A

Capstone Project Report

On

Machine Learning based IDS for Modbus Enabled Industrial Network using Ensemble Learning

Submitted in Partial Fulfilment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science and Information Technology

by

Ms. Pranita Dilip Patil 2260002

Ms. Swati Jaywant Patil 2260003

Ms. Sayali Ravindra Aundhakar 2260004

Ms. Tejal Krishnadev Yadav 2110040

Under The Guidance Of

Prof D.T. Mane

Department of Information Technology

K.E. Society’s

Rajarambapu Institute of Technology, Rajaramnagar

(An Empowered Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

2024-2025

CERTIFICATE

This is to certify that Ms. Pranita Patil, Ms. Swati Patil, Ms. Sayali Aundhkar and Ms. Tejal Yadav has successfully completed the project work and submitted project report on **“Machine learning based IDS for Modbus enabled industrial network using ensemble learning”** for the partial fulfillment of the requirement for the degree of Bachelor of Technology in Computer Science and Information Technology at Rajarambapu Institute of Technology, Rajaramnagar, Dist: Sangli. This final report is the record of the students work carried out under my supervision and guidance.

Prof. D.T.Mane	Dr. A. C. Adamuthe	Dr. P. V. Kadole
Project Guide	**Head IT Dept.**	**Director**

Name and Sign of External Examiner:-

Date:

Place: RIT, Rajaramnagar

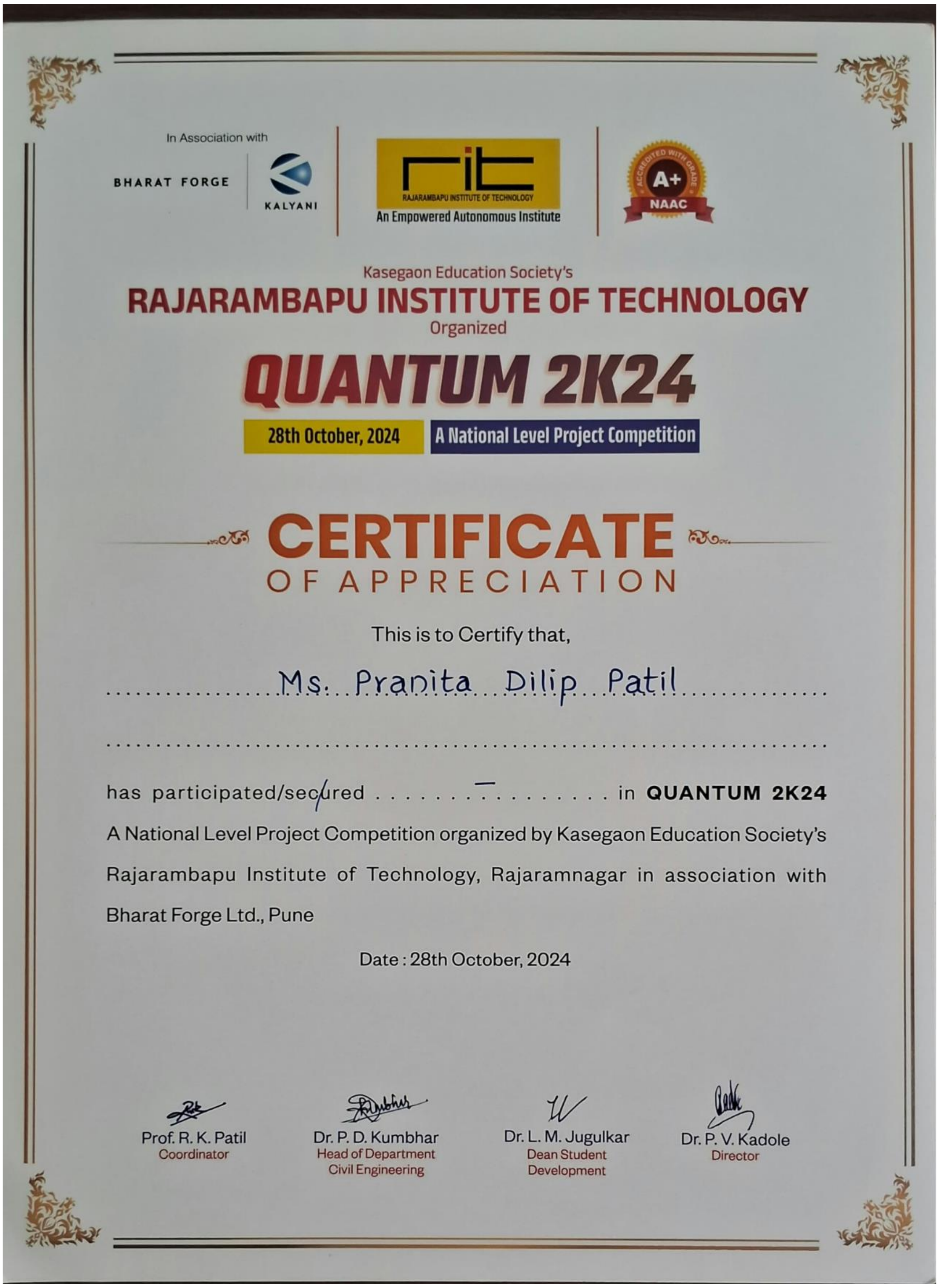
DECLARATION

We declare that this report reflects our thoughts about the subject in our own words. We have sufficiently cited and referenced the original sources, referred or considered in this work. We have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute.

Sr. No	Student Name	Roll No	Signature
1	Pranita Patil	2260002	
2	Swati Patil	2260003	
3	Sayali Aundhkar	2260004	
4	Tejal Yadav	2110040	

Date:

Place: RIT, Rajaramnagar



In Association with

BHARAT FORGE | KALYANI

RAJARAMBAPU INSTITUTE OF TECHNOLOGY
An Empowered Autonomous Institute

ACCREDITED WITH GRADE A+ NAAC

Kasegaon Education Society's

RAJARAMBAPU INSTITUTE OF TECHNOLOGY

Organized

QUANTUM 2K24

28th October, 2024 | A National Level Project Competition

CERTIFICATE OF APPRECIATION

This is to Certify that,

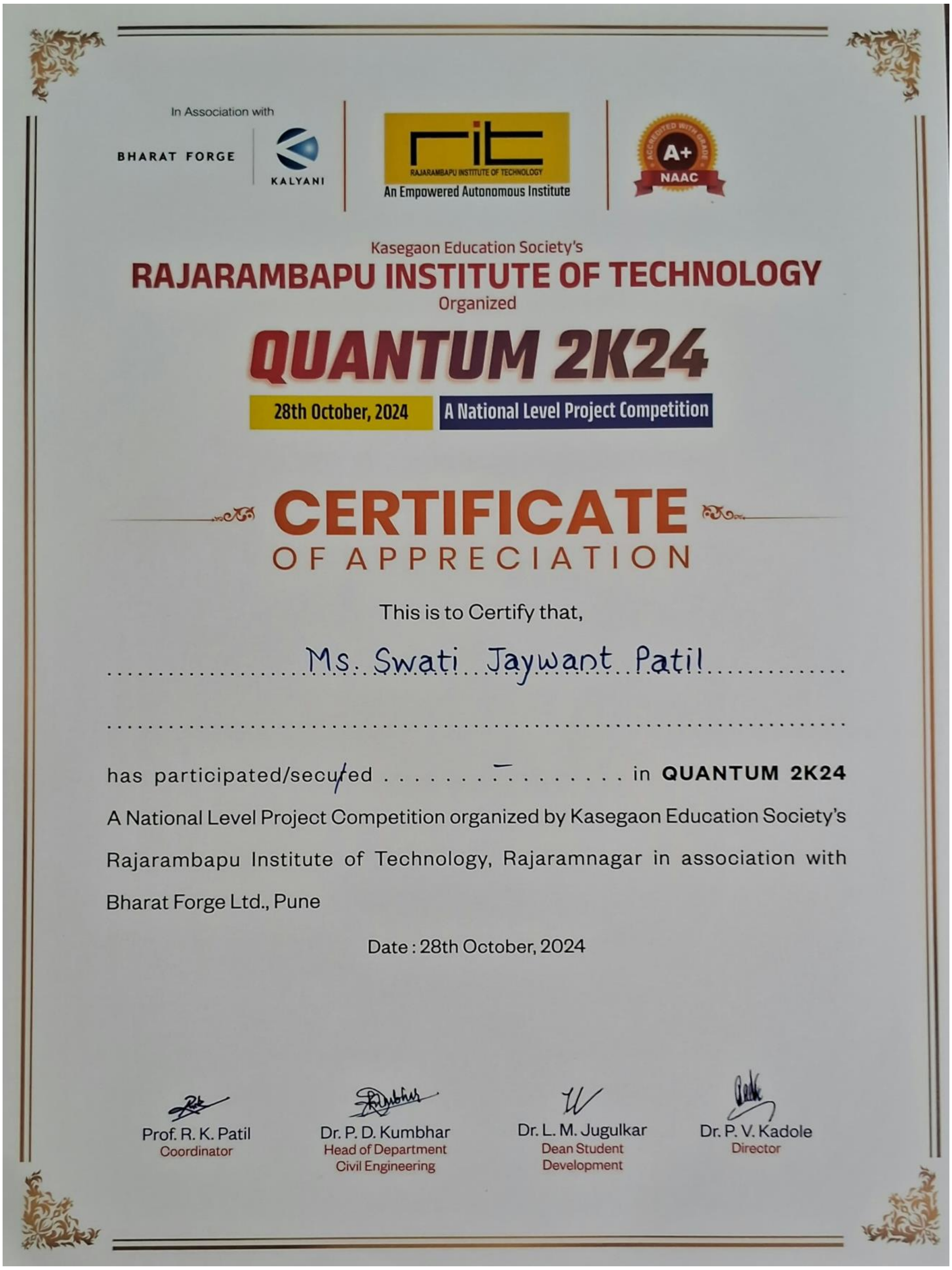
..................Ms. Pranita Dilip Patil..................

....

has participated/secured ....—........ in **QUANTUM 2K24** A National Level Project Competition organized by Kasegaon Education Society's Rajarambapu Institute of Technology, Rajaramnagar in association with Bharat Forge Ltd., Pune

Date : 28th October, 2024

[illegible]	[illegible]	[illegible]	[illegible]
Prof. R. K. Patil	Dr. P. D. Kumbhar	Dr. L. M. Jugulkar	Dr. P. V. Kadole
Coordinator	Head of Department Civil Engineering	Dean Student Development	Director



In Association with

BHARAT FORGE | KALYANI

RAJARAMBAPU INSTITUTE OF TECHNOLOGY
An Empowered Autonomous Institute

ACCREDITED WITH GRADE A+ NAAC

Kasegaon Education Society's

RAJARAMBAPU INSTITUTE OF TECHNOLOGY

Organized

QUANTUM 2K24

28th October, 2024 | A National Level Project Competition

CERTIFICATE OF APPRECIATION

This is to Certify that,

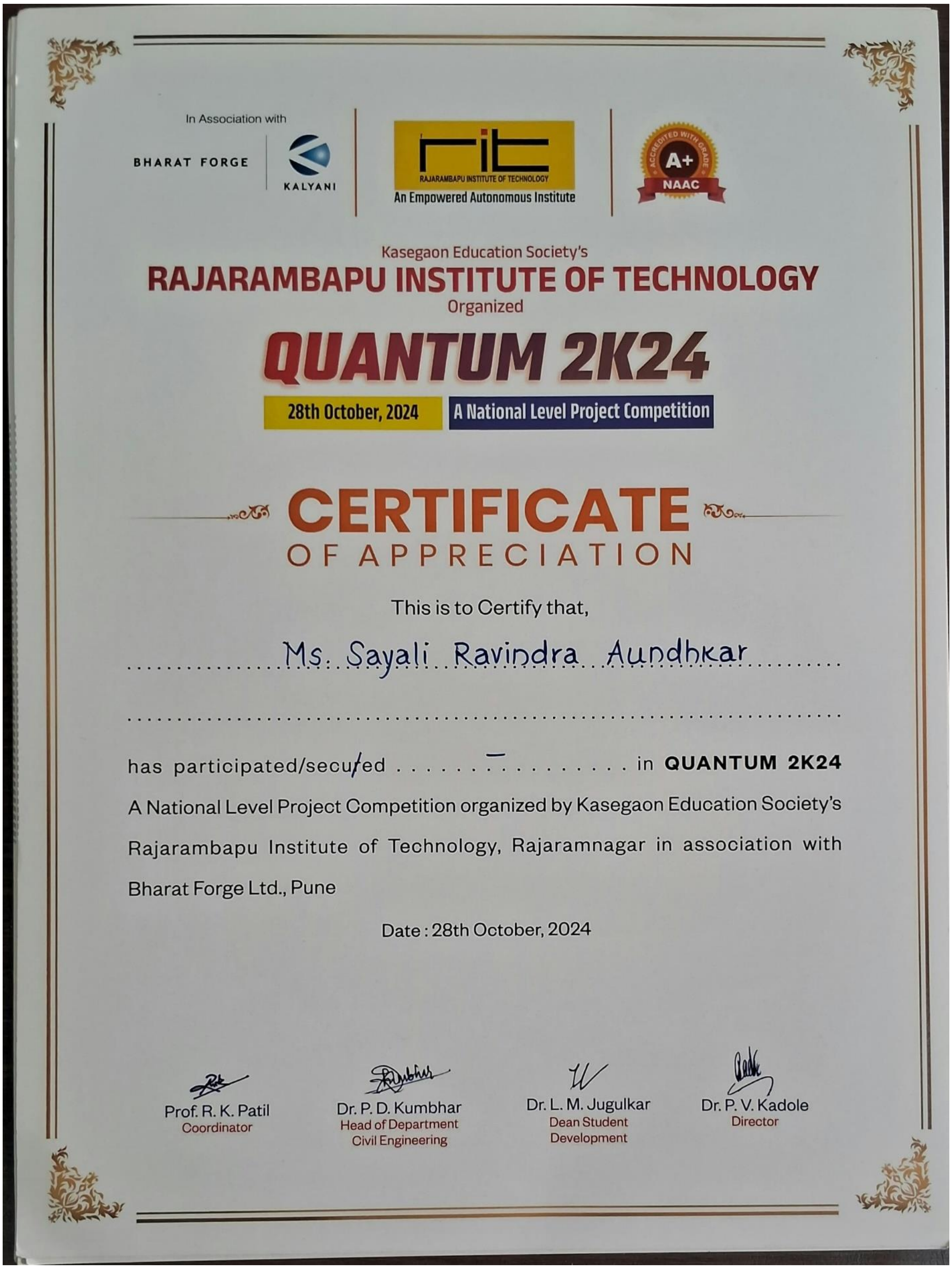
.... Ms. Swati Jaywant Patil ....

....

has participated/secured .... – .... in **QUANTUM 2K24** A National Level Project Competition organized by Kasegaon Education Society's Rajarambapu Institute of Technology, Rajaramnagar in association with Bharat Forge Ltd., Pune

Date : 28th October, 2024

[illegible]	[illegible]	[illegible]	[illegible]
Prof. R. K. Patil	Dr. P. D. Kumbhar	Dr. L. M. Jugulkar	Dr. P. V. Kadole
Coordinator	Head of Department Civil Engineering	Dean Student Development	Director



In Association with

BHARAT FORGE

KALYANI

RAJARAMBAPU INSTITUTE OF TECHNOLOGY

An Empowered Autonomous Institute

ACCREDITED WITH GRADE A+ NAAC

Kasegaon Education Society's

RAJARAMBAPU INSTITUTE OF TECHNOLOGY

Organized

QUANTUM 2K24

28th October, 2024 | A National Level Project Competition

CERTIFICATE OF APPRECIATION

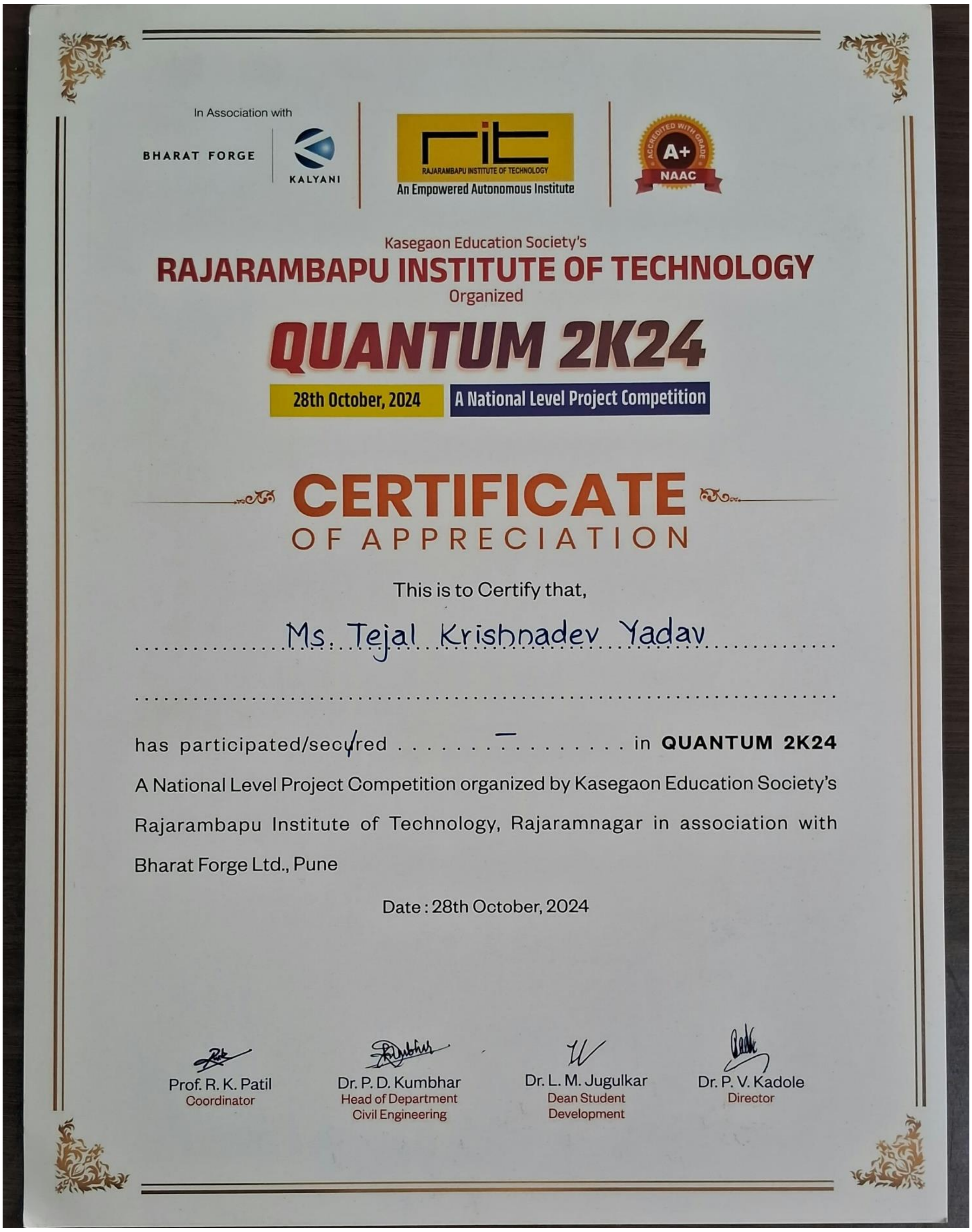
This is to Certify that,

Ms. Sayali Ravindra Aundhkar

has participated/secu~~r~~ed — in **QUANTUM 2K24** A National Level Project Competition organized by Kasegaon Education Society's Rajarambapu Institute of Technology, Rajaramnagar in association with Bharat Forge Ltd., Pune

Date : 28th October, 2024

[illegible]	[illegible]	[illegible]	[illegible]
Prof. R. K. Patil Coordinator	Dr. P. D. Kumbhar Head of Department Civil Engineering	Dr. L. M. Jugulkar Dean Student Development	Dr. P. V. Kadole Director



In Association with

BHARAT FORGE | KALYANI

RAJARAMBAPU INSTITUTE OF TECHNOLOGY
An Empowered Autonomous Institute

ACCREDITED WITH GRADE A+ NAAC

Kasegaon Education Society's

RAJARAMBAPU INSTITUTE OF TECHNOLOGY

Organized

QUANTUM 2K24

28th October, 2024 | A National Level Project Competition

CERTIFICATE OF APPRECIATION

This is to Certify that,

.... Ms. Tejal Krishnadev Yadav ....

....

has participated/secured .... – .... in **QUANTUM 2K24** A National Level Project Competition organized by Kasegaon Education Society's Rajarambapu Institute of Technology, Rajaramnagar in association with Bharat Forge Ltd., Pune

Date : 28th October, 2024

[illegible]	[illegible]	[illegible]	[illegible]
Prof. R. K. Patil	Dr. P. D. Kumbhar	Dr. L. M. Jugulkar	Dr. P. V. Kadole
Coordinator	Head of Department Civil Engineering	Dean Student Development	Director

ACKNOWLEDGEMENT

It is our foremost duty to express our deep sense of gratitude and respect to the guide Prof. D.T.Mane for his uplifting tendency and inspiring us for taking up this project work successful. We are also grateful to Dr. A. C. Adamuthe sir (Head of Department, Information Technology) for providing all necessary facilities to carry out the project work and whose encouraging part has been a perpetual source of information.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Information Technology Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

ABSTRACT

As digital technologies and connected devices proliferate, industrial networks face a growing threat of cyberattacks that compromise their security and reliability. These attacks target critical security pillars like confidentiality, integrity, and availability, posing risks to industrial automation and control systems. Traditional intrusion detection systems (IDS) struggle to detect sophisticated and evolving threats, necessitating innovative approaches. This research proposes a machine learning-based IDS to secure **Modbus-enabled industrial networks**, a protocol widely used in critical infrastructure such as energy, utilities, and manufacturing.

The study evaluates the performance of various machine learning models, including **K-Nearest Neighbors (KNN)**, **Random Forest**, **Support Vector Machine (SVM)**, **AdaBoost**, and an ensemble **stacking classifier**, on the **CIC Modbus Dataset 2023**. This dataset simulates various Modbus-specific attack scenarios like frame stacking, payload injection, and length manipulation, along with benign traffic, providing a robust foundation for model training and evaluation. The stacking classifier achieves the highest accuracy of **99.78%**, outperforming individual models. The results also demonstrate significant improvements in precision, recall, and F1-scores, indicating the effectiveness of ensemble learning in enhancing detection capabilities.

The research emphasizes the importance of integrating decentralized and privacy-preserving frameworks like Federated Learning in future systems to address real-world deployment challenges. By bridging existing gaps, such as the scarcity of real-world datasets, challenges in real-time detection, and lack of comprehensive evaluation metrics, this study contributes a scalable and robust solution to industrial cybersecurity. .

Keywords:Keywords: Intrusion Detection System (IDS), Modbus Protocol, Machine Learning, Stacking Classifier, Cybersecurity, CIC Modbus Dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problems Identification:

Industrial networks are becoming increasingly vulnerable to advanced cyberattacks, particularly those targeting the Modbus communication protocol, which is widely used in critical infrastructure systems like energy, manufacturing, and utilities. Traditional intrusion detection systems (IDS), which primarily rely on static rules and feature filtering, struggle to detect new, evolving, or complex threats. This limitation creates a significant security gap, leaving industrial operations exposed to risks like unauthorized data manipulation, system disruptions, and potential shutdowns.

Modbus networks, due to their simplicity, are especially prone to attacks such as payload injection, frame stacking, and length manipulation. These attacks exploit protocol weaknesses, leading to severe consequences like equipment failure, process delays, or compromised data integrity. The lack of effective real-time detection and the challenge of adapting to novel attack patterns exacerbate the issue.

The need for a robust and adaptive security solution is critical to address these vulnerabilities. A machine learning-based IDS offers the potential to analyze vast amounts of data, recognize complex attack patterns, and adapt to new threats, ensuring the safety and reliability of Modbus-enabled industrial networks.

1.2 Problem Statement:

To develop a machine learning based Intrusion Detection System (for Modbus based industrial networks using ensemble learning with stacking b y combining multiple machine learning models and leveraging their strengths, the system will classify Modbus network traffic into different attack types, improving classification accuracy The goal is to enhance the overall cybersecurity and resilience of industrial networks

1.3 Motivation:

The motivation for the present work titled "Federated Learning (FL) and a Machine Learning-Based Intrusion Detection System (ML-IDS) within Modbus-enabled industrial networks" is to identify the cybersecurity challenges in industrial environments, particularly those using the Modbus communication protocol. This method enables us to the creation of strong intrusion detection systems while prioritizing the protection of private and sensitive data. It mainly targets to A big area where cyberattacks are likely to happen. Adding machine learning makes the cybersecurity defence of Modbus-based industrial networks stronger. The integration of Federated Learning and a Machine Learning-Based Intrusion Detection System aims to create a more adaptive and responsive security framework tailored for Modbus environments. .

1.4 Layout of work:

Data Collection: The project utilized the CIC Modbus Dataset 2023, a dataset specifically designed for Modbus-enabled industrial networks. It includes a mix of benign and malicious network traffic, simulating real-world attack scenarios such as payload injection, query flooding, and frame stacking. Data was collected through methods like network interface capture and Docker bridge capture, ensuring a comprehensive representation of network activity.

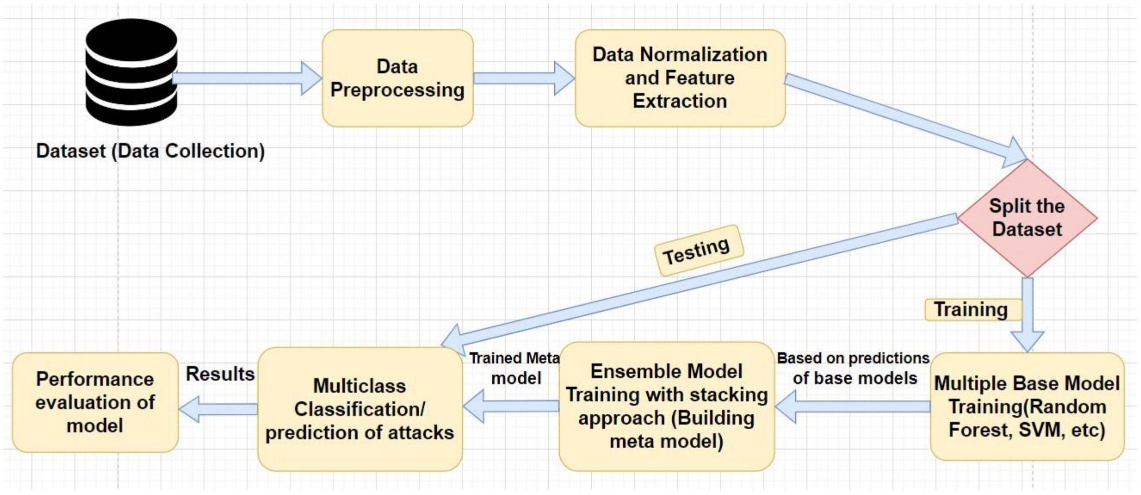

Dataset (Data Collection)
Data Preprocessing
Data Normalization and Feature Extraction
Split the Dataset
Testing
Training
Multiple Base Model Training(Random Forest, SVM, etc)
Based on predictions of base models
Ensemble Model Training with stacking approach (Building meta model)
Trained Meta model
Multiclass Classification/ prediction of attacks
Results
Performance evaluation of model

Figure 1.1: Architecture of proposed system

Data Preprocessing: The dataset was prepared for analysis by structuring and organizing the raw data. Key steps included extracting relevant fields such as timestamps, source/destination IP addresses, and protocol details. Attack logs were formatted to provide a clear distinction between benign and malicious activities, ensuring the data was ready for training machine learning models.

Data Normalization and Feature Extraction: Key features like data length, source/destination ports, and transaction IDs were identified to help detect patterns of attacks. Feature extraction allowed the system to focus on the most important characteristics of the data, aiding in accurate attack detection. While normalization was implied, it played a crucial role in ensuring the dataset was standardized for consistent analysis.

Dataset Splitting: The dataset was divided into training and testing subsets, ensuring that the models learned from one part of the data while being evaluated on another. This split facilitated an accurate assessment of the system's ability to generalize and detect attacks effectively in unseen data. **Multiple Base Model Training:** Several machine learning models, including Random Forest, KNN, SVM, and AdaBoost, were trained on the dataset. These models analyzed the data to identify patterns indicative of both normal traffic and cyberattacks. Each model contributed unique strengths to the detection process, providing diverse insights into network behavior.

Ensemble Model Training (Stacking Strategy) A stacking classifier was implemented to combine the outputs of the base models, creating a meta-model. This approach significantly enhanced accuracy by leveraging the strengths of individual models, achieving an overall accuracy of 99.78%. The stacking classifier demonstrated superior performance compared to any single model, making it a reliable choice for intrusion detection.

Multiclass Classification/Prediction of Attacks The trained stacking classifier was used to classify network traffic into multiple categories, such as benign traffic or specific attack types like length manipulation and payload injection. This stage allowed the system to detect not only whether an attack occurred but also its precise nature, aiding in targeted responses.

Performance Assessment The system's effectiveness was evaluated using metrics such as accuracy, precision, recall, and F1-score. The stacking classifier excelled in all metrics, indicating its ability to accurately detect and classify network traffic. This thorough assessment ensured the reliability and robustness of the intrusion detection system.

Chapter 2

Literature Survey

Federated and Distributed Learning for IDS FL frameworks have been implemented to improve intrusion detection by improving privacy and reducing the communication overhead. DAFL and GöwFed systems dynamically aggregate local models, which result in efficient and accurate detection in IoT and cloud environments [1][7][13]. FL-based models like clustered FL also enhance scalability as it groups devices based on similar data distributions optimising performance in distributed networks [23][25][27]. These techniques do a pretty good job in addressing the privacy concerns and have got high detection accuracies on different platforms.

Hybrid models of ML and DL technologies provide a good solution to detect new sophisticated cyber attacks. CNN and LSTM networks with k-means and Random Forest classifiers are applied to datasets such as NSL-KDD and CIC-IDS2017 [3][9][12]. Such models find a balance between binary and multi-class classification, bridging the gap in intricate attack types. Further research by deep learning in IDS reinforced the fact that hybrid models far outperform traditional systems by both detection rate as well as adaptability for new threats [14][16][3].

These ensemble methods enhance the anomaly detection accuracy, mostly for imbalanced data. Some advanced architectures are Multi Tree and Fusion Net, which consist of Decision Trees, AdaBoost, Random Forest, and SVM for further boosting anomaly detection especially in IoT networks [2][10][18]. These types of methods demonstrate that there are many benefits when utilizing several classifiers regarding dealing with network conditions. Similarly, the federated learning method with weighted optimization adjust the model weights in adaptive manner to attain good performance in real-time [7][24][18] - Benchmark study of different machine learning models IDS that presents the performance effectiveness of k-NN, Decision Tree, and Naive Bayes on the CICIDS2017 dataset. The methods of unsupervised learning are WGAN-div, Info GAN, and k-means clustering. These methods generate synthetic data and identify the anomaly in the emerging pattern of network traffic [5][12][15]. Models that utilized unsupervised approaches were better at accuracy in detecting zero-day attacks and sophisticated intrusions both in cloud and IoT environments [6][19]. The approach enables the system to learn the changes in network behavior hence reducing false positives. Utilizing the NSL-KDD dataset, [16] provides an intrusion detection system that is based on Random Forest algorithms with feature selection that achieved an accuracy rate of 99.68% in all categories of attacks.Lightweight and Explainable IDS for IoT and IIOT Optimization of lightweight models by feature selection technique, comprising RFE with XG Boost and expert-driven approaches, enhances the detection system's accuracy even in resource-constrained IoT environments. Thus, Recursive Feature Elimination using XG Boost with SHAP-based explanation ensures both accuracy and explainability for intrusion detection without obscurity in decision making [28][32][35]. These models are suitable for deployment in real-time in IIOT and Wi-Fi networks, yielding high detection rates with low computation overhead [24][39]. [17]-Exploits ensemble learning and hyperparameter tuning to enhance the anomaly-based IDS for IoT with a gain in precision besides reducing the false positive rate. Comparative and Benchmarking Studies A comparison of the classifier, such as Random Forest, XG Boost, Naïve Bayes, and KNN, reveals that ensemble methods are superior to single approach methods in IoT networks [18][30][31][32]. The experiments show that Random Forest and XG Boost work very well in the detection of sophisticated attacks, such as DoS and Mirai [33][40]. Benchmarking results reflect the need for the models to balance the accuracy of detection with real-time adaptability. [35][39][6] Proposes a model for DDoS attack detection on the UNSW-NB15 dataset improving the speed and accuracy using Random Forest and XG Boost.Data Quality and Multi-Dataset Integration Data quality is the other very important thing of IDS efficiency. Based on the experiment results, BERT and GPT-2 outperform algorithms on mis-labeled and inconsistent data, providing robustness to noise [37][34]. The use of a multi-dataset such as NSL-KDD, UNSW-NB15, and UGR’16 increase the generality of the models with respect to any setting with highly diminished false positives value [36][38]. Such methods ensure the network scenarios and different data sets stay effective with its IDS models.

Chapter 3

Preliminaries

Motivated to design an Intrusion Detection System (IDS) based on the concept of machine learning, we reviewed multiple research papers discussing different existing approaches and techniques. Through this study, we identified that techniques such as the Random Forest algorithm, the KNN algorithm, and Support Vector Machines (SVM) are commonly utilized.

3.1 Techniques Utilized for Implementing IDS

3.1.1 Random Forest

Random Forest is a machine learning algorithm widely used for classification and anomaly detection. Its workflow is as follows:

1. **Bootstrapping:** Perform random sampling with replacement to create multiple subsets of data points.

2. **Creating Decision Trees:** Create a decision tree for each data subset. At each node, consider a random subset of features for splitting.

3. **Construction of Decision Tree:** Use a recursive algorithm to split nodes based on criteria such as Gini impurity. Continue until the tree reaches maximum depth or

further splitting yields minimal improvement.

4. **Individual Tree Predictions:** Each constructed decision tree makes predictions on new data points.

5. **Final Prediction:** For classification tasks, the final prediction is based on the majority class predicted by the ensemble of decision trees.

3.1.2 K-Nearest Neighbors (KNN)

KNN is a simple and effective algorithm used for classifying attacks in a simulated Modbus-enabled industrial network. Its workflow is as follows:

1. **Data Preparation:** Start with a labeled training dataset containing features and corresponding labels.

2. **Calculate Distances:** For a new input data point, calculate distances to all points in the training set.

3. **Select Nearest Neighbors:** Identify the K closest data points based on shortest distances.

4. **Classify or Predict:** Assign a class to the input data point based on the majority class among the K nearest neighbors.

5. **Output the Result:** Return the predicted class label for the input data point.

3.1.3 Support Vector Machine (SVM)

SVM is primarily applied for classification problems, especially with high-dimensional data. The algorithm workflow is as follows:

1. **Data Collection:** Collect labeled data points containing features and class labels.

2. **Feature Extraction and Selection:** Gather and preprocess appropriate features for the model.
3. **Model Training:** Find a hyperplane (decision boundary) that separates predefined classes to make them more distinguishable.
4. **Optimization:** Maximize the distance between support vectors while minimizing classification error.
5. **Model Evaluation:** Assess the model's performance using appropriate metrics.
6. **Prediction:** Apply the trained SVM model on new data points to predict their class labels.
7. **Deployment:** Once satisfactory accuracy is reached, deploy the model on unseen data for real-time prediction.

3.1.4 AdaBoost Algorithm

AdaBoost (Adaptive Boosting) is a supervised ensemble learning method that combines multiple weak classifiers to form a strong classifier. Its workflow is as follows:

1. **Initialization:**
   - Start with a set of N training samples $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, where x_i are the input features and y_i the output labels.
   - Initialize weights $w_i = \frac{1}{N}$ for each training sample to make all samples equally important initially.
2. **Training Weak Learners:**
   - Train a weak learner, such as a decision stump, using the current weight distribution.

- Minimize the weighted classification error:

$$\varepsilon = \sum_{i=1}^{N} w_i \cdot I(y_i \neq h(x_i)),$$

where $I(\cdot)$ is the indicator function, and $h(x_i)$ is the prediction of the weak learner.

3. **Error Calculation and Classifier Weighting:**

   - If $\varepsilon > 0.5$, discard the weak learner as its generalization is poorer than random guessing.
   - Compute α, the weight of the weak learner:

$$\alpha = \frac{1}{2} \ln \frac{1 - \varepsilon}{\varepsilon}.$$

4. **Weight Update:**

   - Update weights for training samples:

$$w_i \leftarrow w_i \cdot e^{-\alpha \cdot y_i \cdot h(x_i)}.$$

   - Normalize weights by dividing each weight by their sum:

$$w_i = \frac{w_i}{\sum_{i=1}^{N} w_i}.$$

5. **Final Model Construction:**

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x) \right),$$

where T is the total number of iterations, and $H(x)$ is the final prediction.

3.1.5 Stacking Classifier

Stacking is an ensemble learning technique that combines predictions from multiple base models to improve classification performance. Its workflow is as follows:

1. **Base Model Selection:**
    - Choose diverse base models, such as Random Forest, KNN, and SVM, to capture different aspects of the data.
    - Ensure each base model excels in a specific aspect to provide varied insights to the ensemble.
2. **Training Base Models:**
    - Train each base model on the same training dataset to predict outputs independently.
    - Optimize each model for its respective algorithm to achieve the best performance.
3. **Generate Meta-Features:**
    - Use the trained base models to generate predictions on both the training and test datasets.
    - These predictions form secondary data, called meta-features.
4. **Training the Meta-Model:**
    - Train a meta-model (e.g., Logistic Regression) using the meta-features as inputs.
    - The meta-model learns to combine the predictions of the base models to maximize overall accuracy.

5. **Final Prediction:**

    - For new input data, the base models generate predictions, which are fed into the meta-model.
    - The meta-model combines these predictions to produce the final output.

Chapter 4

Implemented Work

4.1 Proposed System

The proposed solution is to build a machine learning model to classify the various types of attacks in Modbus-based industrial networks using the ensemble learning stacking approach. This model enables effective classification of different types of attacks to ensure the accuracy of classification of detected attacks, which helps to improve the security in Modbus-based industrial networks. The approach is data-driven and includes the following key components:

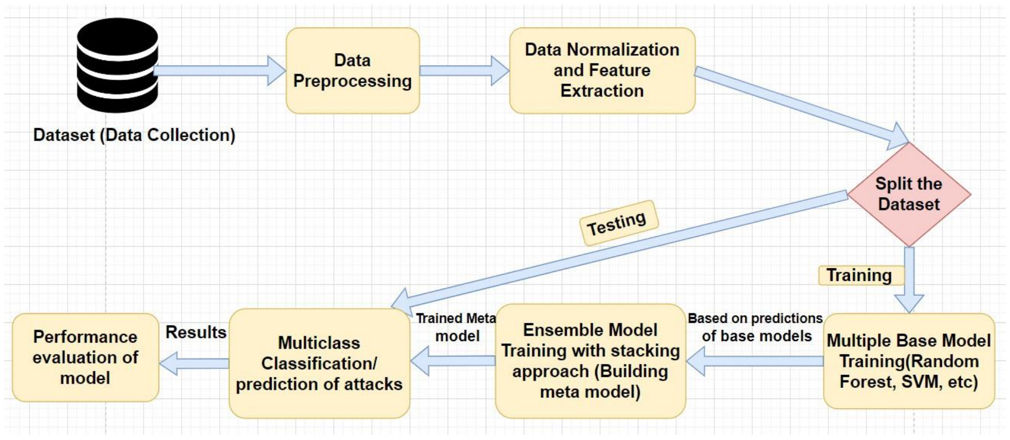

Dataset (Data Collection)
Data Preprocessing
Data Normalization and Feature Extraction
Split the Dataset
Testing
Training
Multiple Base Model Training(Random Forest, SVM, etc)
Based on predictions of base models
Ensemble Model Training with stacking approach (Building meta model)
Trained Meta model
Multiclass Classification/ prediction of attacks
Results
Performance evaluation of model

Figure 4.1: Proposed System Architecture

4.1.1 Data Collection: CIC Modbus Dataset 2023

We have selected the **CIC Modbus Dataset 2023** for our research work. Below is a brief description of the dataset, representing how it was created and its contents. This dataset supports our research in intrusion detection for industrial networks using the Modbus protocol.

- **Dataset Composition**

The CIC Modbus Dataset 2023 comprises two types of data:

- **Attack Data:** Network captures simulating various Modbus protocol attacks, such as reconnaissance, query flooding, payload manipulation, and more, based on the MITRE ICS ATT&CK framework.
- **Benign Data:** Legitimate Modbus communication captured in a substation environment.

- **Dataset Creation Architecture**

The architecture used to build the CIC Modbus Dataset 2023 is as follows:

- **Simulated Substation Network:** Developed in a Docker environment with containers representing:
  - Intelligent Electronic Devices (IEDs)
  - Supervisory Control and Data Acquisition (SCADA) Human-Machine Interfaces (HMIs)
- **Device Interactions:**
  - IEDs change voltage values periodically or in response to SCADA HMI requests.
  - SCADA HMIs adjust based on IED data and over/undervoltage conditions.

- **Security Levels:** Devices are categorized as:
  - **Secure:** Contain detection code.
  - **Insecure:** Operate only with scripts.

- **Data Collection Process**

The data collection process for building the CIC Modbus Dataset 2023 involved:

- **Network Interface Capture:** Individual device traffic captured using tcpdump.
- **Docker Bridge Capture:** Comprehensive capture of all network communications.
- **Attack Scenarios:** Attacks executed from external devices, compromised IEDs, and compromised HMIs.

- **Dataset Formats**

The CIC Modbus Dataset 2023 contains data in the following file formats:

- **PCAP Files:** Captured network traffic, divided into 100MB chunks.
- **CSV Logs:** Timestamped attack logs with fields such as Target IP, Attack Type, and Transaction ID.

- **Applications of the Dataset**
  - **Substation Security:** Development of trust models and security assessments for substation devices.
  - **Machine Learning:** Utilized for anomaly detection, classification, and clustering to enhance security in Modbus networks.

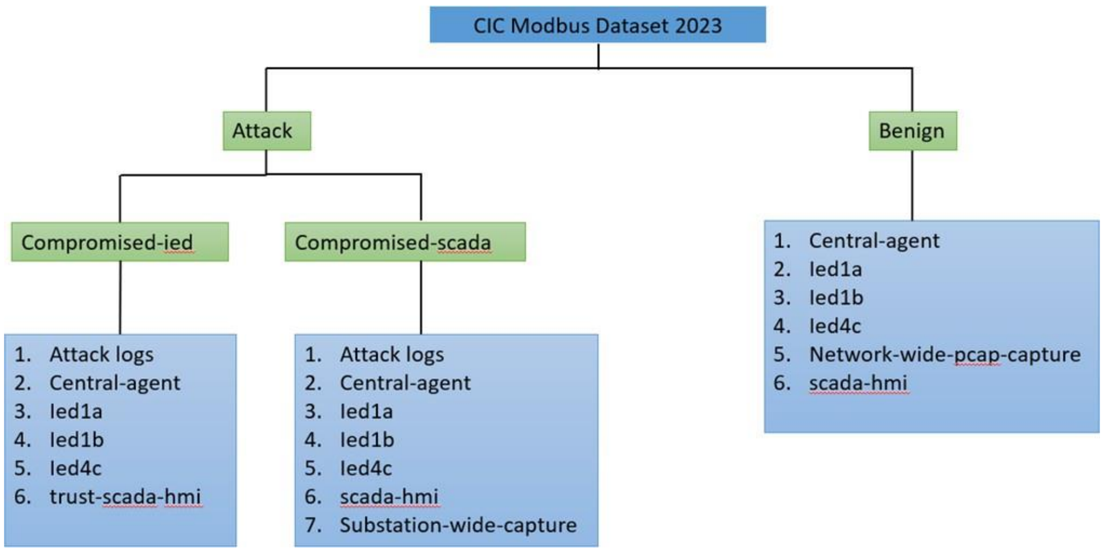


Figure 4.2: Dataset File structure

4.2 Dataset File Structure

- All PCAP files are present except in the Attack Logs folder.
- There are approximately 101 PCAP files in different folders and around 30 log files.

4.2.1 Attack Logs

The attack log files contain the following columns:

Table 4.1: Columns of Attack Logs files

Feature No.	Feature Name
1	Timestamp
2	Target IP
3	Attack
4	Transaction ID

- **Timestamp:** The exact time at which the event or attack activity occurred.
- **TargetIP:** The IP address of the device targeted in the specific event or attack. Each IP corresponds to a specific device within the simulated substation network.

- **Attack:** Describes the type of attack or action being performed. It provides details about the activity, such as "Starting Payload Injection..." or "Payload Injection.. Complete."
- **TransactionID:** A unique identifier for each transaction or session associated with an event. It helps track the sequence of actions within the same attack scenario.

4.2.2 PCAP Files

The PCAP files contain the following columns:

Table 4.2: Columns of PCAP files

Feature No.	Feature Name
1	Timestamp
2	Source IP
3	Destination IP
4	Source Port
5	Destination Port
6	Data Length

- **Timestamp:** The time when the network packet was captured, often recorded in hours and seconds format.
- **Source_IP:** The IP address of the device that sent the network packet.
- **Destination_IP:** The IP address of the device that received the network packet.
- **Source_Port:** The port number on the source device that initiated the network communication.
- **Destination_Port:** The port number on the destination device that received the network communication.
- **Protocol:** The communication protocol used for the packet, such as TCP (Transmission Control Protocol).

4.3 Description of Attack Scenarios

Table 4.3: Class Labels and Number of Instances

Class Labels	**Number of Instances**
Benign	2,000,000
Brute force or specific coil. Address: 1	1,289,879
Brute force or specific – Complete	1,289,711
Starting Query flooding..	4,854
Starting Frame Stacking...	4,809
Frame Stacking... Complete	4,809
False Data Injection Attack	4,364
Query Flooding	4,120
Query flooding. Complete	4,120
Starting Payload Injection...	3,996
Payload Injection.. Complete	3,996
Delay response: Sending response to client	3,375
Query flooding.. Complete	2,532
Baseline Replay: In position	2,379
Starting Length manipulation...	1,947
Length manipulation.. Complete	1,947
Replay – Complete	1,720
Replay	1,720
Length manipulation – Complete	1,388
Length manipulation	1,388
Recon. Complete	1,276
Stacked Modbus Frames	1,044
Stacked Modbus Frames - Complete	1,044
Payload injection	920
Payload injection - Complete	920
Recon. Range: 2000	638
Recon. Range: 125	638
Brute force or specific coil.Address: 8	344
Brute force or specific coil.Address: 14	344
Brute force or specific coil.Address: 13	344

Below is a detailed description of the attack scenarios present in the dataset:

1. **Benign:** Normal Modbus communication traffic serves as a baseline for analyzing deviations or anomalies.
2. **Brute Force on Specific Coil:** Repeatedly targeting specific coil addresses to alter or access device states, potentially disrupting operations or gaining unauthorized control.
3. **Query Flooding:** Sending a high volume of Modbus requests to overwhelm the target device, causing slowdowns or denial of service (DoS).
4. **Frame Stacking:** Sending multiple Modbus frames in a single packet, leading to confusion, buffer overflows, or crashes.
5. **False Data Injection:** Injecting incorrect data into packets, causing incorrect actions or disruptions.
6. **Payload Injection:** Inserting unauthorized commands or data into packets to alter device behavior or compromise network integrity.
7. **Delay Response:** Intentionally slowing down responses, disrupting real-time operations.
8. **Baseline Replay:** Replaying previously captured legitimate traffic, tricking systems into interpreting outdated data as current.
9. **Length Manipulation:** Altering packet length fields, leading to misinterpretation, buffer overflows, or crashes.
10. **Replay:** Sending previously captured traffic back into the network, forcing unintended actions.

4.4 Data Preprocessing

The following steps were taken to combine all files by mapping the data records:

4.4.1 Data Collection and Input Preparation

Goal: To collect and preprocess raw data in an organized format for further analysis.

Steps:

1. Raw PCAP files are converted into CSV format using packet processing tools (not shown in the code but assumed as a preprocessing step).
2. Attack logs are stored in CSV files across various subdirectories (e.g., *compromised-ied, compromised-scada*).

4.4.2 Load and Merge Attack Logs

Goal: To take various sources of attack logs and merge them into a sensible structure to achieve labeling objectives.

Actions:

1. The load_attack_logs function iterates over directories containing CSV files for attack logs.
2. Each file is read into a Pandas DataFrame and stored in a list.
3. Attack logs are merged into a single DataFrame for uniform access during the labeling process.

4.4.3 Label PCAP Data with Attack Information

Objective: Enhance the PCAP data with labels indicating whether a packet is an attack or benign.

Steps:

1. The label_pcap_data_with_attack function reads each PCAP CSV file from the directory.
2. Each packet is initially labeled as *benign*.
3. Timestamps and destination IPs are matched against records in the combined attack logs.
4. If a match is detected, the AttackType field is updated with the appropriate label.
5. Labeled data from different directories are combined into a uniform dataset.

4.4.4 Joining Attack and Benign Data

Goal: Integrate attack and benign datasets into a unified dataset.

Steps:

1. Process attack-labeled PCAP data (e.g., *compromised-ied_processed*, *compromised-scada_processed*).
2. Similarly, process benign PCAP data (e.g., *central-agent_processed*, *ied1a_processed*).
3. Concatenate attack and benign datasets to form the complete labeled dataset.

4.4.5 Data Conversion for Acceleration

Goal: Enable GPU acceleration for computational efficiency.

Procedure:

1. Convert the attack logs DataFrame to a NumPy array using Pandas.
2. Convert the NumPy array to a CuPy array for GPU-accelerated operations.

4.4.6 Saving the Final Dataset

Goal: Save the processed and labeled data for downstream analysis.

Steps:

1. Save the combined labeled dataset as a CSV file (combined_labeled_data_final.csv).
2. Ensure proper balancing to prevent bias in machine learning models.

4.5 Handling Imbalanced Data

Table 4.4 highlights the class imbalance observed in the dataset.

Class	Frequency	Proportion (%)
Benign	87,388,542	98.5
Attack Class A	1,200,000	1.35
Attack Class B	500,000	0.15

Table 4.4: Class Imbalance in the Dataset

Steps Taken:

1. **Data Resampling:**
   - Oversample minority classes using techniques such as SMOTE.
   - Undersample the majority class (Benign) to balance the dataset.
2. **Class Weight Adjustment:**
   - Adjust the loss function to assign higher weights to underrepresented classes during model training.
3. **Use of Anomaly Detection Models:**
   - Train models that inherently address imbalance by focusing on detecting rare patterns.

4.6 Steps Taken to Address Imbalance

4.6.1 Analyzing the Data Distribution

The initial analysis revealed a highly imbalanced dataset with the following class distributions:

- **Benign Records:** 87,388,542 (majority class)
- **Attack Records:** Spread across multiple categories, with counts ranging from a few thousand to less than a thousand for certain attack types.

This skew necessitates balancing the dataset to ensure fair representation of all classes during model training.

4.6.2 Data Splitting of Benign and Attack Data

The dataset was split into two main categories:

- **Benign Data:** All records identified as benign.
- **Attack Data:** All records corresponding to various attack types.

This separation enables independent manipulation of the majority (benign) and minority (attack) classes.

4.6.3 Undersampling the Benign Class

To balance the dataset:

- A random sample of **2 million records** was extracted from the benign class.
- This reduced size maintains diversity while significantly lowering computational overhead.
- Random sampling ensures unbiased selection of benign records.

4.6.4 Combining and Balancing the Dataset

The reduced benign dataset was combined with the entire attack dataset:

- Attack records were **not undersampled** to preserve their diversity and detail.
- The combined dataset was shuffled to avoid order-based biases during model training.

4.6.5 Analyzing the Balanced Dataset

The resulting dataset achieved a more balanced distribution:

- **Benign Records:** 2 million
- **Attack Records:** Counts remained unchanged, ranging from hundreds to thousands based on attack type.

This approach ensures sufficient weight is given to attack records during model training while maintaining diversity across all classes.

4.7 Data Distribution: Before and After Balancing

Table 4.5 provides a summary of the class distributions before and after balancing.

Class Type	**Initial Count**	**After Balancing**
Benign Records	87,388,542	2,000,000
Attack Records (total)	Various (hundreds to thousands)	Various (unchanged)

Table 4.5: Class Distribution Before and After Balancing

4.8 Techniques Used in Data Processing

4.8.1 1. Handling Missing Values

To address missing values in crucial attributes such as Source_IP, Destination_IP, Source_Port, Destination_Port, Protocol, Data_Length, TargetIP, and TransactionID, predefined replacements were used:

- **Categorical attributes:** Replaced with the string "unknown".
- **Numerical attributes:** Replaced with zeros.

This approach maintains the integrity of the dataset by ensuring no rows are dropped due to missing data.

4.8.2 2. Time Stamp Transformation

The Timestamp column was transformed into numerical values representing total seconds from the origin. This transformation:

- Converts string-based timestamps into a numerical format.
- Facilitates better utilization of temporal features in machine learning models.

4.8.3 3. Categorical Encoding

Categorical features such as Source_IP, Destination_IP, Protocol, AttackType, and TargetIP were encoded using label encoding:

- Each unique category was mapped to a numerical value.
- Label encoding preserves uniqueness across categories.
- A consistent mapping was maintained for the AttackType attribute, ensuring reliable encoding throughout.

4.8.4 4. Data Split for SMOTE

To handle class imbalance, the dataset was split as follows:

- **Excluded Data:** Records from dominant classes (e.g., benign) excluded from the SMOTE process.
- **SMOTE-Applicable Data:** Records from minority classes oversampled using SMOTE.

This distinction prevents artificial dominance by ensuring that the majority classes are not oversampled excessively.

4.8.5 5. Feature Standardization

Features in the SMOTE-applicable dataset were standardized:

- Transformed to have a mean of 0 and a standard deviation of 1.
- Standardization normalizes numerical features, enhancing model performance.

Encoded labels in the AttackType column were also standardized, ensuring consistent numerical representation of attack types and benign labels.

4.9 Exploratory Data Analysis (EDA)

The preprocessing steps enabled an effective exploratory data analysis phase, providing valuable insights into:

- Data distributions and imbalances.
- Relationships among features.
- Identification of patterns and anomalies.

The EDA results facilitated informed decision-making during model development.

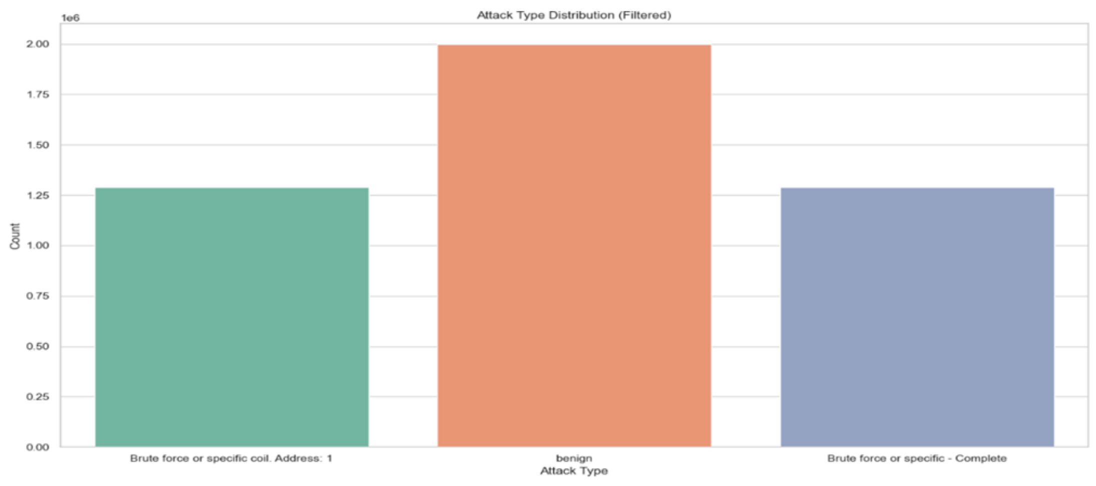


Figure 4.3: Distribution of Attack Types(Most occured)

4.10 Distribution of Attack Type (Most Occurred)

The graph presents a bar chart illustrating the distribution of attack types in the dataset after filtering. Three attack types dominate the dataset:

- **Benign:** This attack type, which represents normal traffic, has the largest count, nearing 2 million instances, significantly outnumbering other attack types.
- **Brute Force on Specific Coil Address: 1:** This attack type appears second, with a count just over 1.25 million, indicating a considerable presence in the dataset.
- **Brute Force on Specific Coil - Complete:** This attack type follows with a count of approximately 1.25 million, similar to the previous one, highlighting its frequency in the dataset.

The chart visually emphasizes the imbalance in the dataset, with benign traffic significantly outweighing the attack types, though these three specific attacks are the most prevalent among the attack types present.

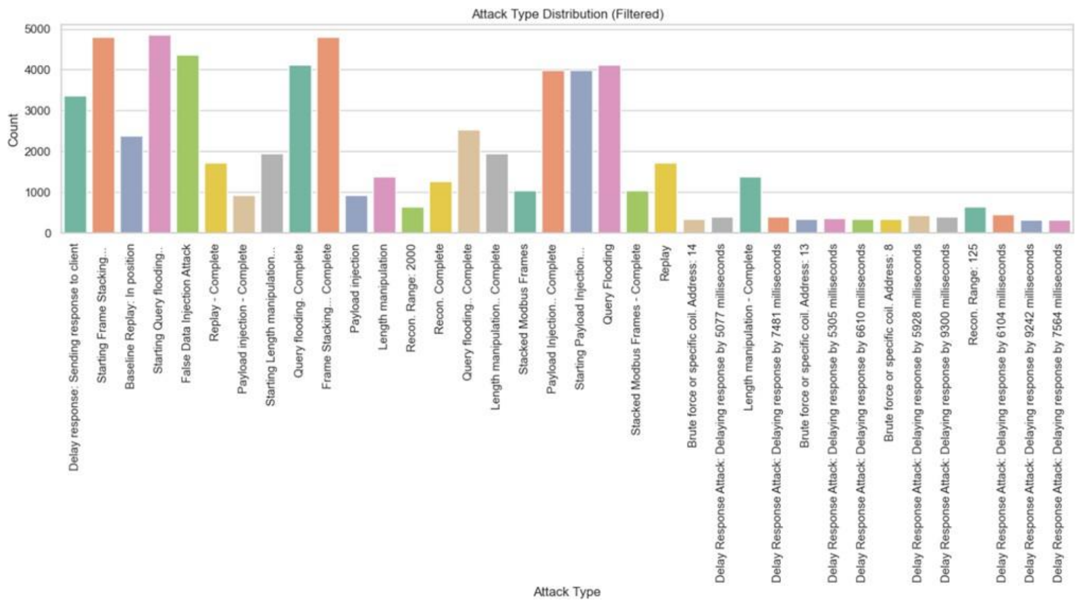


Figure 4.4: Distribution of Attack Types

4.11 Distribution of Attack Types

The graph showcases the distribution of various attack types in the dataset after filtering. It reveals a diverse range of attacks, with the most frequent ones highlighted.

4.12 Top 10 Source and Destination IP Addresses

This figure contains two bar plots, showing the frequency distribution of the top 10 Source IPs and Destination IPs in the dataset:

Top 10 Source IPs (Left Plot)

- The most frequent Source IP is **185.175.0.3**, accounting for over 2 million occurrences, indicating it is the primary source of traffic or activities in the dataset.

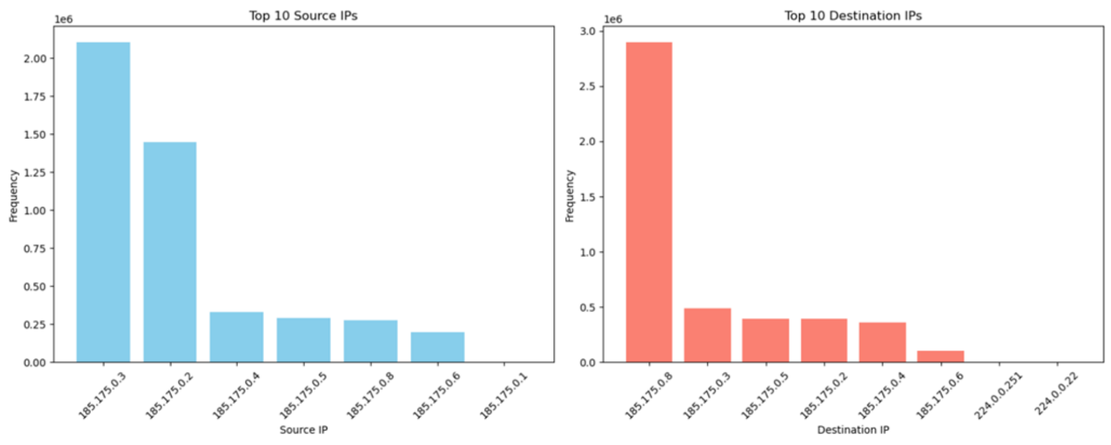


Figure 4.5: Top 10 Source and Destination IP Addresses

- The second most frequent Source IP is **185.175.0.2**, with over 1.5 million occurrences.
- Other Source IPs (**185.175.0.4**, **185.175.0.5**, **185.175.0.8**, etc.) have significantly lower frequencies, indicating they are less involved in the dataset's network activities.

Top 10 Destination IPs (Right Plot)

- The most frequent Destination IP is **185.175.0.8**, with nearly 3 million occurrences, making it the most targeted or interacted destination in the dataset.
- The second most frequent Destination IP is **185.175.0.3**, with fewer occurrences compared to **185.175.0.8**.
- Other Destination IPs (**185.175.0.2**, **185.175.0.4**, **224.0.251**, etc.) appear with much lower frequencies.

4.13 Top 10 Source and Destination Ports

This figure consists of four subplots, providing insights into the frequency distribution of source and destination ports in benign and attack traffic:

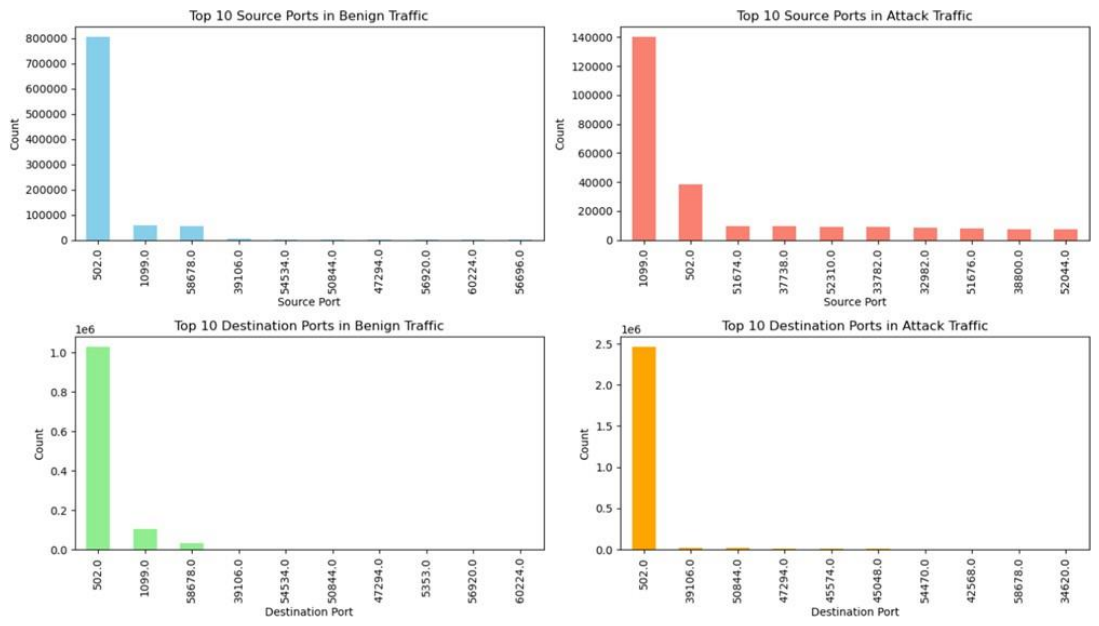


Figure 4.6: Top 10 Source and Destination Ports

Top Left: Source Ports in Benign Traffic

- Port **502.0** dominates benign traffic with nearly 800,000 occurrences, reflecting its frequent use, possibly for Modbus communication.
- Other source ports, such as **1099.0** and **58678.0**, are much less frequent, indicating secondary roles in benign communication.

Top Right: Source Ports in Attack Traffic

- Port **1099.0** is the most common source port in attack traffic, with over 140,000 occurrences, suggesting its potential targeting or exploitation.
- Other ports, including **502.0** and **51674.0**, are also present but with significantly lower frequencies, indicating a diverse attack profile.

Bottom Left: Destination Ports in Benign Traffic

- Similar to the source ports in benign traffic, port **502.0** is overwhelmingly the most frequent destination port, with over 1 million occurrences.
- Secondary destination ports like **1099.0** and **58678.0** appear with minimal usage.

Bottom Right: Destination Ports in Attack Traffic

- Port **502.0** again dominates as the destination port in attack traffic, with over 2.5 million occurrences, reinforcing its importance in the dataset as a critical Modbus communication endpoint.
- Other destination ports, such as **39106.0** and **50844.0**, are present in smaller quantities, hinting at additional targeted services.

4.14 Distribution of Data Length

Below histogram displays the Data Length Distribution in benign versus attack traffic, comparing the frequency of different packet sizes.

Key Observations:

- **Dominance of Small Data Lengths:** Most packets, both in benign and attack traffic, have data lengths concentrated in the lower range (below 100).
- **Slight Variation in Distributions:** While the distributions of benign and attack traffic overlap significantly, attack traffic shows additional activity in slightly different ranges, possibly indicating anomalous or malicious patterns in data length.

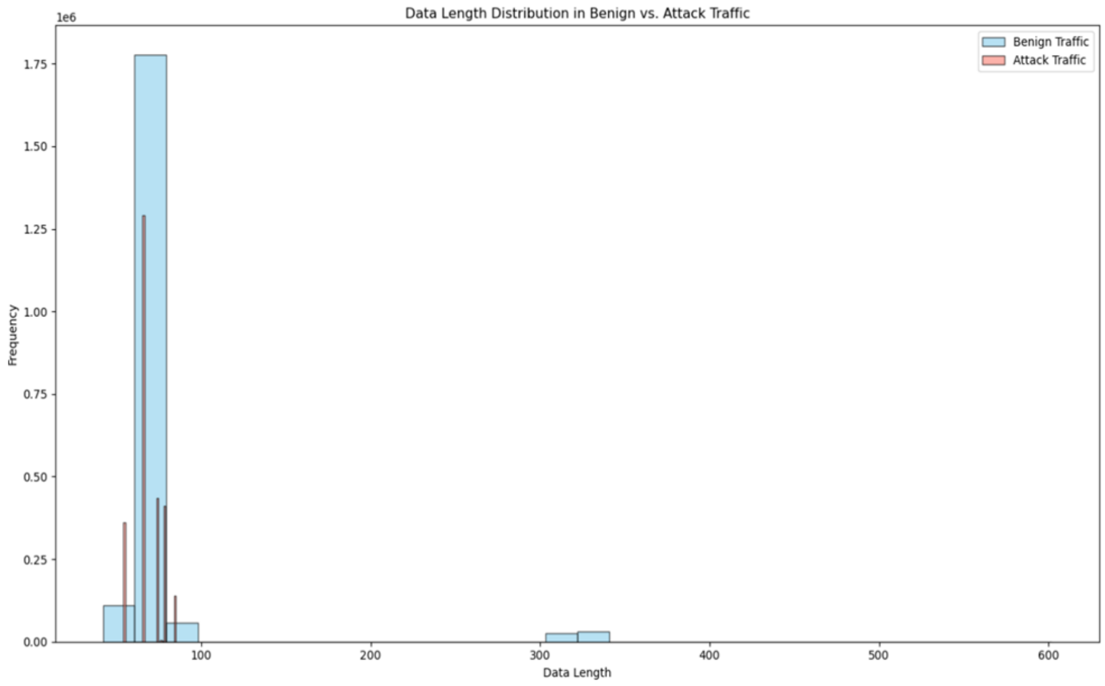


Figure 4.7: Distribution of Data Length

- **Rare Larger Data Lengths:** Both benign and attack traffic show minimal occurrences of packets with larger data lengths (above 300). These could correspond to less common operations or specialized communication.

4.15 Time-Based Analysis of Benign vs. Attack Types

This line chart compares the volume of benign and attack-related network traffic over time.

Observations:

- **Blue Line (Benign Traffic):** Consistently high around the 30,000 mark, with slight fluctuations throughout the day.
- **Red Line (Attack Traffic):** Significantly lower compared to benign traffic, with occasional spikes (e.g., around 12:00), generally remaining below 5,000.

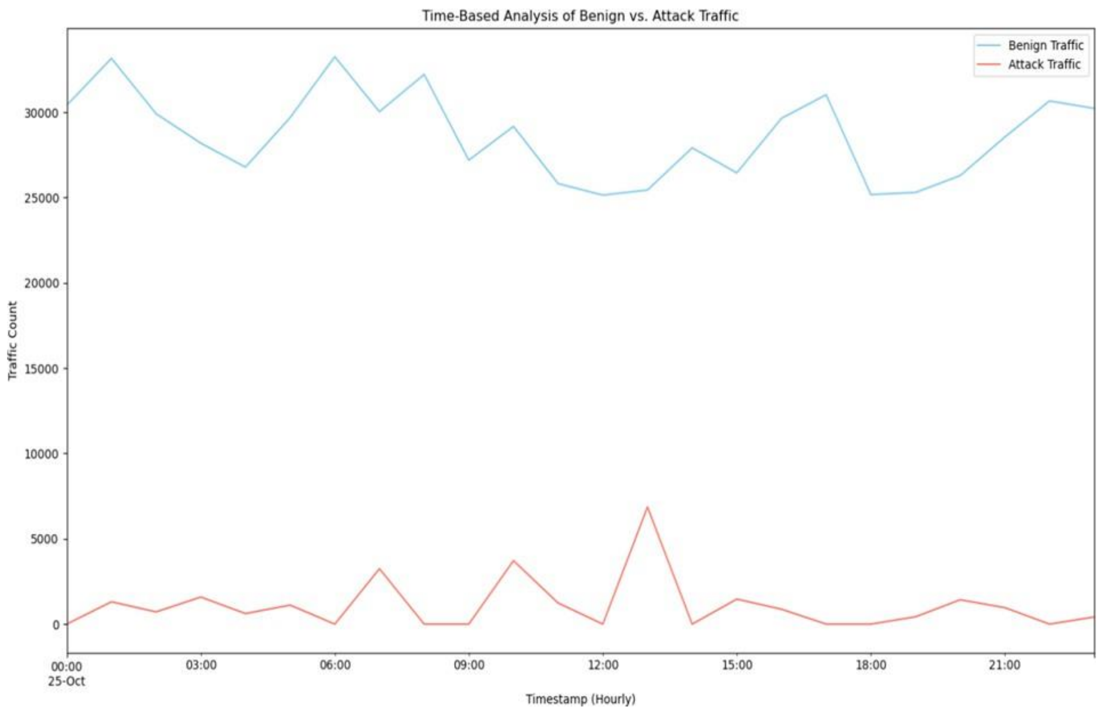


Figure 4.8: Time based analysis of benign vs Attack types

4.16 Splitting Data into Training and Testing Phases

The dataset was divided into two parts:

- **Training Data (70%):** Used to train the model to learn patterns and behaviors.
- **Testing Data (30%):** Reserved to evaluate the model's performance on unseen data.

This approach ensures a fair evaluation of the model's ability to generalize to new, unseen data.

4.16.1 Model Building

In our project, we utilized four machine learning models as base learners in an ensemble learning framework. Below are the base models and their roles:

- **Support Vector Machine (SVM):** Finds the optimal decision boundary to separate benign and attack traffic. It is effective in handling high-dimensional data.
- **Random Forest (RF):** An ensemble of decision trees that captures complex relationships and handles class imbalance effectively.
- **K-Nearest Neighbors (KNN):** Classifies data points based on their similarity to the nearest neighbors, making it useful for detecting anomalies.
- **Adaptive Boosting (AdaBoost):** Focuses on correcting misclassified instances by assigning higher weights to them, improving detection of subtle or rare attack patterns.

Base Models Training

The steps followed to build and train the base models were:

- **SVM:** Utilized with a linear kernel to find the optimal hyperplane for classifying benign and attack traffic.
- **Random Forest:** Built as an ensemble of decision trees to handle complex relationships and class imbalances.
- **KNN:** A similarity-based classifier that classifies instances based on their nearest neighbors.
- **AdaBoost:** An iterative boosting method that improves performance by focusing on misclassified instances.

Each base model was trained independently on the training set, and performance metrics such as accuracy, confusion matrix, and classification report were calculated to evaluate their individual contributions.

Stacking Ensemble Approach

To further enhance predictive performance, we implemented a **StackingClassifier**, which combines the predictions of the base models.

- **Base Models:** SVM, Random Forest, KNN, and AdaBoost provided predictions as input for the meta-model.
- **Meta-Model:** Logistic Regression was used to learn the best combination of predictions from the base models.
- **Cross-Validation:** 5-fold cross-validation was used during training to reduce overfitting and ensure robust performance.

Chapter 5

Results and Analysis

Base Models Performance:

The following models were tested: KNN, SVM, AdaBoost, and Random Forest. The following table illustrates the accuracy, in percent, achieved by each of these models on the test dataset, separately:

Table 5.1: Accuracy Comparison of Models

Model	**Accuracy (%)**
KNN	99.54
SVM	99.34
AdaBoost Models	98.83
Random Forest	99.52
Stacking Classifier	99.78

- **KNN (99.54%):** It had the highest accuracy of the base models. It suggests that the KNN model has correctly captured the relationship between the data points that is playing a critical role in the stacking ensemble.
- **SVM (99.34%):** SVM with a linear kernel was slightly worse than KNN but robust in any case. It showcases the capability of SVM to find a hyperplane that can distinguish benign from attack traffic very effectively.

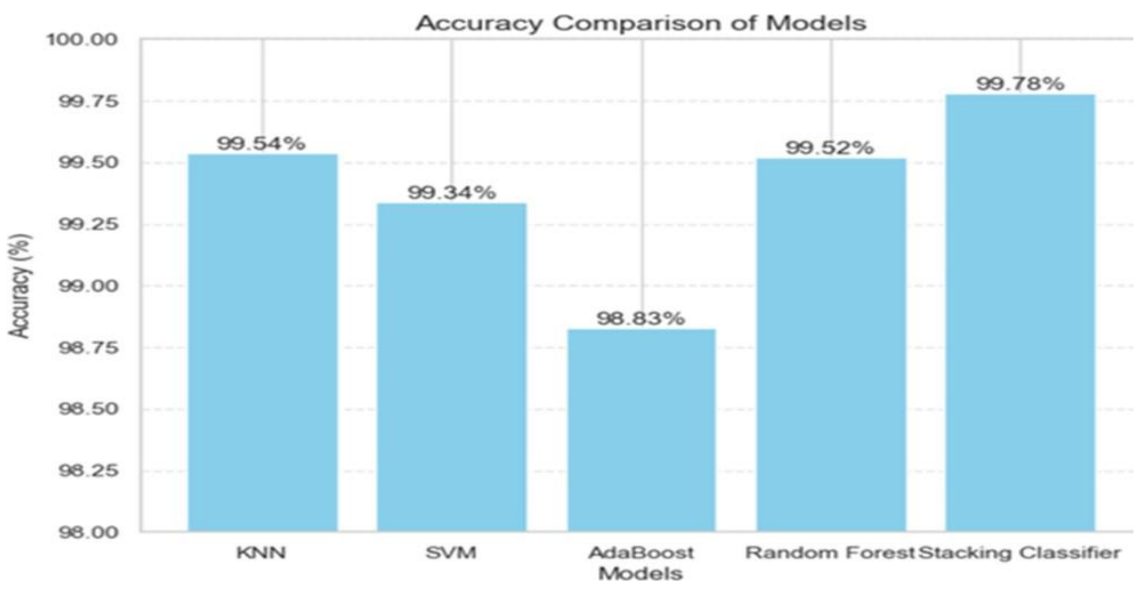


Figure 5.1: Accuracy comparison of base models

- **AdaBoost (98.83%):** It performed slightly low than other models, due to the nature of iterative boosting in AdaBoost which is a bit sensitive to noise present in the dataset.
- **Random Forest (99.52%):** It performed comparable to KNN, thereby showing that the ensemble of decision trees is efficient to handle the complexity of the dataset.
- **Stacking Classifier (99.78%):** The stacking classifier performed better than any of the individual base models. This leads to the conclusion that there exists an effective encapsulation in this particular meta-model, Logistic Regression, of the optimal integration that occurred from the synthesis of these predictions.

Practical Implication

The stacking classifier is very reliable for use in network intrusion detection in addition to increased accuracy. Fundamentally, this has made stacking methodology stronger than the employment of a single model. The figure shows the role of every basic model used in the stacking methodology and its contribution to improving overall accuracy.

5.1 Comparison of Precision, Recall, and F1 Score

The table below, **"Comparison of Precision, Recall, and F1-Score Across Models,"** compares the evaluation metrics for the different base models used as well as the stacking classifier in our intrusion detection system. These metrics—Precision, Recall, and F1-Score—are extremely important in determining the effectiveness of a classification model, especially in domains like intrusion detection where a false positive and a false negative result in high damage.

Table 5.2: Comparison of Precision, Recall, and F1-Score Across Models

Model	**Precision**	**Recall**	**F1-Score**
KNN	0.77	0.76	0.76
SVM	0.55	0.50	0.50
AdaBoost Models	0.10	0.14	0.12
Random Forest	0.77	0.78	0.78
Stacking Classifier	0.90	0.89	0.89

The Stacking Classifier obtains the best performances, with excellent precision and recall and an F1-score very close to 0.90. This is a very good demonstration of how this stacking method may enhance the quality of intrusion detection in a Modbus network.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Our findings indicate that with connectivity through Modbus in an industrial network, the stacking-based intrusion detection system, which utilizes KNN, SVM, AdaBoost, and Random Forest models, significantly improves the precision of detection. The sys- tem achieves remarkable values for precision, recall, and F1-scores, with an outstanding accuracy of 99.78%.

It proves to be highly efficient for detecting intrusions related to Modbus, while also being adaptive to emerging threats. This highlights its potential for improved feature se- lection and preprocessing techniques, further enhancing its detection capabilities. Future research in this area could lead to the development of a scalable and reliable security framework tailored for industrial networks.

6.2 Future Work

- **Extend Dataset:** Collect more diversified data to build a more accurate and flexible model that can generalize effectively to various scenarios.
- **Real-Time Detection:** Evaluate the intrusion detection system in real-time environments, enabling immediate action against potential attacks.
- **Support More Protocols:** Expand the compatibility of the intrusion detection system to include additional industrial protocols, such as DNP3 and OPC-UA.
- **Adaptive Learning:** Continuously improve the intrusion detection system through adaptive learning, incorporating newly discovered patterns of attacks into the model.

Chapter 7

References

[1] D. Alsalman, "A comparative study of anomaly detection techniques for IoT se- curity using adaptive machine learning for IoT threats," IEEE Access, vol. 12, pp. 14719–14730, Mar. 2024.

[2]C. Parian, T. Guldimann, and S. Bhatia, “Fooling the master: Exploiting weaknesses in the modbus protocol,” Proc. Comput. Sci., vol. 171, pp. 2453–2458, Jan. 2020.

[3] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning," IEEE Access, vol. 9, pp. 75729–75740, May 2021.

[4]M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, “Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recom- mendations,” Int. J. Crit. Infrastructure Protection, vol. 38, Sep. 2022.

[5]M. Ring, S.Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “Asurvey of network- based intrusion detection data sets,” Comput. Secur., vol. 86, pp. 147–167, Sep. 2019.

[6]T. Alladi, V. Chamola, and S. Zeadally, “Industrial control systems: Cyberattack trends and countermeasures,” Comput. Commun., vol. 155, pp. 1–8, Apr. 2020.

[7] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," IEEE Access, vol. 7, pp. 82512–82521, Jun. 2019.

[8]F. A. Md. Zaki and T. S. Chin, “FWFS: Selecting robust features towards reliable and

stable traffic classifier in SDN,” IEEE Access, vol. 7, pp. 166011–166020, 2019.

[9] J. Li, X. Tong, J. Liu, and L. Cheng, "An efficient federated learning system for network intrusion detection," IEEE Systems Journal, vol. 17, no. 2, pp. 2455–2464, Jun. 2023.

[10]Z.-H. Zhou and X.-Y. Liu, “On multi-class cost-sensitive learning,” Comput. Intell., vol. 26, no. 3, pp. 232–257, 2010

[11] K. Samunnisa, G. Sunil Vijaya Kumar, K. Madhavi, “Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods,” Measurement: Sensors, vol. 25, pp. 100612, 2023.

[12] M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, and I. Berrada, "Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems," Expert Systems with Applications, vol. 213, p. 121000, 2023

[13] L. Zhang, X. Liu, and Y. Li, “Federated learning for privacy-preserving intrusion detection systems in the Internet of Things,” Journal of Network and Computer Applications, vol. 164, pp. 102676, Apr. 2023.

[14] R. Sharma, A. Kumar, and S. R. Biradar, “Federated learning for network intrusion detection: A survey and future directions,” Journal of Computer Networks and Communications, vol. 2021, pp. 1-14, Jul. 2021.

[15] S. Duque and M. N. Omar, "Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS)," International Journal of Computer Applications, vol. 109, no. 13, pp. 15-19, Jan. 2023.

[16] N. Sharma and P. Singh, "Random Forest Modeling for Network Intrusion Detection System," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, pp. 45–56, Mar. 2025.

[17] E. Ozdogan, “On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things,” IEEE Access, vol. 12, pp. 34567–34578, Jul. 2025.

[18]M. A. Ferrag, L. Maglaras, and H. Janicke, "A Machine Learning-Based Intrusion

Detection for Detecting Internet of Things Network Attacks," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9042–9053, Dec. 2019

[20] A. Gupta, S. M. Gupta, and S. P. Singh, “Federated machine learning for intrusion detection in IoT-based smart cities,” Wireless Communications and Mobile Computing, vol. 2021, pp. 1-12, Nov. 2021.

[21] K. Samunnisa, G. Sunil Vijaya Kumar , K. Madhavi, “Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods,” Measurement: Sensors 25 (2023) 100612.

[22] Shaashwat Agrawal , Sagnik Sarkar , Ons Aouedi , Gokul Yenduri , Kandaraj Piamrat ,Mamoun Alazab, Sweta Bhattacharya , Praveen Kumar Reddy Maddikunta ,Thippa Reddy Gadekallu, “Federated Learning for intrusion detection system: Concepts, challenges and future directions,” Computer Communications 195 (2022) 346–361.

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

[24] Xabier Sáez-de-Cámara , Jose Luis Flores, Cristóbal Arellano, Aitor Urbieta, Urko Zurutuza, "Clustered fe derate d learning architecture for network anomaly detection in large scale heterogeneous IoT networks," Computers Security 131 (2023) 103299.

[25] Milan Samantaray, Ram Chandra Barik, Anil Kumar Biswal, “A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems,” Decision Analytics Journal 11 (2024) 100478.

[26] Aitor Belenguer, Jose A. Pascual, Javier Navaridas , “GöwFed : A novel federated network intrusion detection system,” Journal of Network and Computer Applications 217 (2023) 103653.

[27] Danish Javeed , Muhammad Shahid Saeed , Muhammad Adil , Prabhat Kumar ,Alireza Jolfaei, " A federated learning-based zero trust intrusion detection system for Internetof Things," Ad Hoc Networks 162 (2024) 103540.

[28] Sreekanth Vadigi , Kamalakanta Sethi, Dinesh Mohanty , Shom Prasad Das ,Padmalochan Bera , " Federated reinforcement learning based intrusion detection system

using dynamic attention mechanism," Journal of Information Security and Applications 78 (2023) 103608.

[29] Sreekanth Vadigi , Kamalakanta Sethi , Dinesh Mohanty , Shom Prasad Das , Padmalochan Bera , "A comprehensive review of AI based intrusion detection system," Measurement: Sensors 28 (2023) 100827.

[30] Ayesha Rahman , Ghulam Mustafa , Abdul Qayyum Khan , Muhammad Abid ,Muhammad Hanif Durad , Launch of denial of service attacks on the modbus/TCP protocol and development of its protection mechanisms, international journal of critical infrastructure protection 39 (2022) 100568.

[31] E. Ozdogan, "A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems," IEEE Access, vol. 12, pp. 46785–46809, Apr. 2024.

[32] M. He, Y. Huang, X. Wang, P. Wei, and X. Wang, "A Lightweight and Efficient IoT Intrusion Detection Method Based on Feature Grouping," IEEE Internet Things J., vol. 11, no. 2, pp. 2935–2949, Jan. 2024.

[33] O. Alghushairy, R. Alsini, Z. Alhassan, A. A. Alshdadi, A. Banjar, A. Yafoz, and X. Ma, "An Efficient Support Vector Machine Algorithm Based Network Outlier Detection System," IEEE Access, vol. 12, pp. 24428–24441, Feb. 2024.

[34] M. M. Alani and A. I. Awad, "An Intelligent Two-Layer Intrusion Detection System for the Internet of Things," IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 683–692, Jan. 2023.

[35] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," IEEE Access, vol. 11, pp. 80348–80362, Aug. 2023.

[36] N. Tran, H. Chen, J. Bhuyan, and J. Ding, "Data Curation and Quality Evalua- tion for Machine Learning-Based Cyber Intrusion Detection," IEEE Access, vol. 10, pp. 121900–121920, Oct. 2022.

[37] N. Tran, H. Chen, J. Bhuyan, and J. Ding, "Data Curation and Quality Evalua- tion for Machine Learning-Based Cyber Intrusion Detection," IEEE Access, vol. 10, pp.

121900–121920, Oct. 2022.

[38] E. Chatzoglou, G. Kambourakis, C. Kolias, and C. Smiliotopoulos, "Pick Quality Over Quantity: Expert Feature Selection and Data Preprocessing for 802.11 Intrusion Detection Systems," IEEE Access, vol. 10, pp. 64761–64777, June 2022

[39] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security," Big Data Mining and Analytics, vol. 6, no. 3, pp. 273–287, September 2023.

[40] Siddhartha Deb Roy, Graduate Student Member IEEE. Sanjoy Debbarma. Senior Member IEEE. Adnan Iqbal, A Decentralized Intrusion Detection System for Security of Generation Control, IEEE Internet Of Things Journal, VOL. 9, NO. 19, 1 October 2022

Activity Chart

Activity	Duration in Days	End Date	Month 2024								
			Feb	Mar	Apr	May	Jun	Jul	Oct	Nov	Dec
Team Formation project proposal/Problem identification	14	29/01/2024	🟦								
Literature Review and study of existing systems.	10	29/02/2024		🟦							
Synopsis Submission and Presentation	6	01/03/2024		🟦							
System Requirements specification	10	30/03/2024			🟦						
Project Design Phase Evaluation	20	20/04/2024				🟦					
Project Phase-I Report Preparation	10	01/05/2024					🟦				
Coding 50% completion	20	22/05/2024					🟦				
Project Phase-I Evaluation	5	28/05/2024					🟦				
Self-Study on relevant technology	7	05/06/2024						🟦			
100% Coding	30	26/10/2024							🟦		
Writing Research Paper	25	10/11/2024								🟦	
Prepare and Publish Research Paper on H Index Journal	15	27/11/2024								🟦	
Report Preparation	10	29/11/2024									🟦
Final Report Preparation and Submission	1	01/12/2024									🟦
Final Presentation	1	07/12/2024									🟦

IDS

ORIGINALITY REPORT

7%

SIMILARITY INDEX

PRIMARY SOURCES

1	www.coursehero.com Internet	134 words — 2%
2	ijarcet.org Internet	63 words — 1%
3	ir.juit.ac.in:8080 Internet	37 words — < 1%
4	www.mdpi.com Internet	25 words — < 1%
5	link.springer.com Internet	21 words — < 1%
6	scholarworks.aub.edu.lb Internet	20 words — < 1%
7	u-aizu.repo.nii.ac.jp Internet	20 words — < 1%
8	ijeer.forexjournal.co.in Internet	17 words — < 1%
9	Norberto Garcia, Tomas Alcaniz, Aurora González-Vidal, Jorge Bernal Bernabe, Diego Rivera, Antonio Skarmeta. "Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial	16 words — < 1%

Intelligence", Journal of Network and Computer Applications, 2020
Crossref

10	atharvacoe.ac.in Internet	15 words — < 1%
11	kipdf.com Internet	15 words — < 1%
12	www.e3s-conferences.org Internet	15 words — < 1%
13	www.ijisae.org Internet	14 words — < 1%
14	ijsrcseit.com Internet	13 words — < 1%
15	jad.shahroodut.ac.ir Internet	13 words — < 1%
16	cs.nju.edu.cn Internet	12 words — < 1%
17	repositorio.ufmg.br Internet	12 words — < 1%
18	thesai.org Internet	12 words — < 1%
19	www.frontiersin.org Internet	12 words — < 1%
20	www.geeksforgeeks.org Internet	12 words — < 1%

21	Quang Huy Hoangp, Phuong Linh Nguyenp, Tien Dat Do, Anh Vu Tran, Thuy Anh Nguyen, Thanh Trung Nguyen, Viet Dung Nguyen. "Leukemia Classification using Principal Component Analysis and Ensemble Learning on Gene Expression Data", 2023 International Conference on Advanced Technologies for Communications (ATC), 2023 Crossref	11 words — < 1%
22	prism.ucalgary.ca Internet	11 words — < 1%
23	www.int-arch-photogramm-remote-sens-spatial-inf-sci.net Internet	11 words — < 1%
24	G.P.C. Venkata Krishna, D. Vivekananda Reddy. "Machine learning-enhanced hybrid cryptography and image steganography algorithm for securing cloud data", Journal of Intelligent & Fuzzy Systems, 2023 Crossref	10 words — < 1%
25	Kai Xu, Zemin Li, Nan Liang, Fanchun Kong, Shaobo Lei, Shengjie Wang, Agyemang Paul, Zhefu Wu. "Research on Multi-Layer Defense against DDoS Attacks in Intelligent Distribution Networks", Electronics, 2024 Crossref	10 words — < 1%
26	Kunpeng Wang, Jingmei Li, Weifei Wu. "A novel transfer extreme learning machine from multiple sources for intrusion detection", Peer-to-Peer Networking and Applications, 2023 Crossref	10 words — < 1%
27	www.science.gov Internet	10 words — < 1%