. Successfully contributing to improving the security of OT/ICS networks can help protect critical infrastructure, prevention from financial loss and ensuring public safety which is very critical and importance.

1.3 Research Question

1.3.1 Research Problem

OT/ICS environments are critical infrastructures whose operational disruption causes huge financial losses and impact to public safety. The security space for OT/ICS systems is not as researched and mature as the IT space (Jeffrey, Tan and Villar, 2024a). In a 2023 Survey by OTORIO and ServiceNow (OTORIO, 2023) where the survey was about understanding the current state OT/ICS security, where 58% percentage of the respondents mentioned the risk severity level to OT/ICS networks as Critical and 53% of the respondents didn't have OT/ICS security in place for their OT/ICS networks. Hence it is very important to research into methods for securing the OT/ICS environments without having the need for making infrastructural-level changes and without having to put over-head on the OT/ICS system.

1.3.2 Research Question

1.3.2.1 Why is it important?

OT/ICS environments are critical infrastructures, which exist to monitor, regulate and control critical industrial processes like water supply management, energy power grids, transportation, manufacturing, oil and gas etc. Cyberattacks on these environments cause huge disruption resulting in financial loss and impact on public safety. SANS Institute 2023 ICS/OT Cybersecurity Survey (Parsons, 2023) states that 25% of the survey responders considered cybersecurity threats against ICS/OT environment to be of Critical and 43.9% considered them to be of High severity. Hence it is important to look into securing OT/ICS environments.

1.3.2.2 What can be done to address the problem?

In SANS Institute 2023 ICS/OT Cybersecurity Survey (Parsons, 2023), respondents mentioned the top 3 items of most importance for OT/ICS cybersecurity, out of which one was detection of threats entering or navigating the OT/ICS network. Hence for the detection of threats we will be looking into developing an Anomaly Detection System using Machine Learning (ML) methods.

1.3.2.3 How will you carry out the work?

An Ensemble Learning model will be developed where Ensemble Learning is the method of combining the predictions of multiple individually trained machine learning models to improve the overall performance of the model. The individual machine learning models will be selected based on the best performing algorithm. The final developed Ensemble Learning model will be used as an Anomaly Detection System in OT/ICS environments. The performance of Base Classifiers (Individual Algorithms) and the Ensemble Learning model will be evaluated using Accuracy, Precision, Recall, F1-Score, False Positive Rate (FPR) and False Negative Rate (FNR).

RQ: Can an Ensemble Learning Model constructed from top-performing base classifiers, effectively serve as an Anomaly Detection System in OT/ICS environments, achieving high accuracy while maintaining low False Positive Rates (FPR) and False Negative Rates (FNR)?

1.4 Limitations

As we are using Machine Learning to develop an Anomaly Detection System, the performance of the model highly depends on the quality of the dataset used. Here the limitation arises as in the OT/ICS space good quality labelled datasets are difficult to find as creating datasets or capturing real-scenario traffic from an actual testbed requires expertise. And if we get a good quality dataset it might have the issue of the dataset being small. Hence the first issue to tackle would be to find a good quality dataset or improve an already existing dataset to a good quality.

1.5 Structure Of the Report

This research paper discusses the development and implementation of an Anomaly Detection System using Ensemble Learning methods. **Section 1. Introduction** presents the background of the research topic, why it is important, the research problem and research question along with the limitation of the method of the paper. **Section 2. Literature Review** discusses different research papers where researchers have developed Anomaly Detection Systems using different machine learning algorithms, using supervised, unsupervised algorithm and using ensemble learning method. **Research Methodology is discussed in Section 3.** where

the equipment used for the experiment, algorithm selection, dataset selection process and information about the dataset is presented. **Section 4. Design Specification** discusses machine learning techniques and code flow. **Section 5. Implementation** discusses the implementation carried out for developing Anomaly Detection using Ensemble Learning methods. **Section 6. Evaluation** presents the metrics used for evaluation of the developed model, discussion about the output of the models, experiment contribution and critical analysis of the experiment and the methods used. **Section 7. Conclusion and Future Work** discusses the conclusion of the research and future works.

2 Literature Review

Author (S, Selvan and Ramkumar, 2023) focuses on implementation of Intrusion Detection System using anomaly-based detection method where they use Pearson Correlation for feature selection from the dataset and Deep Neural Network for anomaly detection. The dataset used in this paper is a popular dataset called HAI 3.0 also known as HIL-based Augmented ICS Security Dataset (ICS (Industrial Control System) Security Dataset, 2022), which is a simulation of steam turbine power production and pumped-storage hydropower facility. The dataset consists of total 361,200 datapoint (rows) and 86 features (columns). Their proposed method achieved an accuracy of 99.24%. The strength of the proposed model is that it uses Pearson Correlation for feature selection which helps in removing irrelevant or similar features from the dataset that might affect the accuracy. The machine learning model used in this paper is Deep Neural Network which sometimes cause obstacle due to their high computational cost and complex implementation.

Author (Choi and Kim, 2024) proposed an unsupervised learning model for detecting anomalous traffic. The paper utilized Principal Component Analysis (PCA) for feature selection and K-Means algorithm for clustering the data and a composite autoencoder model with Convolutional Neural Network (CNN), Rectified Linear Unit (ReLU) and Long Short-Term Memory (LSTM) as layers in the autoencoder. The dataset used for this experiment is the HIL-based Augmented ICS Security Dataset (ICS (Industrial Control System) Security Dataset, 2022) from which the Attack label column was removed as the author is using unsupervised learning models. The strength of the proposed model is that it uses an unsupervised model which does not requires pre-labelled dataset which is a big issue in the OT/ICS space as generating labelled datasets is a difficult task as it requires significant efforts and expertise. Also using an unsupervised algorithm makes model highly adaptable to different OT/ICS environments. But unsupervised algorithms are very prone to false positive detections which is a big deal when developing an anomaly detection.

Author (Araya et al., 2017) makes use of an Ensemble learning model for building a pattern-based anomaly detector called the – "Collective Contextual Anomaly Detection Using Sliding Window" (CCAD-SW) framework where multiple anomaly detection classifiers like pattern-based and prediction-based anomaly classifiers are used. The paper also proposes an ensemble anomaly detection (EAD) framework for enhancing the anomaly detection capacity where the framework makes use of several classifiers using the majority voting technique. The CCAD-SW framework makes use of an autoencoder with two prediction classifiers implemented using Support Vector Regression (SVR) and Random Forest. The proposed paper makes use of the dataset provided by Powersmiths company (Eldridge, 2019) which focuses on producing sensor devices where the company collects data from these sensors which is used as the dataset. Though this paper by objective/goal does not focuses on threat anomaly detection but the methods used in this paper for detecting anomalous energy usage

can be used for developing a threat anomaly detection model. Results of the paper conclude that ensemble learning methods are more effective in anomaly detection than individual algorithms.

Author (Jeffrey, Tan and Villar, 2024a), discuss how the security industry for IT environments is matured as compared to OT/ICS industry and how the ICS environments school of thought is when considering security. The author mentions that the IT security industry is very matured for example the industry has many large vendors like Checkpoint, Cisco, Palo-Alto, Microsoft, Sophos, etc providing various types of security tools whereas the OT/ICS industry lacks such tools as the ICS infrastructures differ hugely when it comes to standard of protocols and hardware. Also, the incorrect school of thought where they keep security on the back-track making the incorrect assumption of their environment being isolated from the hostile internet network and them being on a trusted network. In the paper a total of 310 papers and online articles were reviewed of which the top 5 publishers were IEEE - 47%, ScienceDirect - 15%, Springer - 12%, ACM - 9%, MDPI - 4%, all others - 13%, where the common two security solutions being stated for securing the OT/ICS environments were Security by design and Anomaly Detection/Threat Detection. Security by design can only be used in scenarios where a new OT/ICS infrastructure is being built which is infeasible as OT/ICS infrastructures have a very long life. The second option which is developing an Anomaly Detection System which seems to be achievable as it doesn't require any infrastructural level change. The author discusses about the challenge in OT/ICS space where the datasets available have a large amount of normal/benign traffic and a small amount of attack traffic which makes the dataset imbalanced hence the author proposes that the anomaly detection must be modelled as a one-class problem where the model is trained on the normal data which is readily available and anything deviating from the trained class is tagged as anomaly. The author proposes an approach of building an anomaly detection method using a hybrid model framework where a single classifier model with one-class SVM and one-class KNN model is used. The dataset used in this paper is a proprietary dataset, that was generated by the author programmatically using only benign. The testbed from where the dataset was generated is based on a scaled-down pilot system for a commercial greenhouse facility. The 1-class SVM model provided an accuracy of 68% where the model suffered from high false positives of 37% and high false negative rate of 32%. The 1-class KNN model provided an accuracy of 98% but this model also had excessive false positive prediction. The weakness of this paper is that due to the use of one-class models, the results have a considerable amount of false positives which is a big obstacle while developing anomaly detection.

The paper (Jeffrey, Tan and Villar, 2024b) discusses the challenges with machine learning for developing an anomaly detection for OT/ICS environment where the datasets are imbalanced having a large amount of normal traffic and small amount of attack traffic. Due to this the anomaly detection models developed have less predictive accuracy, high false positive or false negative predictions. Another point the author makes is about typically the developed anomaly model use one or two machine learning algorithms which are manual selected by the researcher due to which the results might be influenced by the limitation of the selected algorithm. Hence the author proposes an Ensemble Learning model which utilizes multiple algorithms which provide improved predictive performance when compared to single machine learning algorithm. Even minute improvements in the predictive performance of the model are highly desirable in OT/ICS environments which are extremely intolerant of false positives or false negative predictions. This strategy tackles the issue of bias introduced due to the imbalanced dataset, as combining multiple classifiers/algorithms leads to a strong

model. Hence the author proposes an ensemble learning anomaly detection model, where the base classifiers used are Logistic Regression, Support Vector Machine, Naive Bayes, Multi-Level Perceptron and K-Nearest Neighbour. The paper uses two publicly available dataset – "Edge-IIoTset2023 (Ferrag et al., 2022)" and "CICIoT2023 (Pinto et al., 2023)". The Edge-IIoTset2023 dataset was developed recently in 2023 which was collected from a testbed consisting of seven layers. The dataset consists of attacks including Mirai-udpplain, MITM-ArpSpoofing, DNS_Spoofing, Recon-PingSweep, Recon-PortScan, Recon-OSScan, Recon-HostDiscovery, XSS, CommandInjection, VulnerabilityScan, Backdoor_Malware, BrowserHijacking, DictionaryBruteForce, SqlInjection, and Uploading_attack. The version of dataset that the paper uses consists of 2,291,201 datapoints (rows) and 63 features (columns) of data where 85.9% is normal data and 14.1% is anomaly data. The second dataset used is CICIoT2023 which was developed by the Canadian Insitute for Cybersecurity where the dataset consists of 33 different cyber attacks against 105 different devices. The attacks that are included in this dataset are DDoS, DoS, Recon, Web-based, Brute Force, spoofing, and Mirai. The version of dataset that is used has 2,867,734 datapoints (rows) and 46 features (columns) of data where 61.7% of the data is anomaly data and 38.3% of the data is normal data.

In conclusion from the literature review of related work, we found out how the IT industry is much more mature in terms of security than the OT/ICS industry and how the merger of OT and IT has exposed the OT/ICS space to various cyber threats. To tackle these cyber threats the only two effective solutions discovered by a review of 310 papers were security by design and anomaly detection. Where developing an anomaly detection system seems practical and requires less effort than security by design. Diving into developing an anomaly detection for OT/ICS environment we found out about the challenges faced while developing anomaly detection where the problem is the availability of good OT/ICS datasets where the available datasets are highly imbalanced where the normal traffic is more when compared to the attack traffic in the dataset. The next phase in developing anomaly detection using machine learning algorithms is whether to use supervised or unsupervised algorithms where supervised algorithms require a labelled dataset and unsupervised algorithms do not require a labelled dataset. The use of unsupervised algorithms in developing an anomaly detection model is very dominant as it is difficult to find a labelled dataset which has a considerable amount of data points but unsupervised algorithms have excessive false positive and false negative prediction which is a big trade-off when it comes to anomaly detection. Whereas supervised algorithms perform better than unsupervised algorithms in the false positive and false negative predictions the only problem is that supervised algorithms require a labelled dataset with enough data points. Next, we found out that using and combining multiple algorithms than a single algorithm is much more effective in terms of accuracy, false positive and false negative predictions. Hence developing an anomaly detection model using the ensemble learning method (Combing multiple algorithms) is better. These were the findings discovered via the literature review of related work research papers.

3 Research Methodology

3.1 Equipment used in the research.

The main equipment used for implementing this research is an ASUS laptop on which Jupyter Notebook was ran via Anaconda Navigator. Below are the device and software specifications:

Laptop Model: ASUS TUF FX506HF-HN076W

Device name: LAPTOP-GFH5JVBM
Processor: 11th Gen Intel(R) Core (TM) i5-11260H @ 2.60GHz 2.61 GHz
Installed RAM: 16.0 GB (15.7 GB usable)
System type: 64-bit operating system, x64-based processor

Edition: Windows 11 Home Single Language
Version: 23H2
Installed on: 19-07-2023
OS build: 22631.3880
Experience: Windows Feature Experience Pack 1000.22700.1020.0

Anaconda Navigator Version: 2.5.0
Jupyter Notebook Version: 6.5.4

Python Version: 3.11.5
Pandas version: 2.0.3
Numpy version: 1.24.3
Scikit-learn version: 1.3.0

3.2 Dataset Selection and Transformation

3.2.1 Dataset Selection

As discussed in the literature review section, the datasets available have multiple issues like them being very imbalanced where the amount of normal traffic in the dataset is way more when compared to the attack traffic which affects the final output of the model. Second, that the availability of a good OT/ICS dataset is scarce as it requires great expertise to develop a simulated dataset. Third, it is difficult to find labelled datasets with enough data points, that won't affect the model results. Fourth, for our specific case, we need to find an OT/ICS dataset where the data are in appropriate format which for our research is in CSV format.

So, we had to find a dataset that tackles all the above-mentioned issue, hence we researched and tested multiple datasets. First, we tested the most used dataset we observed during our literature review which is the HAI (HIL-based Augmented ICS) Security Dataset (ICS (Industrial Control System) Security Dataset, 2022), where we used the latest version of the dataset which is version – "haiend-23.05" released on 31st May 2023. The HAI dataset is collected from a testbed with a Hardware-In-The-Loop (HIL) simulator which it simulates steam turbine power generation and pumped-storage hydropower generation. The dataset consists of a total of 11,80,800 data points out of which Normal Traffic Datapoints are 896400 and the attack traffic data points are 284400, where the split of normal and attack traffic in terms of percentage is 75.9% and 24.1% respectively. As this dataset looked promising, we carried out the data pre-processing phase for the dataset where we cleaned, performed feature selection and transformed the dataset and trained the ensemble learning model on the dataset, and the results obtained from the dataset were very high which we found to be odd. This issue could have been due to the split of attack and normal data points or the total data points count/size of the dataset or the feature values. Hence, we decided to try a different dataset which did not have the above-mentioned weaknesses as we wanted to be sure that these high results are correct and not due to some weakness in the dataset.

The next dataset we tried was from the Canadian Institute for Cybersecurity called – "CIC Modbus dataset 2023 (Canadian Institute for Cybersecurity, 2023; Kwasi Boakye-Boateng,

Ghorbani and Arash Habibi Lashkari, 2023)" where the dataset was generated from Wireshark network captures collected from a simulated testbed. Captures obtained from a simulated Docker environment testbed where the Docker containers were created to represent IEDs and SCADA HMIs. The dataset consists of various types of Modbus protocol attacks which include reconnaissance, query flooding, loading payloads, delay response, modify length parameters, false data injection, stacking Modbus frames, brute force write and baseline replay attacks. During the data screening phase of the dataset, we discovered a major issue which was the dataset was not in an appropriate format and all the data was in PCAP files which we tried converting into CSV files but the features in the PCAPs were not useful, hence we had to drop this dataset as well.

The next dataset we reviewed was the – "ICS-Flow Dataset (Dehlaghi-Ghadim et al., 2023)" where the dataset is generated using a bottle filling factory as a testbed simulation which includes the ICS that controls the equipment within the factory that controls components such as pipes, valves, conveyor belt and a water tank to fill the bottle with water from the tank. The attacks in the dataset are IP-Scan, Port-Scan, Replay, DDoS and MitM attacks. During the data screening phase, we discovered that the total number of data points in the dataset is 45,808 out of which the normal traffic is 30,326 and the attack traffic is 15,482. The count of data points in the dataset were very low, hence this dataset was also dropped.

The next dataset we reviewed was the – "Electra Dataset (Perception, 2016)" which was generated from network traffic of an electrical substation. The Electra Dataset had two versions of the dataset where one dataset had Modbus protocol and the other dataset had S7Comm protocol. The Modbus dataset contained 16,28,9277 labelled data points and the S7Comm dataset contained a total of 38,70,98466 labelled data points. As the S7Comm dataset contained more data points we selected the S7Comm dataset for further review. The dataset contained 7 attack types which are – Response, Command, Replay, Read, Write, False Error Response and MitM attack. This dataset looked promising as it checked all the boxes where the data points in the dataset were more than enough in count and were labelled and the dataset attack and normal data points could be under sampled/reduced to create a balanced dataset, hence we went ahead with the Electra Dataset.

Table 1: Dataset Summary.

Dataset	Datapoints	Normal Datapoints	Attack Datapoints	Features	Labelled	Attack Types	Reason
HAIend-23.05	11,80,800	896400 (75.9%)	284400 (24.1%)	226	Yes	42	The amount of Datapoints were not sufficient.
CIC Modbus Dataset 2023	N/A	N/A	N/A	N/A	N/A	N/A	Dataset not in the appropriate Format
ICS-Flow Dataset	45,808	30326 (66.20%)	15482 (33.80%)	43	Yes	5	The amount of Datapoints were not sufficient.
Electra Modbus Dataset	1,62,89,277	13894323 (85.34%)	2394954 (14.66%)	10	Yes	7	Less Datapoint as compared to S7Commm Dataset.
Electra S7Comm Dataset	38,70,98,466	264464599 (68.33%)	122633867 (31.67%)	10	Yes	7	Checked all the boxes of a good quality dataset, hence this dataset was selected.

3.2.2 Algorithm Type Selection

Once the dataset was finalised the next step was to select whether to use supervised or unsupervised algorithms. As discussed in the literature review section, researchers used unsupervised algorithms majorly due to the reason because they did not have a good quality labelled dataset with enough data points. Also found during the review of research papers that one major issue with using unsupervised algorithms is that the output always have a high probability of excessive/high false positive and false negative detection (Jeffrey, Tan and Villar, 2024a). As we had a labelled balanced dataset with large amount of datapoints and to avoid the high amount of false positive and false negative detection which for an anomaly detection is a big issue, we decided to use supervised algorithms in our ensemble learning model.

3.2.3 Dataset

The Electra dataset is generated from the network traffic of an electric traction substation used in a real high-speed railway area. The dataset is recorded under normal and under attack operations. The dataset is created in a realistic scenario with industrial devices such as Programmable-Logic Controllers (PLCs) and a SCADA system which are controlled by industrial S7Comm protocol. The main purpose of the testbed is to convert the electric power of the network to voltage, current and frequency conditions to supply it to the railway system. It is also used for converting the three-phased alternating current into a single phase with a lower frequency needed for railway electrification systems (Perception, 2016).

To accomplish all the tasks mentioned above the electric traction substation has 5 PLCs where 1 is the master PLC and the remaining 4 are the slave PLCs, a SCADA system which consists of a Nanobox and an HMI that communications via OPC protocol, a switch for the interconnection of different devices and a firewall to protect the substation from the attacks (Perception, 2016).

Table 2: Features in the Electra S7Comm Dataset (Perception, 2016)

Feature	Description	Data type
time	Timestamp	Integer
smac	Source MAC address	String
dmac	Destination MAC address	String
sip	Source IP address	String
dip	Destination IP address	String
request	Indicates whether the packet is a request	Integer
fc	Function code	Integer
error	Indicates whether there has been an error in reading/writing operation	Integer
address	Memory address to perform read/write operation	Integer
data	Data transmitted or received	Integer
label	Label for attacks and normal samples	String

Table 3: Electra S7Comm Dataset

Traffic	Traffic Type	Datapoints	Total Datapoints
Normal Traffic (68.33%)		264464599	387098466
Attack Traffic (31.67%)	MITM_UNALTERED	117050911	
	FALSE_ERROR_RESPONSE_ATTACK	1664107	
	COMMAND_ATTACK	575122	
	RESPONSE_ATTACK	144892	
	WRITE_ATTACK	2235772	
	READ_ATTACK	936306	
	REPLAY_ATTACK	26757	

3.2.4 Dataset Pre-Processing

The first challenge in pre-processing the dataset was the dataset size which was huge where the CSV file of the dataset was of 34.2 GB. Because of the large size of the dataset and due to the laptop having 16 GB of RAM the dataset couldn't load entirely in one data frame, hence we had to create chunks of the dataset and then perform pre-processing on them.

The first step was to under sample/reduce the dataset where the split between the normal traffic and attack traffic is 50-50 which will create a balanced dataset. We had to try different iterations of datapoint count where the dataset could load entirely in a single data frame, and we could still have enough RAM capacity left on the laptop to train the ensemble models. So initially first we reduced the total data points from 387 million to 247 million, then 20 million and finally 10 million which worked properly according to our hardware capacity.

Then comes the stage of feature selection where we select features of importance to keep and remove the irrelevant feature/column, which were – 'smac', 'dmac', 'sip' and 'dip' which were removed using the drop() method of the Pandas library as these features had string values and not integer values. Then as the dataset had multiple categorial attack class, under the column – "label", hence we performed label encoding where we mapped the – "NORMAL" value with integer value – "0" and attack traffic values with integer – "1". After that, the original label column was dropped from the data frame using the drop() method. Then the dataset was screened for infinity values which were then replaced with NaN (Not an Integer) values using the replace() and inf() method and then the data points containing the NaN values and missing values were dropped from the data frame.

4 Design Specification

We decided to use supervised algorithms for our anomaly detection model and as our problem is a binary classification hence, we trained 7 majorly used supervised algorithms for binary classification on our Electra S7Comm dataset which has 10 million data points where the split between normal and attack data points is 50%. The dataset is split into two parts where one part is the training data, and the other part is the testing data and the split for this is 70% training data and 30% testing data. This is the most common split use in machine learning hence this split percentage was used. After training and predicting the 7 supervised algorithms, all the algorithms with the best-balanced outputs are selected for ensemble learning.

In ensemble learning, multiple base models (algorithms) are combined to obtain better and improved performance by combining the outputs of the base models. This technique helps in obtaining better predictive outcomes when compared to individual models as ensemble learners are more robust to noise and outliers in the dataset as the errors or weaknesses of one mode are compensated by the others. Ensemble learning method - Voting, Bagging, Boosting and Stacking were used as the ensemble techniques as all these methods utilize and combine the base model predictions in different ways. In Voting ensemble technique, it combines the predictions of multiple models and makes the final prediction on the bases of which class got the majority vote, this type of voting is called Hard Voting. Stacking ensemble technique combines the predictions of the base classifiers and uses these predictions as input for the Meta-Model where Meta Model is another machine learning algorithm, generally Logistic Regression is used as the Meta Model algorithm. Bagging ensemble technique creates multiple subsets of the dataset via bootstrapping technique and then a machine learning algorithm, generally Decision Tree is trained on each of these subsets and then the final prediction is made via majority voting of individual predictions as our problem is a classification problem. In Boosting ensemble technique base classifiers are trained sequentially where every next model focuses on improving the errors made by the previous base classifier and the final prediction is the weighted sum of the individual models prediction where higher weights is given to better performing models (Simplilearn, 2021).

4.1 Machine Learning Code Flow

Figure 1: Machine Learning Code Flow.

In Figure 1. the actual machine code flow is presented, where after the data-preprocessing we go through the Base Classifier phase then we perform the first evaluation of Base Classifier models to find which models performance are better than the others and then those Base classifiers are selected for the Ensemble Technique phase, where the Ensemble Techniques are initiated and trained using the selected Base Classifiers and then at the last stage the evaluation phase is again performed but this time with Ensemble Techniques instead of Base Classifiers.

5 Implementation

5.1 Tools and Language

Python is the most used language when dealing with Machine Learning, as it has various libraries for performing machine learning task with efficiency, use-friendliness and ease. Hence Python language version 3.11.5 was used for this research. Python was ran in Jupyter

Notebook environment version 6.5.4 which is a web based development environment which allows to run code in individual cells allowing for greater data exploration and experimenting with different techniques. Jupyter Notebook was run via Anaconda Navigator version 2.5.0. For data pre-processing and machine learning python libraries pandas and numpy and libraries under the sklearn package were utilized.

5.2 Implementation Phases

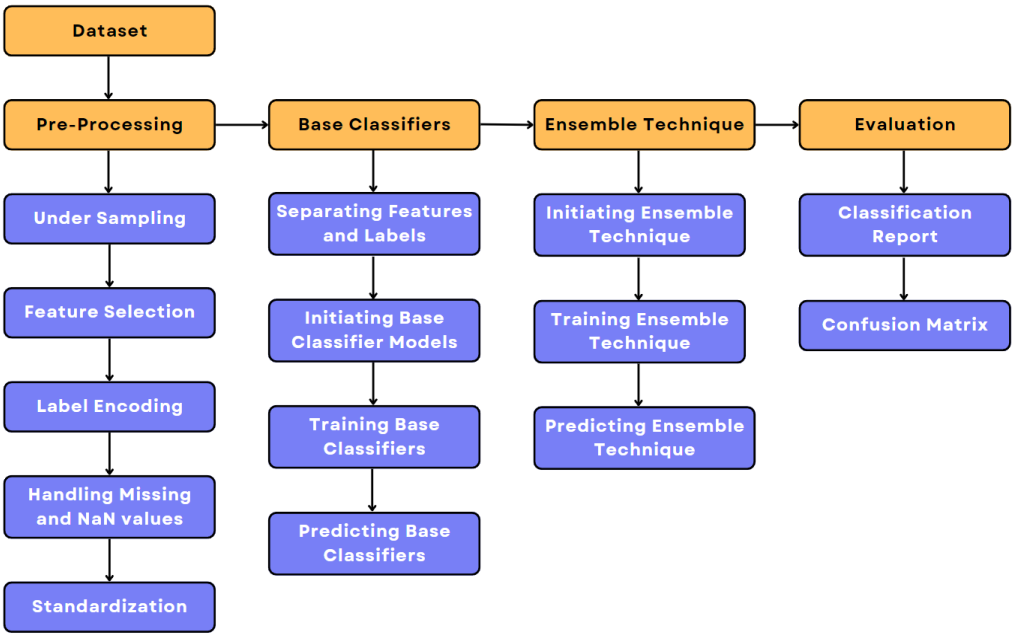

Figure 2: Different Implementation Stages.

Figure 2. presents the different phases of implementation from Data Pre-Processing to final Evaluation. Below we have described each phase in detail.

5.2.1 Data Pre-Processing

First step performed in pre-processing phase is the under sampling of the dataset where the datapoints of the dataset are reduced from 387 million to 10 million, this is done so that the dataset can be loaded directly in a data frame to perform further pre-processing. Another thing carried out during under sampling is that the Normal and Attack traffic are under sampled in such a way that the proportion of both the classes data points are equal, resulting in a balanced dataset.

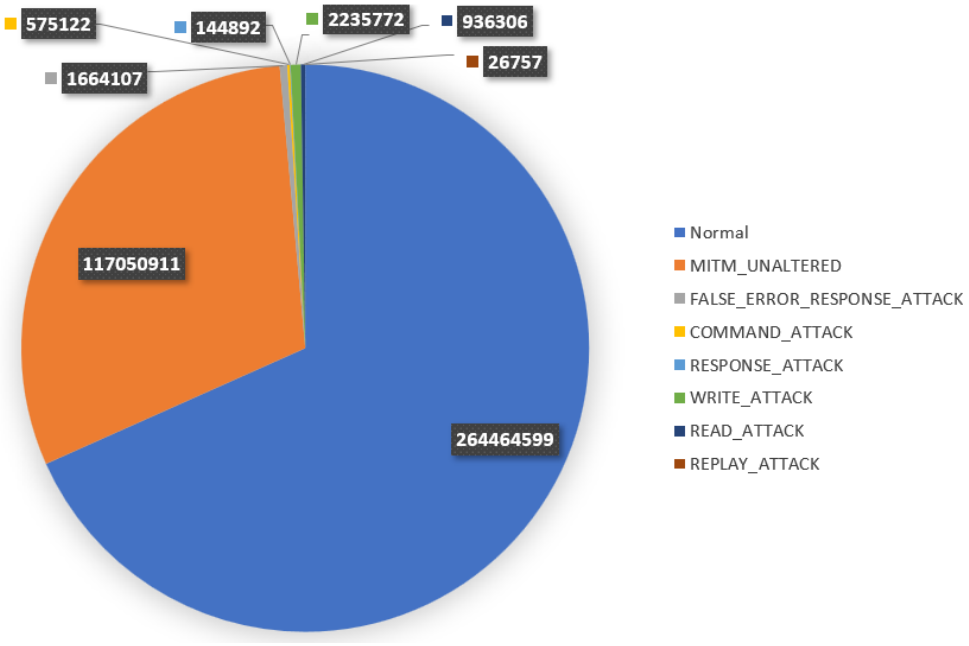

Figure 3: Datapoint Distribution before Under Sampling.

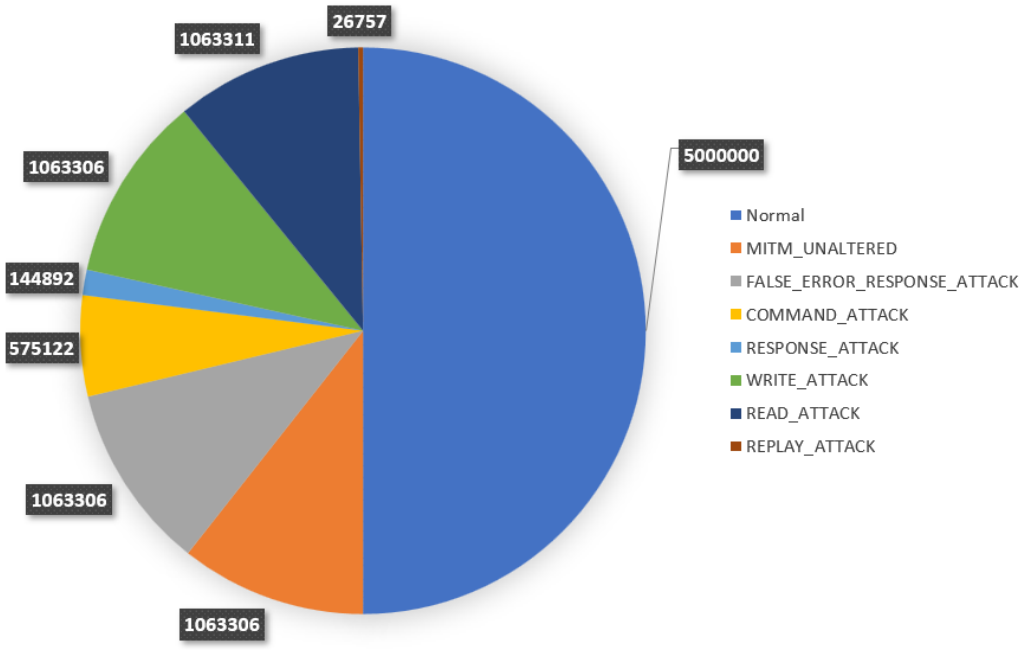

Figure 4: Datapoint Distribution after Under Sampling.

Figure 3. presents the data points distribution from the original dataset, where the data is imbalanced as the Normal class has the majority of datapoints. After under sampling the dataset, it becomes balanced as displayed in Figure 4. where the Normal class has half a million data points and the different types of attacks combined are half a million.

Second step is to perform feature selection, where during data exploration we reviewed the features and their values. Where we kept all the features containing integer values and removed feature with string values.

Then we performed label encoding as the labels in our dataset are string values, we have to convert them to integer values as machine learning algorithms that we have selected only work with integer values, hence we mapped the value – 0 to NORMAL and 1 to ATTACK traffic.

After label encoding, we screen the dataset for any missing value or NaN (Not a Number) values and remove the data points with missing or Nan values.

Finally, we perform Standardization on the dataset features where standardization transforms the feature values of the dataset to have a mean of zero and a standard deviation of one. Standardization helps in bringing the feature values on a similar scale. It helps in improving machine learning models performance.

5.2.2 Base Classifiers

First step in the Base Classifier phase is to separate the dataset into training subset and testing subset. We have used the most common split percentage that is used which is dividing the dataset into 70% Training data and 30% testing data. Where the machine learning models are trained on the 70% training subset and for verifying the model's performance the remaining 30% test subset is used for model prediction.

Then we initiate all the base classifier models, train them on the training subset and then perform prediction using the test subset.

5.2.3 Ensemble Technique

In the Ensemble Technique phase, we initiate the ensemble techniques, train them on the training subset and then make predictions using test subsets.

5.2.4 Evaluation Phase

In the Evaluation Phase, we initiate the classification report and extract the metrics – Precision, Recall and F1-Score of the positive class that is the attack class, and calculate accuracy. Then we initiate the confusion matrix and extract True Positive, True Negative, False Positive and False Negative values and from these values metrics like False Positive Rate (FPR) and False Negative Rate (FNR) are calculated. All the above is done for all the base classifiers and the ensemble techniques.

6 Evaluation

For evaluating the performance of the ensemble learning models and the base models the selected metrics are the Accuracy, Precision, Recall, F-1 Score, False Positive Rate (FPR) and False Negative Rate (FNR).

True Positive (TP) states the number of records that were correctly predicted as positive by the algorithm. False Positive (FP) states the number of records that were incorrectly predicted as positive by the algorithm. True Negative (TN) states the number of records correctly predicted as negative by the algorithm. False Negative (FN) states the number of records incorrectly predicted as negative by the algorithm.

The Accuracy metric is defined as the percentage of correctly predicted instances out of total instances. i.e., Accuracy = TP + TN / TP + TN + FP + FN (Shah, 2023).

The Precision metric is the percentage of True Positive predictions out of all the positive predictions made by the model. i.e., Precision = TP / TP + FP (Shah, 2023).

The Recall metric also known as True Positive Rate (TPR), is the percentage of True Positive predictions out of all the actual positive instances in the dataset. i.e., Recall = TP / TP + FN (Shah, 2023).

The F1-score is the mean of precision and recall. i.e., 2 x (Precision x Recall / Precision + Recall) (Shah, 2023).

The False Positive Rate (FPR) is the percentage that an actual negative instance is incorrectly classified as a Positive instance. i.e., FPR = FP / FP + TN (BMJ, 2023).
The False Negative Rate (FNR) is the percentage that a positive instance is classified as a Negative instance. i.e., FNR = FN / TP + FN (BMJ, 2023).

Table 4: Base Classifiers Output.

Algorithms	Accuracy	Precision	Recall	F1-Score	FPR	FNR
Random Forest	**0.997441**	0.997555	**0.997329**	**0.997442**	0.002448	**0.002671**
Logistic Regression	0.871415	**1.000000**	0.743000	0.852553	**0.000000**	0.257000
XGBOOST	**0.998326**	0.997094	**0.999568**	**0.998330**	0.002917	**0.000432**
KNN	**0.998944**	**0.998532**	**0.999360**	**0.998945**	**0.001472**	**0.000640**
MLP	0.994953	0.995980	0.993925	0.994951	0.004017	0.006075
DT	**0.999791**	**0.999843**	**0.999740**	**0.999791**	**0.000157**	**0.000260**
NB	0.875626	**1.000000**	0.751417	0.858067	**0.000000**	0.248583

Table 5: Ensemble Models Output.

Algorithms	Accuracy	Precision	Recall	F1-Score	FPR	FNR
Voting	0.999041	0.998766	0.999318	0.999042	0.001236	0.000682
Stacking	0.999838	0.999829	**0.999847**	0.999838	0.000171	**0.000153**
Bagging	**0.999840**	**0.999863**	0.999816	**0.999840**	**0.000137**	0.000184
Boosting	0.999735	0.999704	0.999766	0.999735	0.000297	0.000234

6.1 Discussion

6.1.1 Output Discussion

First, we trained and predicted 7 supervised algorithms on our Electra S7Comm dataset where the output of these models is presented in Table 4. Out of these 7 supervised algorithms, Random Forest, XGBOOST, KNN and Decision Trees outperformed the other algorithms and hence these 4 algorithms were selected as base classifiers for the ensemble learning model.

For ensemble learning, we selected Voting, Stacking, Bagging and Boosting as each of these ensemble techniques works differently as explained in Section 4.1. Table 5. displays the output of the ensemble technique where Stacking and Bagging ensemble techniques performed better than Voting and Boosting. Stacking had the lowest False Negative Rate

(FNR) but Bagging performed better in other metrics as well as maintaining a low False Negative Rate (FNR) and False Positive Rate (FPR), hence Bagging ensemble technique is selected as the Ensemble Learning Technique for Anomaly Detection System.

For developing an effective anomaly detection, the False Positive Rate (FPR) and especially the False Negative Rate (FNR) should be as low as possible. The FPR rate equates to the model categorizing normal traffic as attack traffic which in real-world scenarios would trigger an alert to the security team and when the FPR rate is high the security teams are bombarded with false positive alerts and due to this overburden of alerts the actual critical attack alerts are not prioritised or missed. Whereas if the FNR rate is high the actual attack traffic is categorized as normal traffic and no alerts are triggered to the security team, hence the threat goes undetected which is a big security risk. Hence from the results of Ensemble Learning Techniques, we selected the Bagging technique which has the **lowest FPR rate (0.000137%)** and **low FNR rate (0.000184).** This approach represented in the paper can be used to deploy an anomaly detection system in OT/ICS environment using an ensemble learning method where the ensemble technique used is bagging technique, where the anomaly detection system can help in detecting anomalies leading to early detection of threats.

6.1.2 Critical Analysis of the Experiment

The main issue working with machine learning is that the output or the performance of the model is highly dependent on the quality of the dataset. Hence the research method carried out in this paper for developing an Anomaly Detection System, considered what qualities make a valid and good dataset and applied all of those to the dataset that was used in the research, like making the dataset balanced, reducing the variance of the dataset via performing standardization, label encoding, feature selection, selecting a large enough dataset.

In the initial phase, we selected the Haiend-23.05 dataset (ICS (Industrial Control System) Security Dataset, 2022) which yielded us high output. However, we were sceptical about these high outputs and hence just to be sure that the high output was not something we obtained due to some issue with the dataset, hence we went ahead and reviewed and tried different datasets where we got similar results. The output of these trial-and-error datasets can be viewed in the Appendix section.

As we wanted to keep the False Positive Rate (FPR) and False Negative Rate (FNR) low, we selected the use of supervised machine learning algorithms for the base classifiers of the ensemble learning model. Our target class had two classes which are Normal traffic and Attack traffic, which made it a binary problem and hence we considered supervised algorithms that perform well for binary classification problems. And the best performers from these algorithms were selected for Ensemble Learning. For Ensemble Learning techniques, we selected Voting, Bagging, Boosting and Stacking, where the reason for the choice was that these techniques work in different way from each other and hence we could find out which method of combining the base classifier gave us the best results. We carried out all of these best practices to make sure we don’t yield high result due to low quality dataset or due to weak-performing algorithms.

One improvement could be to obtain a dataset with a few more features, which can solidify that the results obtained from the implemented model in this research paper are correct and not because of dataset quality issue.

6.1.3 Experiment Contribution

From an academic perspective, findings from this research highlight the effectiveness of using the bagging ensemble technique in an anomaly detection system. The findings from these research papers can be used as a benchmark where other researcher can compare their results against the approach followed in this paper. The findings of this research paper may inspire other researchers to further improve the results from the paper or research about the bagging ensemble technique.

From the practitioner's perspective, the method presented in the research paper can be implemented as an anomaly detection system in real-world OT/ICS environments where due to the model's low False Positive Rate(FPR) fewer alarms will be triggered to the security teams resulting in the teams to focus on actual threats and due to the low False Negative Rate(FNR) fewer anomalies/threats would be missed resulting in more effective detection and timely response to actual threats which will prevent operational disruptions in the OT/ICS environments where a little downtime can also cause a huge damage and financial loss.

7 Conclusion and Future Work

The aim of this research was to develop an Anomaly Detection System using Ensemble Learning Model for OT/ICS environments, where the model achieves high accuracy while maintaining low False Positive Rates (FPR) and False Negative Rates (FNR). Results demonstrate that the model presented in this research paper achieved high accuracy of **0.999840%**. The experiment was successful where the Ensemble Technique outperformed other ensemble techniques where it yielded the highest accuracy along with maintain low FPR and FNR score which are **0.000137%** and **0.000184** respectively. The limitation of this study was the lack of quality OT/ICS datasets where the datasets had large amount of data points and features along with the dataset being labelled.

This research can potentially improve the state of OT/ICS security in detection of anomalies/threats in their environments resulting in prevention of cyberthreats leading to no operational disruption for these critical infrastructures. This research can be improved by using OT/ICS datasets with more features to make the results of the Ensemble Learning Model more convincing.

References

Araya, D.B., Grolinger, K., ElYamany, H.F., Capretz, M.A.M. and Bitsuamlak, G. (2017). An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, [online] 144, pp.191–206. doi:https://doi.org/10.1016/j.enbuild.2017.02.058.

BMJ (2023). *Definitions and formulae for calculating measures of test accuracy*. [online] Available at: https://www.bmj.com/content/bmj/suppl/2020/05/12/bmj.m1808.DC1/watj056527.ww1.pdf.

Canadian Institute for Cybersecurity (2023). *Modbus 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*. [online] www.unb.ca. Available at: https://www.unb.ca/cic/datasets/modbus-2023.html [Accessed 11 Aug. 2024].

Choi, W.-H. and Kim, J. (2024). Unsupervised Learning Approach for Anomaly Detection in Industrial Control Systems. *Applied System Innovation*, [online] 7(2), p.18. doi:https://doi.org/10.3390/asi7020018.

Cybersecurity & Infrastructure Security Agency (2021). *Cyber-Attack Against Ukrainian Critical Infrastructure*. [online] Cybersecurity and Infrastructure Security Agency CISA. Available at: https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01 [Accessed 11 Aug. 2024].

Dehlaghi-Ghadim, A., Moghadam, M.H., Balador, A. and Hansson, H. (2023). Anomaly Detection Dataset for Industrial Control Systems. doi:https://doi.org/10.48550/arxiv.2305.09678.

Eldridge, L. (2019). *Powersmiths | Transformers, PDUs, Submeters, Resource Management*. [online] Powersmiths | Power for the Future. Available at: https://ww2.powersmiths.com/index.php?q=content/powesmiths/about-us [Accessed 11 Aug. 2024].

Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L. and Janicke, H. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access*, pp.1–1. doi:https://doi.org/10.1109/access.2022.3165809.

ICS (Industrial Control System) Security Dataset (2022). *HAI (HIL-based Augmented ICS) Security Dataset*. [online] GitHub. Available at: https://github.com/icsdataset/hai [Accessed 11 Aug. 2024].

Jeffrey, N., Tan, Q. and Villar, J.R. (2024a). A hybrid methodology for anomaly detection in Cyber–Physical Systems. *Neurocomputing*, 568, pp.127068–127068. doi:https://doi.org/10.1016/j.neucom.2023.127068.

Jeffrey, N., Tan, Q. and Villar, J.R. (2024b). Using Ensemble Learning for Anomaly Detection in Cyber–Physical Systems. *Electronics*, [online] 13(7), p.1391. doi:https://doi.org/10.3390/electronics13071391.

Kushner, D. (2013). *The Real Story of Stuxnet*. [online] IEEE Spectrum. Available at: https://spectrum.ieee.org/the-real-story-of-stuxnet [Accessed 11 Aug. 2024].

Kwasi Boakye-Boateng, Ghorbani, A.A. and Arash Habibi Lashkari (2023). Securing Substations with Trust, Risk Posture, and Multi-Agent Systems: A Comprehensive Approach. doi:https://doi.org/10.1109/pst58708.2023.10320154.

OTORIO (2023). *OTORIO - OT security insights survey*. [online] Otorio.com. Available at: https://go.otorio.com/ot-security-survey-04-23 [Accessed 11 Aug. 2024].

Parsons, D. (2023). *SANS ICS/OT Cybersecurity Survey: 2023's Challenges and Tomorrow's Defenses*. [online] www.sans.org. Available at: https://www.securityweek.com/wp-content/uploads/2023/09/sans-OT-survey.pdf [Accessed 11 Aug. 2024].

Perception (2016). *Electra dataset: Anomaly detection ICS dataset*. [online] Inf.um.es. Available at: http://perception.inf.um.es/ICS-datasets/ [Accessed 11 Aug. 2024].

Pinto, C., Sajjad Dadkhah, Ferreira, R., Alireza Zohourian, Lu, R. and Ghorbani, A.A. (2023). CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. 23(13), pp.5941–5941. doi:https://doi.org/10.3390/s23135941.

S, P.S., Selvan, E. and Ramkumar, M.P. (2023). Anomaly-based Intrusion Detection System for ICS. doi:https://doi.org/10.1109/icccnt56998.2023.10307058.

Shah, D. (2023). *Top Performance Metrics in Machine Learning: A Comprehensive Guide*. [online] www.v7labs.com. Available at: https://www.v7labs.com/blog/performance-metrics-in-machine-learning [Accessed 11 Aug. 2024].

Simplilearn (2021). *What Is Ensemble Learning? Understanding Machine Learning Techniques*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/ensemble-learning-article [Accessed 11 Aug. 2024].

Storm, J.-M., Siv Hilde Houmb, Pallavi Kaliyar, Laszlo Erdodi and Janne Merete Hagen (2023). Testing Commercial Intrusion Detection Systems for Industrial Control Systems in a Substation Hardware in the Loop Testlab. *Electronics*, 13(1), pp.60–60. doi:https://doi.org/10.3390/electronics13010060.

Appendix

Output for Haiend-23.05 HIL-based Augmented ICS Security Dataset: -

Base Classifiers Metrics:

	Algorithms	Accuracy	Precision	Recall	F1-Score	FPR	FNR
0	Random Forest	0.999994	1.000000	0.999977	0.999988	0.000000	0.000023
1	Logistic Regression	0.833009	0.706430	0.526896	0.603596	0.069637	0.473104
2	XGBOOST	0.999997	1.000000	0.999988	0.999994	0.000000	0.000012
3	KNN	0.999918	0.999930	0.999731	0.999830	0.000022	0.000269
4	MLP	0.999915	0.999673	0.999977	0.999825	0.000104	0.000023
5	DT	0.999983	0.999977	0.999953	0.999965	0.000007	0.000047
6	NB	0.760976	0.840102	0.011617	0.022918	0.000703	0.988383

Table 6: Haiend-23.05 Base Classifiers Output.

```
Ensemble Classifiers Metrics:
  Algorithms  Accuracy  Precision    Recall  F1-Score       FPR       FNR
0     Voting  0.999994   1.000000  0.999977  0.999988  0.000000  0.000023
1   Stacking  0.999997   1.000000  0.999988  0.999994  0.000000  0.000012
2    Bagging  0.999983   0.999977  0.999953  0.999965  0.000007  0.000047
3   Boosting  0.999980   0.999965  0.999953  0.999959  0.000011  0.000047
```

Table 7: Haiend-23.05 Ensemble Techniques Output.

Output for ICS-Flow Dataset: -

```
Base Classifiers Metrics:
            Algorithms  Accuracy  Precision    Recall  F1-Score       FPR  \
0        Random Forest  0.999194   0.997040  0.997040  0.997040  0.000466
1  Logistic Regression  0.976789   0.976855  0.849615  0.908803  0.003172
2              XGBOOST  0.999355   0.997632  0.997632  0.997632  0.000373
3                  KNN  0.998791   0.997031  0.994079  0.995553  0.000466
4                  MLP  0.999033   0.997626  0.995263  0.996443  0.000373
5                   DT  0.998146   0.994069  0.992303  0.993185  0.000933
6                   NB  0.948823   0.960664  0.650681  0.775856  0.004198

        FNR
0  0.002960
1  0.150385
2  0.002368
3  0.005921
4  0.004737
5  0.007697
6  0.349319
```

Table 8: ICS-Flow Base Classifiers Output.

```
Ensemble Classifiers Metrics:
  Algorithms  Accuracy  Precision    Recall  F1-Score       FPR       FNR
0     Voting  0.999436   0.998814  0.997040  0.997926  0.000187  0.002960
1   Stacking  0.999436   0.998223  0.997632  0.997927  0.000280  0.002368
2    Bagging  0.998711   0.997620  0.992895  0.995252  0.000373  0.007105
3   Boosting  0.998146   0.994069  0.992303  0.993185  0.000933  0.007697
```

Table 9: ICS-Flow Ensemble Techniques Output.