I.J. Wireless and Microwave Technologies, **2025, 6, 54-70**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2025.06.05

Ensemble Learning-Based Intrusion Detection System for Modbus-Enabled Industrial Networks

Dadaso T. Mane
Department of Information Technology, Kasegaon Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: dadaso.mane@ritindia.edu
ORCID iD: https://orcid.org/0000-0003-0382-8789

Vijay H. Kalmani*
Department of Computer Science and Engineering, Kasegaon Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: vijaykalmani@gmail.com
ORCID iD: https://orcid.org/0000-0003-0738-3211
*Corresponding Author

Sayali Aundhakar
Department of Information Technology, Kasegaon Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: sayaliaundhkar2798@gmail.com
ORCID iD: https://orcid.org/0009-0003-3968-231X

Pranita Patil
Department of Information Technology, Kasegaon Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: pranitapatil502@gmail.com
ORCID iD: https://orcid.org/0009-0006-3840-0044

Swati Patil
Department of Information Technology, Kasegaon Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: swatipatil3034@gmail.com
ORCID iD: https://orcid.org/0009-0002-4611-9923

Tejal Yadav
Department of Information Technology, Kasegoan Education Society's, Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, Maharashtra 415415, India
E-mail: tejaly319@gmail.com
ORCID iD: https://orcid.org/0009-0008-0186-240X

Received: 22 January, 2025; Revised: 01 May, 2025; Accepted: 26 June, 2025; Published: 08 December, 2025

Abstract: Industrial Control Systems (ICS) and Modbus-enabled networks are facing escalating threats from sophisticated cyber-attacks, while current Intrusion Detection Systems (IDS) struggle to identify intricate and adaptive attacks. This study envisions an ensemble learning-based IDS for Modbus-enabled industrial networks using a real-like Modbus 2023 dataset for industrial networks. The proposed IDS combines four base classifiers, namely K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and Adaptive Boosting (AdaBoost), using the stack ensemble framework, where Logistic Regression acts as the meta-classifier. Preprocessing involved PCAP capture and attack log synchronization, feature normalization, and one-hot encoding for balanced and accurate model training. Experimental evaluation demonstrated that the ensemble model has a 99.78% detection accuracy while outperforming the base individual models in terms of precision, recall, and F1-score. The results indicate the efficiency of ensemble learning for enhanced accuracy detection and false-positive reduction for Modbus networks. Future research will consider real-time testing, feature elimination, and explainable AI for higher operational deployment and scalability.

Index Terms: Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA), Intrusion Detection System (IDS), Machine Learning, Ensemble Learning, Support Vector Machine (SVM), Random Forest, Adaptive boosting, K-Nearest Neighbors, Stacking Classifier

1. Introduction

Recent studies have highlighted the increased cybersecurity threats to Industrial Control Systems (ICS) and Modbus-enabled networks beyond 2022. The integration of Internet of Things (IoT) devices with ancient ICS technology has increased the attack surface area while introducing these systems to highly advanced threats such as ransomware and Advanced Persistent Threats (APTs) [1]. Legacy infrastructure in ICS often uses ancient technologies and protocols, such as Modbus, which are highly susceptible and lack the security capabilities required to counter modern-day cyber threats effectively [2]. Classic Intrusion Detection Systems (IDS) and firewalls have several inadequacies in this scenario. Classic IDS frequently produce a large number of false positives, such that security staff experience alarm fatigue, often causing genuine threats to be bypassed unknowingly [3]. In addition, these systems rely primarily on static rules or signatures for threat detection, rendering them less useful in the face of new attack plans that have yet to be indexed elsewhere [4]. Most IDS products also have no comprehension of the dedicated industrial protocols and behaviors that exist in ICS applications, such that their signature detection capabilities are hindered in these environments when it comes to discovering complicated malicious activities [2,5]. Considering these inadequacies, Machine Learning (ML) and ensemble-based IDS techniques have become critical solutions. ML models have adaptive learning patterns, such that new patterns and strange occurrences are detected easily and effectively, which is very important in the context of fast-paced industrial environments [6]. Anomaly detection using ML provides the possibility of baselining regular operational patterns, such that strenuous attempts are detected that may otherwise evade traditional methods [7]. Ensemble methods that combine several models offer enhanced accuracy for bout detection while minimizing the number of false positives by exploiting the advantages of all the varied algorithms included [8]. These ML-infused systems also have real-time processing capabilities that are required for maintaining the security of fast-paced ICS environments [9].

Based on this motivation, the current research proposes a new Ensemble Learning-Based Intrusion Detection System (IDS) designed for Modbus-enabled industrial networks. Unlike previous studies that mainly address generic IoT or IT datasets, this study uses the CIC Modbus 2023 dataset, which simulates real industrial communication traffic and a variety of Modbus-specialized attacks, such as payload injection, query flood, and delay-response attacks. The new model combines four underlying classifiers--Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Adaptive Boosting (AdaBoost)--in a stacking ensemble framework using Logistic Regression as the meta-classifier. This ensemble setup takes advantage of the complementary capabilities of various algorithms to achieve higher accuracy and generality in the detection of intrusions. Experimental work validates that the proposed stacking ensemble IDS attains a very high 99.78% detection accuracy, which surpasses individual models in precision, recall, and F1-score measures. In addition to enhanced Modbus-specialized security, this research work contributes a scalable and computationally light setup amenable for real-time industrial deployment.

To guide this work, the subsequent primary research queries are answered to closely align the goals with the proposed ensemble-based IDS framework:

1. How can ensemble learning methods be streamlined to detect Modbus protocol–specific attacks in industrial control networks?
2. The improvements in accuracy, precision, and false-positive rate that may be obtained by using a stacking ensemble model versus individual classifiers such as RF, KNN, SVM, and AdaBoost.
3. How efficient will the proposed framework be in scaling for real-time execution in Modbus-based ICS environments with minimal computational capabilities?

The remainder of this paper is organized as follows. Section 2 discusses related work on machine learning, deep learning, and Modbus-specialized IDS frameworks. Section 3 details the CIC Modbus 2023 dataset and its preprocessing. Section 4 details the proposed ensemble stacking methodology. Section 5 presents the experimental results and analysis. Section 6 concludes with key insights and future work.

2. Literature Survey

Recent studies have established that the most popular algorithms for machines, such as Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost, are increasingly utilized to penetrate ICS and SCADA networks. The analysis of the algorithms on several datasets established their high detection ability and resilience in detecting intrusions. For example, a comparison between RF and SVM using a SCADA gas pipeline dataset provided by Mississippi State University proved that the Random Forest model produced a high F1 score of

over 99% for a very high level of reliability during intrusion detection [10]. Likewise, the Decision Tree (DT) and KNN models demonstrated impressive performance in SCADA networks when the models were assessed for accuracy, training time, and prediction time using publicly sourced cybersecurity datasets [11]. These results further attest to the applicability of ML algorithms for ICS networks while maintaining the limitations of dealing with zero-day attacks, real-time adaptation, and managing computational overhead [12]. To resolve these inadequacies, the utilization of deep and hybrid learning paradigms has been increasingly studied. Models such as BiLSTM-CNN [13], CNN-LSTM [14], and DL-IDS [15] have achieved high accuracy through joint temporal–spatial feature extraction while significantly minimizing false positives using datasets such as UNSW-NB15, NSL-KDD, and CICIDS2017. These developments together create a strong foundation for the building of adaptive and intelligent intrusion detection networks while also establishing the necessity for methods better suited for industry protocols such as Modbus.

A list of preliminary notable machine learning and hybrid solutions for ICS and SCADA environments is presented in Table 1.

Table 1. Overview of Machine Learning and Hybrid IDS Methods Applied in ICS/SCADA Networks

Technique / Model	Dataset Used	Performance Highlights	Key Observations / Limitations
Random Forest (RF), Support Vector Machine (SVM) [10]	SCADA gas pipeline dataset (Mississippi State Univ.)	RF achieved F1 > 99%	High detection accuracy but lacks real-time adaptability.
Decision Tree (DT), KNN [11]	Public cybersecurity datasets	Efficient accuracy and prediction speed	Limited robustness for zero-day attacks.
Comparative ML study (RF, SVM, KNN) [12]	Multiple ICS datasets	High detection precision	Computational overhead and difficulty in real-time use.
BiLSTM-CNN Hybrid [13]	UNSW-NB15	High accuracy, low false positives	Complex architecture; higher training cost.
CNN-LSTM Hybrid [14]	KDD CUP99, NSL-KDD, UNSW-NB15	High accuracy; adaptable to IIoT	Dataset dependence; limited Modbus evaluation.
DL-IDS (CNN-LSTM with category weight optimization) [15]	CICIDS2017	98.67% overall accuracy	Needs validation in ICS/SCADA context.

Further development of these advancements has led to ensemble and stack-based techniques becoming the primary research interest for enhanced robustness, accuracy, and generalization capability. A bagging and boosting hybrid technique using SHAP-based feature selection showed a top accuracy of 98.47% and 96.19% F1-score by enhancing interpretability and performance for high-dimensional data [16]. Similarly, a stacked classifier using weight-based feature selection mechanisms achieved 99.87% accuracy and 99.88% precision for the NSL-KDD and CSE-CIC-IDS2018 datasets by defeating conventional stack-based algorithms using high-performance base-learner selection [17]. The Self-Adaptive Stacking Ensemble Model (SSEM) improved accuracy and recall using genetic algorithm-based optimization [18], whereas ensemble strategies using varied feature selection and stochastic combination enhanced opponent perturbation resilience [19]. Concurrently, some studies have also ventured towards anomaly detection under industrial protocols such as Modbus and DNP3. Studies using Modbus-based SCADA datasets have proven that rule-based, ML, and semantic inspection methods can successfully detect intricate replay and multi-command attacks [20–22]. Despite this, key challenges include the absence of standardized Modbus datasets that hinder research replication, limited detection of progressive threats owing to accuracy limitations, and excessive computational overhead for real-time applications [7,23,24]. A comparison tabulation of ensemble and Modbus-specific IDS research work is presented in Table 2.

Apart from these ensemble solutions, the latest research focuses on lightweight, adaptive, and federated IDSs that are specially developed to perform the best in edge computing and IIoT environments. Privacy-respecting federated learning mechanisms, such as the Privacy Protection Scheme for Federated Learning under Edge Computing (PPFLEC), achieved a 40% boost in performance over conventional mechanisms [25], while hierarchical federated models reduced the risk of privacy leakage and enhanced the throughput for the detection of anomalies in IIoT devices [26]. Lightweight Collaborative Intrusion Detection Systems (CIDS) that blend federated learning and blockchain achieved 97.65% accuracy and 98.81% precision using the CICIoT2023 dataset [27]. Furthermore, blockchain-assisted schemes fortified authentication and safe sharing of data using hierarchical encryption [28], while the FIDChain architecture combined blockchain and federated learning for privacy-preserving IDS in IoT healthcare [29]. Each of these studies indicates a distinct shift in IDS research towards scalable, privacy-aware, and computationally lean solutions. Despite these studies, research that specifically focuses on Modbus-based ICS networks remains scarce, highlighting the need for a high-performance ensemble framework, such as that outlined in this study, which is capable of achieving a higher detection accuracy while retaining real-time industrial applicability.

Concurrently with federated learning strategies, ensemble learning persistently shows its efficacy in multiple fields as an effective method for enhancing generalizability and model stability. Recent ensemble learning research also supports the applicability of hybrid and stacking models in improving model stability and predictive precision. Complain [30] introduced an ensemble with an automated hyperparameter search based on the isolation forest for improving the detection of fraudulent activities, showcasing high adaptability and accuracy through the combination of several classifiers including KNN, Random Forest, and Logistic Regression. Another work [31] concerned ensemble-based models such as Random Forest, XGBoost, and Logistic Regression in predicting the global risk of diabetes,

realizing high efficiency and explicability with distributed learning. Another related work [32] presented a neural ensemble with the integration of Random Forest, XGBoost, Support Vector Regression, and Linear Regression in order to maximize the efficiency of big data distribution, revealing significant improvement in processing precision and computational efficiency. Analogously, a stacking ensemble model [33] with the integration of SVM, RF, KNN, XGBoost, and Logistic Regression reached a precision in the detection of heart attack up to 97.48%, supporting the applicability of ensemble techniques in the processing of heterogeneous and multi-source data. In summary, the papers collected share the importance of ensemble techniques in enhancing generalizability, scalability, and decision confidence in the fields of industry and healthcare.

Table 2. Comparative Analysis of Ensemble and Modbus-Specific IDS Models

Approach / Model	Dataset / Protocol	Performance	Limitations / Observations
Hybrid Bagging + Boosting with SHAP [16]	General IDS datasets	98.47% accuracy, 96.19% F1	Improved interpretability; not Modbus-specific.
Weighted Stacking Classifier [17]	NSL-KDD, CSE-CIC-IDS2018	99.87% accuracy, 99.88% precision	Excellent accuracy; offline evaluation only.
Self-Adaptive Stacking Ensemble Model (SSEM) [18]	Nine benchmark datasets	Superior accuracy & recall	Needs domain-specific adaptation for ICS.
Diverse Feature Selection + Stochastic Aggregation [19]	General IDS datasets	High robustness	Limited validation in industrial environments.
ML & rule-based Modbus IDS [20–22]	Modbus SCADA datasets	Effective replay & multi-command detection	Dataset scarcity; computational complexity.
Ensemble-based IDS for industrial networks [7,23,24]	Multiple ICS datasets	Improved detection efficiency	No standardized Modbus dataset; limited real-time testing.

2.1 Research Gaps

Recent work on Intrusion Detection Systems (IDS) has identified several open issues, especially for Modbus-centered research, dataset variety, ensemble stacking verification, and conformable real-time industrial testing. Despite the great advances in using machine and deep learning for industrial cybersecurity, several constraints hinder the adoption of efficient, scalable, and context-sensitive IDS solutions in real Modbus-enabled environments.

(i) *Lack of Modbus-Focused Studies*: The Modbus protocol, which has extensive applicability in Supervisory Control and Data Acquisition (SCADA) and industrial systems, has no intrinsic security mechanisms and is susceptible to a variety of assault vectors, such as spoofing, flooding, and replay. Although a few studies have simulated Modbus vulnerabilities and designed the placement of Intrusion Detection Systems (IDS) through deep packet inspection [21,34], there is a critical gap in the literature regarding Modbus-specific security solutions. Relatively few investigations have constructed efficient detection frameworks specifically designed for discovering Modbus network traffic or assessed their efficiency using prototype industrial datasets, representing a large gap in the prevention of protocol-oriented intrusion.

(ii) *Lack of Dataset Diversity*: A major limitation of IDS research is the absence of varied and complete datasets that characterize the full spectrum of attack varieties and operational environments found in current industrial networks. Datasets such as KDD 99, NSL-KDD, and UNSW-NB15 are still popular benchmarks but are unable to characterize the dynamic nature and protocol-specific issues of real-world industrial environments [2,35]. Most datasets lack Modbus-related traffic, new attack scenarios, or time-sensitive behaviors characteristic of operational control systems. For these inadequacies, it is imperative to develop and maintain new datasets that are current, large in size but representative, and that characterize realistic industrial communication patterns along with threat vectors for effective training and testing of models.

(iii) *Limited Ensemble Stacking Validation*: Although ensemble stacking methods that combine the predictions of several models have shown great promise for enhancing the accuracy and resilience of IDSs, real-world experimentation of such methods in operational or near-operational environments is lacking [36,37]. Most experiments are performed by offline evaluation using static datasets, which prevents insight into the behavior of ensemble models under fluctuating industrial workloads, latency requirements, and noise levels. Large-scale testing and cross-domain verification are necessary to evaluate the futurity, extensibility, and survivability of stacking-based ensemble IDS frameworks in fluctuating and diverse industrial systems.

(iv) *Absence of Real-Time Industrial Evaluation:* A serious limitation of current IDS research is the absence of real-time testing in live or near-live industrial environments. Most models are proven merely in simulation environments or pre-recorded datasets while disregarding the execution challenges of latency, throughput, and false alarm management that characterize real ICS scenarios [38,39]. A real-time test will is required to test its responsiveness and stability under realistic time constraints and traffic levels. It is crucial to develop effective mechanisms for the real-time deployment and tracking of IDS frameworks for industrial readiness and reliability.

A concise description of such recognized gaps, their implications, and the contributions of the proposed study towards their filling are presented in Table 3.

Table 3. Summary of identified research gaps and their relevance to the proposed study

Research Aspect	Observed Limitation / Challenge	Implications for IDS Research	Addressed in this Study
Modbus-Focused Studies [21,34]	Modbus lacks built-in security, leading to vulnerabilities like spoofing, flooding, and replay. Limited Modbus-specific IDS studies exist.	Lack of comprehensive frameworks leaves protocol-level threats unaddressed.	Targets Modbus-enabled networks using the CIC Modbus 2023 dataset and ensemble learning.
Dataset Diversity [2,35]	Commonly used datasets (KDD 99, NSL-KDD, UNSW-NB15) are outdated and lack industrial protocol data.	Limits IDS generalization and realism; lacks Modbus-related attack representation.	Utilizes CIC Modbus 2023 — a realistic, industrial dataset with diverse attack types.
Ensemble Stacking Validation [36,37]	Ensemble models rarely tested in real-world or diverse industrial networks.	Performance under variable workloads remains uncertain.	Implements and validates stacking ensemble (RF, KNN, SVM, AdaBoost + Logistic Regression) on Modbus data.
Real-Time Industrial Evaluation [38,39]	Most IDSs evaluated offline, lacking real-time industrial validation.	Reduces practical applicability; real-time behavior untested.	Proposes real-time Modbus detection framework designed for industrial deployment.

3. Research Motivation

The growing employment of Modbus-enabled industrial networks for key infrastructure avers the requirement for powerful IDS systems that move beyond the confines of traditional systems. Existing IDS solutions are incapable of discovering advanced and evolving attack patterns, produce several false positives, and are computationally expensive in resource-limited environments. Furthermore, such issues are further evidenced by the fact that datasets are outdated and threats particular to Modbus are insufficiently targeted, for instance, payload injection and query flooding. This study addresses these voids by using the CIC Modbus 2023 data and a superior machine learning method, ensemble learning using a stacking technique. The proposed IDS places all classifiers, such as Random Forest, K-Nearest Neighbors, Support Vector Machines, and Adaptive Boosting, all under a Logistic Regression meta-model that achieves state-of-the-art detection accuracy using robustness and real-time applicability. It guarantees superior feature selection and lower computation time using a scalable procedure for industrial network security.

4. Methodology

This paper aims to present a Machine Learning-oriented methodology for the detection and classification of intrusions in Modbus-enabled industrial networks. The methodology encompasses the following steps: gathering and appropriate preparation of the dataset for training the machine learning models; and utilization of various ensemble learning methods that attempt to aggregate the strengths of several algorithms in a single attempt to gain greater accuracy and higher reliability in the detection of intrusions. The introduced methodology can discover various types of Modbus protocol attacks and lower false positives by introducing greater robustness into the model. The following section details the steps and techniques adopted for the construction and testing of the proposed system.

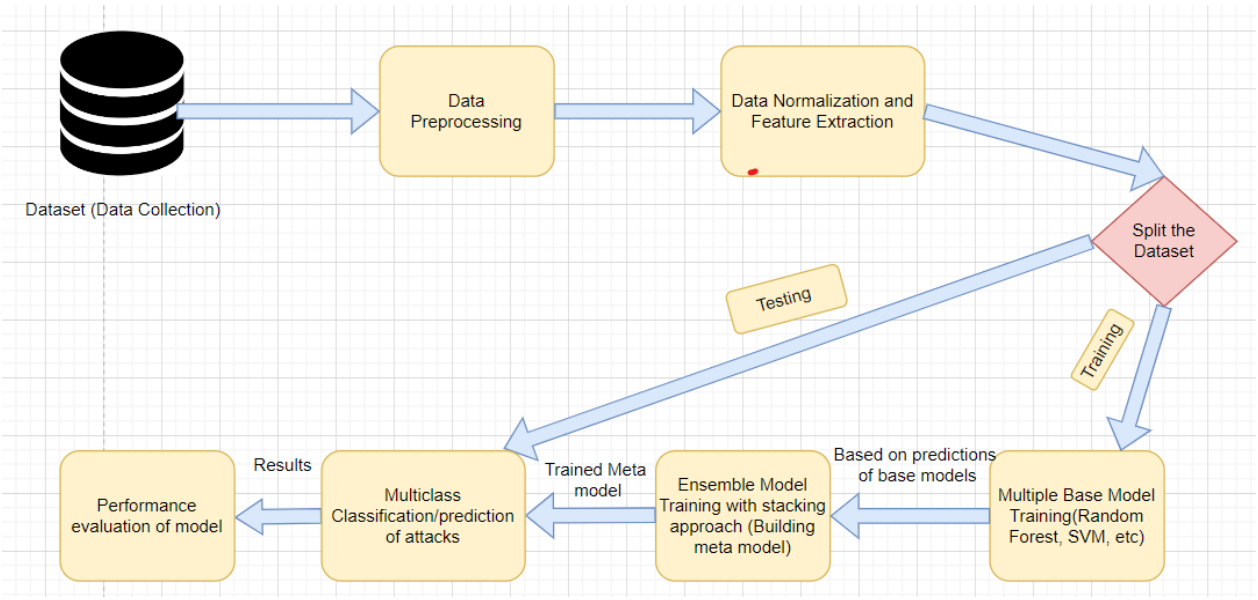

Fig. 1. Ensemble Stacking Approach for Modbus Attack Detection

Fig.1 shows the organized work style of the proposed ensemble stacking framework-based machine-learning-based intruder detection system (IDS). The procedure begins with data accumulation, followed by stringent preprocessing to make the data qualitative through cleansing, normalization, and feature encoding. The prepared dataset was divided into training and testing partitions to ensure the model evaluation was sound. Then, several distinct base leaners are trained autonomously from the training datasets, such that unique feature–pattern connections and choice borders are learned by them. Finally, the stacking layer or meta-learner is learned using the base model predictions as input features, such that the between-model dependencies are learned and the individual shortcomings are compensated for. This hierarchical ensemble technique improves the discriminative power, strength, and efficiency of the IDS in discovering and categorizing cyberattacks based on the Modbus protocol.

4.1 Novelty of the Research

Our work proposes a new way of escalating IDS techniques in industrial networks enabled by Modbus using a stacking-based ensemble learning technique, as shown in Fig.2. We could not find any previous studies that have attempted to employ the technique of stacking for the multi-class classification of industrial network attacks enabled by Modbus. It begins using a training set, training four varied base models: SVM, KNN, Random Forest, and AdaBoost. Each model ran a separate analysis of the data to identify possible attacks on Modbus. These models have come up with their individual predictions—P1, P2, P3, and P4—to each of which are captured varied patterns and understanding in the data. These individual predictions are given to a meta-classifier, for which this study applies a Linear Regression model. It takes the combined outputs of all base models and learns to combine their predictions in the most optimal manner.

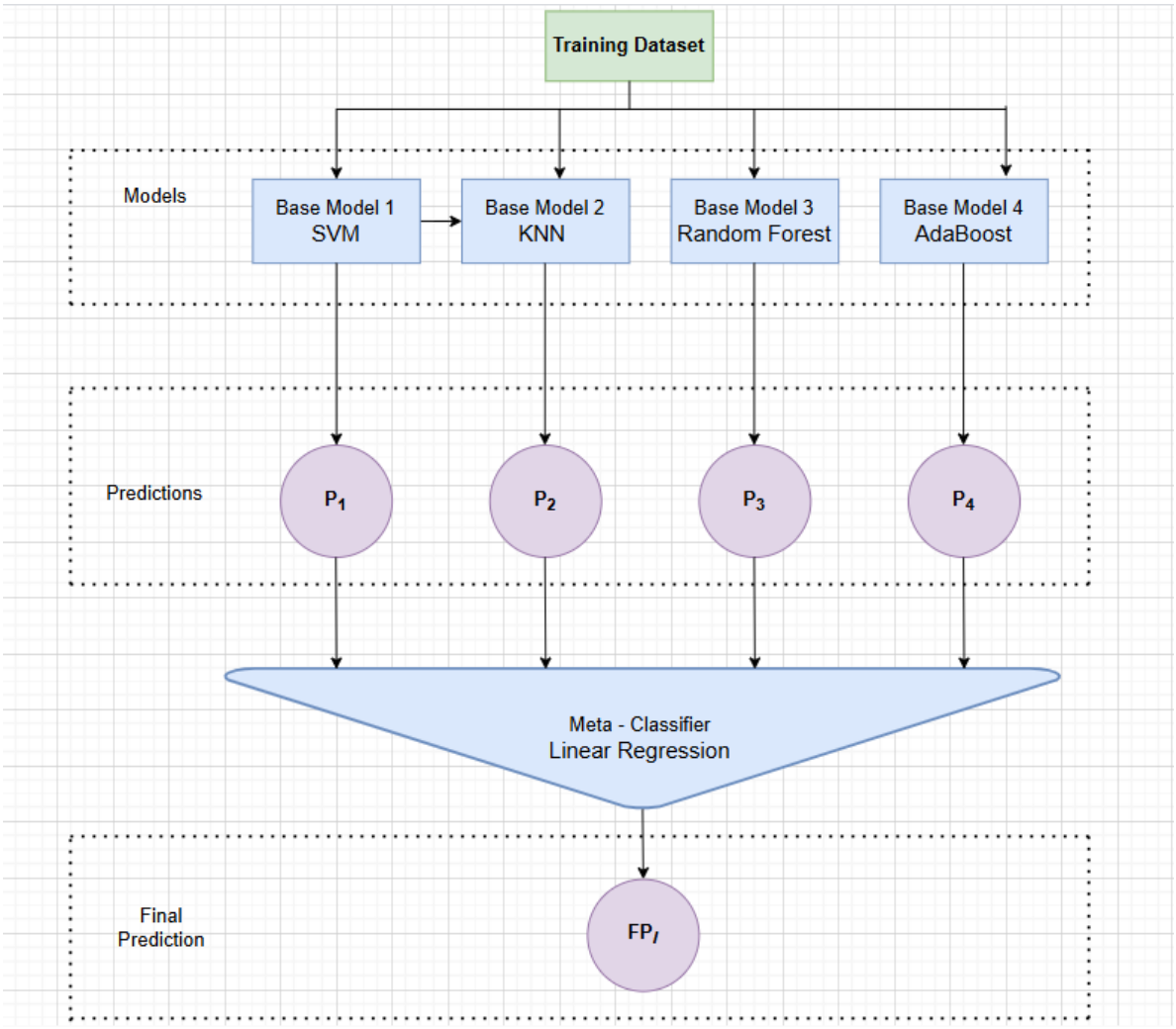

Fig. 2. Novel Stacking Ensemble Approach for Modbus Attack Detection in Industrial Networks

The overall outcome is a stack of these base models that come up with a single final guess that is more accurate and stronger owing to the complementary nature of the base model strengths. This ensemble learning technique, by means of stacking, represented in the diagram, remarkably elevates the performance of the IDS by facilitating enhanced classification accuracy and reliability. Thus, by embracing this novel technique, we bridge a major research gap and provide a potent solution for industrial network attack detection using Modbus.

While the model retains a typical stacking ensemble architecture, this study contributes refinement expressly for industrial networks enabled by Modbus. The contributions of this study are as follows: (i) leveraging the CIC Modbus 2023 dataset, which comprises realistic attack traffic by Modbus; (ii) integrating four diverse classifiers whose complementary capabilities boost protocol-level attack detection, such as query flooding, delay-response events, and payload injection; and (iii) refining the meta-learner using cross-validated probability inputs to boost overall precision while limiting false positives. These constrained improvements ensure the applicability of the ensemble to realistic industrial environments while maintaining consistency with its original name and focus.

4.2 Data Collection and Description

We selected the CIC Modbus Dataset 2023[40] for the research work for a machine learning-based intrusion detection system for modbus-enabled industrial networks using ensemble learning with a stacking approach.

The CIC Modbus Dataset 2023 served as the basis for our research on Modbus intrusion detection in industrial networks. It contains two principal classes: legitimate Modbus communication and benign data, as well as various attacks under the MITRE ICS ATT&CK framework, including reconnaissance, query flooding, and payload manipulation. It was generated on a virtual substation network hosting SCADA HMIs and IEDs using Docker containers. Targets will be both secure and unsecured devices, while the collection techniques include network traffic capture through TCPdump and Docker Bridge. The dataset was provided as CSV timestamped logs with attack details and PCAP network traffic.

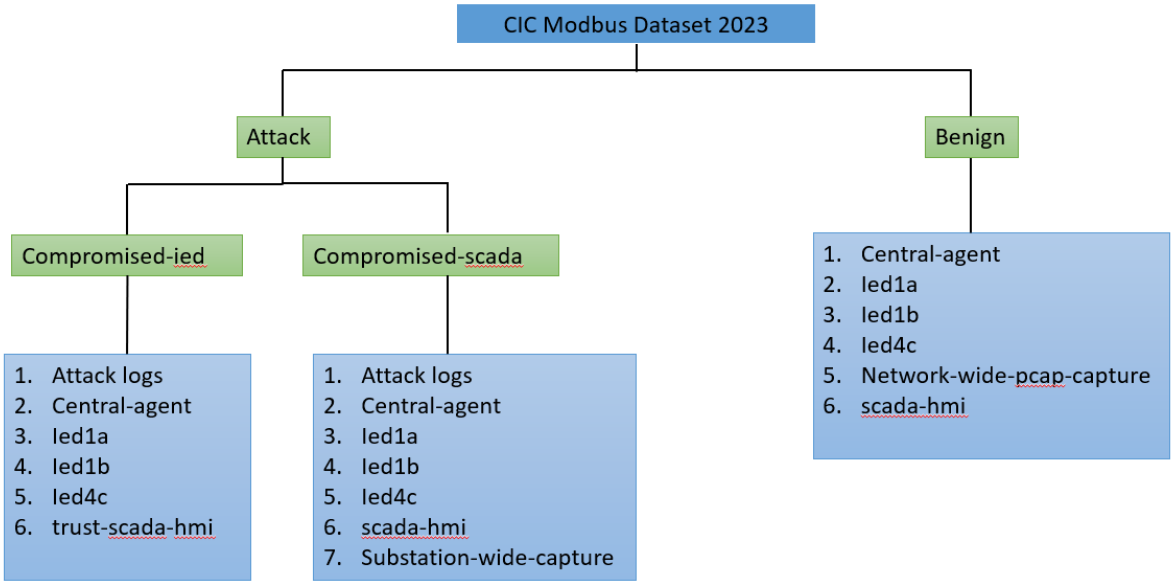

Fig. 3. Hierarchical Structure of the CIC Modbus Dataset 2023, Showing Attack and Benign Categories

Fig.3 illustrates the organizational structure of the dataset. This consists of several subdirectories, as shown in the figure below. Each directory contains pcap files, except for the attack logs folder. In summary, there were approximately 101 pcap files distributed throughout the folders and approximately 30 log files.
This is the information regarding files data-

A. Log file column description:

1. Timestamp: The time at which an incident or attack occurred.
2. Target IP Targeted device-specific IP of the event of attacks targeted in the case of simulation of making the network substation, and every distinct IP exists for different particular devices.
3. Attack: This is the type of attack or action that occurs. It will say something related to an activity, such as the initiation or completion of an attack. For instance, "Payload Injection. Starting" or "Payload Injection. Complete."
4. Transaction ID is a unique identifier associated with events that signifies every transaction/session or attack scenario involved. This is tracked within the action sequence of the same event.
5. Attack log files describe the different types of attacks found in various attack log files.

B. PCAP file column description:

1. Timestamp: The time at which the network packet was captured, usually in hours.
2. In second's format.
3. Source IP: This is the IP address of the source device.

4. Destination_IP: IP address of the device that receives the network packet.
5. Source_Port: The port on the source device that originated the network communication.
6. Destination_Port: The destination port number at the destination device, which accepts network communication.
7. This explains the communication protocol of the packet, such as TCP or Transmission Control Protocol.
8. Data Length: The Size, in bytes, of the data in the packet.

4.3 Data Preprocessing

Our data were preprocessed by combining two different file formats: PCAP files and CSV assault logs. For easier data analysis and combination, the PCAP files were first transformed into the PCAP-to-CSV format. The transformed files have columns such as the following: Protocol (e.g., TCP), Data Length (in bytes, the size of the package), Source IP (IP address of the originating device), Destination IP (IP address of the receiving device), Destination Port (originating communication port number), and Timestamp (time in seconds the network package has been captured). There were already columns in the pre-provided attack logs in CSV, such as Timestamp or time of attack, or Target IP, which is a device-oriented IP that has been attacked by the assault, Attack, a description of the attack activity such as "Payload Injection. Starting" and Transaction ID, which is a unique number for the attack occurrences. The two sources were combined into a single, uniform dataset by aligning the Timestamp and Target IP fields from both. Care was taken during formatting to ensure that it was in the same format. This enabled precise synchronization. A combined dataset was prepared, where the attack column from the logs became the dependent variable, but the added information from the PCAP and the attack logs was preserved for further analysis. This has enabled a complete description of network activities and their corresponding attack occurrences, thereby facilitating subsequent predictive and analytical work.

The detailed process that we pursued to merge all the files by matching the data records is as follows:

1. Data Collection and Preparation for Input: Raw PCAP files were also transformed into CSV format using packet processing tools, and attack logs were preserved in the CSV files in several subdirectories (e.g., compromised-ied, compromised-scada).

2. Initializing and Merging Attack Logs: The files are imported recursively by walking the directories that have attack log CSV files. All the attack logs are combined for consistent access while the labeling occurs.

3. Associating PCAP Data with Attack Information: Reading each PCAP CSV file in the given directory. For each PCAP packet trace record: It is marked initially as "benign." The Timestamp and Destination IP from the packet were matched against the records in the aggregated attack logs. If a match is established, the field of the given PCAP's AttackType is filled in by the attacker's label from the logs. The labeled PCAP data from all directories were aggregated into a single dataset.

4. Combining the Attack and Benign Data: Table 4 defines the columns included in the PCAP files, and the Attack-labeled PCAP data from various sources (e.g., compromised-ied_processed, compromised-scada_processed) were aggregated. Likewise, Table 5 explains that benign PCAP data (e.g., central_agent _processed, ieda_processesd) are labeled and processed similarly. Both the attack and benign datasets were combined to create the fully labeled dataset.

5. Data Conversion for Acceleration: The attack log dataframe is also cast to a NumPy array by pandas for compatibility when running GPU-based libraries. The NumPy array was also cast to a CuPy array for computation using the GPU.

6. Saving the Last Dataset: The aggregated labeled dataset was also stored as a CSV file for training and testing the machine learning models. After merging the data into a single csv file, we obtained total 90,027,742 approx. records and after managing the higher number of benign records to match with attacks records, we obtained 26,44,009 number of records in a final data csv file.

Table 4. Columns and Descriptions of Features from PCAP Files Used in Analysis

Feature No.	Feature Name.
1	Timestamp
2	Source IP
3	Destination_IP
4	Source_Port
5	Destination_Port
6	Data Length

Table 5. Features Present in Processed and Labeled Benign PCAP Data

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

Table 6 describes the number of records present in the final data CSV file with respect to the specific attack types. We used these records to build the model and to interpret the results.

Table 6. Class Distribution Used for Model Training and Evaluation

Class labels	Number of instances	Class labels	Number of instances
benign	2000000	Length manipulation	1388
Brute force or specific coil. Address: 1	1289879	Recon. Complete	1276
Brute force or specific – Complete	1289711	Stacked Modbus Frames	1044
Starting Query flooding..	4854	Stacked Modbus Frames - Complete	1044
Starting Frame Stacking...	4809	Payload injection	920
Frame Stacking... Complete	4809	Payload injection - Complete	920
False Data Injection Attack	4364	Recon. Range: 2000	638
Query Flooding	4120	Recon. Range: 125	638
Query flooding. Complete	4120	Delay Response Attack: Delaying response by 6104 milliseconds	451
Starting Payload Injection...	3996	Delay Response Attack: Delaying response by 5928 milliseconds	429
Payload Injection.. Complete	3996	Delay Response Attack: Delaying response by 7481 milliseconds	402
Delay response: Sending response to client	3375	Delay Response Attack: Delaying response by 5077 milliseconds	398
Query flooding.. Complete	2532	Delay Response Attack: Delaying response by 9300 milliseconds	394
Baseline Replay: In position	2379	Delay Response Attack: Delaying response by 5305 milliseconds	358
Starting Length manipulation...	1947	Delay Response Attack: Delaying response by 6610 milliseconds	352
Length manipulation.. Complete	1947	Brute force or specific coil. Address: 8	344
Replay – Complete	1720	Brute force or specific coil. Address: 14	344
Replay	1720	Brute force or specific coil. Address: 13	344
Length manipulation – Complete	1388	Delay Response Attack: Delaying response by 9242 milliseconds	332
Length manipulation	1388	Delay Response Attack: Delaying response by 7564 milliseconds	327

4.4 Splitting data in Training and Testing phase

In our research effort, we divided our dataset into two phases: training and test data. We used 70% of the dataset strictly for training our model. This means that the model learns the patterns and characteristics from this subset. To test its performance on never-before-seen data, we allocated the remaining 30% for testing the model. This approach ensures fairness in the assessment of a model's accuracy and its ability to make generalizations.

4.5 Model Building

A. Machine Learning Classifiers under Base Learners

For our research experiment, we employed four machine learning models as base learners for the ensemble learning framework. Short explanations for each model and the particular role that they serve in the experiment are provided below.

1) *Support Vector Machine (SVM)*: SVM is a powerful algorithm that is highly effective for binary and multi-class classifications. For our project, it helps us obtain the optimum decision boundary to distinguish between benign and attack traffic using features. SVM can easily perform on high-dimensional network intrusion detection data.
2) *Random Forest (RF):* Random Forest is a combination of decision trees that reduces prediction inaccuracy by making predictions using a group of several trees. Our project has successfully embraced the complexity of data and class imbalance, which translates into successful predictions for both benign and attack traffic.
3) *K-Nearest Neighbors (KNN)*: This is a very basic but efficient method in which the classification of data points relies on how close they are to their nearest neighbors. For our project, in the detection of patterns, KNN is useful because each instance of the traffic is being matched to instances that are similar in the training dataset and hence is useful for the detection of anomalies.
4) *Adaptive Boosting (AdaBoost):* AdaBoost incorporates several weak learners, commonly in the form of decision stumps, to develop a strong classifier. In this project, the errors in classification are adjusted by giving increased weight to such and thereby building strength for the detection of infrequent or rare patterns of attack.

Hence, through these heterogeneous basic models, we utilize the specific benefits within our ensemble learning methodology to achieve higher accuracy and generalization for network intrusion detection. We used the following procedure to develop the machine learning model targeted for the intrusion detection system:

B. Base Models Training

Four machine learning models were used as base learners in the stacking ensemble.

1. *Support Vector Machine (SVM):* Utilized with a linear kernel to find the optimal hyperplane for distinguishing between benign and attack traffic.
2. *Random Forest:* An ensemble of decision trees that captures complex relationships and effectively handles class imbalances.
3. *KNN:* This algorithm uses a similarity-driven approach to classify each instance by referring to its nearest neighbors.
4. *Adaptive boosting or AdaBoost* is an iterative algorithm that places weight on instances misclassified during its steps to achieve better performance.

All models were independently trained on the training dataset. The accuracy scores, confusion matrix, and classification report for each model were calculated to test them uniquely.

C. Stacking Ensemble Approach

We employed a StackingClassifier for the best prediction performance, as the advantages of using several models are combined.

- Overall Models: The forecasts of the four overall models, Support Vector Machine, Random Forest, K-Nearest Neighbors and AdaBoost, were utilized as the input of the meta-model.
- Meta-Model: The meta-model utilized was a logistic classifier for logistic regression to obtain the optimum selection of predictions by the base models.
- Cross-Validation: For the stacking technique, 5-fold cross-validation was used in the training to prevent overfitting and obtain complete performance.

D. Mathematical Equations for Stacking Ensemble Approach

In the stacking ensemble approach, the following are denoted:

- X as the input feature vector for the Modbus network traffic.
- $\{M_1, M_2, \ldots, M_n\}$ as the base classifiers, where M_1 is the K-Nearest Neighbors (KNN) model, M_2 is the Support Vector Machine (SVM), M_3 is the Random Forest, and M_4 is AdaBoost.
- f(x) is the function learned by each base model.
- $h_i(x)$ is the prediction of the i-th base model M_i for an input x.
- M_{meta} is a meta-classifier that learns from the combined predictions of each base model.

a) Step 1: Predictions from Base Models

Each base model M_i produces a prediction $h_i(x)$ for input x as follows:

$$hi(x) = Mi(x) \forall i \in \{1,2,\ldots,n\} \quad (1)$$

b) Step 2: Formulating the Meta-Classifier Input

The outputs from each base model are used to create a new feature vector H(x), where

$$H(x) = [h1(x), h2(x), \ldots, hn(x)] \quad (2)$$

This vector H(x) represents the predictions of the base models and serves as the input to the meta-classifier.

c) Step 3: Final Prediction from Meta-Classifier

The meta-classifier M_{meta} takes H(x) as its input and produces a final output y for classification:

$$y = Mmeta(H(x)) = Mmeta([h1(x), h2(x), \ldots, hn(x)]) \quad (3)$$

d) Step 4: Ensemble Prediction

For classification, the end guess y may be computed by majority vote (in the event of classification) or mean (in the event of regression, but in IDS, it is generally classification). For binary classification, the meta-classifier may employ logistic regression or any other appropriate classifier to combine the predictions.

Algorithm 1 describes the entire preprocessing pipeline followed by the raw CIC Modbus 2023 dataset to guarantee data quality and consistency prior to model training. Both the PCAP network captures and attack log files are raw datasets that need to be synchronized and combined. This step cleans, integrates, and labels the data by aligning records using timestamps and target IP address pairs to properly align benign and attack instances of traffic. Feature extraction attempts to identify pertinent Modbus communication parameters, such as source and destination IPs, ports, transaction IDs, data lengths, and timing in packets. Numerical attributes were scaled using a scaling function, whereas categorical attributes were encoded to maintain compatibility across ensemble classifiers. The dataset was further split into training and testing partitions in a 70:30 ratio to facilitate the evaluation of models that are both robust and unbiased. This step ensures that the subsequent modeling process is based on balanced and properly labeled data that closely resemble real Modbus communication behavior.

Algorithm 1: Data Preprocessing and Feature Engineering

```
Begin
Input: Raw CIC Modbus 2023 dataset (PCAP + CSV attack logs)
Output: Preprocessed and labeled dataset (X, y)
Procedure: Data Preparation for Intrusion Detection
 Data Loading:
    - Load the attack log CSV and PCAP-derived CSV files.
  Data Cleaning:
    - Remove duplicates and handle missing values.
  Data Integration:
    - Match records by Timestamp and Target IP to merge attack logs with PCAP data.
  Feature Extraction:
    - Extract features: Timestamp, Source IP, Destination IP, Ports, Data Length.
  Labeling:
    - Assign attack or benign labels based on matched logs.
  Feature Scaling:
    - Apply StandardScaler to normalize numerical attributes.
  Encoding:
    - Apply OneHotEncoder or LabelEncoder to categorical fields.
  Data Splitting:
    - Split into training (70%) and testing (30%) subsets.
End
```

Algorithm 2: Ensemble Learning Stacking Approach for Intrusion Detection

```
Begin
Input: Preprocessed dataset (X_train, y_train, X_test, y_test)
Output: Predicted labels and performance metrics
Procedure: Ensemble Learning-based Intrusion Detection and Classification (ELIDC)
  Base Models Setup:
    - Base Models = {SVM, KNN, AdaBoost, RandomForest}
  Meta-Model Setup:
    - MetaModel = Logistic Regression
  Stacking Construction:
    - StackingClassifier = StackingClassifier(
        estimators=[('knn', KNN),
                    ('svm', SVM),
                    ('ada', AdaBoost),
                    ('rf', RandomForest)],
        final_estimator=MetaModel, cv=5)
  Model Training:
    - Train the ensemble using training data.
  Prediction:
    - Predict labels on test set.
  Evaluation:
    - Calculate Accuracy, Precision, Recall, and F1-Score.
    - Generate classification report and confusion matrix.
End
```

Algorithm 2 illustrates the intrinsic detection mechanism that applies ensemble learning to combine a varied number of machine learning classifiers to yield improved accuracy and reliability in detection. Four base classifiers—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and Adaptive Boosting (AdaBoost)—are trained in isolation to yield diverse decision boundaries for the Modbus network traffic. The probabilistic outputs of these base models are also combined in the context of a stacking ensemble through the utilization of Logistic Regression as the meta-classifier that learned the optimum base prediction weighing and base prediction interaction patterns. A 5-fold cross-validation technique was utilized to prevent overfitting and generalize over unseen attack scenarios. The ensemble model obtained was evaluated using accuracy, precision, recall, and F1-score measures to approximate its efficacy in discriminating between benign and malicious Modbus traffic. This stacked learning model significantly improved the overall detection performance and reduced false positives compared to the individual classifiers.

5. Result and Analysis

The performance shown in Table 7 indicates the efficiency of the machine learning approaches to create a sufficiently strong intrusion detection system. As indicated in Table 4, we tried a wide variety of models, such as KNN, SVM, AdaBoost, and Random Forest, as base models and then aggregated their power by applying a stacking ensemble. This benefit is showed by a classifier using a meta-model such as Logistic Regression for a stacking such that utilized in a stacking classifier, in having the power to aggregate various base models that are able to identify various patterns and features of the information. In the focus on the aspect of making the ensemble methods successful for achieving the collaborative effects and the aspect of diversity for achieving the success in network traffic classification and intrusion detection. The results confirm that intelligent ensemble learning methods are appropriate for improving the efficiency and reliability of intrusion detection methods.

Notably, the overall accuracy remains higher than 99%, but the precision and recall of the models change because of the class imbalance nature of the dataset. This is not a result of a computational mistake but a result of the varied detection performance between the benign and the number of attack classes.

Table 7. Performance Comparison Demonstrating the Effectiveness of the Stacking Classifier

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	99.54	77	76	76
SVM	99.34	50	55	50
AdaBoost	98.83	10	14	12
Random Forest	99.52	77	78	78
Stacking Classifier	99.78	90	89	89

Fig. 4. Accuracy Results for Base Models and the Stacking Ensemble Method

Fig.4 indicates the efficiency of various ML models that have been implemented in the intrusion detection system and the accuracy values provided for those models. Compared to the four basic models, the model with the highest accuracy is KNN at 99.54%, followed by Random Forest at an accuracy of 99.52%. Both models are highly efficient in dealing with the complex patterns and relationships observed in the dataset. Support Vector Machine (SVM) 99.34%. Again, SVM has proven its efficiency in discriminating between good and bad traffic by using a hyperplane. AdaBoost achieved 98.83%. This is slightly lower because it is highly sensitive to the noisy data. The stacked classifier, which combined the results of these base models using logistic regression, was better than all the individual models, achieving 99.78%. These results verify the benefits of ensemble learning, particularly when the stacking technique takes advantage of the strengths engineered in each base model to provide the most valid and accurate response. These results show that the utilization of various models for the purpose of intrusion detection yields efficiency and robust network security.

All results were obtained using 5-fold cross-validation over the CIC Modbus 2023 dataset run on GPU-accelerated machinery. Although no pilot plant evaluation in real time was feasible under the purview of this research, the experimental setup closely emulated real Modbus network traffic and provided statistically valid proof of performance.

5.1 Implication of results

Our results, depicted in Fig. 5, provide a good estimate of the machine learning algorithms for building a strong intrusion detection system. This is done by prioritizing individual and overall performance when ensemble methods are considered. We evaluated several base models, such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), AdaBoost, and the Random Forest, to determine how they could be useful in terms of accurately labeling network traffic and detecting intrusions. All the models were evaluated using the test dataset to assess accuracy, and for the base models, the KNN realized a 99.54% accuracy level; this indicates that the model is very efficient in terms of apprehending the local relationships as well as similarities in the data, thereby making it very useful for this particular operation. With a 99.52% accuracy level, the Random Forest model also worked very effectively, which is an indication of the prowess of such a model in terms of dealing with highly complicated nonlinear relationships in a dataset. A 99.34% accuracy level was achieved using the linear kernel by the SVM, which is a true indication of its powerful separable nature in discriminating between benign and malicious traffic by its nature to discern that a particular hyperplane separates the classes. However, AdaBoost achieved an accuracy level of 98.83%, which could have been slightly worse and likely was very noise-sensitive in the dataset because the generation of the weights by AdaBoost occurs in a very iterative manner.

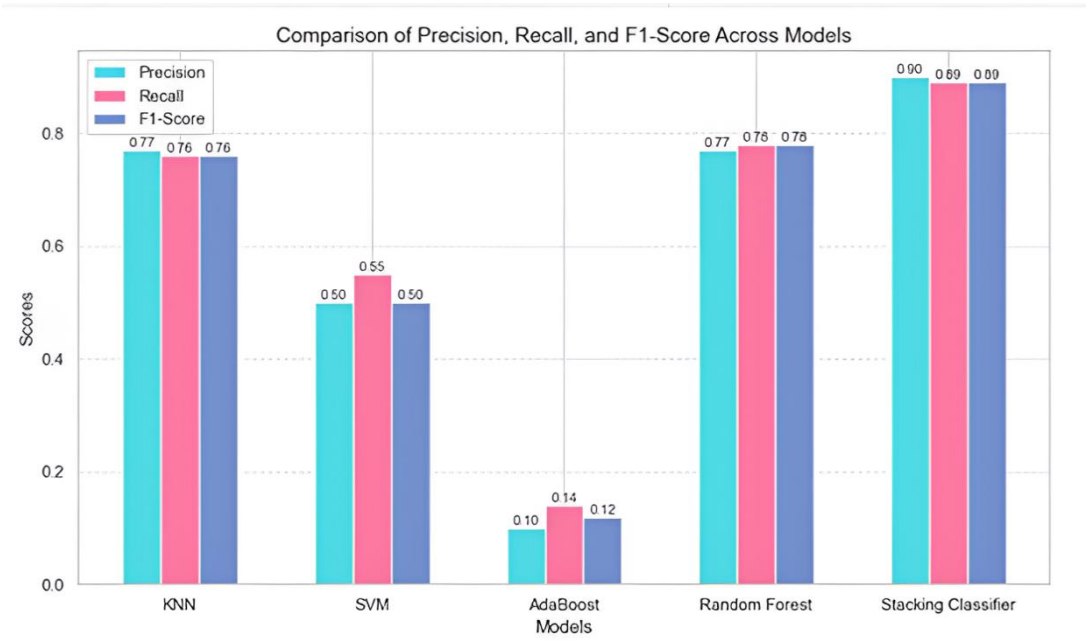

Fig. 5. Performance Comparison Highlighting the Benefits of the Stacking Ensemble Approach

To further optimize performance, we also implemented a stacking ensemble technique such that the predictions of these base models were stacked using Logistic Regression as the meta-model. The accuracy of the stacking classifier improved and was better than that of the individual models by 99.78%. This demonstrates that the strength of the ensemble technique has been enhanced by combining the various strengths of the base models. For example, the strength of local patterns by KNN, the precision in linear class separation by SVM, the emphasis of AdaBoost on the instances that are incorrectly classified, and the aptitude of Random Forest in depicting complicated patterns in combination all together resulted in the enhanced performance of the stacking classifier.

Therefore, our findings present model variety as a function of ensemble learning. The stacking procedure takes advantage of the unique strengths endowed in each of the base models, such that the system is better suited to being more accurate and more reliable in identifying intrusions. This agrees with the basic objective of our research: to achieve a robust mechanism for the detection of intrusions that can sufficiently manage the diversities of network traffic with high accuracy. With higher accuracy in stacking, this classifier is perfectly suited for application in the practice of intrusion detection, which plays a pivotal role in the detection and prevention of cyber threats for the sustenance of network security. Such results hold the potential for leapfrogging in ensemble learning methods in the detection of intrusion systems to unveil the potentiality in the classification and identification of intrusions with great accuracy and reliability.

The Stacking Classifier achieved optimum results, with good accuracy, recall, and F1-score of approximately 0.90. This indicates how successful the stack technique is in refining the accuracy of the detection of intrusions in Modbus networks.

6. Conclusion

This study introduces a Modbus-enabled industrial network ensemble learning-based intrusion detection system (IDS). Using a combination of the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and Adaptive Boosting (AdaBoost) classifiers in a stacking ensemble context, the approached realized a detection precision of 99.78% using the CIC Modbus 2023 dataset. The model revealed considerable enhancements in precision, recall, and F1-score over individual classifiers and substantiated the fact that ensemble learning optimally utilizes the complementary capabilities of various diverse algorithms. The outcomes provide good empirical evidence that ensemble-based IDS models are highly appropriate for solving the escalating cybersecurity threats in industrial environments.

In addition to performance, the findings indicate several realistic implications. First, realistic Modbus datasets guarantee that the proposed framework covers realistic industrial communication behavior that is applicable to realistic SCADA systems. Second, improved accuracy and lower false-positive rate indicate the model's potential to reduce operational interruptions when implemented in real-time monitoring applications and human inspection work. Third, the ensemble's malleability to identify multi-class attack variants indicates its resilience in discriminating between subtle Modbus threats, such as payload manipulation, query flood, and delay-response attacks. Collectively, these results demonstrate that ensemble learning is a viable and evidence-supported industrial cybersecurity strategy.

Future work will extend this research in three primary directions. The first is real-time verification, where the ensemble model is implemented inside a live Modbus communication system to assess the latency, throughput, and response constancy. The second direction is feature optimization and XAI, using interpretability models such as SHAP or LIME to provide insights into model decisions for greater operator confidence. The third direction is cross-protocol generalization-allies for the same ensemble concept to other industrial protocols such as DNP3 or IEC 60870—with the aim to build a homogenous intrusion detection mechanism for heterogeneous industrial control environments. These directions will further develop the scalability, transparency, and operational preparedness of ensemble learning-based IDS solutions for next-generation industrial networks.

References

[1] A. Andersson and Å. Grönlund, “A Conceptual Framework for E-Learning in Developing Countries: A Critical Review of Research Challenges,” E J Info Sys Dev Countries, vol. 38, no. 1, pp. 1–16, July 2009, doi: 10.1002/j.1681-4835.2009.tb00271.x.

[2] M. S. Mohammed and H. A. Talib, “Using Machine Learning Algorithms in Intrusion Detection Systems: A Review,” Tikrit J. Pure Sci., vol. 29, no. 3, pp. 63–74, June 2024, doi: 10.25130/tjps.v29i3.1553.

[3] A. M. Y. Koay, R. K. L. Ko, H. Hettema, and K. Radke, “Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges,” J Intell Inf Syst, vol. 60, no. 2, pp. 377–405, Oct. 2022, doi: 10.1007/s10844-022-00753-1.

[4] A. Alharbi et al., “Analyzing the Impact of Cyber Security Related Attributes for Intrusion Detection Systems,” Sustainability, vol. 13, no. 22, p. 12337, Nov. 2021, doi: 10.3390/su132212337.

[5] M. A. Ayub, W. A. Johnson, A. Siraj, and D. A. Talbert, “Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning,” Institute Of Electrical Electronics Engineers, Mar. 2020, pp. 1–6. doi: 10.1109/ciss48834.2020.1570617116.

[6] S. Raghavan, “Digital forensic research: current state of the art,” CSIT, vol. 1, no. 1, pp. 91–114, Nov. 2012, doi: 10.1007/s40012-012-0008-7.

[7] A. H. Ali et al., “Unveiling machine learning strategies and considerations in intrusion detection systems: a comprehensive survey,” Front. Comput. Sci., vol. 6, June 2024, doi: 10.3389/fcomp.2024.1387354.

[8] B. A Kadheem Hammood and A. T Sadiq, “ENSEMBLE MACHINE LEARNING APPROACH FOR IOT INTRUSION DETECTION SYSTEMS,” ijci, vol. 49, no. 2, pp. 93–99, Dec. 2023, doi: 10.25195/ijci.v49i2.458.

[9] N. Thapa, B. Gokaraju, D. B. Kc, K. Roy, and Z. Liu, “Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems,” Future Internet, vol. 12, no. 10, p. 167, Sept. 2020, doi: 10.3390/fi12100167.

[10] R. Lopez Perez, R. Soua, F. Adamsky, and T. Engel, “Machine Learning for Reliable Network Attack Detection in SCADA Systems,” Institute Of Electrical Electronics Engineers, Aug. 2018. doi: 10.1109/trustcom/bigdatase.2018.00094.

[11] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Efficient Classification of Enciphered SCADA Network Traffic in Smart Factory Using Decision Tree Algorithm," IEEE Access, vol. 9, pp. 154892–154901, Jan. 2021, doi: 10.1109/access.2021.3127560.
[12] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure," Sensors, vol. 23, no. 5, p. 2415, Feb. 2023, doi: 10.3390/s23052415.
[13] S. Sadhwani, M. A. H. Khan, P. M. Pawar, and R. Muthalagu, "BiLSTM-CNN Hybrid Intrusion Detection System for IoT Application," Jan. 03, 2024, Research Square Platform Llc. doi: 10.21203/rs.3.rs-3820775/v1.
[14] J. Du, Y. Hu, K. Yang, and L. Jiang, "NIDS-CNNLSTM: Network Intrusion Detection Classification Model Based on Deep Learning," IEEE Access, vol. 11, pp. 24808–24821, Jan. 2023, doi: 10.1109/access.2023.3254915.
[15] P. Sun et al., "DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System," Security and Communication Networks, vol. 2020, pp. 1–11, Aug. 2020, doi: 10.1155/2020/8890306.
[16] U. Ahmed et al., "Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems," Sci Rep, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-81151-1.
[17] R. Zhao, X. Wen, Y. Mu, and L. Zou, "A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier," IEEE Access, vol. 10, pp. 71414–71426, Jan. 2022, doi: 10.1109/access.2022.3186975.
[18] W. Jiang, D. Shao, L. Ma, Z. Chen, Y. Xiang, and J. Zhang, "SSEM: A Novel Self-Adaptive Stacking Ensemble Model for Classification," IEEE Access, vol. 7, pp. 120337–120349, Jan. 2019, doi: 10.1109/access.2019.2933262.
[19] F. Zhang, Z. Ren, and K. Li, "Improving Adversarial Robustness of Ensemble Classifiers by Diversified Feature Selection and Stochastic Aggregation," Mathematics, vol. 12, no. 6, p. 834, Mar. 2024, doi: 10.3390/math12060834.
[20] T. Morris and W. Gao, "Industrial Control System Traffic Data Sets for Intrusion Detection Research," Springer, 2014, pp. 65–78. doi: 10.1007/978-3-662-45355-1_5.
[21] W. Yusheng et al., "Intrusion Detection of Industrial Control System Based on Modbus TCP Protocol," Institute Of Electrical Electronics Engineers, Mar. 2017, pp. 156–162. doi: 10.1109/isads.2017.29.
[22] I. N. Fovino, A. Trombetta, T. De Lacheze Murel, M. Masera, and A. Carcano, "Modbus/DNP3 State-Based Intrusion Detection System," Institute Of Electrical Electronics Engineers, Apr. 2010. doi: 10.1109/aina.2010.86.
[23] A. Khraisat, P. Vamplew, J. Kamruzzaman, and I. Gondal, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecur, vol. 2, no. 1, July 2019, doi: 10.1186/s42400-019-0038-7.
[24] Z. Azam, M. N. Huda, and M. M. Islam, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," IEEE Access, vol. 11, pp. 80348–80391, Jan. 2023, doi: 10.1109/access.2023.3296444.
[25] R. Wang, M. Karuppiah, Z. Zhang, X. Li, P. Vijayakumar, and J. Lai, "Privacy-Preserving Federated Learning for Internet of Medical Things under Edge Computing.," IEEE J. Biomed. Health Inform., vol. PP, no. 2, pp. 854–865, Feb. 2023, doi: 10.1109/jbhi.2022.3157725.
[26] X. Wang et al., "Towards Accurate Anomaly Detection in Industrial Internet of Things Using Hierarchical Federated Learning," IEEE Internet Things J., vol. 9, no. 10, pp. 7110–7119, Apr. 2021, doi: 10.1109/jiot.2021.3074382.
[27] A. A. Wardana, P. Sukarno, and G. Kołaczek, "Lightweight, Trust-Managing, and Privacy-Preserving Collaborative Intrusion Detection for Internet of Things," Applied Sciences, vol. 14, no. 10, p. 4109, May 2024, doi: 10.3390/app14104109.
[28] A. Sasikumar et al., "Blockchain-Assisted Hierarchical Attribute-Based Encryption Scheme for Secure Information Sharing in Industrial Internet of Things," IEEE Access, vol. 12, pp. 12586–12601, Jan. 2024, doi: 10.1109/access.2024.3354846.
[29] E. Ashraf, N. F. F. Areed, H. Salem, E. H. Abdelhay, and A. Farouk, "FIDChain: Federated Intrusion Detection System for Blockchain-Enabled IoT Healthcare Applications.," Healthcare, vol. 10, no. 6, p. 1110, June 2022, doi: 10.3390/healthcare10061110.
[30] K. A. Kumar, A. Dhar, and I. Chauhan, "Enhanced Credit Card fraud Detection Using iForest Classifier of ensemble Learning with Automated Hyperparameter Tuning," International Journal of Education and Management Engineering, vol. 15, no. 1, pp. 52–60, Feb. 2025, doi: 10.5815/ijeme.2025.01.05.
[31] D. Uhryn, V. Vysotska, D. Zadorozhna, M. Spodaryk, K. Hazdiuk, and Z. Hu, "Intelligent application for predicting diabetes spread risk in the world based on machine learning," International Journal of Intelligent Systems and Applications, vol. 17, no. 3, pp. 90–144, May 2025, doi: 10.5815/ijisa.2025.03.06.
[32] N. Fayzullo, R. Akbar, and Y. Sherzodjon, "Determining the number of effective distributions based on neural network ensemble," International Journal of Intelligent Systems and Applications, vol. 17, no. 4, pp. 69–77, Jul. 2025, doi: 10.5815/ijisa.2025.04.07.
[33] S. A. Hamim, R. S. Aftab, M. Ahmed, F. Faiza, and M. F. Mridha, "Advanced heart attack prediction using a stacked ensemble machine learning model and diverse data integration," I.J. Intelligent Systems And Applications, pp. 49–67, 2025, doi: 10.5815/ijisa.2025.05.04.
[34] G. Ravikumar, J. R. Babu, A. Singh, M. Govindarasu, and A. Moataz A, "D-IDS for Cyber-Physical DER Modbus System - Architecture, Modeling, Testbed-based Evaluation," Institute Of Electrical Electronics Engineers, Oct. 2020, pp. 153–159. doi: 10.1109/rws50334.2020.9241259.
[35] P. Dini, A. Begni, K. Gasmi, A. Elhanashi, Q. Zheng, and S. Saponara, "Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity," Applied Sciences, vol. 13, no. 13, p. 7507, June 2023, doi: 10.3390/app13137507.
[36] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid," Institute Of Electrical Electronics Engineers, May 2021, pp. 129–135. doi: 10.1109/eit51626.2021.9491891.
[37] M. Alalhareth and S.-C. Hong, "Enhancing the Internet of Medical Things (IoMT) Security with Meta-Learning: A Performance-Driven Approach for Ensemble Intrusion Detection Systems.," Sensors, vol. 24, no. 11, p. 3519, May 2024, doi: 10.3390/s24113519.
[38] P. Radoglou Grammatikis, P. Sarigiannidis, E. Panaousis, and G. Efstathopoulos, "ARIES: A Novel Multivariate Intrusion Detection System for Smart Grid.," Sensors, vol. 20, no. 18, p. 5305, Sept. 2020, doi: 10.3390/s20185305.
[39] W. Lee, A. Thomas, N. Balwalli, Y. Zhang, S. Saluja, and J. B. D. Cabrera, "Performance Adaptation in Real-Time Intrusion Detection Systems," Springer Berlin Heidelberg, 2002, pp. 252–273. doi: 10.1007/3-540-36084-0_14.

[40] Kwasi Boakye-Boateng, Ali A. Ghorbani, and Arash Habibi Lashkari, "Securing Substations with Trust, Risk Posture and Multi-Agent Systems: A Comprehensive Approach," 20th International Conference on Privacy, Security and Trust (PST), Copenhagen, Denmark, August. 2023.

Authors' Profiles

Dadaso T. Mane has pursued M.E. in Computer Engineering from Savitribai Phule University of Pune. Currently, he is working as an Assistant Professor in the Department of Information Technology at Rajarambapu Institute of Technology, Rajaramnagar, Sakharale, MH state, India. He has 15 years of academic experience. He is having a life membership of ISTE. His area of interest is Data Science, Machine Learning, and Engineering Education. He has published total 6 research papers in the National / International Journal and Conferences. He has received ISTE National level Gold medal as a Guide for a project titled "Accident Detection & Prevention system using GPS/GSM Technology" at Periyaar Maniammai University, Thanjavaur, TamilNadu (2012-13).

Vijay H. Kalmani, Ph.D., is a Professor in the Department of Computer Science and Engineering at Rajarambapu Institute of Technology, Rajaramnagar, Maharashtra, India. He obtained his M.Tech in Computer Science and Engineering from Gogte Institute of Technology, Belagavi, Affiliated with Visvesvaraya Technological University, Belagavi, Karnataka, India and earned his Ph.D. in Computer Science and Engineering from Suresh Gyan Vihar University, Jaipur, India, in 2016. He has published over 25 papers in international journals and conferences. His areas of expertise include Cloud Security, Artificial Intelligence, and Machine Learning. He has successfully guided three research scholars to the completion of their Ph.D. degrees.

Sayali Aundhakar is an undergraduate student pursuing a Bachelor of Technology (B.Tech) degree in Computer Science and Information Technology at Rajarambapu Institute of Technology, Affiliated with Shivaji University, Kolhapur, India. Her specialization areas include Big Data Analytics and Network Security.

Pranita Patil is an undergraduate student pursuing a Bachelor of Technology (B.Tech) degree in Computer Science and Information Technology at Rajarambapu Institute of Technology, Affiliated with Shivaji University, Kolhapur, India. Her areas of specialization include Machine Learning and Cloud Security. Her academic interests focus on predictive modeling, data analysis, and intelligent systems.

Swati Patil is an undergraduate student pursuing a Bachelor of Technology (B.Tech) degree in Computer Science and Information Technology at Rajarambapu Institute of Technology, Affiliated with Shivaji University, Kolhapur, India. Her areas of specialization include Machine Learning and Cryptograpgy. Her academic interests include intelligent data processing and pattern recognition.

Tejal Yadav is an undergraduate student pursuing a Bachelor of Technology (B.Tech) degree in Computer Science and Information Technology at Rajarambapu Institute of Technology, Affiliated with Shivaji University, Kolhapur, India. Her areas of specialization include Machine Learning, and Information Security. Her academic interests include algorithm optimization, real-time data processing, and the application of machine learning.

How to cite this paper: Dadaso T. Mane, Vijay H. Kalmani, Sayali Aundhakar, Pranita Patil, Swati Patil, Tejal Yadav, "Ensemble Learning-Based Intrusion Detection System for Modbus-Enabled Industrial Networks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.15, No.6, pp. 54-70, 2025. DOI:10.5815/ijwmt.2025.06.05