National College of Ireland

Cloud-Based Intrusion Detection using Super Learner Ensemble in ICS/SCADA

MSc Research Project
MSc in Cloud Computing

Mohammed Zubair Shaik

Student ID: x23228946

School of Computing

National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mohammed Zubair Shaik
Student ID:	23228946
Programme:	MSc in Cloud Computing **Year:** 2024-2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/08/2025
Project Title:	Cloud-Based Intrusion Detection using Super Learner Ensemble in ICS/SCADA

Word Count:6887 **Page Count: 19**

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

Signature: Mohammed Zubair

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Cloud-Based Intrusion Detection using Super Learner Ensemble in ICS/SCADA

Mohammed Zubair Shaik
Student ID: x23228946
MSc Research Project | MSc in Cloud Computing
Email: x23228946@student.ncirl.ie

Abstract

In cloud-integrated industrial environments, Intrusion Detection Systems (IDS) play a pivotal role in safeguarding Industrial Control Systems (ICS) and SCADA networks. Traditional IDS models often lack the scalability, adaptability, and performance required to handle heterogeneous and dynamic Industrial Internet of things (IIoT) data streams. Addressing these challenges, this study proposes a cloud-based IDS leveraging the ToN_IoT dataset and a Super Learner ensemble model. The system integrates Random Forest, Logistic Regression, and Naïve Bayes as base learners with Gradient Boosting as the meta-learner, achieving superior accuracy (95%) and efficient latency-throughput performance. Motivated by the increasing sophistication of cyber threats and the limitations of static signature-based methods, this approach enhances generalization and real-time detection capability. The entire pipeline is deployed on AWS (EC2, Cloud9, and S3), enabling scalable, accessible, and real-time operation. This work demonstrates the effectiveness of stacking ensemble techniques in cloud environments for ICS security and sets a benchmark for future intelligent IDS solutions.

Keywords: Intrusion Detection System, ToN_IoT Dataset, Super Learning, Cloud Deployment

1 Introduction

1.1 Motivation

The rapid integration of cloud computing and Industrial Internet of Things (IIoT) technologies into Industrial Control Systems (ICS) and SCADA architectures has drastically increased operational efficiency—but it has also introduced complex cybersecurity risks. Traditional security mechanisms often fail to address the dynamic, heterogeneous nature of these interconnected environments, making them vulnerable to advanced threats like ransomware, data injection, and denial-of-service attacks. Motivated by these concerns, this

research aims to develop a robust, cloud-based intrusion detection system (IDS) that leverages machine learning and ensemble techniques for real-time threat detection. The choice of the ToN_IoT dataset ensures realism and relevance to modern IIoT ecosystems, while the deployment of the solution on AWS provides scalability and accessibility. Furthermore, using a Super Learner ensemble allows for higher accuracy and generalization compared to single-model approaches.

1.2 Research Question

How effectively can a Super Learner ensemble model, deployed on AWS cloud infrastructure and trained on the ToN_IoT dataset, detect and classify multiple types of cyberattacks in ICS/SCADA systems in terms of measurable performance metrics such as accuracy, latency and throughput?

1.3 Objectives of the Research

The research objectives for this report are:

1. To develop and evaluate a cloud-deployed intrusion detection system (IDS) for ICS/SCADA environments using the ToN_IoT dataset and machine learning techniques, with a focus on multiclass attack classification.
2. To implement and compare multiple machine learning algorithms—including Random Forest, Gradient Boosting, Naïve Bayes, and Logistic Regression—and integrate them into a Super Learner ensemble model to improve detection accuracy, latency, and throughput.
3. To deploy the IDS pipeline on AWS cloud infrastructure (EC2, Cloud9, and S3), ensuring real-time processing, scalability, and accessibility while analyzing feature importance and class imbalance handling for robust model interpretability and performance.

1.4 Outline of the Report

This report is structured into the following sections:

1. Introduction: Introduces the research motivation, objectives, and research question focusing on cloud-based IDS in ICS/SCADA using Super Learner ensemble models.
2. Related Work: Reviews existing literature on intrusion detection, cloud-IoT security, ML/DL techniques.
3. Research Methodology: Describes the dataset (ToN_IoT), preprocessing steps, visualization, feature importance analysis, and data balancing techniques used.
4. Design Specification: Presents the end-to-end system architecture, detailing how data flows from collection to intrusion detection output in a cloud environment.

5. Implementation: Explains the technical implementation of all ML models and their deployment on AWS services like EC2, Cloud9, and S3.
6. Evaluation: Evaluates model performance using accuracy, latency, throughput, and compares results with baseline studies to validate improvements.
7. Conclusion and Future Work: Summarizes the outcomes, highlights key achievements, and proposes future enhancements such as real-time streaming and advanced deep learning models.

2 Related Work

2.1 Emergence of Threat Monitoring in Smart Systems

The transformation of conventional infrastructure to smart, interconnected spaces with the help of the Internet of Things (IoT) has severely increased the cyber exposure area Djenna et al. (2021). The integration of cloud and IoT in modern systems is very important because IoT devices generate data, and they are transmitted to the cloud where they are processed centrally, improving efficiency of operations and, at the same time, providing highly advanced targets of attack by malicious actors Ajayi (2025). These machines, many of which lack the computational capacity and also have more dated firmware, are poorly positioned to resist advanced persistent threats on their own. Conventional intrusion detection systems that support static enterprise networks are not scalable as well as flexible enough to support the dynamism of IoT environments supported by the cloud Sanetra et al. (2025). The continuous nature of data exchange, combined with the habit of devices to exchange information with remote cloud-hosted apps Shukla et al. (2023), exposes it to the possibility of malicious actors recognizing the vulnerability of the endpoint, or intercepting data streams with various malignant data. The use of AI and automatic technologies in specific exploits is becoming more and more common, which is why current intelligent cloud-based intrusion detection systems are becoming a necessity.

2.2 Structural Complexities of IoT Security

The IoT protection in cloud-assisted structures has multi-dimensional problems to its assets as a result of the peculiarities of the organizational structure of IoT implementation Jurcut (2020). These networks have a very diverse type of devices, starting with environmental sensors and smart meters to industrial actuators, and possessing various hardware capabilities, communication protocols, and operating conditions. In contrast to the regular networks that applied uniform architecture, an IoT network is mostly established in unpredictable environments, and in remote or unsecured areas. Most of these gadgets do not have the processing power to take on a combination of key encryption, gadget-level firewalls and automation of patching Micheal (2025). Thus, they rely mostly on the cloud-based analytics, storage, and centralized control. Nonetheless, this vulnerability expands the area of attack, particularly due to compromised or unsecured communication channels, when the data is relayed by the cloud servers. Telemetry data and device credentials can also be leaked by improper configuration of cloud storage and cloud API in general Ademilua (2021). The absence of standardization of IoT platforms also contributes to inability to have standard

policies and lifecycle management of the devices. Also, with dynamic connection / de-connection of devices on the network, the dynamics adds more complexity to the system to be monitored and controlled. The misalignments are also exploited by attackers by means of spoofing, firmware injection, or denial-of-service attacks intended to exhaust cloud-hosted applications.

2.3 Evolution of Anomaly Detection in Industrial Networks

The incorporation of IoT with cloud computing in industrial settings like SCADA (Supervisory Control and Data Acquisition) and ICS (Industrial Control Systems) has brought about certain efficiency as well as complexity Ara (2022). Such systems which were historically closed and deterministic in nature are now becoming more networked in relation to cloud platforms together with centralized data management and analytics and distant control. In spite of the increased visibility and predictive maintenance capabilities, this transition increases the risks to the industrial networks by exposing it to new cyber threats. The former detection frameworks were based on the static and rule-based intrusion measures, which could only detect known signatures. However, these mechanisms are not very useful against new threats, particularly in IoT-cloud extremely dynamic environments. With the evolution of attack vectors to take advantage of both cloud-side and hardware vulnerabilities, the industry has tried moving towards anomaly detection methods that run on artificial intelligence capabilities. Anomaly detection models, developed in the cloud, use historical data in the scale of network traffic logs, telemetry streams, and control signals to form a notion of the normal operational environment Nwachukwu et al. (2024). These models can then be used to track real-time data streams to detect violations, and detect things like ransomware, sensor compromise, or compromised data injection attacks. Notably, the cloud supplies the computing power that is necessary to support processing the intricate models and large quantities of ingested data without overwhelming the local industrial network.

2.4 ML Techniques for Network Intrusion Detection in ICS/SCADA

Mesbah et al. (2023) has suggested SCADA honeypots to provide a proactive cybersecurity layer, monitoring malicious behaviour against industrial control systems (ICS) and supervisory control and data acquisition (SCADA) networks and detect and analyse this malicious behaviour. The researchers plan to increase cybersecurity in Operational Technology (OT) systems, where availability is most important because the industry process affects all other factors. The researchers used Conpot, an open-source honeypot framework that provides honeypots that emulate real ICS/SCADA devices and standards to conduct research and learn about attack vectors, attacker behaviors, the most targeted protocols, and how threats are introduced. The presented method will allow early detection of cyber intrusion and understanding of the vulnerabilities that have been exploited by the attacker. The biggest problem that had to be overcome was realistic simulation of ICS environments in a manner that did not expose real infrastructure to risk.

Mubarak et al. (2021) has come up with an anomaly-based Intrusion Detection System (IDS) in Industrial Control Systems (ICS), specifically the SCADA environment, with different machine learning algorithms that detect cyber threats depending on their network activities.

This study will help the research team create an effective and automated method that would detect malicious activity in the critical industrial infrastructure with the view of classification and prediction of anomalies in ICS traffic. The approach being suggested will consist of examining identified labeled publicly available sets of data that involves ICS network traffic settings then examining the packets and retrieving the flow-based characteristics of the traffic as part of behavior profiling, especially when carried out on a port-by-port basis. Several machine learning models, such as Logistic Regression, KNN, Naive Bayes, Decision Tree, Random Forest, Artificial Neural Networks (ANN) and SVMs were tested based on their classification challenges. Random Forest and Decision Tree models were the ones with the highest training accuracies (96.18% and 96.16% respectively), but had dropped significantly in the test accuracy (both around 81.9%), and are therefore somewhat indicative of being overfitted.

Rakas et al. (2020) have put forward a categorical evaluation framework of intrusion detection systems (IDSs) that are specifically directed towards SCADA networks in the aim of assessing state-of-the-art and to identify open challenges as well as research directions to focus on. The paper is concentrated on the peculiarities of the SCADA systems, e.g., real-time requirements, specific protocols, and traffic peculiarities that make it highly inappropriate to apply generic IDS systems. The proposed solution involves conducting an extensive analysis and assessment of 26 scientific articles published in 2015-2019 with respect to such criteria as detection methods, targeted protocols, implementation tooling, test frameworks, and performance measurements. The main strength of the study is that it focuses on implementation maturity and applicability to Future Internet environments. Among the challenges that were encountered is the comparison of studies having different methodologies and inconsistent metrics of evaluation. Although the review offers important insights and implementation plan, it is short of limitations given it is based on the literature of the past, thereby missing the emerging technology, or that which is not published.

A hybrid intrusion detection system (IDS) has been suggested by Ahakonye et al. (2023) by integrating Chi-square feature selection that is combined with Modified Decision Tree (MDT) classification, which enhances accuracy with minimal computation cost. It examines the increasing risk on Industrial Internet of Things (IIoT) and SCADA systems, where the current IDS systems tends to have problems with computational complexity and real-time dynamics. The suggested method is executed in three stages: data preparation (consisting of preparation and normalization), the fusion of features through the Chi-square method, identification of the most relevant attributes, and anomaly detection with the help of the MDT classifier. One of the crucial concerns tackled was the trade-off between detection precision and practicality of the system in terms of qualities required to operate in real time. Nevertheless, one of the possible drawbacks is the particularity of features selection and trained classifiers that can be potentially reconfigured based on collections of various SCADA architecture or changing attack patterns.

2.5 DL Techniques for Network Intrusion Detection

Ashiku and Dagli (2021) has put forward an adaptive and resilient Network Intrusion Detection System (IDS) utilizing deep learning constructs, specifically Deep Neural Networks (DNNs), to increase the rate at which known and zero-day cyberattacks are recognized and instilled. The aim of the research is to respond to the emerging sensitivities

due to the increased dependency of computing systems on security, which necessitates the traditional protection measures to be inadequate because of higher interdependence and interoperability of these systems. This proposed solution exploits DNNs to their advantage, since they can be trained based on the made complex patterns in the network traffic data and be adaptive to the changing attack trends. The investigation proves hopeful outcomes in terms of accuracy of detection and fewer false positives. One weakness is the use of dummy attack data, which might not reflect the random nature of zero-days attacks and thus an issue of generalizability and robustness of the model in production networks.

A recent study Kocher and Kumar (2021) has outlined an exhaustive abstractive review of exploring the comparison between traditional Machine Learning (ML) environments, like the Artificial Neural Network (ANN), Support Vector machines (SVM), fuzzy logic, swarm intelligence, and evolutionary computation with the Deep Learning (DL) technique to the scenario of intrusion detection systems (IDS). This study is aimed at filling the gap that still exists in literature on the relative effectiveness of these strategies and given the relative emergence of DL methods as a result of their capacity to analyse large and complex data. The paper is a review-analytical work on the recent literature and base datasets, performance benchs, and examples of DL to intrusion detection. The shortcoming of the approach is that it is a literature-based review that has not been confirmed in experiments and the findings examined depend on the results provided in the literature which may also differ with experimental conditions and used dataset.

The meta-survey and taxonomy of the deep learning-based intrusion detection system (IDs) schemes to improve the security of the network and host concerning the rising technical sophistication of cyber threats have been proposed by Lansky et al. (2021). The objective of the study is to examine the manner in which the different deep learning techniques become incorporated into the IDS architectures and present a structured insight on how they can be utilized to accomplish the accurate detection of intrusion. The literature review entails a long-range of concepts of IDS as well as a set of deep learning models; then, the existing IDS frameworks can be classified by the particular types of deep learning they implement: CNNs, RNNs, and autoencoders. The paper presents a comparative study of these frameworks, their advantages in enhancing the detection accuracy and their ability to ensure they can be adapted to the emerging threats. One of the major contributions of the paper denotes the finding that deep learning can be a great tool to increasing efficiency in IDS. A weakness is however its theoretical nature, that is, it cannot be benchmarked empirically to a composite experimental system and therefore to directly compare the performance among studies or evaluate ability to implement in the real world.

The paper by Kasongo (2023) addresses the issue of Internet of Things (IoT) cybersecurity by presenting a proposed Deep Learning-based Intrusion Detection System (IDS) that can help protect these environments against numerous cyberattacks. The increased risk of intrusions of the IoT devices is a topical issue which the study will address as the undetected vulnerability of the system can lead to extensive disruption of the services and financial losses. The suggested solution relies on a four-layer deep Fully Connected (FC) Artificial Neural Network to localize malignant traffic in real-time, where the system is developed to be communication protocol agnostic, thus, sheltering more complex deployments across varied IoT platforms. The model was evaluated on simulated and real intrusions prevention scenario and demonstrated high detection accuracy of 93.74% with precisions, recall, and F1-scores of 93.71, 93.82 and 93.47 respectively. The detection system is able to detect different forms of attacks such as Blackhole, DDoS, Sinkhole, Wormhole and Opportunistic Services. The

downside, though, is that the system demonstrates a high performance in the strictly controlled settings, whereas its capabilities in the large-scale, heterogeneous real world IoT networks where unknown types of attacks or extremely dynamic threat environment are possible are not fully clear and can be subject to validation.

In Khraisat et al. (2020), the authors suggested a Hybrid Intrusion Detection System (HIDS), which is used to enhance intrusion detection capabilities by stacking the ability of C5.0 Decision Tree classifier (which denotes the Signature-based IDS) with One-Class Support Vector Machine (OC-SVM) (which denotes the Anomaly-based IDS). The study aims to come up with a powerful system that can recognize the known and zero-day attacks whose detection is not precise by the traditional single-classifier systems since the modern viruses are polymorphic and metamorphic. The HIDS that has been suggested takes advantage of both the accuracy of the C5.0 model in classification and the outlier recognition ability of OC-SVM to limit false positives and increase the detection rates. It was also tested on standard datasets-NSL-KDD and ADFA and had the maximum accuracy of 83.24% as compared to individual classifiers. Nevertheless, as the results are better, the method still has several limitations (moderate accuracy in comparison to newer methods of deep learning and stacking results in computational overhead), so it is not ideal to use in the real-time context without additional optimization.

2.6 Research Niche Summary

This related work review highlights the evolution of intrusion detection in ICS/SCADA, emphasizing the shift from static, rule-based methods to AI-driven anomaly detection leveraging ML and DL. It identifies limitations in existing approaches, including scalability issues, overfitting, lack of real-time readiness, and poor adaptability to heterogeneous IIoT environments. While prior studies explored honeypots, hybrid models, feature selection, and deep neural architectures, most lacked cloud deployment, comprehensive latency–throughput evaluation, and explainability. This creates a niche for a cloud-based, Super Learner ensemble IDS using the ToN_IoT dataset—offering high accuracy, real-time performance, scalability, and interpretability tailored to the unique demands of industrial networks.

3 Research Methodology

3.1 Dataset Description

The ToN_IoT dataset is a systematic and realistic benchmark developed by Cyber Range and IoT Labs at the UNSW Canberra to validate cybersecurity solutions like intrusion detection systems (IDS), threat intelligence, malware detection, fraud detection, privacy-preservation techniques, and digital forensics, adversarial machine learning, and threat hunting applications. The data is heterogeneous in nature with lower levels and sensors of IoT and IIoT, telemetry data, Windows 7 and Windows 10 operating system logs, TLS Ubuntu 14 and Ubuntu 18 operating system logs, and network traffic records. It was developed based on simulated Industry 4.0 testbedelling of various networks of virtual machines and interconnected systems with Windows, Linux, and Kali OS operating systems to simulate many-layered IoT-Cloud-Edge/Fog architecture. Several cyberattacks were leveled on this environment including the use of Denial of Service (DoS), Distributed Denial of Service (DDoS), and ransomware attacks to attack gateway systems of the IoT, web applications, and endpoint systems. Data collection was done in parallel such that a diverse and balanced distribution of both normal and malicious activities were collected in several domains.

Consequently, the ToN_IoT dataset lends itself as an excellent and extendable training and test dataset for machine learning models in order to improve the security and survivability of the current generation of Industrial Control Systems and SCADA infrastructures.

3.2 Data Preprocessing

During data preprocessing step, the objective was to prepare ToN_IoT data to be run through multi-class classification. The binary column that classified the data as either benign or malicious was also removed since the aim was to identify the type of attack. Some of the categorical features present in the dataset had to be transformed into numeric because the machine learning algorithms do not work well on such features. These categorical columns i.e. protocol types, connection states and IP addresses were selected as these columns lacked float64 and int64 data types. These were then labeled coded in form of Label Encoding where the unique category would be assigned a numeric code. This transformation gave the current model a greater ability to interpret categorical variables. Also, the dataset was not entirely devoid of the problem of class imbalance as some types of attacks were underrepresented as compared to others. Label encoding and subsequent resampling or weighting helped to overcome this imbalance in training the model. These characteristics included the numerals like duration, numbers of bytes sent and received, packets, and attributes like DNS or HTTP, were maintained as in the original formats. This in-depth preprocessing process managed to clean the dataset, to make it numerically consistent, and structurally aligned to be used on the next step of feature selection and model development on the field of machine learning.

Figure 1: Correlation Matrix

3.3 Data Visualization

Figure 1 presents a correlation heatmap, which visualizes the pairwise correlation coefficients between various features in the dataset. Each cell in the heatmap displays a value between -1 and 1, indicating the strength and direction of linear relationships. Yellow represents high positive correlation (e.g., src_bytes and src_ip_bytes ≈ 1.00), while dark purple indicates low or negative correlation. Strong correlations (values close to ±1) suggest redundancy, which can influence model performance if not addressed. Features like dns_qtype, proto, and

dst_port show weaker correlations with others, indicating independent contribution. This heatmap aids in identifying multicollinearity and selecting meaningful features.

Figure 2 illustrates a bar chart representing the distribution of values in the target column type, which includes multiple attack classes and the normal class. The dataset is highly imbalanced, with the normal class having the highest frequency—approximately 2500+ instances. In contrast, all attack types such as scanning, xss, injection, ddos, password, rce, ransomware, and backdoor have significantly fewer instances, typically ranging between 100 to 300 each. This imbalance can adversely impact model training, favoring the majority class. This visualization highlights the necessity for appropriate data balancing techniques during preprocessing for fair and effective classification.

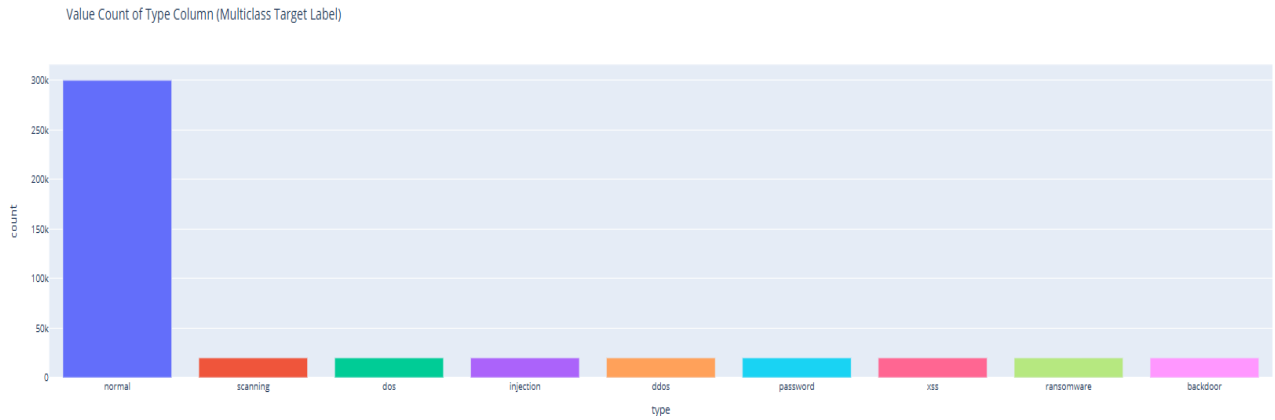

Figure 2: Bar Chart Showing Value Counts of Multiclass Target Labels

Figure 3 displays a bar chart representing the distribution of binary labels in the dataset before converting to a multiclass format. The label column includes two classes: 0 (normal) and 1 (attack). The chart shows a significant class imbalance, with approximately 300k instances labeled as 0 (normal) shown in blue, and 160k instances labeled as 1 (attack) shown in red. This imbalance suggests that the dataset contains almost twice as many benign samples compared to malicious ones, which could bias a machine learning model toward predicting the majority class unless balanced through resampling or weighted algorithms.

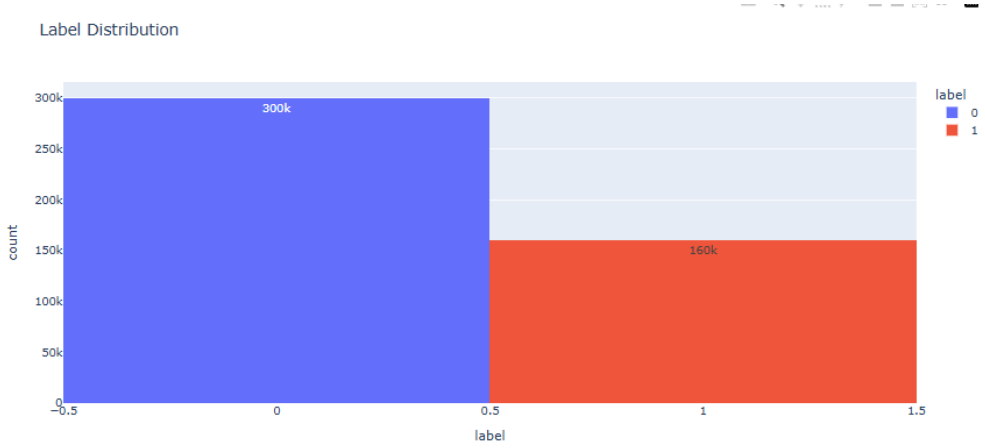

Figure 3: Bar Chart of Binary Label Distribution

Figure 4 presents a donut chart showing the distribution of network protocols used in the dataset. The chart reveals that TCP (Transmission Control Protocol) dominates the dataset with 61.2% of the traffic, followed by UDP (User Datagram Protocol) with 37.5%, and a minimal share of 1.27% belonging to ICMP (Internet Control Message Protocol). This protocol distribution provides insight into the nature of communication patterns in the dataset, where TCP-based communications are most prevalent. Understanding protocol proportions is essential in intrusion detection, as different attack types may exploit specific protocols, influencing feature relevance and model training.

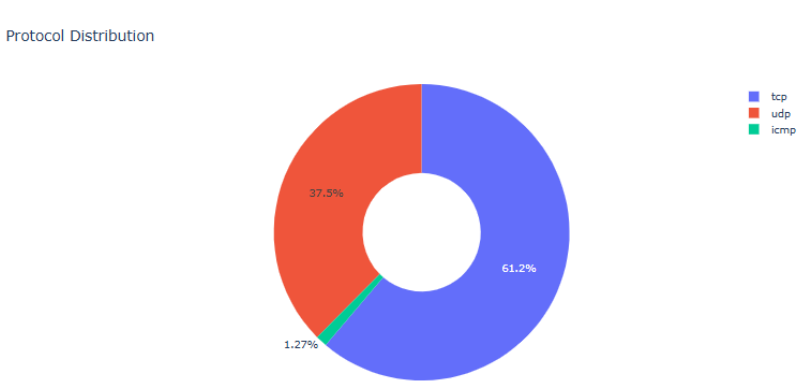

Figure 4: Donut Chart of Protocol Distribution

Figure 5 illustrates a bar chart representing the distribution of various connection states in the network traffic data. The most frequent connection state is SF (Normal completion of connection) with around 125k instances, followed by S0 (connection attempt seen, no reply) and OTH (miscellaneous states), both exceeding 110k counts. Other notable states include REJ (~95k), SHR, and RSTR. Less common states like RSTO, RSTOS0, S2, and S3 each appear fewer than 5k times. This distribution highlights the dominance of complete and attempted connection states, which are vital for understanding typical and anomalous network behaviors in intrusion detection systems.

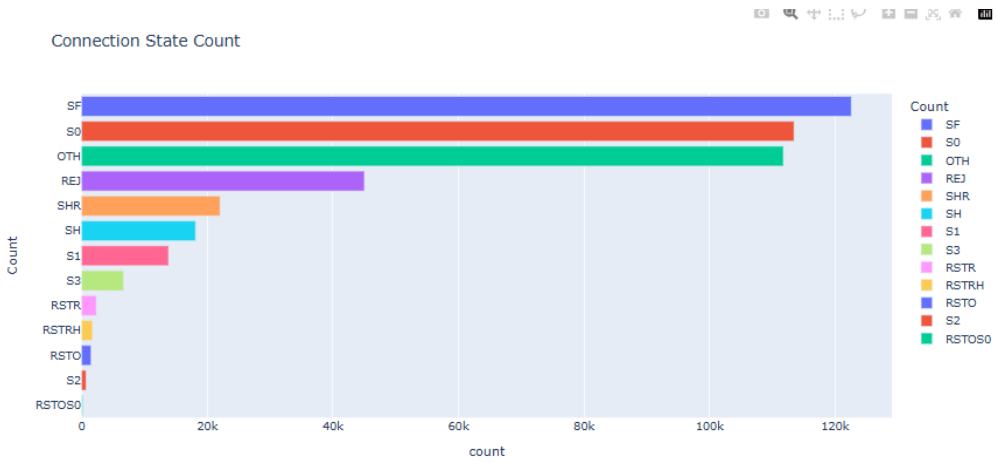

Figure 5: Bar Chart of Connection State Count

3.4 Feature Importance

In finding the most crucial characteristics leading to intrusion detection, feature importance analysis was done using Random Forest Classifier. On the preprocessed dataset, a RandomForestClassifier was trained to an optimum of 3 depth, where maximum depth is the number of levels a tree should be allowed to grow, in this case 3 to allow performance and fulfill objectives of explainability in the built model. Once the model was fitted, the scores of the feature importances were obtained with the feature_importances_ attribute, which gives us the quantitative result of how important every feature lead to the predictive power of the model. The scores were also joined with the names of the features they refer to, to create a DataFrame, and it became easier to read and represent the results visually. The DataFrame was arranged in descending order to show the top favourite features which include source bytes, destination bytes, duration, and DNS based parameters which are very important in distinguishing between normal traffic and different kind of cyberattacks. This ranking can not only be used to help determine which attributes are the most important but also will allow dimensionality reduction, which will allow computational performance to increase, as well as possibly increase the accuracy of the model. The produced list of ranked feature importance was matched in a graph format and placed in the specified AWS S3 bucket in the format that could further be loaded and deciphered with other pieces of the relevant data. This action can have a major contribution to the explainability of the model and the improvement of the complete intrusion detection pipeline.

Figure 6 displays a bar chart showing the feature importance scores derived from a Decision Tree classifier. The most influential feature is src_ip, with a dominant importance score of over 0.35, followed by proto, dst_port, and dst_ip, each contributing scores between 0.10 to 0.15. Other moderately important features include src_ip_bytes, src_pkts, and conn_state. Features like http_status_code, http_response_body_len, and missed_bytes had near-zero importance, indicating minimal influence on classification outcomes. This analysis helps prioritize features during model training, highlighting which variables significantly impact decision-making in network intrusion detection.

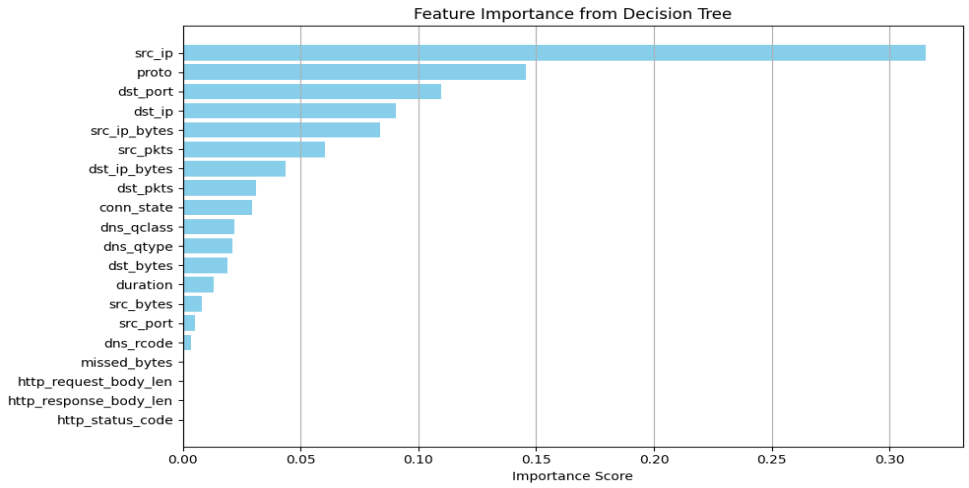

Figure 6: Bar Chart of Feature Importance from Decision Tree

3.5 Data Balancing

Figure 7 is a bar chart visualizing the class distribution of the binary label column before applying any data balancing techniques. Figure 8 presents a bar chart showing the value count of the label column after applying an undersampling technique to balance the dataset.

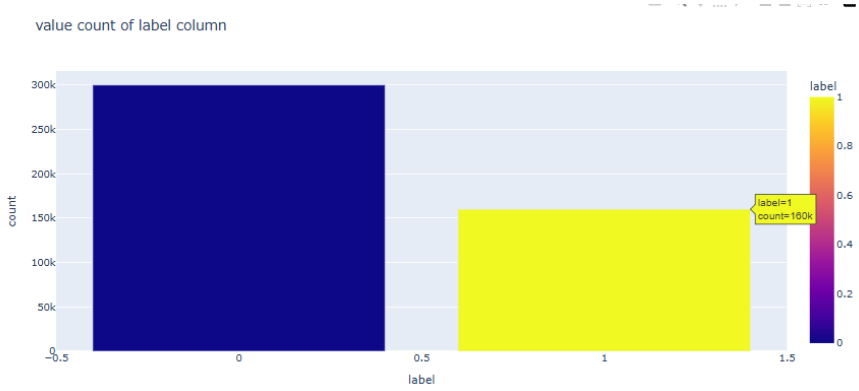

Figure 7: Bar Chart of Label Distribution Before Data Balancing

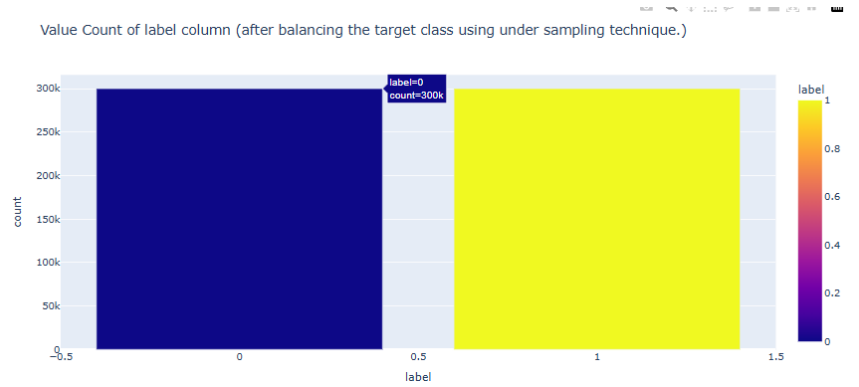

Figure 8: Bar Chart of Label Distribution After Data Balancing (Undersampling)

4 Design Specification

Figure 9 illustrates the end-to-end system architecture for the proposed cloud-based intrusion detection system in ICS/SCADA environments. The process begins with ICS/SCADA Network Traffic, where raw telemetry and packet data are collected from IIoT-enabled industrial control systems. This data is stored in the AWS S3 Bucket, serving as a scalable and centralized repository for the ToN_IoT dataset. The AWS Cloud9 Environment acts as the development and execution platform, enabling secure code implementation and pipeline integration. The pipeline initiates with Data Preprocessing, including cleaning, encoding, and undersampling to handle noise and class imbalance. Following this, Feature Selection is carried out using Random Forest and Decision Tree methods to identify impactful attributes.

The core Model Training phase integrates multiple classifiers—Random Forest, Logistic Regression, and Naive Bayes—into a Super Learner Ensemble using Gradient Boosting as the meta-model. The trained model is then subjected to rigorous Model Evaluation, which assesses accuracy, latency, and throughput. The final stage outputs Intrusion Detection Results, classifying real-time traffic into normal or specific attack types. This modular, cloud-deployed architecture ensures scalability, efficiency, and robustness, making it well-suited for modern industrial cybersecurity applications.

Figure 9: System Architecture Diagram

5 Implementation

5.1 Implementation of ML Models

This section shows implementation for all ml models having maximum depth, L1 regularizer and all.

5.1.1. Random Forest Classifier

To avoid complex and overfitting model, a Random Forest Classifier was tuned with maximum depth of 3 and 6 estimators. It was trained on training data (X_train, y_train) and its performance was measured on the test set. Random Forest is a machine learning algorithm that aims at building numerous decision trees and combining their prediction to enhance an increased accuracy and reduce the variance. This is because its noise as well as its capabilities to work with high-dimensional datasets makes it a good framework in intrusion detection. The results of the model were then combined to form the ensemble Super Learner model to increase the performance of the whole model due to the multi-modelling voting and decision combination techniques comprised.

5.1.2 Gradient Boosting Classifier

To find a model that reduces the error of prediction in each iteration, a shallow depth of 2 and three estimators were used by the Gradient Boosting method. In this model, we need to make improvements on the residuals of the earlier models and in the process, it also enhances predictive power. The standalone performance of the test set then generated predictions after

training on the given dataset. Gradient Boosting is also useful when it comes to imbalanced and noisy data, so it will be a good solution to cybersecurity issues. It also has been deployed as a final meta-model in Super Learner ensemble because it has the ability to capture and model complex relationships between features.

5.1.3 Naïve Bayes Classifier

The Naïve Bayes model was implemented using the GaussianNB classifier, which assumes features follow a Gaussian distribution. It was trained on the full dataset and tested for classification performance. This algorithm is highly efficient, scalable, and well-suited for real-time prediction in intrusion detection systems. It performs particularly well with high-dimensional input data and independent features. Despite its simplicity, it provides strong baseline performance and was also evaluated for latency and throughput, achieving fast inference time. Due to its lightweight architecture, it was also used as a base model in the stacking ensemble to balance complexity and speed.

5.1.4 Logistic Regression with L1 Regularization

The Logistic Regression model was implemented with L1 regularization using the liblinear solver. L1 penalty (Lasso) helps in feature selection by shrinking irrelevant feature weights to zero, thus enhancing generalization. The model was trained on the processed dataset and tested for classification accuracy. Logistic Regression is a strong baseline classifier in intrusion detection due to its interpretability and low computational cost. It is capable of efficiently modeling the probability of attack classes, making it suitable for multiclass classification scenarios. This model was also included as a base learner in the ensemble Super Learner to contribute to decision diversity.

5.1.5 Super Learner (Stacked Ensemble Model)

The Super Learner model is an ensemble technique combining the predictions of multiple base learners—Random Forest, Logistic Regression, and Naïve Bayes—using Gradient Boosting as the meta-model. It was trained through out-of-fold predictions generated via 5-fold cross-validation, which allowed the meta-model to learn from diverse model outputs without overfitting. After fitting base models on full training data, the meta-model was trained on the stacked outputs. This method aims to achieve better generalization and accuracy by leveraging the strengths of individual models. Performance was evaluated using accuracy, latency (per prediction), and throughput (predictions per second), showing effective scalability and efficiency.

5.2 Implementation of AWS Setup

Figure 10 demonstrates how the AWS EC2 instance utilized in this research has been deployed in a cloud environment with intrusion detection models put into effect. The EC2 is

setup and used in the t2.large type, which gives it enough computer power to take on the job of machine learning such as training and evaluation. This example is the central space in which preprocessing of data, training of the model, and its performance are to be performed. It allows an accessible and scalable infrastructure to host the intrusion detection pipeline with Python and Pandas as well as scikit-learn. The cloud architecture makes the setup flexible, remotely accessible and allows the ease of experimenting with models.

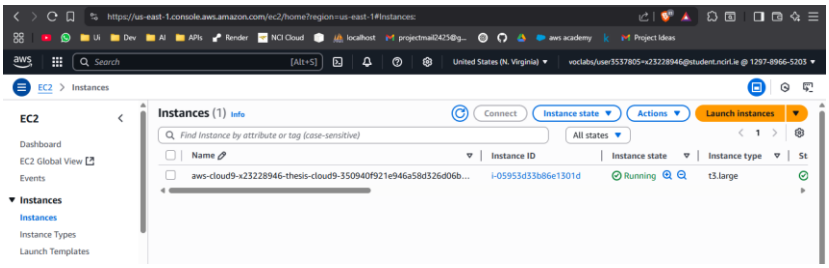

Figure 10: AWS EC2 Instance Setup for Intrusion Detection System Deployment

As shown in Figure 11, the AWS Cloud9 system is the setting utilized in this work, which facilitated intrusion detection model construction and execution. The Cloud9 interface is connected with the EC2 instance and it creates secure browser-based Integrated Development Environment (IDE) of real-time coding, debugging, and machine learning workflow management. It facilitates the smooth connection of industry datasets that are stored on the S3, running the Python code and connecting with the AWS coded additional tools. This development environment reduces complexities of developing an application without local configuration and crucial in the execution of pre-processing, feature selection, model-training, and model-testing. It optimizes productivity, security and cooperation within the cloud based IDS chain.

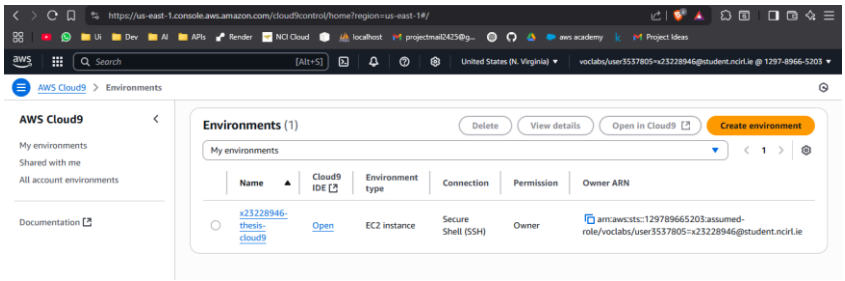

Figure 11: AWS Cloud9 IDE Environment for Development and Execution

The configuration of AWS S3 bucket applied in this research is illustrated in Figure 12, with the purpose of storing datasets and output graphs in the process of implementing intrusion detection system. The bucket is a globally accessible scaled storage point where data can be handled effectively in the cloud. It will have CSV training and testing files on machine learning models and PNG graph files produced in the process of exploratory data analysis and evaluation. With such configuration, it offers intact storage and accessible environments on

EC2 and Cloud9 platforms readily, facilitating effective integration with the machine learning system into the IDS pipeline, hence end-to-end cloud-hosted IDS functioning.

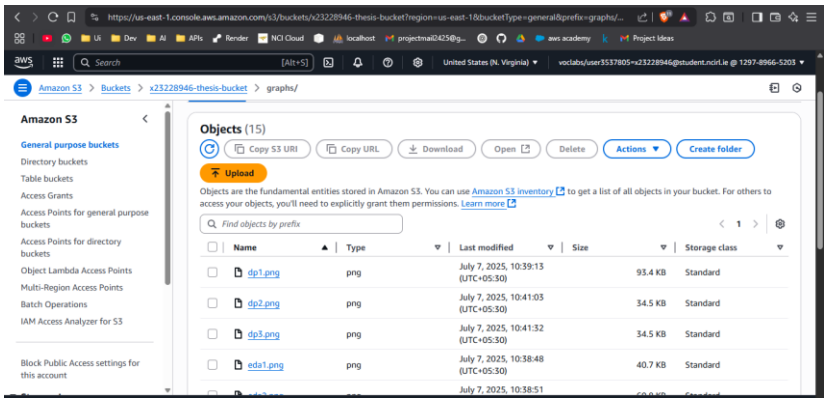

Figure 12: AWS S3 Bucket for Dataset and Graph Storage

6 Evaluation

6.1 Experiment 1: Model metrics of ML Models

Table 1 compares the accuracy performance of various machine learning models used in this study for intrusion detection in ICS/SCADA systems. Among the models, the Super Learner ensemble model achieved the highest accuracy of 95%, outperforming individual models by combining their strengths. Random Forest and Gradient Boosting also performed well, with accuracies of 94% and 93% respectively, showing strong classification capabilities. Logistic Regression achieved a moderate accuracy of 83%, while Naïve Bayes had the lowest performance at 66%, indicating limited effectiveness on this dataset. The Super Learner is identified as the most robust and reliable model for this task. This table also shows other metrics also like precision, recall and all for all respective five models like precision of random forest is 0.94.

Table 1: Model Metrics Comparison Table

Model	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)
Random Forest	0.94	0.94	0.94	0.94
Gradient Boosting	0.93	0.94	0.93	0.93
Naïve Bayes	0.66	0.73	0.66	0.64
Logistic Regression	0.83	0.85	0.83	0.83
Super Learner (Stacked)	**0.95**	0.95	0.95	0.95

6.2 Experiment 2: Latency and Throughput of ML Models

Table 2 compares the latency and throughput performance of five machine learning models used for intrusion detection in the ICS/SCADA environment. Logistic Regression achieved the best overall performance with the lowest latency of 1.5395 ms and the highest throughput of 10,012,263.38 predictions per second. While Super Learner achieved a high throughput, its latency was significantly higher (12.9756 ms). Naïve Bayes also performed well in terms of latency but lagged in throughput.

Table 2: Model-wise Latency and Throughput Comparison

Model	Latency per Prediction (ms)	Throughput (Predictions/sec)
Random Forest	2.1412	4,979,978.63
Gradient Boosting	2.0067	4,672,278.04
Naïve Bayes	1.7026	3,802,460.45
Logistic Regression	1.5395	10,012,263.38
Super Learner	12.9756	8,928,166.89

6.3 Comparative Analysis with Base Work

Compared to the base study by Khraisat et al. (2020), this study introduces a more robust and real-world applicable intrusion detection system tailored for ICS/SCADA environments. While the base paper employed the older UNSW-NB15 dataset and a hybrid model combining C5.0 and One-Class SVM with limited deployment focus, our study leverages the more relevant and recent ToN_IoT dataset that better represents IIoT and SCADA-specific threats. Our stacked ensemble model, comprising Random Forest, Logistic Regression, Naïve Bayes, and Gradient Boosting as a meta-learner, outperforms the base approach with a higher accuracy of 95%, reduced latency, and enhanced throughput. Furthermore, by deploying the system on AWS and integrating interpretability through feature importance analysis, this study approach ensures scalability, real-time detection capability, and improved operational relevance.

Table 3: Comparison Table: My Study vs. Base Paper

Feature	Base Paper: Khraisat et al. (2020)	My Study
Dataset Used	UNSW-NB15	ToN_IoT (more recent, tailored for IoT/IIoT/SCADA)
Model Approach	Deep Learning (Stacked C5 + One-Class SVM Hybrid IDS)	Stacked Ensemble (Random Forest, Logistic Regression, NB with GB as meta-model)
Deployment	Local evaluation (non-cloud)	Deployed on AWS (EC2, S3, Cloud9)

Evaluation Metrics	Accuracy (83.24%), general discussion on efficiency	Accuracy (95%), Latency, Throughput
Real-time Readiness	Limited discussion on real-time deployment	Cloud-integrated, performance-tested for real-time
Explainability	Limited	Feature importance + correlation heatmaps for interpretability
Model Efficiency	Higher overhead due to SVM and hybrid model stacking	Lightweight, optimized ensemble with low latency
Security Focus	General IDS	ICS/SCADA-specific intrusion detection

7 Conclusion and Future Work

7.1 Conclusion

This study has managed to design a cloud-based intrusion detection system with ICS/SCADA systems, based on the dataset on ToN_IoT data and a stacked ensemble machine learning scenario. Random Forest, Logistic Regression, Naive Bayes, and Gradient Boosting have been integrated into the system, delivering a high accuracy of 95%. The system also demonstrates good latency and throughput performance. AWS services such as EC2, S3, and Cloud9 deployment was used to provide scalability and the ability to process things in real-time. The model also offered some interpretability as it allowed the understanding of the most important features that showed significant indicators of an attack. The results shows that the proposed Super Learner ensemble addresses the research question, proving robust and efficient for modern industrial cybersecurity.

7.2 Limitations and Future Works

Although good results were recorded, this study had its limitations. Tested and trained on fixed information never having to integrate real-time streaming or live threat situations the current model has been made obsolete. Also, the system is yet to be tested in real ICS/SCADA settings within operational restrictions. Future work will include integration of real-time streaming of data out of industrial systems to provide increased responsiveness with regard to detection. More complex models like LSTM and GRU could be used to recognise time dependencies in atacks and perhaps using the Transformer could have better results when working with sequences. Unsupervised anomaly detection can be performed with the use of autoencoders, and GNNs can represent intricate patterns of communication relating devices to one another. Furthermore, federated learning techniques can be regarded as deploying secure, privacy-preserving model training at distributed industrial nodes with no need of sharing raw data. These innovations are to enhance flexibility, accuracy and versatility of the intrusion detection system.

References

1. Ashiku, L. and Dagli, C., 2021. Network intrusion detection system using deep learning. *Procedia Computer Science*, *185*, pp.239-247.
2. Kocher, G. and Kumar, G., 2021. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, *25*(15), pp.9731-9763.
3. Lansky, J., Ali, S., Mohammadi, M., Majeed, M.K., Karim, S.H.T., Rashidi, S., Hosseinzadeh, M. and Rahmani, A.M., 2021. Deep learning-based intrusion detection systems: a systematic review. *IEEE Access*, *9*, pp.101574-101599.
4. Kasongo, S.M., 2023. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, *199*, pp.113-125.
5. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. and Alazab, A., 2020. Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. *Electronics*, *9*(1), p.173.
6. Mesbah, M., Elsayed, M.S., Jurcut, A.D. and Azer, M., 2023. Analysis of ICS and SCADA systems attacks using honeypots. *Future Internet*, *15*(7), p.241.
7. Mubarak, S., Habaebi, M.H., Islam, M.R., Rahman, F.D.A. and Tahir, M., 2021. Anomaly Detection in ICS Datasets with Machine Learning Algorithms. *Computer Systems Science & Engineering*, *37*(1).
8. Rakas, S.V.B., Stojanović, M.D. and Marković-Petrović, J.D., 2020. A review of research work on network-based scada intrusion detection systems. *IEEE Access*, *8*, pp.93083-93108.
9. Ahakonye, L.A.C., Nwakanma, C.I., Lee, J.M. and Kim, D.S., 2023. SCADA intrusion detection scheme exploiting the fusion of modified decision tree and Chi-square feature selection. *Internet of Things*, *21*, p.100676.
10. Djenna, A., Harous, S. and Saidouni, D.E., 2021. Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. *Applied sciences*, *11*(10), p.4580.
11. Ajayi, R., 2025. Integrating IoT and cloud computing for continuous process optimization in real-time systems. *Int J Res Publ Rev*, *6*(1), pp.2540-2558.
12. Kumar, A., Radhakrishnan, R., Sumithra, M., Kaliyaperumal, P., Balusamy, B. and Benedetto, F., 2025. A Scalable Hybrid Autoencoder–Extreme Learning Machine Framework for Adaptive Intrusion Detection in High-Dimensional Networks. *Future Internet*, *17*(5), p.221.
13. Shukla, S., Hassan, M.F., Tran, D.C., Akbar, R., Paputungan, I.V. and Khan, M.K., 2023. Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR). *Cluster Computing*, *26*(5), pp.2657-2680.
14. Jurcut, A.D., Ranaweera, P. and Xu, L., 2020. Introduction to IoT security. *IoT security: advances in authentication*, pp.27-64.
15. Micheal, D., 2025. Resilient Cyber Defense: A Multilayer Approach to Preventing Intrusions in Distributed Environments Using Encryption and Deep Learning.
16. Ademilua, D.A., 2021. Cloud Security in the Era of Big Data and IoT: A Review of Emerging Risks and Protective Technologies. *Communication In Physical Sciences*, *7*(4), pp.590-604.
17. Ara, A., 2022, May. Security in supervisory control and data acquisition (SCADA) based industrial control systems: challenges and solutions. In *IOP Conference Series: Earth and Environmental Science* (Vol. 1026, No. 1, p. 012030). IOP Publishing.
18. Nwachukwu, C., Durodola-Tunde, K. and Akwiwu-Uzoma, C., 2024. AI-driven anomaly detection in cloud computing environments. *International Journal of Science and Research Archive*, *13*(2), pp.692-710.