Mini Project Report

"Analysis of Different Types of Attack in Industrial Automation Systems"

Submitted by:

Ankit Kumar (20BCS017)
Aryan Tejiyan (20BCS021)
Bazil Shaikh (20BCS029)
Tanzeem (20BCS132)

Under Guidance of :

Dr. Malay Kumar,
Assistant Professor

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Indian Institute of Information Technology, Dharwad

(08/05/2023)

CERTIFICATE

It is to certify that the work contained in the project report titled "Analysis of Different Types of Attack in Industrial Automation Systems" by Ankit Kumar (20BCS017), Aryan Tejiyan (20BCS021), Bazil Shaikh (20BCS029) and Tanzeem (20BCS132) has been submitted for the fulfillment of the requirement for the degree of "Bachelor of Technology in Computer Science & Engineering". Their work has been found satisfactory and hereby approved for the submission.

Signature of the Supervisor
Dr. Malay Kumar
Assistant Professor
Department of Computer Science and Engineering
IIIT-Dharwad

DECLARATION

We declare that this written project submission represents our ideas and has not been taken from any other source. We have mentioned and cited the resources of any other publisher wherever needed in the report. We also declare that this work is done by following the principles of academic honesty and integrity and we have done this work with utmost sincerity towards our profession and also towards the institute. We moreover understand the consequences of falsifying any information of data and figures, hence we have tried to come up and present the best possible response and results.

Ankit Kumar (20BCS017)
Aryan Tejiyan (20BCS021)
Bazil Shaikh (20BCS029)
Tanzeem (20BCS132)

ACKNOWLEDGEMENT

We would like to express our sincere thanks to Dr. Malay Kumar, Assistant Professor (Dept of Computer Science and Engineering) Dr Sadhvi Manerikar, Seminar Coordinator for their constant support and guidance throughout the project. It is their constant effort which helped us in completion of the project well and on time. We would also like to thank the Department of Computer Science IIIT-Dharwad for the motivation and encouragement, sincere thanks to Dr. Malay for providing us with the guidelines and resources to successfully complete this report.

TABLE OF CONTENTS

1. INTRODUCTION
    - 1.1 Motive
2. OBJECTIVE
3. RELATED WORK
    - 3.1 Recent Attacks
4. DATASET DESCRIPTION
    - 4.1 Gas pipeline system & Water storage tank system datasets
    - 4.2 Description of attack
    - 4.3 Data Set Organization
    - 4.4 Data visualization
        - 4.4.1 Heatmap
        - 4.4.2 Scatter Plot
5. METHODOLOGY
    - 5.1 ML approaches
        - 5.1.1 SVM
        - 5.1.2 Decision tree
        - 5.1.3 Naive Bayes
        - 5.1.4 Ensemble Methods
        - 5.1.5 Random Forest
    - 5.2 Deep Learning approach
        - 5.2.1 Neural Network
        - 5.2.2 Deep Belief Network
6. SIMULATION
    - 6.1 ML approach
    - 6.2 Deep Learning approach
7. CONCLUSION
8. REFERENCE

ABSTRACT

In recent years, we have witnessed a huge growth in the number of Internet of Things (IoT) and edge devices being used in our everyday activities. Supervisory control and data acquisition (SCADA) networks are widely used by industrial automation and control systems (IACS). However, their integration into IACS is vulnerable to various cyber-attacks. This demands the security of these devices from cyber-attacks to be improved to protect its users. For years, Machine Learning (ML) techniques have been used to analyze the various attacks on SCADA systems with the aim of increasing their reliability/ robustness. Among the earlier ML techniques XG boost, Gradient boost, Random Forest ensemble technique and SVM performed well. In recent years, Deep Learning (DL) techniques have been used in an attempt to build more reliable systems. In this project, we aim to analyze the effectiveness of DBNs and NNs in enhancing the security of IACS and SCADA systems. We will compare the accuracy and performance of these advanced techniques with traditional machine learning algorithms using real-world gas-pipeline data. Our goal is to demonstrate that DBNs and NNs can provide more reliable security for IACS and SCADA systems by learning from the large volumes of data generated by these systems.

The results of this project will provide insights into the potential of advanced machine learning techniques in enhancing the security of critical infrastructures. By identifying the most effective machine learning techniques for SCADA systems, we can improve the reliability and resilience of these systems against cyber-attacks. Ultimately, this project will contribute to the development of more secure and robust IACS and SCADA systems, ensuring the safety of workers, the environment, and the public.

Keywords - Neural Network, Deep Belief Network, IACS, IOT, SCADA.

1.INTRODUCTION

The Internet of Things (IoT) refers to the interconnected network of devices, sensors, and machines that can communicate with each other and exchange data over the internet. This technology has a wide range of applications, including in the field of industrial automation and control.

IoT (Internet of Things) devices are increasingly being used to connect various physical objects to the internet, allowing them to be remotely monitored, controlled, and managed. However, as with any internet-connected device, IoT devices can also be vulnerable to cyber-attacks, which can compromise their security and privacy.

To prevent IoT devices from being targeted in cyber-attacks, it is essential to implement strong security measures such as regular firmware updates, strong authentication protocols, and secure network configurations.

Industrial Automation Control Systems (IACS) are increasingly being targeted by cyber attackers seeking to disrupt or damage critical infrastructure. These attacks can have serious consequences, including safety risks, financial losses, and damage to public trust.

To prevent cyber-attacks, organizations should implement a robust cybersecurity strategy that includes regular risk assessments, employee training, network segmentation, access controls, and incident response planning.

SCADA (Supervisory Control and Data Acquisition) is a form of control system used to monitor and control industrial operations in Industrial Automation Control Systems (IACS). In over three decades of their existence, Supervisory Control and Data Acquisition (SCADA) systems have undergone massive changes in their capabilities, structures, functionality and even in general perception and their role in the overall Industrial control system (ICS). SCADA systems are often made up of hardware and software components that work together to deliver real-time data on the status of industrial processes and allow operators to remotely control those operations.

In Industrial IoT (IIoT) networks, there are massive amount of supervisory control and data acquisition (SCADA)-based IACS.

The management and control of industrial processes including power generation, water treatment, and oil refining depend heavily on Industrial Automation and Control Systems (IACS), especially Supervisory Control and Data Acquisition (SCADA) systems [11]. However, these systems are susceptible to cyber-attacks due to their reliance on networked technologies and interconnectedness. These assaults run the risk of endangering people's safety, physically harming machinery, and interfering with business operations. IACS and SCADA systems must therefore build efficient intrusion detection and prevention solutions.

IACS security has long employed traditional machine learning techniques, but these algorithms' efficiency is constrained by the complexity and variability of IACS data [12]. In comparison to conventional machine learning algorithms, more sophisticated techniques like Deep Belief Networks (DBNs) and Neural Networks (NNs) are able to learn complex patterns and correlations within data [13].

In this project, we examine how well DBNs and NNs work to improve the security of IACS and SCADA systems. Using actual SCADA data, we will compare the precision and effectiveness of these sophisticated techniques with those of conventional machine learning algorithms. Our objective is to show that by learning from the massive amounts of data produced by these systems, DBNs and NNs can provide more dependable security for IACS and SCADA systems.

The findings of this project will shed light on the potential of cutting-edge machine learning methods to improve the security of crucial infrastructures. We can increase the dependability and resistance of SCADA systems against cyberattacks by determining the most efficient machine learning methods. In the end, this project will help to design IACS and SCADA systems that are more reliable and secure, assuring the safety of workers, the environment, and the general public.

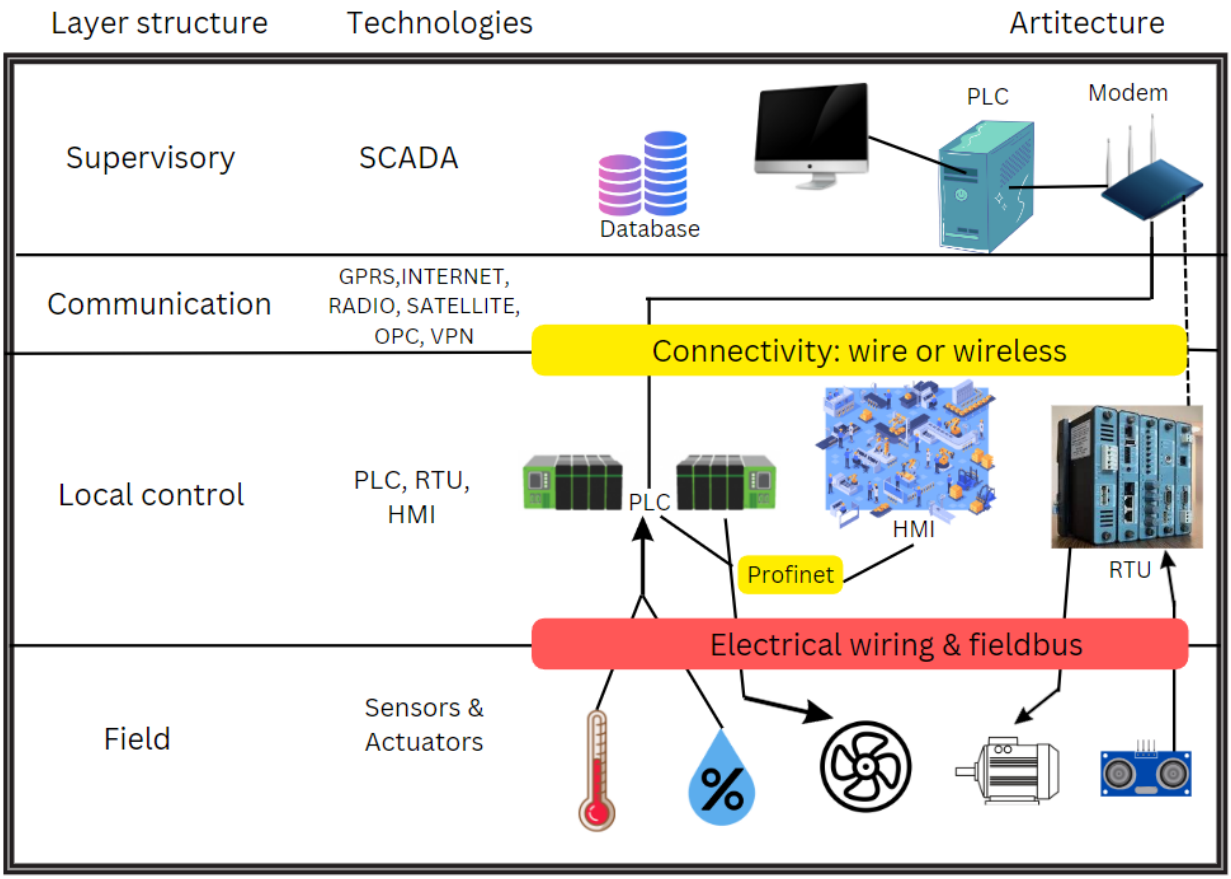


Fig 1: The SCADA architecture.

According to **Figure 1**, remote terminal units (RTUs) or PLCs are connected to field equipment (such as sensors, actuators, pumps, switches, turbines, etc.) in a local network where input data are received and processed. Based on the output of control logic that executes inside PLCs or RTUs, actions are initiated.

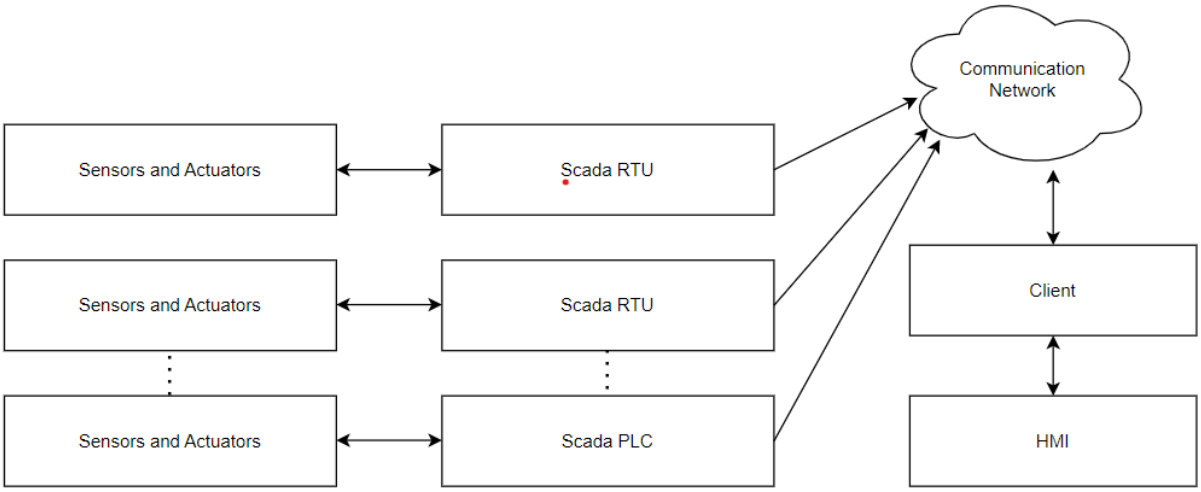


Fig 2 – Block Diagram of Scada System

Figure 2, shows that the physical layer contains programmable logic controllers (PLCs) and remote terminal units (RTUs), which can gather information from the sensors. The cyber layer is used to monitor and control the various devices in local control layer. Also, they include the cyber-attacks detection systems. Corporate layer is an IT system, which can support business processes and push management decisions.

1.1. MOTIVATION

- The analysis of assaults on industrial automation systems is motivated by an understanding of the risks and weaknesses associated with these essential systems. Industrial automation systems are used to regulate and manage a variety of industrial processes, including production of energy and transportation. There could be severe consequences if these systems are successfully attacked, including physical harm, financial losses, and even fatalities.

- Over the past ten years, attacks on industrial automation systems have considerably increased. Industrial control systems (ICS) were the target of 54% more attacks in 2018 than the previous year and 220% more attacks in 2017. According to a Kaspersky Lab investigation, this is the case. In 2020, there were over 2,000 ICS recorded cyberattacks, with the manufacturing and energy sectors as the main targets.

- By analyzing previous assaults on industrial automation systems, we can discover typical attack pathways and develop strategies to thwart and mitigate these attacks. We can also anticipate and be ready for future attacks by understanding the goals and techniques used by attackers.

2. OBJECTIVES

- The objective of developing a neural network model for the analysis of attacks on industrial automation systems is to improve the cybersecurity of these vital systems by identifying and reducing potential risks.
- Identifying the strengths and weaknesses of the implemented deep belief network in detecting different types of attacks.
- By examining data from SCADA and ICS systems using machine learning techniques, a neural network model can aid in the identification of potential assaults and the detection of odd behaviour. This makes it possible to identify potential hazards before they have a chance to cause real damage.
- After been trained on a sizable dataset of typical system behaviour, the neural network model can be used to detect variations from the norm that may be indicators of an attack.
- By analysing system data in real-time, the model may spot potential risks, alerting operators or security personnel so they can take the appropriate security measures to thwart the attack.
- Proposing improvements to the deep belief network to enhance its performance in detecting different types of attacks.
- Assessing the efficiency of deep learning in identifying attacks on industrial automation systems.

3. RELATED WORK

Today, cyber-security issues are receiving a lot of attention from both the academic and professional communities. Large-scale traffic information prevents typical attack detection systems from successfully addressing the needs of security offence and defence. Numerous machine learning approaches have significantly improved the field of detecting cyberattacks during the last few decades.

Due to the rising Internet traffic and massive amounts of data, smaller machine learning algorithms are unable to manage these complicated security challenges. As a result, because neural networks may be used to investigate and benefit from complex correlations in collected datasets, building neural network-based algorithms for detecting cyberattacks becomes a hot topic.

A water treatment plant neural network-based control system was created by Huda et al. in "Neural network-based control of a water treatment plant" [4]. The system was able to keep the water quality within allowable bounds while using the fewest chemicals and resources possible.

Liu et al.'s "An integrated neural network-based approach for fault diagnosis and fault-tolerant control of an industrial process" [5] suggested such an integrated neural network-based method. The process was kept within safe operating limits thanks to the system's ability to detect and diagnose defects in real-time and adopt fault-tolerant control techniques.

Author	Algorithm Used	Findings
Shahriari, M. et al	Multi-layer feedforward neural network	Effectively detect and classify different types of faults that may occur in an industrial robot.
Tao, Y. et al	Neural network-based soft sensor	robust and can handle variations in wastewater composition and operating conditions.
M. Masud et al	1. feature extraction module(CNN & LSTM) 2. classification module(MLP).	Deep learning-based IDSs have shown promise for detecting cyber attacks in SCADA networks.
M. Rashid et al	decision trees, support vector machines, artificial neural networks, deep learning models, and ensemble methods	Hybrid approaches that combine signature-based and anomaly-based methods can improve the detection accuracy and reduce the false positive rate.
K. -D. Lu et al	PEO-DBN, ENPEO-DBN	The approach is robust to noisy data and can handle missing data effectively.
Liu, W. et al	Neural network-based fault tolerant control algorithm	The proposed neural network-based approach can effectively diagnose faults and implement fault-tolerant control for an industrial process.
Shahriari, M. et al	Neural network-based Control algorithm	The control algorithm based on the neural network model can improve the performance of the dryer in terms of energy consumption and drying time.
D. Dzung et al	Decision trees, support vector machines, and neural networks(CNN & RNN)	Highlight the importance of data preprocessing and feature selection for effective machine learning in ICS security.

3.1. Some Recent Attacks:

SCADA (Supervisory Control and Data Acquisition) and ICS (Industrial Control Systems) are computer systems used to monitor and control industrial processes. Unfortunately, these systems are often vulnerable to cyber-attacks, which can have serious consequences for industrial operations:

- **The Ukrainian power grid attack:** The Ukrainian power grid attack, which occurred on December 23, 2015, is considered one of the most significant cyberattacks on critical infrastructure to date. The attack resulted in a widespread power outage that affected over 225,000 people in the western region of Ukraine. They used a sophisticated malware called BlackEnergy (Fig b.1) to infiltrate the systems of three different electricity distribution companies.

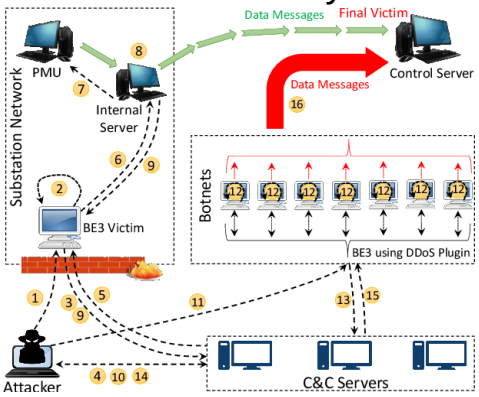


Fig b.1. View of working of Black Energy Malware

- **Saudi Arabian petrochemical plant:** The attack on a Saudi Arabian petrochemical plant in 2017, which used a variant of the Shamoon malware to cause significant damage to the plant's computer systems. The attack caused a shutdown of the plant's operations for several weeks, disrupting production and resulting in significant financial losses.

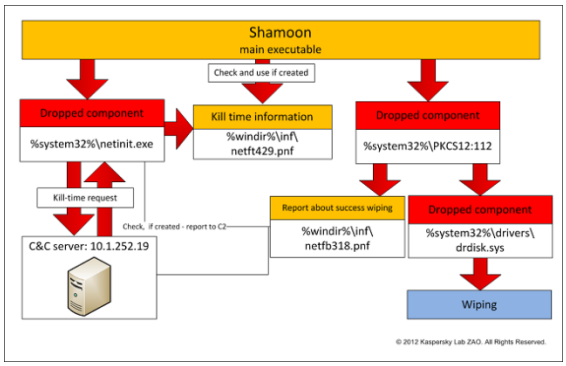


Fig b.2. View of working of Shamoon Malware

- **Stuxnet:** In 2010, a highly advanced computer worm specifically targeted Iran's nuclear programme. Stuxnet particularly targeted the centrifuges utilised in Iran's uranium enrichment facilities, and it is commonly believed that it was created jointly by the United States and Israel. To physically harm the centrifuges, Stuxnet used zero-day flaws in Windows and Siemens PLCs (Programmable Logic Controllers).

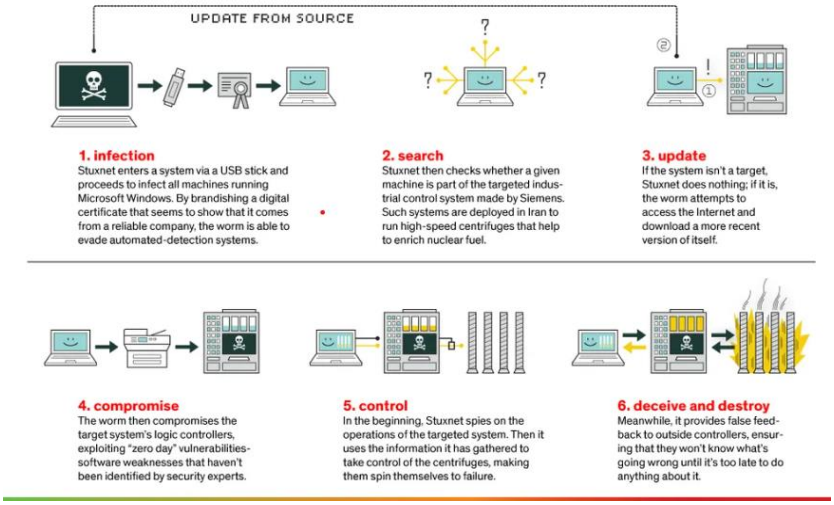


Fig b.3. View of working of Stuxnet

4. DATASET DESCRIPTION

4.1 Gas pipeline system & Water storage tank system data set

The dataset that is used in this project is the Gas pipeline dataset & Water storage tank system that we get from the google dataset website.The data sets described in this project were captured using a network data logger, which monitored and stored MODBUS traffic from a RS-232 connection. Two laboratory-scale SCADA systems were used: a gas pipeline and water storage tank. It is a real SCADA network data of gas pipeline system & Water storage tank system. Before implementing the proposed methods to detect the cyber-attacks, the data preprocessing has been done. The detailed data preprocessing is given as follows. To reduce the influence on the different data dimensions on experimental results, the data is normalized by using StandardScaler. The data sets presented in this project include network traffic, process control and process measurement features from normal operations and attacks against the SCADA system. The attacks are grouped into four classes: (i) reconnaissance; (ii) response injection; (iii) command injection; and (iv) denial-of-service (DoS).

4.2 Detailed description of these attack types is given as follows.

1. **NMRI attack:** Through the previously known information of network servers and devices, the attackers can inject random invalid information to the packet. But they cannot achieve the information of the underlying process being monitored and controlled.
2. **CMRI attack:** Here, the attacks have full information of SCADA network and devices, so that they can mask the actual state of the physical process and cause bad influence of the feedback control process.
3. **MSCI attack:** Through the actuators operation, the state of the physical system (e.g., ON/OFF) can be controlled. MSCI attack can change the state of the register, which may cause the operation of actuators incorrectly.
4. **MPCI attack:** In this type of attack, the attack can alter the set points parameters in field devices, e.g., PLC. After changing these parameters (e.g., PID controller's parameters), the controller performance will be influenced or even does not work at all.
5. **MFCI attack:** If the attacks have the knowledge of the built-in protocol functions provided by the manufacturers for diagnostic purposes, they may abuse the functions.

6. **DoS attack:** DoS attack targets communications links and system programs to stop part of SCADA network. Sometimes, the attackers can change a packet and send it to the filed devices, which may result in crash for the operating system.
7. **Recon attack:** This attack collects SCADA system information, maps the network structure, as well as identifies device characteristics.

4.3 Data Set Organization

Two categories of features are present in the data sets: network traffic features and payload content features. Network traffic features describe the communications patterns in SCADA systems. Payload content features describe the current state of the SCADA system; they are useful for detecting attacks that cause devices (e.g., PLCs) to behave abnormally.
Network traffic features include the device address, function code, length of packet, packet error checking information and time intervals between packets. Payload content features include sensor measurements, supervisory control inputs and distributed control states.

Network Traffic Features-

Attributes	Description
command_address	Device ID in command packet
response_address	Device ID in response packet
command_memory	Memory start position in command packet
response_memory	Memory start position in response packet
command_memory_count	Number of memory bytes for R/W command
response_memory_count	Number of memory bytes for R/W response
command_length	Total length of command packet
response_length	Total length of response packet
time	Time interval between two packets
crc_rate	CRC error rate

List of payload attributes-

Feature Name	Description
comm fun	Value of command function code
response fun	Value of response function code
sub function	Value of sub-function code in the command/response
measurement	Pipeline pressure or water level
control mode	Automatic, manual or shutdown
pump state	Compressor/pump state
manual pump setting	Manual mode compressor/pump setting
label	Manual classification of the instance

It has one column named result that consists of 8 classes-

Label Name	Label Value	Label Description
Normal	0	Instance is not part of an attack
NMRI	1	Naive malicious response injection attack
CMRI	2	Complex malicious response injection attack
MSCI	3	Malicious state command injection attack
MPCI	4	Malicious parameter command injection attack
MFCI	5	Malicious function command injection attack
DoS	6	Denial-of-service attack
Reconnaissance	7	Reconnaissance attack

Unique features of Gas pipeline dataset

Feature Name	Description
set point	Target pressure in the gas pipeline
control scheme	Control scheme of the gas pipeline
solenoid state	State of solenoid used to open the gas relief valve
gain	Gain parameter value of the PID controller
reset	Reset parameter value of the PID controller
dead band	Dead band parameter value of the PID controller
rate	Rate parameter value of the PID controller
cycle time	Cycle time parameter value of the PID controller

Unique features of water storage tank system dataset

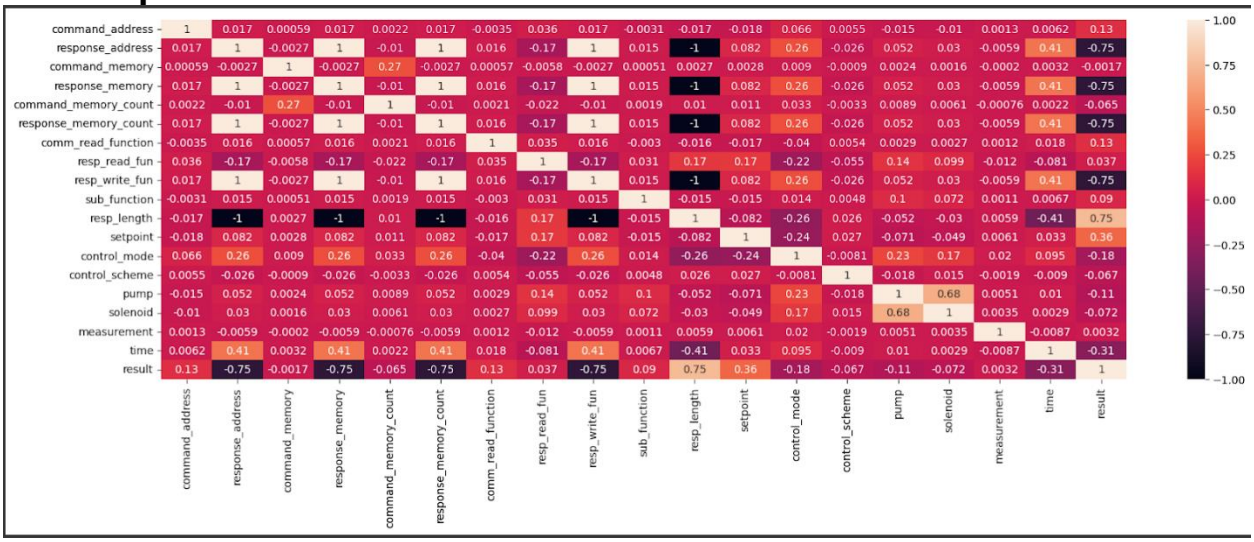
Feature Name	Description
HH	Value of HH setpoint
H	Value of H setpoint
L	Value of L setpoint
LL	Value of LL setpoint

4.4 Data visualization

Data visualization is an important step in the data analysis process that can provide valuable insights and help in making informed decisions.

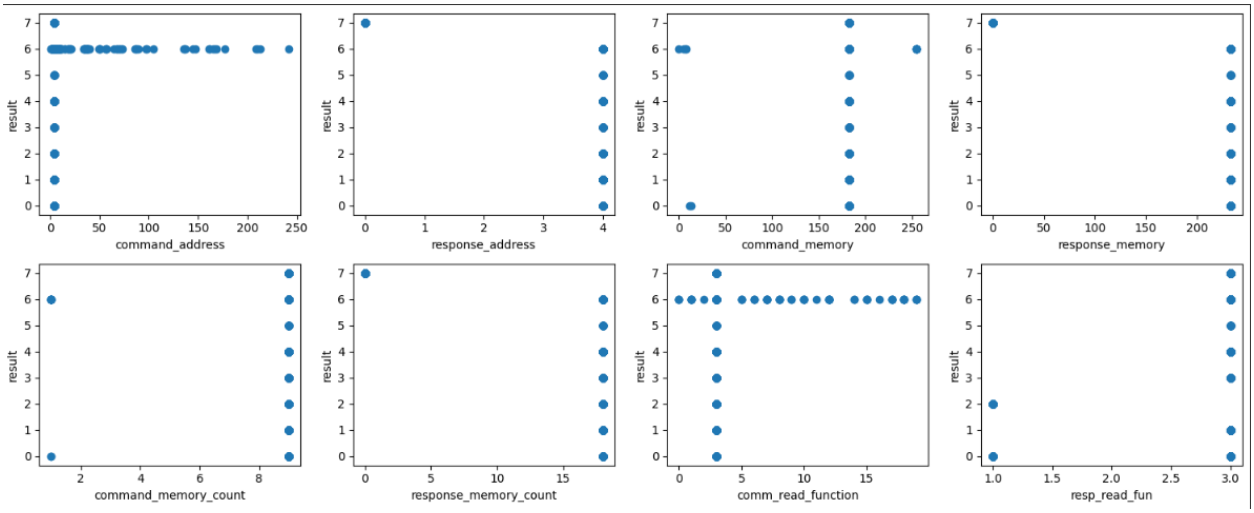
4.4.1 Gas pipeline dataset

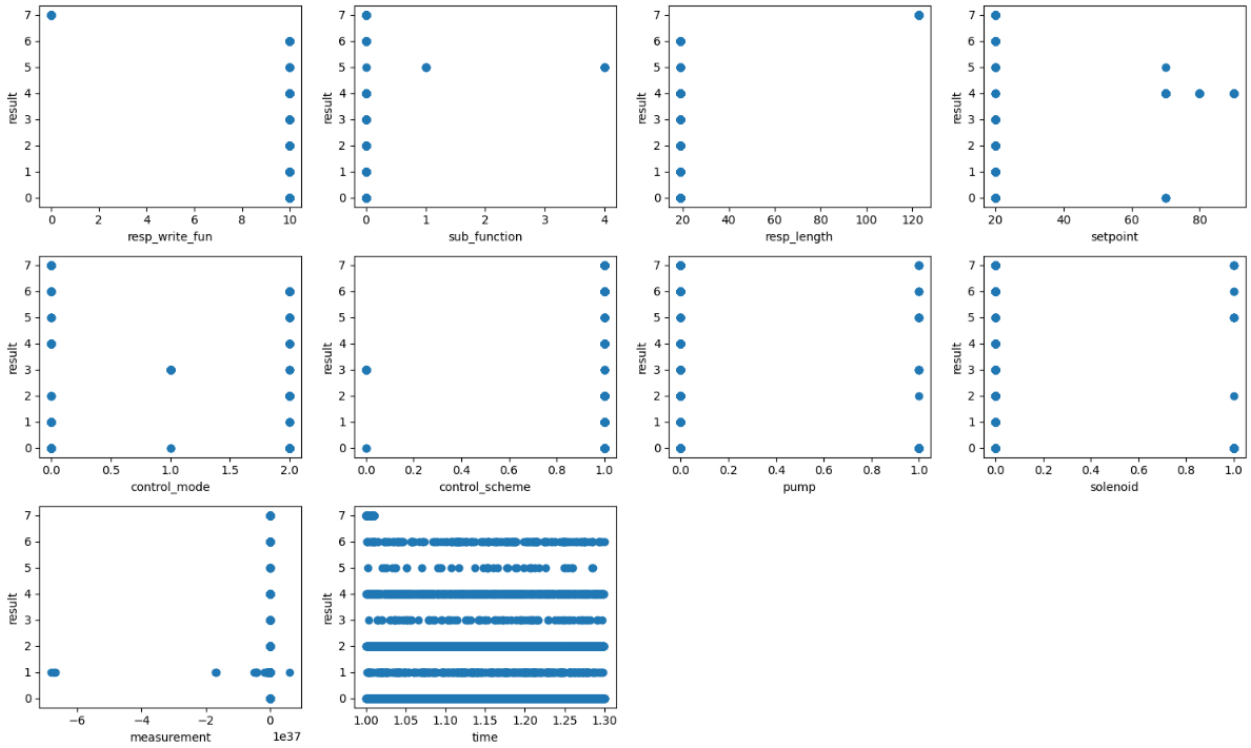
Heatmap



With help of codes we plotted a heatmap using the Seaborn library to visualize the correlation between the columns 'command_address', 'response_address', 'command_memory', 'response_memory', 'command_memory_count', 'response_memory_count', 'comm_read_function', 'resp_read_fun', 'resp_write_fun', 'sub_function', 'resp_length', 'setpoint', 'control_mode', 'control_scheme', 'pump', 'solenoid', 'measurement', 'time', and 'result' from the 'data' DataFrame:

Scatter plot (input features vs target feature ’result’)





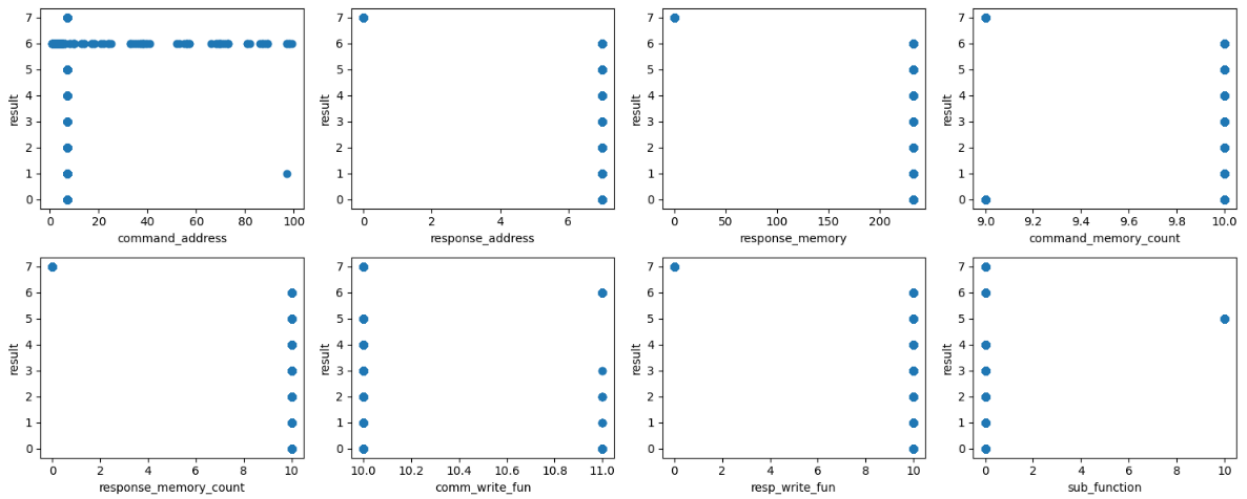
With the help of python code we created a scatter plot for a selected column (result) against all the other columns in a given dataset (gas pipeline system). It first creates a list of column names by extracting the column names from the dataset. It then removes the selected column from this list. It calculates the number of rows and columns required to show all the scatter plots and creates a figure with subplots accordingly. It flattens the 2D axes array into a 1D array and iterates through every column in the list of column names. For each column, it creates a scatter plot between that column and the selected column and sets the x and y labels accordingly. It then hides any unused subplots and displays the final figure. **This code can be useful for quickly visualizing the relationship between a selected column and all the other columns in a dataset.**

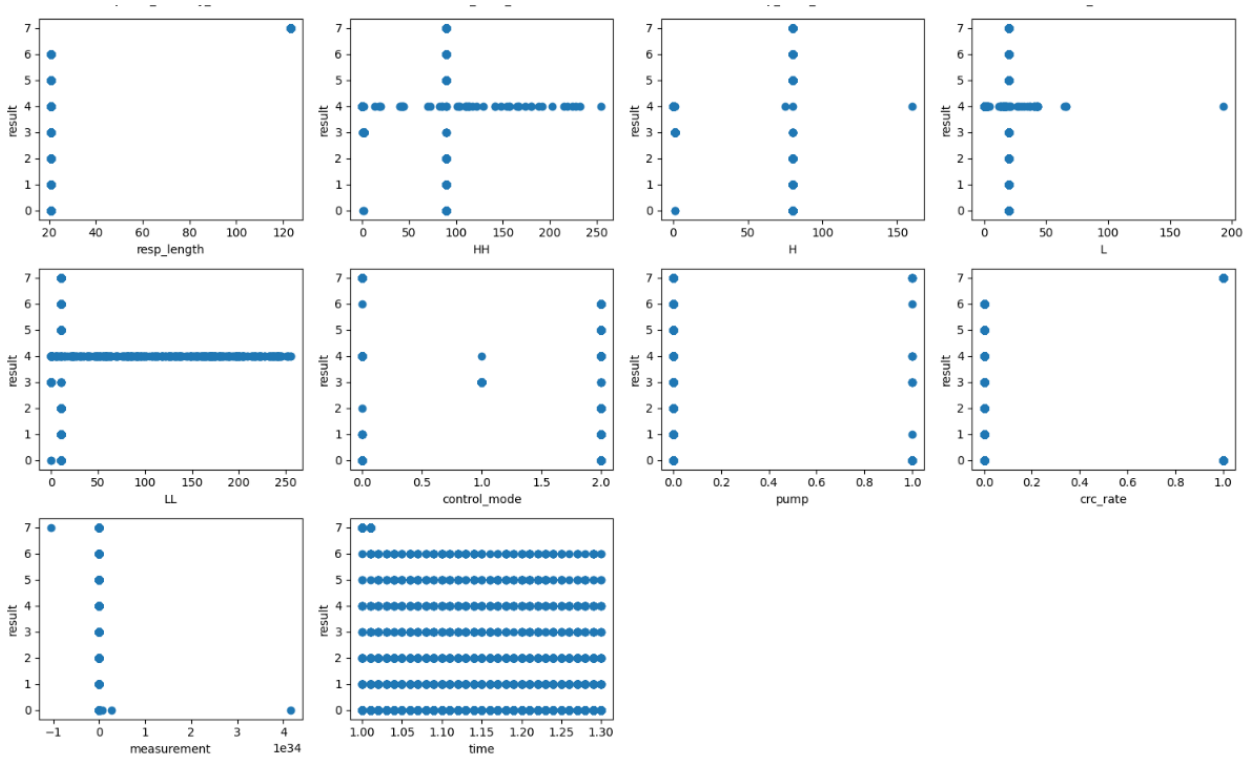
4.4.2 Water storage tank system

Heatmap

The dataset used for this visualization has multiple features, including command_address, response_address, command_memory, response_memory, command_memory_count, response_memory_count, comm_read_function, resp_read_fun, resp_write_fun, sub_function, resp_length, setpoint, control_mode, control_scheme, pump, solenoid, measurement, time, and result. These features are plotted against each other to identify the correlation between them. The corr() function is used to calculate the correlation between features and the resulting values are plotted on the heatmap. The annot=True parameter is used to display the correlation values on the heatmap. This visualization helps to identify which features are positively or negatively correlated with each other, and how strong the correlation is.

Scatter plot (input features vs target feature 'result')





With the help of python code we created a scatter plot for a selected column (result) against all the other columns in a given dataset (water storage tank). It first creates a list of column names by extracting the column names from the dataset. It then removes the selected column from this list. It calculates the number of rows and columns required to show all the scatter plots and creates a figure with subplots accordingly. It flattens the 2D axes array into a 1D array and iterates through every column in the list of column names. For each column, it creates a scatter plot between that column and the selected column and sets the x and y labels accordingly. It then hides any unused subplots and displays the final figure. **This code can be useful for quickly visualizing the relationship between a selected column and all the other columns in a dataset.**

5. METHODOLOGY

5.1 Machine learning approaches

Developing ML algorithms to detect cyber attacks on IAS is a complex process that requires careful consideration of several factors. The methodology outlined in this report can be used as a guide to develop ML algorithms for the analysis of different types of cyber attacks on IAS. By leveraging ML algorithms, it is possible to enhance the security of IAS and prevent cyber attacks.

The following steps was used to develop ML algorithms for the analysis of different types of cyber attacks on IAS:

Data Collection: As we collected data related to cyber attacks on IAS from the google dataset website.

Data Preprocessing: After collecting the data we preprocessed. This involved cleaning the data, removing duplicates, handling missing values, and transforming the data into a format that can be used for ML algorithms.Then we normalized the data using StandardScaler.

Feature Selection: Feature selection is an essential step in developing ML algorithms. It involves selecting the most relevant features that can help in detecting cyber attacks on IAS. The selected features should be able to differentiate between normal and abnormal behavior in IAS.
So in this Dataset we selected the result column as a target feature that has 8 types of attack.Then we deleted 8 columns that had the same value in the entire column.

Algorithm Selection: There are several ML algorithms that we used for the analysis of cyber attacks on IAS.The selection of an appropriate algorithm depends on the type of data, the number of features, and the complexity of the model.After analyzing the dataset we conclude that it is a multi-classification dataset. Here we used algorithms including Decision Trees, Random Forest, Support Vector Machines (SVM), Naive Bayes, Boosting and bagging machine learning techniques.

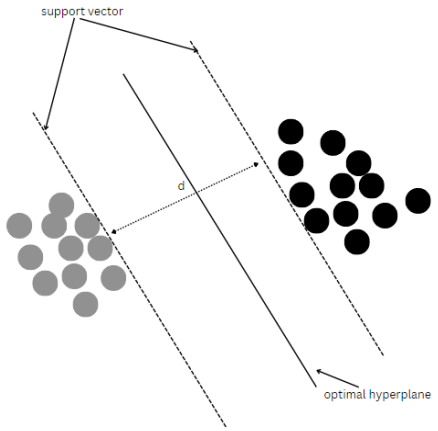
Model Training: After selecting the algorithm, the model needs to be trained on the preprocessed data. The model's performance was evaluated using various metrics such as accuracy, precision, recall, and F1 score.

Model Evaluation: The trained model needs to be evaluated on a separate test set to ensure that it can detect cyber attacks on IAS accurately. If the model is not performing well, then it needs to be fine-tuned. In our project, based on the Accuracy we evaluate the model.

Deployment: Once the model is trained and evaluated, it can be deployed in a real-world environment. The model needs to be continuously monitored and updated to adapt to changing cyber threats.

5.1.1 SVM(Support vector machine)

- It is used for supervised machine learning problem where we try to find a hyperplane that best separates the two classes.
- "Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges.
- However, it is mostly used in classification problems.
- **SVM Math: finding margin(d)**
    - Equation of Hyperplane: L: $W^T X+b=0$
    - Equation of +ve hyperplane as: L1: $W^T X+b=+1$
    - Equation of +ve hyperplane as: L2: $W^T X+b=-1$



5.1.2 Decision tree

- Decision tree builds classification or regression models in the form of a tree structure.
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with decision nodes and leaf nodes.
    - A decision node has two or more branches.

    - Leaf node represents a classification or decision. The top most decision node in a tree which corresponds to the best predictor called root node.
- Decision trees can handle both categorical and numerical data.
- The decision tree algorithm makes decisions by recursively partitioning the input data based on the feature that provides the most information gain. The information gain is calculated using the entropy measure, which is defined as:
  Entropy = $-\sum_i P(i)\log_2 P(i)$

- The information gain for a feature is then calculated as:

  Information gain = Entropy before split - Entropy after split

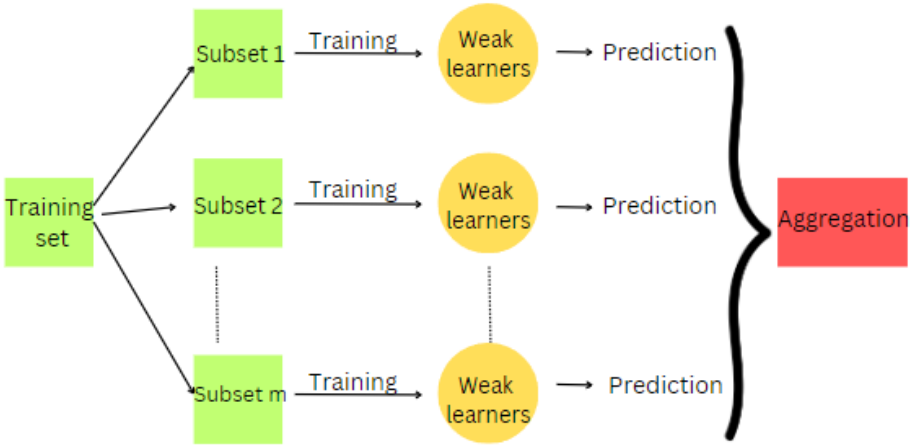
5.1.3 Naive Bayes

- Naive Bayes classification is a probabilistic algorithm that uses Bayes' theorem to calculate the probability of a data point belonging to a particular class. It assumes that the features of the data point are independent of each other, hence the term "naive".
- The Naive Bayes classification algorithm uses the following mathematical formula to predict the class of a new data point:
- **P(class | features) = (P(class) * P(features | class)) / P(features)**

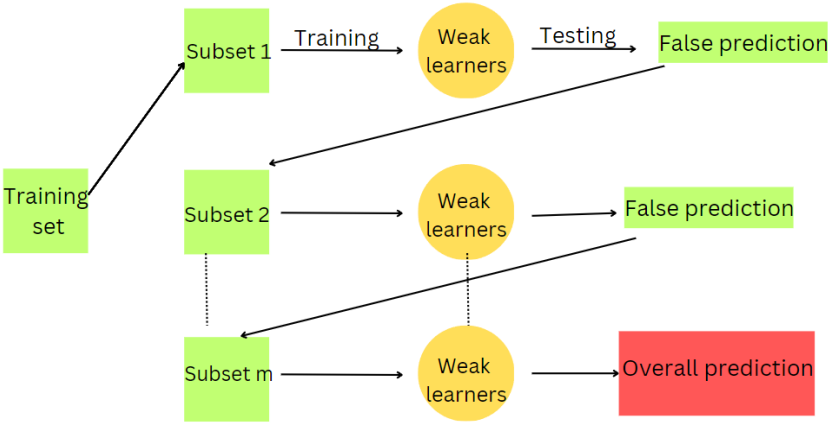
5.1.4 Ensemble Methods: Bagging and Boosting

- The weakness of individual models in machine learning is a concern.
- In other words, their forecast accuracy is typically poor.
- We blend many models to create one with a superior performance to help alleviate this issue. Ensemble learning is the name given to this technique.
- Weak learners are the individual models that we assemble. Because they either have a large bias or a high variance, we refer to them as weak learners.
- The performance of a model is enhanced primarily in three ways:
    - By reducing the variance of weak learners
    - By reducing the bias of weak learners,
    - By improving the overall accuracy of strong learners.

- Approach in Ensemble learning:
  - **Bagging**: used to reduce the variance of weak learners

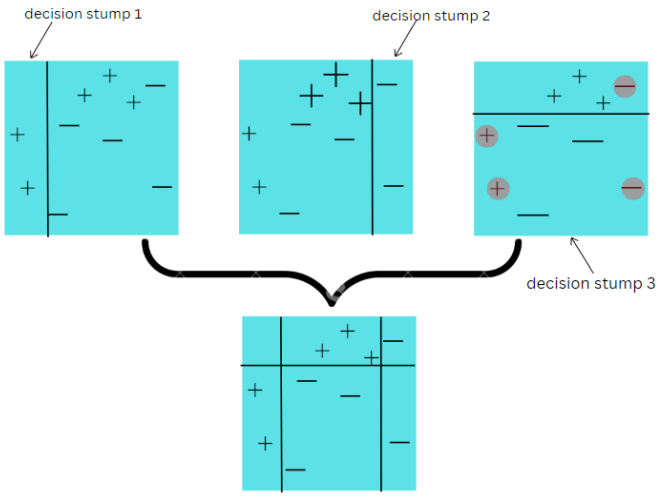


  - **Boosting**: used to reduce the bias of weak learners



5.1.4.1 AdaBoost (Adaptive Boosting)

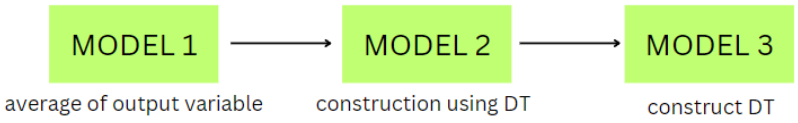
- Adaboost is a stagewise additive method.
- Three important points to understand adaboost algorithm:
  - Weak Learners- A weak learner produces a classifier which is only slightly more accurate than random classification.
  - Decision stumps- A decision tree with a single node operating on one input variable, the output of which makes a prediction directly.
  - +1 and -1
    - For positive class we use +1
    - For negative class we use -1



Classification of misclassified points

5.1.4.2 Gradient Tree Boosting

- It is a boosting algorithm.
- It works in a sequential stage wise addition.



Final PredBoost
= M1 prediction + 0.1*Model 2 prediction + 0.1*Model 3 prediction

5.1.4.3 XGBoost

- **XGboost** is eXtream Gradient Boost.
- XGboost is good in terms of
  - Speed: parallelization, cache optimization, out of memory computation
  - Performance: Regularization, Auto pruning, Missing values control
- It is developed on top of the gradient boost.
- **The important parameters in XGboost are:**
  - **λ**: Regularization parameter helps in pruning and prediction
  - **γ**: gamma helps in pruning
  - **η**: learning rate (0.1-0.3)

5.1.5 Random Forest

- Random Forest is one of the most popular and commonly used algorithms by Data Scientists.
- Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems.
- It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
- Random Forest uses a bagging approach of ensemble learning.

Random Forest: Hyperparameters

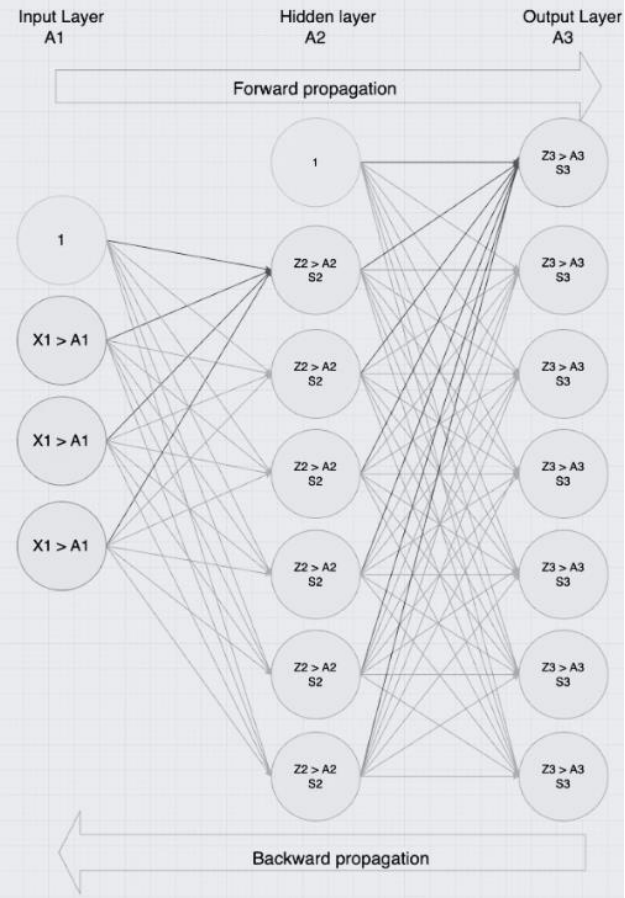
- n_estimators: Number of trees the algorithm builds before averaging the predictions.
- max_features: Maximum number of features random forest considers splitting a node.
- mini_sample_leaf: Determines the minimum number of leaves required to split an internal node.
- criterion: How to split the node in each tree? (Entropy/Gini impurity/Log Loss)
- max_leaf_nodes: Maximum leaf nodes in each tree

5.2 Deep learning approaches

Deep learning (DL) has been increasingly used for the analysis of different types of cyber attacks on industrial automation systems (IAS) due to its ability to learn from data and detect patterns that are difficult to capture with traditional methods. DL approaches can help to detect and classify different types of cyber attacks on IAS, including but not limited to, denial of service attacks, malware attacks, and man-in-the-middle attacks.

5.2.1 Neural Networks

Neural networks (NNs) are a powerful class of machine learning models that have been used for the analysis of different types of attacks on industrial automation systems (IAS). NNs can be used to detect anomalies in IAS data, classify attacks, and predict future attacks based on historical data.



There are six key ideas in a neural network-

Weights: Weight is the parameter within a neural network that transforms input data within the network's hidden layers.
Layers: Our network will have three tiers.
Forward propagation: Obtain Z and A using the features and weights
Back propagation: Using the results of forward propagation and weights, do back propagation to obtain S.
Cost: calculating each weight's cost or gradient
C(y, o) = $\frac{1}{N}\sum_{i=0}^{n}.(y_i\text{-}o_i)^2$
Gradient descent: Find the appropriate weight or hypothesis for the gradient decline

Neural networks vs. deep learning-
It can be misleading because the terms "deep learning" and "neural networks" are frequently used interchangeably in speech. It's important to remember that the "deep" in deep learning just denotes the number of layers

in a neural network. A neural network with more than three layers, including the inputs and outputs, is referred to as a "deep learning algorithm." Simply put, a simple neural network is one that just contains two or three layers.

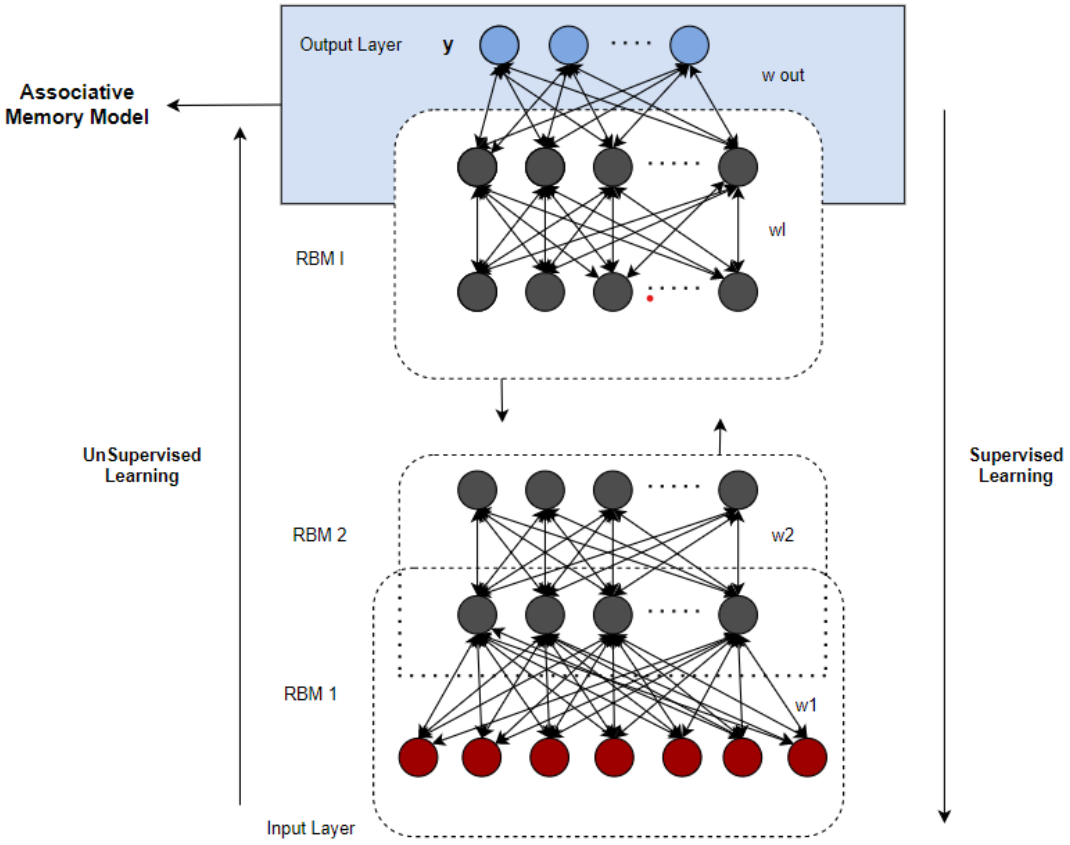
5.2.2 Deep Belief Networks

Deep Belief Networks (DBNs) are a type of deep learning model that consists of multiple layers of Restricted Boltzmann Machines (RBMs). DBNs are able to learn complex hierarchical representations of data and have been successfully applied in various applications such as image and speech recognition, natural language processing and fault detection.

Deep belief networks have the following two key characteristics:

- The top-down, generative weights that specify how the variables in one layer depend on the variables in the layer above can be efficiently learned layer by layer.
- After learning, a single bottom-up run that starts with an observed data vector in the bottom layer and applies the generative weights in the opposite direction can be used to infer the values of the latent variables in each layer.

The key components of a DBN for supervised learning include RBMs, layer-wise pretraining, fine-tuning, an output layer, and a cost function. By using these components, a DBN can be trained to perform classification tasks on labeled data.



Restricted Boltzmann Machines (RBMs): In a supervised DBN, the input to the RBMs is the labeled training data.
Layer-wise pretraining: The layers of the DBN are pretrained using an unsupervised learning algorithm such as Contrastive Divergence. The pretraining process involves learning the weights of the RBMs to maximize the likelihood of the input data.
Fine-tuning: Once the layers of the DBN are pretrained, the entire network is fine-tuned using a supervised learning algorithm such as backpropagation. The fine-tuning process involves adjusting the weights of the DBN to minimize the error between the predicted output and the actual output.
Output layer: The output layer of the DBN is typically a softmax layer for classification tasks.
Cost function: The cost function is used to measure the difference between the predicted output and the actual output. For classification tasks, the cross-entropy loss function is commonly used.
Cross Entropy Loss: $\mathbf{L(\hat{()}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)}$

WORKING OF DBN MODEL:

- The DBN is trained via greedy learning techniques. The greedy learning method learns decreasing generating weights layer by layer.

  Upd $W_{11} = W_{11} + L*(P(H_{11} = 1|V) - P(V_1 = 1|H_1))$

- Gibbs sampling steps are carried out by DBNs on the top two hidden levels. The RBM outlined by the two upper hidden layers is sampled ***at*** this stage.
- DBNs use a single ancestral sampling run through the rest of the model to extract a sample from the visible units.
- DBNs are aware that a single pass from bottom to top can deduce the values of the variables inherent in each layer.[16]

DBN: Algorithm

1. *Input:*
   a. *Training dataset: X_train,Y_train*
   b. *Number of hidden layers e*
   c. Number of nodes in each hidden layer H
   d. Learning rate α
   e. number of epochs E
2. *Steps:*
   a. Pre-train the DBN model by unsupervised learning using Contrastive Divergence algorithm
   b. for l ← 1 to L do
   c. Train the RBM in layer l using X_train and the output of layer l-1 as input, with H nodes and α learning rate, for E epochs.
   d. Compute the output of layer l by passing the input through the trained RBM.
   e. end for
   f. Concatenate the output of all hidden layers as the input of the softmax classifier.
   g. Train the softmax classifier using X_train and y_train as input, with softmax activation function and categorical cross-entropy loss function, for E epochs.

*Ou*tput: Trained DBN Model

6. Simulation:

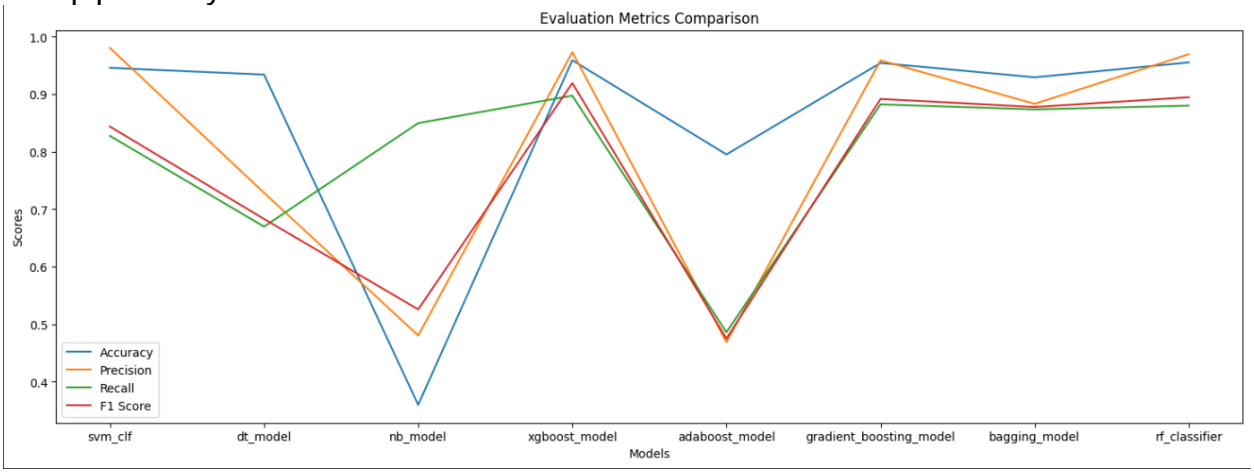
6.1 Simulation of machine learning approach

Evaluation Metrics

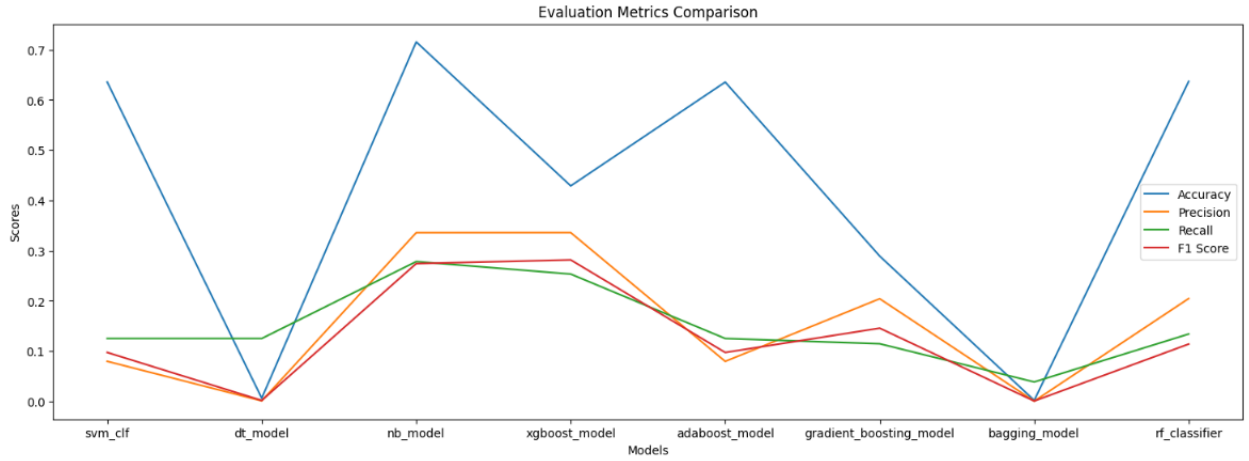
Evaluation metrics are tools or measures used to assess the performance of a machine learning model. They help to determine how well a model is able to make accurate predictions on new data. Some common evaluation metrics used in machine learning include:

1. **Accuracy:** This measures the percentage of correctly predicted instances among all instances.
2. **Precision:** This measures the percentage of true positive instances among all instances predicted as positive.
3. **Recall:** This measures the percentage of true positive instances among all actual positive instances.
4. **F1 Score:** This is the harmonic mean of precision and recall. It is used to balance precision and recall in cases where there may be a trade-off between the two.

Gas pipeline system

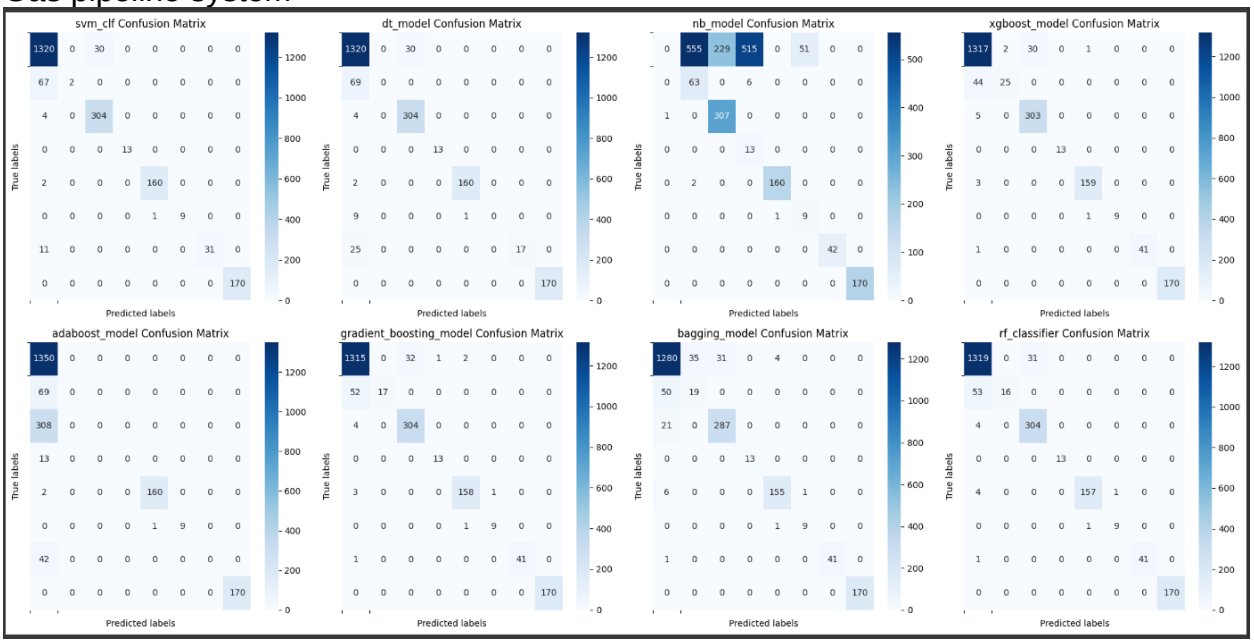


Water storage tank system

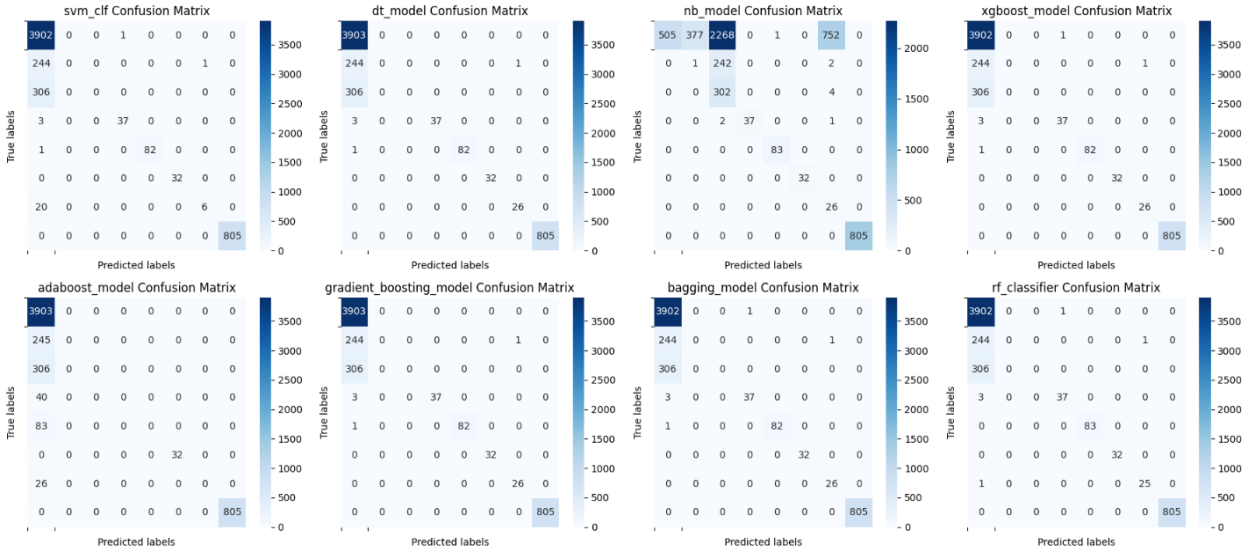


5.Confusion Matrix: It is a table used to describe the performance of a classification model. It shows the number of correct and incorrect predictions made by the model compared to the actual outcomes.

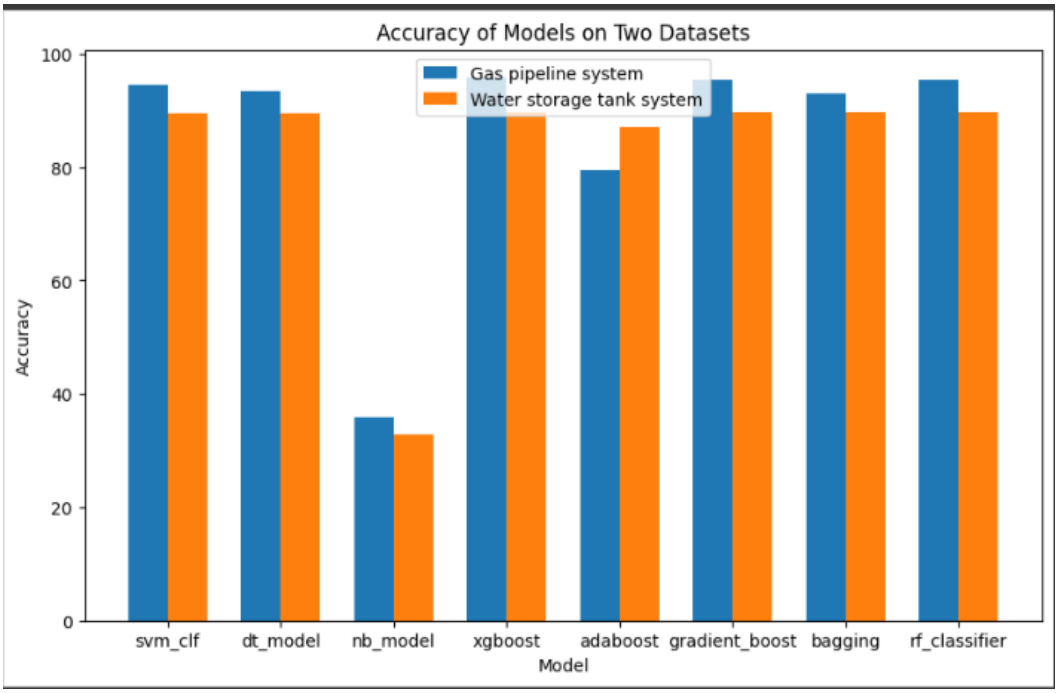
Gas pipeline system



Water storage tank system



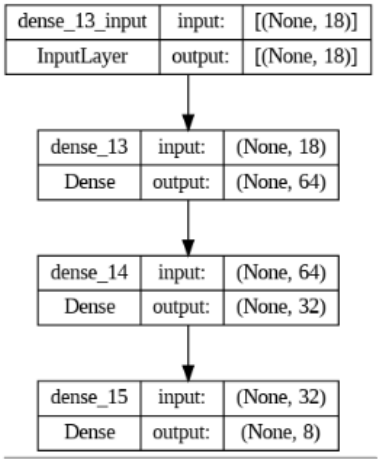
Comparison of accuracies of different machine learning model that we implemented:



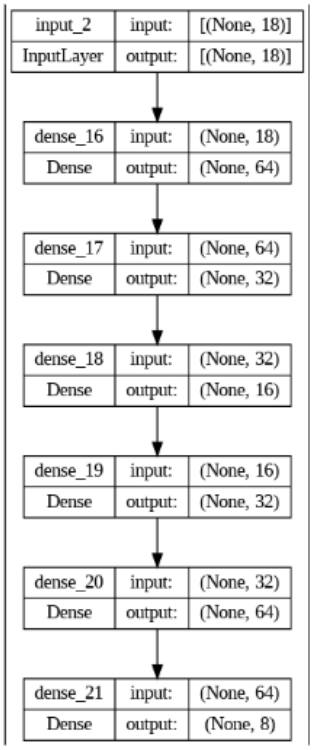
Machine Learning (ML) techniques have been used to analyze the various attacks on SCADA systems with the aim of increasing their reliability/ robustness. Among all the implemented ML techniques XG boost, Gradient boost, Random Forest ensemble technique and SVM performed well.

6.2 Simulation of Deep learning approach

Visualize neural network architectures-



Neural network



Deep belief network

Neural network

Input layer: Dense layer with 64 units (neurons) and input_dim equal to the number of features in the input data
Hidden layer: Dense layer with 32 units and 'relu' activation function
Output layer: Dense layer with number of units equal to the number of classes in the target variable, and 'softmax' activation function to obtain class probabilities.

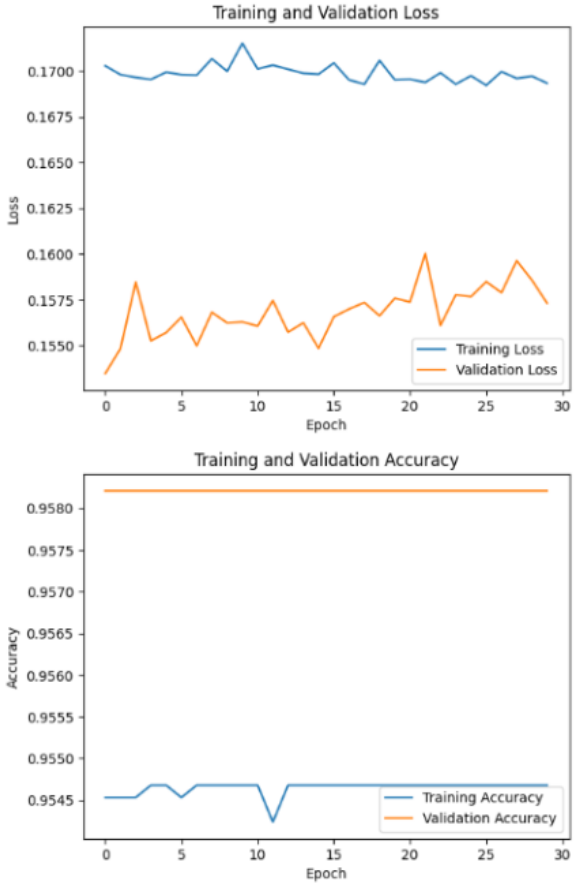
Deep belief network

Input layer: accepts input data with shape X2_train_scaled.shape[1:]
Encoded layer 1: a fully connected layer with 64 neurons and ReLU activation function, with L2 regularization of 0.001
Encoded layer 2: a fully connected layer with 32 neurons and ReLU activation function, with L2 regularization of 0.001
Encoded layer 3: a fully connected layer with 16 neurons and ReLU activation function, with L2 regularization of 0.001
Decoded layer 1: a fully connected layer with 32 neurons and ReLU activation function, with L2 regularization of 0.001
Decoded layer 2: a fully connected layer with 64 neurons and ReLU activation function, with L2 regularization of 0.001
Output layer: a fully connected layer with y2_train_categorical.shape[1] neurons and softmax activation function

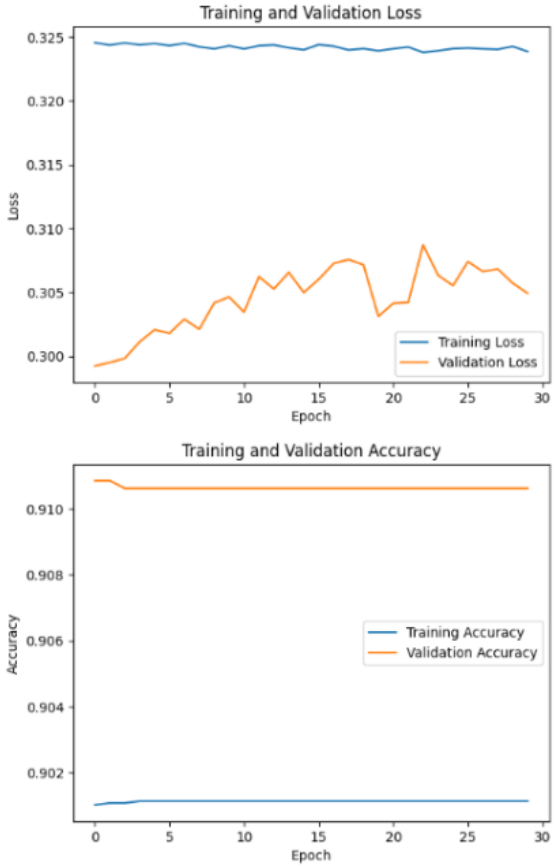
The plotting of training and validation loss and accuracy over epochs using Matplotlib:

Visualizing the training and validation loss and accuracy over epochs can help in determining if the model is overfitting or underfitting.

Neural network model



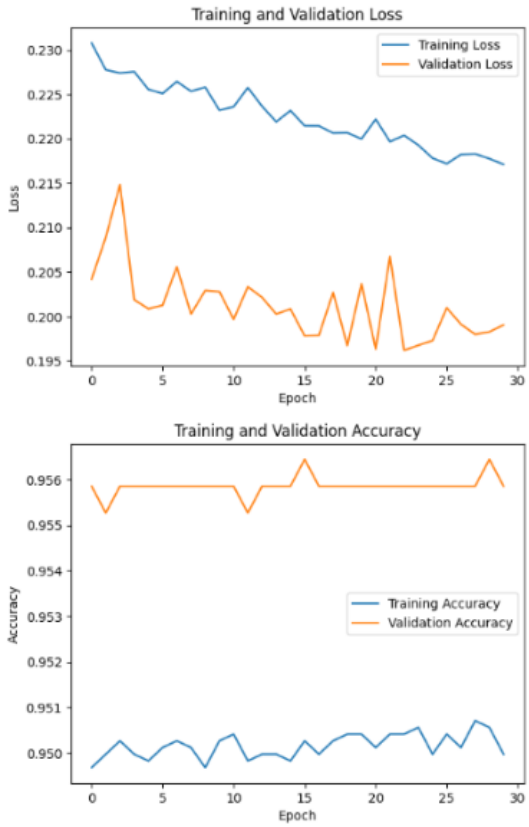
Gas pipeline system dataset



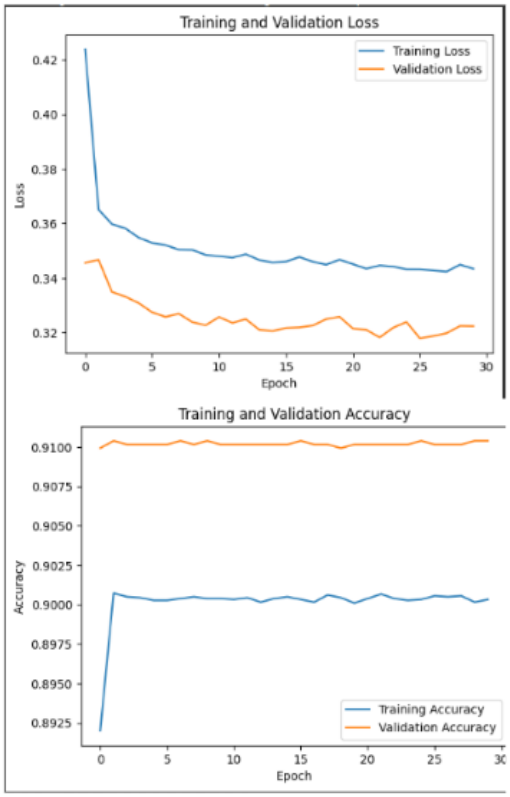
Water storage tank system dataset

The first plot shows the training loss and validation loss over each epoch, with the x-axis representing the epoch number and the y-axis representing the loss value. The second plot shows the training accuracy and validation accuracy over each epoch, with the x-axis representing the epoch number and the y-axis representing the accuracy value. These plots are useful for visualizing the performance of the neural network over the course of training, and can be used to identify issues such as overfitting or underfitting.

Deep Belief Network model



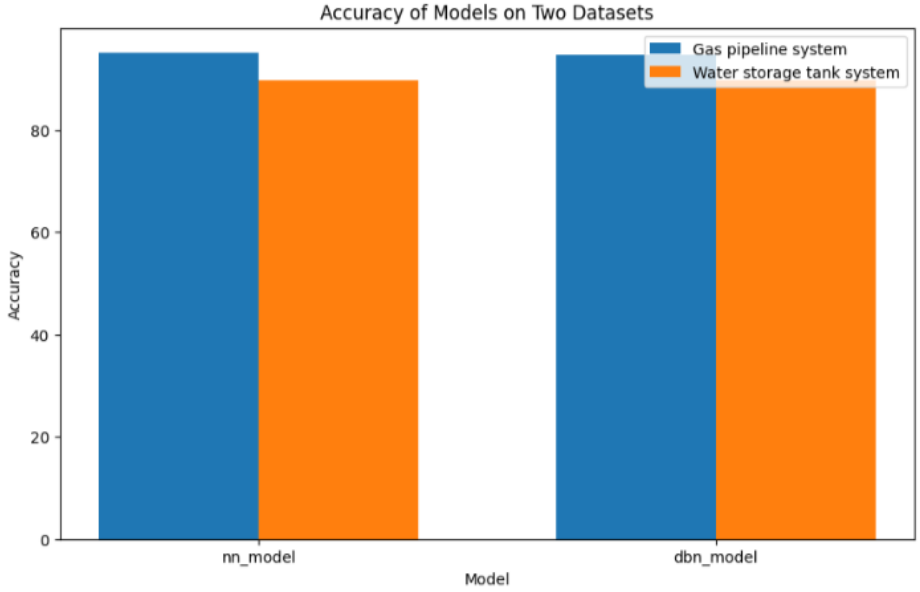
Gas pipeline system dataset



Water storage tank system dataset

These plots are useful for understanding how well the DBN model is learning from the training data and how well it generalizes to the validation data. If the training loss and accuracy are decreasing and increasing, respectively, while the validation loss and accuracy are also decreasing and increasing, respectively, this suggests that the model is learning from the training data and generalizing well to new data. If the validation loss and accuracy start to plateau or even increase while the training loss and accuracy continue to improve, this suggests that the model is overfitting to the training data and not generalizing well to new data.

Comparison of accuracies of Neural Network(NN) and Deep belief Neural Network(DBN) -



A grouped bar plot comparing the accuracy of two models, "nn_model" and "dbn_model," on two different datasets: "Gas pipeline system" and "Water storage tank system." This plot is useful for visually comparing the performance of two models on different datasets.

7.CONCLUSION

- With the development of novel technologies and the integration of enormous number of IoT devices in IACS, the internet traffic is increased sharply, producing large-scale and multi-dimensional data, which makes the cyber-attacks scenarios more sophisticated.
- Shallow machine learning-based detection methods may fail to exploit the deep relationship and implicit information from these datasets in handling the growing security challenges.
- Deep learning techniques have shown their advantages in various domains such as privacy preserving and classification problems , compared to the shallow machine learning
- In conclusion, the project has demonstrated that deep learning techniques—including neural networks and deep belief networks—are more accurate in spotting cyberattacks than traditional machine learning methods.
- This is because deep learning models can analyze and learn from enormous amounts of complex data, which enables them to spot patterns and anomalies that conventional machine learning models might miss.
- Deep learning techniques will probably be used more frequently in the field of cybersecurity as cyber threats become more sophisticated, improving safety for both individuals and organizations.
- Overall, all the models perform well on gas pipeline dataset as compared to water storage tank dataset.

8. REFERENCES

[1] Shahriari, M., et al. "Neural network-based fault diagnosis of an industrial robot." Journal of Intelligent & Fuzzy Systems 33, no. 3 (2017): 1293-1301.
[2] Tao, Y., et al. "Design of a neural network-based soft sensor for estimating the quality of industrial wastewater treatment." Journal of Cleaner Production 195 (2018): 126-133.
[3] Shahriari, M., et al. "Neural network-based modeling and control of an industrial dryer." International Journal of Control, Automation and Systems 15, no. 1 (2017): 238-246.
[4] Huda, M. N., et al. "Neural network-based control of a water treatment plant." International Journal of Environmental Science and Technology 16, no. 5 (2019): 2515-2526.
[5] Liu, W., et al. "An integrated neural network-based approach for fault diagnosis and fault-tolerant control of an industrial process." Control Engineering Practice 67 (2017): 46-55.
[6]https://blog.schneider-electric.com/cyber-security/2018/08/07/one-year-after-triton- building-ongoing-industry-wide-cyber-resilience/
[7]https://resources.infosecinstitute.com/topic/shamoon-reloaded-mysterious-return-dreaded-wiper/
[8]https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power- grid/
[9] https://iq.opengenus.org/deep-belief-network/
[10] K. -D. Lu, G. -Q. Zeng, X. Luo, J. Weng, W. Luo and Y. Wu, "Evolutionary Deep Belief Network for Cyber-Attack Detection in Industrial Automation and Control System," in IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7618-7627, Nov. 2021, doi: 10.1109/TII.2021.3053304.
[11] "A Survey of Intrusion Detection Systems in SCADA: Challenges and Opportunities" by A. Alrawais, M. Alsaleh, A. Alsharif, and R. Almotairi
[12] "A Review of Machine Learning Approaches for SCADA and ICS Security" by M. Rashid, R. Ali, and M. Sharif
[13] "A Deep Learning-Based Intrusion Detection System for SCADA Networks" by M. Masud and M. Arefin
[14] "Machine Learning in Industrial Control Systems Security" by D. Dzung, V. Nguyen, and T. Dao
[15] "Cybersecurity for Industrial Automation and Control Systems" by NIST Special Publication 800-82.

[16] D. Zheng, Z. Hong, N. Wang, and P. Chen, “An improved LDA-based ELM classification for intrusion detection algorithm in IoT application,” Sensors, vol. 20, no. 6, p. 1706, 2020.
[17] Y. Xin et al., “Machine learning and deep learning methods for cybersecurity,” IEEE Access, vol. 6, pp. 35365–35381, 2018.
[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.
[19] Y. He, G. J. Mendis, and J. Wei, “Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism,” IEEE Trans. Smart Grid, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.
[20] T. Morris and W. Gao, “Industrial control system traffic data sets for intrusion detection research,” in Proc. Int. Conf. Crit. Infrastructure Protection, Berlin, Germany: Springer, 2014, pp. 65–78.
[21] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2015, pp. 4580–4584.
[22] U. Adhikari, S. Pan, T. Morris, R. Borges, and J. Beave, “Industrial control system (ICS) cyber attack datasets,” Accessed: Apr. 2020. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ ics-data-sets