

IDS Report.docx



Document Details

Submission ID

trn:oid:::29427:448226331

Submission Date

Apr 11, 2025, 10:25 PM UTC

Download Date

Apr 11, 2025, 10:27 PM UTC

File Name

IDS_Report.docx

File Size

919.4 KB

51 Pages

11,164 Words

66,380 Characters



0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

- 
1 AI-generated only 0%
 Likely AI-generated text from a large-language model.
- 
2 AI-generated text that was AI-paraphrased 0%
 Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



Machine Learning-Based Intrusion Detection System for Enhanced Network Security

Abstract:

IDS technology functions as an essential system which detects suspicious activities within network-based environments. This research presents a Machine Learning-based IDS framework designed for general-purpose enterprise networks, using the TII-SSRC-23 dataset to train and evaluate multiple supervised classifiers. Models including Random Forest, Gradient Boosting, XGBoost, SVM, KNN, and Logistic Regression were tested. Random Forest and Gradient Boosting emerged as the top performers and were further optimized using GridSearchCV. After tuning, Gradient Boosting achieved the highest accuracy (98.15%), while Random Forest recorded the best ROC AUC score (0.997), indicating strong classification across all thresholds. Feature importance analysis identified Total Forward Packets as the most influential feature for detecting high-volume attacks such as DoS and botnets. Random Forest was selected as the primary model due to its high accuracy, interpretability, and suitability for real-time implementation in diverse network environments. The results validate that ensemble learning models can significantly enhance intrusion detection by reducing false positives and identifying sophisticated threats. This research supports the development of scalable, accurate, and adaptive IDS solutions for enterprise use. Future work will explore integration with encrypted traffic analysis, adversarial robustness, and deep learning techniques such as RNNs and transformers to further improve detection capabilities in dynamic and evolving network environments.

Contents

Abstract:	1
List of Tables	3
List of Figures:	4
Lists of Abbreviations	4
Chapter 1: Introduction	5
1.1 Background	5
1.2 Motivation	6
1.3 Problem Statement	7
1.4 Research Objectives	9
1.5 Research Questions:	9
1.6 Thesis Structure	11
Chapter 2 Literature Review	12
2.1 Introduction	12
2.2 Intrusion Detection Systems (IDS)	12
2.3 Machine Learning for Intrusion Detection	12
2.4 Challenges in ML-Based IDS	13
2.5 Recent Advances and Research Gaps	14
2.6 Conclusion	18
Chapter 3: Research Methodology	19
3.1 Introduction	19
3.2 Research Design	19
3.2.1 High-Level System Architecture	21
3.3 Dataset Selection	23
3.4 Data Preprocessing	24
3.3.1 Handling Missing Values	24
3.3.2 Feature Scaling	24
3.3.3 Encoding Categorical Variables	25
3.4 Model Selection	25
3.5 Model Evaluation	26
3.6 Conclusion	26

Chapter 4 Experimental setup and Implementation	28
4.1 Introduction	28
4.2 Tools and Technologies Used	28
4.3 Data Preprocessing	29
Data visualization after cleaning	29
4.4 Data Splitting:	32
4.4.1 Low-Level System Workflow	32
4.5 Machine Learning Model Implementation for Intrusion Detection	33
4.6 Hyperparameter tuning	35
4.7 Model evaluation	37
4.8 Conclusion	38
Chapter 5: Results and Discussion	40
5.1 Introduction	40
5.2 Results	40
Feature Important analysis	41
5.3 Discussion	43
5.3.1 Limitations	45
5.4 Conclusion	46
Chapter 6 Conclusion and Future Work	46
6.1 Conclusion	46
6.2 Future Work	48
References	48

List of Tables

Table 1 Gaps analysis from previous Researches	16
Table 2 Performance Evaluation Metrics.....	26
Table 3 Tools and Technologies Used	28
Table 4 Model Performance Comparison	34
Table 5 Tuned Model Performance Comparison	35
Table 6 Comparison of Feature Importance with Previous Research	43

List of Figures:

Figure1 Methodology Pipeline for Machine Learning-based IDS	20
Figure 2 high-level design	22
Figure 3 Feature Distribution.....	30
Figure 4 Boxplot of Numerical Features	31
Figure 5 the feature relationships by displaying different clustering patterns.....	31
Figure 6 Protocol Distribution	32
Figure 7 Data splitting	32
Figure 8 Low-Level System design	33
Figure 9 Visual Comparison of Machine Learning Model Performance on IDS Classification ..	35
Figure 10 Confusion matrix of Random forest (tuned)	36
Figure 11 Confusion matrix of gradient boosting (tuned)	37
Figure 12 Tuned Models Evaluation Metrics	38
Figure 13 Figure 13 Feature important for Random forest.....	41
Figure 14 Feature importance for Gradient Bosting	42

Lists of Abbreviations

Abbreviation	Full Form
IDS	Intrusion Detection System
ML	Machine Learning
RF	Random Forest
GB	Gradient Boosting
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
LR	Logistic Regression
XGBoost	Extreme Gradient Boosting
ROC AUC	Receiver Operating Characteristic – Area Under Curve
F1 Score	Harmonic Mean of Precision and Recall
DDoS	Distributed Denial of Service
IoT	Internet of Things
TII-SSRC-23	Technology Innovation Institute – Secure Systems Research Centre Dataset (2023)
CV	Cross-Validation

Chapter 1: Introduction

1.1 Background

Today in hyper connected digital world, there has been an exponential rise in internet enabled systems and services that this means a lot of surface for potential cyberattacks. Top heavy corporations and enterprises cannot operate without network infrastructure, and neither can governments, educational institutes and healthcare providers. Today, and with greater frequency and intensity, cyber threats target the increasingly cyber dependent digital systems that enable both the functions of business, public and military, as well as the lives of citizens, businesses, and individuals (Lee, 2021)

Distributed Denial of Service (DDoS), ransomware, phishing, privilege escalation and zero day exploits have been elevated to more sophisticated and harder to detect attacks than those used with traditional security techniques. Conventional perimeter-based defenses like firewalls and rule based Intrusion Detection Systems (IDS) that are currently deployed to minimize the risk of an attack rely heavily on the existing, predefined attack signatures (Xiao, 2024). Generally, these legacy systems are not able to detect or mitigate known, as well as previously unknown, attack vectors that threaten critical infrastructures.

The limitations of static IDS have made Machine Learning (ML) techniques become a very powerful solution to integrate. IDS based on ML can learn from historical traffic patterns to realize anomalous behavior either in real time or in near real time to detect known and unknown threats. In contrast to traditional systems, ML approaches are essentially data driven, dynamic, and are able to generalize to the extents beyond the rules (Karthikeyan, 2024).

We propose in this study to apply Machine Learning in developing an Intrusion Detection System based on TII-SSRC-23 dataset, a rich pool of labeled network flows that include both normal and attack traffic. The system has an organized pipeline consisting of preprocessing such as data cleaning, normalization, encoding of many categorical features using sklearn one hot encoder, and

correlation based feature selection. Then, it uses several ML algorithms, like Random Forest, Gradient Boosting, XG Boost, Logistic regression, K Nearest Neighbors (KNN), and Support Vector Machines (SVM) in order to classify the traffic and find possible intrusion.

The implemented and tested system is implemented on a Python environment using scikit-learn and XGBoost libraries. To evaluate performance, widely accepted metrics are used to find the most effective model such as: Accuracy, F1 Score, Precision, Recall, ROC AUC. The system tries to reduce the number of false positives while also improving detection accuracy and allowing proactive security measures in the enterprise.

Finally, this research contributes to the best body of knowledge as to how intelligent IDS design should be maintained, providing a scalable and flexible solution that can be included into present day network infrastructures to reinforce security monitoring and response functionality.

1.2 Motivation

In today's world of digital systems within every organization, the concept of network security has become a mission critical issue (Sun, 2024). With the continuous growing of the cyberattacks, from the zero day exploits to the advanced denial of service campaigns, the Intrusion Detection Systems have shown their limitations. Most of these systems are based on static, rule based or signature based mechanisms, which are appropriate only to well known attack patterns. Indeed, traditional IDS has difficulty in detecting novel or previously unseen threat and allows malicious activity to bypass organizational defenses (Aygul, 2024).

One of the promising alternative is the development of an IDS based on Machine Learning (ML), and assuming the use of a data driven methodology to detect an anomalous patterns in network traffic. In contrast with traditional systems, ML models can generalize on existing attack data, and generalize in the sense that they can generalize to recognize new threats and thus improve proactive detection as well as real time responsiveness. ML based systems learn from traffic features with time and then classify malicious behavior better and down the rate of false positives (Sarker, 2023).

The main reason for doing this research is the necessity to develop such a framework of an adaptive, intelligent intrusion detection that works well in a dynamic network environment. Like any ML model, these models need to be trained and validated and a good foundation to get started with such a task is the TII-SSRC-23 dataset, which contains a wide range of labeled attack types and normal network flows. This has diversity and realism which makes it optimal for evaluation of detection systems in enterprise like scenarios.

This study aims to develop and assess a Machine Learning based IDS by using Random forest and Gradient boosting classifier for the sake of enhancement of detection accuracy and minimization of false alarms. The research aims to compare the performance of machine learning in terms of evaluation metrics and show how can machine learning improve scalability, accuracy and reliability of intrusion detection mechanisms. In the end, this work fills the pressing need for more robust, more flexible and immune cybersecurity solutions than those now in vogue.

1.3 Problem Statement

Modern cyberattacks have grown so complex and big that standard Intrusion Detection Systems (IDS) cannot prevent them as they were meant to from safeguarding contemporary network environments. Most times these systems use static rules or known attack signs to identify threats. Signature based IDS will be successful to the attack, which the IDS have faced but they fail when they are not able to detect zero day attack, polymorphic malicious code, and anomalous behavior. Rapid turnover of signatures in a signature database means they do not stay current and systems are kept reactive not proactive.

Before going further, it is obligatory to state the difference between the usual IDS and the traditional IDS. In general, conventional IDS refers to systems that lack intelligent or adaptive learning capabilities and includes rule-based and also signature based detection mechanisms. On the other hand, traditional IDS refers to static, signature based systems that dedicate to detect regular attacks (predefined attack patterns). Even though both of them are not flexible, traditional IDS are weak in terms of their lack of adaptiveness—they are rigid as they do not change with evolving threats or the continuous evolution of network behavior. In modern enterprise

environment where real time adaptability and resilience are imperative, their ability to not be able to self update or learn from new traffic patterns, reduces dramatically their effectiveness.

Furthermore, the rising number of data volumes with the advent of cloud service, mobile connectivity and IoT devices has exhausted the traditional Intrusion Detection Systems (IDS). Because these are rule based systems that depend on manual configurations, they are difficult to handle high speed and high volume of traffic in real time. Thus, in practice, detection delays are the norm and attackers can use long even before security teams can respond. This is very dangerous for highly sensitive sectors like healthcare, finance, and government because those are breaches that can lead to legal, financial, and social damage.

A main concern is that traditional IDS generates high rate of false positives as well as false negatives. Often time, security teams have too much inaccurate threat signals, leading them to alert fatigue and also doing a disservice to responding to the real threats. Traditionally, IDS are designed as being static, limiting their ability to handle new types of traffic patterns and attacker techniques as the environment dynamic changes, i.e., workloads transiently distributed architectures.

For these reasons, and intuitive to advanced attackers, evasion strategies such as traffic encryption, payload obfuscation and protocol manipulation are being used to evade common detection systems. Weakening packet inspection and rule matching through techniques like traffic fragmentation, tunneling and adversarial attacks allow malicious activity be hidden within 'legitimate' network behaviour. Such signature based systems do not inspect the encrypted payloads or detect malicious intent inside of anomalous traffic flow patterns.

Additionally, as the size of the organizations' IT infrastructure grows, so does the requirements for the security systems to be able to increase without sacrificing speed in detecting the security events as well as accuracy. But, as with evolved infrastructure and numerous endpoint devices, traditional IDS are not easy to scale and cannot keep up. Critical systems are vulnerable and unprepared because they cannot adapt and evolve to the emerging threats.

Against this background, it has become evident that conventional IDSs are inadequate. Again, there is an urgent need for an advanced, intelligent IDS framework that can identify known, as well as previously unseen threats in the time the threats emerge. The constant stream of traffic is

another reason why using data driven techniques based on machine learning to continuously learn about patterns of historical and live traffic are a promising alternative to Machine Learning based IDS. Such systems can be made to autonomously adapt to new forms of attacks, substantially reduce false alerts and provide faster and more accurate intrusion detection. Combining supervised and unsupervised ML models results in improved anomaly sensitivity and thus improves the robustness of the network defense mechanisms typical of modern enterprise.

1.4 Research Objectives

The aim of this research is to design and evaluate a Machine Learning-based Intrusion Detection System (ML-IDS) capable of identifying both known and unknown attacks using adaptive, data-driven models trained on real-world enterprise network traffic.

The objective of this research:

- To develop a supervised Machine Learning-based IDS framework using the TII-SSRC-23 dataset, incorporating realistic and labeled network traffic flows for system training and evaluation.
- To implement and compare the performance of multiple machine learning classifiers—including Random Forest, Gradient Boosting, XGBoost, SVM, KNN, and Logistic Regression—within the IDS pipeline.
- To evaluate the detection performance of the proposed ML-IDS using standard classification metrics: Accuracy, F1 Score, Precision, Recall, and ROC AUC, and identify the most effective model for real-time detection.
- To analyze the limitations of traditional IDS approaches by comparing them with ML-based techniques, highlighting how machine learning improves detection speed, accuracy, and adaptability to new threats.
- To investigate the effect of feature preprocessing and selection techniques (such as encoding, scaling, and correlation analysis) on overall model accuracy and system efficiency.

1.5 Research Questions:

This study examines these three critical research questions about intrusion detection challenges together with machine learning solutions evaluation.

RQ1: How machine learning techniques enhance intrusion detection system efficiency by improving their capability for identifying cyber threats.

RQ2: What are the best machine learning approaches that will enable an efficient and scalable intrusion detection system development?

RQ3: What obstacles prevent the implementation of ML-based IDS in actual network environments particularly related to computation and scalability?

1.6 Project Scope

This research is not an oriented research on any specific domain such as IoT, cloud computing or industrial control systems. It is instead directed towards a general purpose enterprise network environment usually seen in organizational, academic, and campus based environments. Such networks usually link a large array of end point and infrastructure collection of devices, for example desktop and laptop computers, smartphones, tablets, servers, printers, network swallows, routers, and VoIP phones. In such highly trafficked environment with different communication protocols, the problem of reliability and availability is essential.

By choosing such a network as the scope of the study, the research maintains wide applicability outside of such constrained environments. Based on the fact that these enterprise networks show widely varying user behavior and access privileges, they become excellent subjects of study for evaluating scalable and adaptive intrusion detection systems (IDS). Its aim is to design a machine learning based IDS to efficiently detect intrusions in real time on a wide scope of normal and abnormal traffic flows in enterprise grade systems.

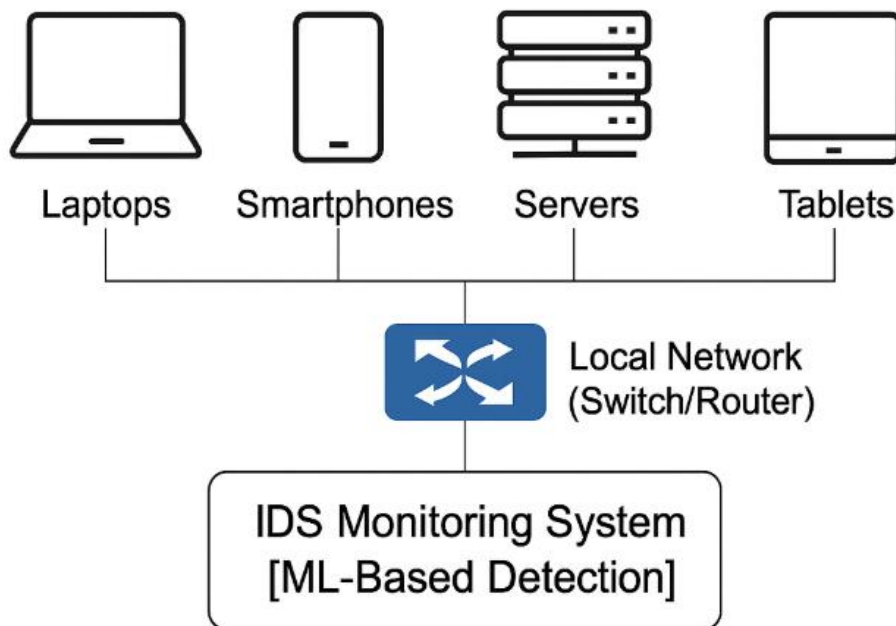


Figure 1 Target Enterprise Network Environment for Testing the IDS

A dataset TII-SSRC-23 (also known as USCB dataset) selected because the traffic patterns captured by it are a variety that can be generated by enterprise devices such as workstations, mobile devices and servers, and it is applicable to evaluate intrusion detection in the environments with all type of endpoints.

1.7 Thesis Structure

The research follows this structure:

Chapter 2: Literature Review – Discusses existing IDS models, machine learning techniques in cybersecurity, and related research.

Chapter 3 outlines the methodology by breaking down the information regarding dataset preprocessing together with feature selection and the ML model types employed.

In Chapter 4 the reader will find technical information about building the ML-based IDS with focuses on model training processes and tuning methods.

The chapter 5 displays experimental findings while assessment of model execution and examination of collected data appear in this section.

In chapter 6 the final part of the study compiles essential findings into a conclusion followed by plans for future research development.

The research motivation along with study objectives receive description in this chapter. The proposed ML-based IDS receives analysis through a sequence of chapters which include theoretical foundation analysis with experimental evaluation and methodological assessment.

Chapter 2 Literature Review

2.1 Introduction

In this chapter, he reviews the literature in intrusion detection systems, the strengths and drawbacks of traditional methodologies, integration of ML based methodologies and the reasons why ML based IDS are not used more often.

2.2 Intrusion Detection Systems (IDS)

According to two common classifications into signature based detection and anomaly based detection. An example of signature based IDS are IDS's which compare the known attack signatures stored in a database. This method is used in tools like Snort and Suricata with high accuracy in defeating previously known threats. Nevertheless, these systems are unable to detect zero day attack or novel threats as this requires constant updates with the signature databases (Heidari, 2024).

At the other side, the IDS comprise anomaly based IDS, which is monitoring the network behavior to detect anomalies on the norms. The stats models and ML are used in these systems to detect unknown attacks. However, emerging threats can be detected using anomaly detection methods, however, this is at the expense of high false positives as legitimate network activities are sometimes misclassified as malicious (Albulayhi, 2022). To deal with this problem, researchers proposed such hybrid IDS models mixing the signature based and anomalous ones, taking benefits from two approaches (Awotunde J. B., 2021).

2.3 Machine Learning for Intrusion Detection

IDS has been applied increasingly to machine learning for improving detection accuracy and adapting to the changes in intrusion detection. IDS based on ML can be treated as supervised, unsupervised and deep learning methods.

A supervised learning techniques to classify the attacks need two things: labeled datasets to train models. To complement literature, decision trees (DT) and random forest (RF) classifiers are popular in IDS because of their interpretability and speed. But DT models is prone to overfitting while RF model needs a lot of computational resource to handle the network data on large scale (Ahmad Z. S., 2021). In addition to Support Vector Machines (SVM), they have also been explored for intrusion detection problems—especially ones that are binary classification problems—but high dimensional data causes them to be less successful for real world problems (Yin, 2021)

K-Means clustering and Autoencoders are applied for unsupervised learning of patterns of network traffic, without requiring any labeled datasets. Anomalies are detected in kmeans clustering by identifying similar data points and identifying outliers. However, this method is effective, yet it comes with predefined cluster numbers, which may affect the detection accuracy (Samunnisa, 2022). A neural network based approach called autoencoders are also used to learn the normal traffic behavior of the networks and the deviations from that. Yet autoencoders are expensive in terms of computational cost for giant scale network monitoring (Lin, 2022) and are prone to overly careful parameters tuning.

In intrusion detection, deep learning approaches especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have good results. However, CNNs are very good at recognizing attack patterns in network traffic at the expense of requiring high computational resources. RNNs, such as Long Short Term Memory (LSTM) networks are used for data sequence and also time series intrusion detection. Yet, they are subject to vanishing gradient issues that can affect their performance on long term dependencies (Yue, 2021). Recently, Graph Neural Networks (GNNs) have been proposed to be a promising approach for intrusion detection in a network since they use graph structures to learn the complex relationship among network nodes (Abdallah, 2021)

2.4 Challenges in ML-Based IDS

While a number of challenges still remain in the field of machine learning in intrusion detection, the above achieves reasonable performance, regardless of the intrusion rule domain used. In fact the data quality and imbalances are one of the worst problems. However, the majority of IDS datasets available in public (such as CICIDS-2017), are extremely imbalanced classes having

normal much more than the attack data. Unfortunately, attacker payload can only be hard to detect and rare (but critical), on the other hand, normal traffic and biased ML models rely on according to (Jose, 2023).

This is another major point of concern. One limitation with anomaly-based IDS is that most of the models generate high false alarms thus creating an alert fatigue for security analysts. One of the key research focus in ML based IDS (Ahmad Z. S., 2021) is to reduce false positives while ensuring high detection accuracy.

It is also challenging to apply real time IDS. High throughput network traffic is essentially challenging to traditional ML models to handle. This problem is addressed by the distributed and federated learning approaches that allow IDS to be at multiple network nodes without centralization of all data (Umer, 2022).

Moreover, adversarial attacks are becoming a big threat to ML based IDS. Attacking network traffic means that attackers are able to do something that was previously impossible, and is able to deceive ML models into misclassifying when they should not and into evasion. Currently, it is an area of ongoing research around the development of robust IDS models that are resistant to adversarial perturbations (Awotunde J. B., 2021)

2.5 Recent Advances and Research Gaps

To enhance IDS performance, researchers have looked into hybrid models combining wide range of ML methods. As an example, ensemble learning constructs by combining multiple classifiers in order to improve the detection accuracy and robustness. In order to reduce computational overhead and improve interpretability, feature selection techniques have been used in this work as well (Almotairi, 2024)

Nevertheless, some points still remain to be addressed through research. Most existing IDSs based on ML are not adaptable to emerging cyber threats. Therefore such adaptive IDS solutions still require continuous learning from the new attack patterns given and need not be retrained frequently. Finally, combining ML based IDS with cloud security frameworks provides scalability and the ability of sharing real time threat intelligence (Lefoane, 2021)

In order to achieve good performance on KDDCUP'99 as well as CIC_Mal_Mem-2022 imbalanced dataset, (Alamin Talukder, 2022) suggested a hybrid approach combining XGBoost and SMOTE. The achieved accuracy was very high with results reported at 99.99% and 100% which indicates good classification of network intrusions. But the high performance metrics are suspiciously good, and in light of the limited evaluation to other more recently available or diverse datasets, these figures are suspiciously good. However, the study would have been aided by demonstrating the benefit of sampling techniques used with such powerful ensemble models as XGBoost for feature selection as well as classification.

As they reflected on the existing anomaly detection methods, (Kale, 2022) introduced a layered detection pipeline integrating GANomaly (a GAN based anomaly detector), K mean clustering and CNNs. Finally, NSL-KDD, CIC-IDS2018, and TON_IoT datasets are used to test this system. The approach succeeds at combining three learning paradigms (unsupervised, semi-supervised, and supervised), but achieved variable success, having modest performance of 67.7–87.2% depending on dataset. This study also points out that high computational complexity is a limiting factor, thus limiting applications in the real time domain. In fact, it demonstrates the application of layered architectures for enhancing the precision of detection of complex network threats.

This (Chua, 2022) research employed dedicated ICS security dataset and focused on models including LSTM, GRU, and Random Forest on ICS. The study examined the detection performance as well as application specific challenges of real time data processing and system constraints with an accuracy of 94.3%. Although encouraging, this has a limitation of being domain specific (limited generalization beyond the enterprise or cloud based networks). The work shows that while valuable, testing such a model requires testing across multiple domains, as there is no way to validate claims regarding ICS environments.

(Ahmed, 2025) used UNSW-NB15 dataset and researched such integration of the signature based IDS with machine learning and Deep learning models like Random Forest, SVM, and LSTM. The paper claims that a hybrid strategy allows better detection at the cost of constant rule update that limits scalability and adaptability for signature based systems. Specific accuracy metrics were not specified, but the study provides a useful view on bridging traditional and AI based approaches in IDS by using ML/DL to supplement static rule-based detection.

(Mohale, 2025) looked into the role of Explainable Artificial Intelligence (XAI) in increasing transparency and trust in IDS models in an examination of the literature in this systematic review. Despite that, the study critically evaluated different XAI frameworks and their integration with the existing ML models, but there was no reporting on the performance metrics. Integration complexity was one limiting factor and more so when we went to integrate the XAI in real time detection pipelines. Yet, the research indicates the rising interest in the interpretability in security ML and the potentials of XAI for greater analyst trust and decision making.

To explore typologically about intrusion detection, (Hassini, 2024) evaluated several classifiers such as XGBoost, Extra Trees, and Deep SVDD on the TII-SSRC-23 dataset. Thus, the study demonstrated a strong baseline for IDS performance, with the accuracies of 98.79% (XGBoost) established. Finally, it identified model-specific trade-offs such as SVDD's resource requirements and the dependence of Extra Trees on redundancy in the features. The study was further important to show the critical importance of flow and protocol based features to improve detection accuracy.

Table 1 Gaps analysis from previous Researches

Reference	Technique	Model	Accuracy	Dataset	Limitation	Key Findings
(Alamin Talukder, 2022)	Hybrid ML and DL Model	SMOTE + XGBoost	99.99%, 100%	KDDCUP'99, CIC-MalMem-2022	Potential overfitting	Combines SMOTE and XGBoost for feature selection
(Kale, 2022)	Hybrid Deep Learning Framework	GANomaly + K-means + CNN	NSL-KDD, 87.2%, CIC-IDS2018, 67.7%, TON_IoT (CIC-IDS2018), 86.5% (NSL-KDD)		High computational cost	Integrates unsupervised, semi-supervised, and supervised learning

(Chua, 2022)	ML for Industrial Control Systems	LSTM, GRU, RF	94.3%	ICS Security Dataset	Limited to ICS environments	Provides applications, challenges, and recommendations
(Ahmed, 2025)	Signature-Based IDS with ML/DL	RF, SVM, LSTM	Not mentioned	UNSW-NB15	Requires constant updates	Combines ML and DL for intrusion detection
(Mohale, 2025)	Explainable AI in IDS	XAI-based ML models	Not reported	CIC-IDS-2023	Integration complexity	Systematic review of XAI in IDS
(Herzalla, 2023)	Typological Exploration for IDS	XGBoost, Extra Trees, Deep SVDD	98.79% (XGBoost), 93.36% (Extra Trees), 97.84% (Deep SVDD)	TII-SSRC-23	Model-specific limitations	Established baseline IDS performance; Identified critical IDS features
(Hassini, 2024)	Intrusion detection using CNN1D	Lightweight CNN1D	100% Accuracy, Precision, and F1-score	TII-SSRC-23	The study fails to demonstrate how deployment challenges specifically affecting	The proposal introduces an efficient lightweight CNN1D model specifically for securing IIoT security while

					edge devices would be addressed.	CTGAN is used to balance data for enhanced performance.
--	--	--	--	--	----------------------------------	---

2.6 Conclusion

The literature on intrusion detection systems has been reviewed, particularly machine learning based approaches that were the main interest of this chapter. However, traditional IDS solutions are efficient at identifying known attacks but do not work well in detecting new attacks. However, ML based IDS has potential to make great progress, but it is hindered by false positives, scalability and adversarial robustness problems. Future research can be aimed to develop adaptive, scalable and adversarially robust IDS frameworks that could deal effectively with the contemporary cyber threats. In the next chapter, research methodology is presented with associated dataset, feature selection process, as well as machine learning models applied for IDS evaluation.

Chapter 3: Research Methodology

3.1 Introduction

The development of a machine learning-based intrusion detection system (IDS) for network security enhancement follows the research methodology described here. The research design explains how the authors approached their work together with their selection of dataset and data preprocessing steps and their methods for feature selection and model selection and evaluation metrics and implementation system.

3.2 Research Design

The research methodology implements quantitative experimental method to identify malicious network traffic through supervised machine learning methods. The main goal focuses on creating an intrusion detection system that recognizes differences between standard network behavior and harmful network activity. Various stages in the research methodology create an efficient systematic workflow that will guide the research process show in figure 2.

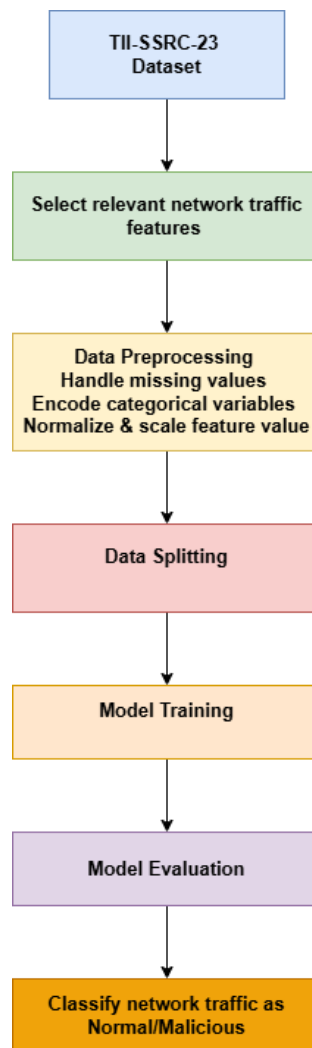


Figure2 Methodology Pipeline for Machine Learning-based IDS

The initial part of the process requires acquiring network traffic data followed by data conditioning to suit machine learning needs. The preprocessing stage includes three key steps that involve treating missing values for stability and applying numerical scaling approaches and converting non-numeric data through categorical encoding for machine learning compatibility. Model input quality improves substantially through proper preprocessing of data because it directly affects predictive performance results.

After preprocessing the dataset operations proceed to data splitting procedures which separate the data into training and testing subsets. The established 80:20 data split enables models to gain knowledge from most original data without sacrificing any evaluation data. A distinct partitioning

of data serves as a necessity to determine how the model performs when presented with new information.

Model selection and training constitute the third phase of implementation that involves using a Random Forest Classifier as part of a machine learning pipeline. The Random Forest algorithm becomes the selection because it demonstrates three vital characteristics including robust operation alongside high-dimensional data input and ability to detect intricate traffic patterns. Through pipeline processing the crucial data preparation techniques get automated which results in simplified training operations.

The evaluation process includes testing the trained model by measuring accuracy in combination with F1-score, precision recall Roc AUC and creating classification reports. Accuracy calculates total correct predictions but the F1-score combines precision with recall to effectively deal with datasets that are unbalanced. The classification report provides extensive details regarding the correct identification of normal and malicious traffic instances by the model. The research design structure enables a thorough method to develop security-enhancing network intrusion detection systems.

The system being developed is a Machine Learning-based Intrusion Detection System (ML-IDS) tested using the TII-SSRC-23 dataset within the Kaggle cloud environment. It operates in a simulated enterprise-like network context using Python and scikit-learn pipelines.

3.2.1 High-Level System Architecture

The general architecture of the proposed Machine Learning based Intrusion Detection System (IDS) is shown in Fig. 3. Raw network traffic is ingested and features are extracted then processed with operations such as scaling, cleaning, encoding etc. Then, the system features selection is conducted by using correlation and importance measures, and then the data is sent to a group of machine learning classifiers such as Random Forest, Decision Tree, SVM, KNN, Logistic Regression, XGBoost, and Gradient Boosting. At the last stage, the final outputs of the detection are in the forms of alerts, logs, or classification reports based on model predictions.

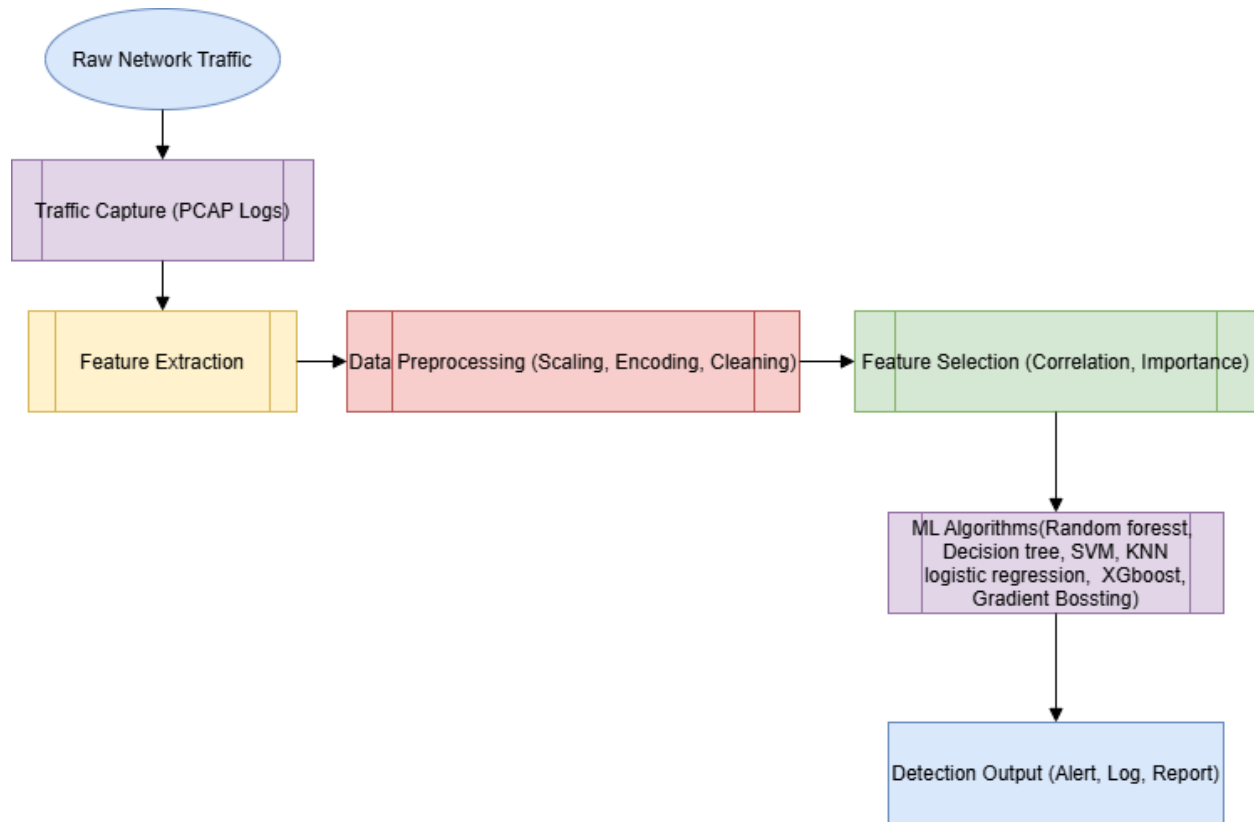


Figure 3 high-level design

Overall working of high level design are mentioned below:

1. Raw Network Traffic

This is the real time data received from various devices and applications in the network as a data packet. It is composed of both normal and potentially malicious traffic.

2. Traffic Capture (PCAP Logs)

This module collects the raw packets in PCAP (Packet Capture) format, it is captured using packet sniffing tools. However, it records the log of traffic needed for feature extraction.

3. Feature Extraction

Raw packet logs are the source where the key attributes are extracted including protocol type, flow duration and packet counts. The machine learning model takes in the inputs of these features.

4. Data preprocessing

Handling of missing values, scaling of numerical data using scaling and encoding of categorical variables to numerical format are steps that make part of this step as we prepare the data for model training.

5. Feature Selection (Correlation, Importance)

The main goal is to reduce the dimensionality of the data by selecting only the most useful features based on correlation and feature importance. A better model efficiency and prediction accuracy is achieved.

6. ML Algorithms (Random Forest, Decision Tree, SVM, KNN, Logistic Regression, XGBoost, Gradient Boosting)

Different supervised learning classification methods are trained and compared. The selected features are processed by each model to learn and predict malicious pattern in the network.

7. Alert, Log, Report (Detection Output)

Results such as intrusion alerts, forensic logs logs or classification reports of detection accuracy and type of threats are generated by the system.

3.3 Dataset Selection

The researchers decided on TII-SSRC-23 dataset because it offers a detailed representation of network traffic data containing both normal and malicious activities. The dataset includes a wide array of network flow characteristics needed for training an intrusion detection model. The dataset enables classification through supervised learning techniques because it provides instances with labeled values between normal and suspicious activities.

Multiple important features present in the dataset contribute to effective attack pattern recognition. The essential components among all features include:

The identification of communication protocols through the Protocol Type measurement is vital for traffic behavior evaluation.

The measurement of flow duration shows how long network connections last thus helping detect normal from abnormal traffic patterns in networks.

Total Forward Packets enables the discovery of potential flooding attacks by counting packets transmitted by the source system to its destination.

The detector utilizes this metric to determine backward packet quantity which supports analysis of abnormal two-way transmissions.

The dataset selection was made because it contains authentic network traffic data which allows the trained model to effectively adapt to varied cyber threat types. The proposed research uses this dataset to construct a strong intrusion detection system that will recognize security threats during live operations.

3.4 Data Preprocessing

Building an appropriate intrusion detection model requires data preprocessing to create an effective dataset ready for use. The study employs missing value handling as well as feature scaling and categorical encoding to keep the data reliable and consistent.

3.4.1 Handling Missing Values

Network traffic datasets face frequent data loss incidents from three main sources which are packet loss and network congestion together with incomplete logging protocols. The presence of uncleaned missing values disrupts model training processes which results in misidentified records. For numerical features the investigators used mean or median imputation while categorical features received the most frequent value (mode) as replacement. The removal of records containing substantial missing data served to minimize potential bias. Integrity checks on data combined with model reliability improvements are the direct results of this method implementation.

3.4.2 Feature Scaling

The diverse ranges of network features produce prediction bias because of different values. The measurement period of Flow Duration spans between milliseconds and seconds whereas packets follow varying number sequences from low to high counts. StandardScaler standardized the features to achieve equal contribution weights by creating a uniform value scale. The technique stops larger attributes from controlling the model performance by creating more efficient systems.

3.4.3 Encoding Categorical Variables

Because most machine learning algorithms need numerical input the categorical features like protocol type (TCP, UDP, ICMP) must undergo a numerical formatting process before usage. The conversion of categorical variables enables efficient classification by the model among different network traffic patterns.

The variables are transformed into numbers through Label Encoding which utilizes unique numerical tags for each category (TCP becomes 0 whereas UDP becomes 1 and ICMP is assigned as 2). The encoding bears an efficient design but generates unexpected order connections between classification groups.

The conversion of categorical variables through encoding practices helps the intrusion detection system achieve better model interpretability while also maintaining correct network traffic classification ability. The transformed categorical values enable the model to employ every accessible piece of data for detection purposes.

The preprocessing methods refine the dataset so it achieves better quality which meets machine learning requirements. The intrusion detection system achieves dependable and accurate detection of network traffic attacks because it handles missing values properly and performs feature rescaling and encoding operations.

3.5 Model Selection

An evaluation process of different machine learning models helps determine which method provides the best detection solution for network intrusions. The Random Forest Classifier functions as the main model selection because it demonstrates outstanding performance alongside its capability for processing diverse data dimensions and exhibits strong resilience against overfitting. The use of Decision Trees enables developers to see their models but leads to these models being easily exploited by training data. Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The Support Vector Machine (SVM) shows high ability in differentiating classes yet it needs high computational capabilities. The predictive accuracy improvement of Gradient Boosting Classifier depends on repeated decision tree enhancements while requiring detailed parameter modifications.

3.5 Model Evaluation

Table 2 Performance Evaluation Metrics

Metric	Definition	Formula
Accuracy	Measures the proportion of total correct predictions among all predictions made.	$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
Precision	Proportion of correctly predicted positive observations out of total predicted positives.	$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
Recall	Proportion of actual positive cases that were correctly identified by the model.	$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
F1 Score	Harmonic mean of Precision and Recall, useful in imbalanced datasets.	$\text{F1 Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$
ROC AUC	Measures the classifier's ability to distinguish between classes across all thresholds.	<i>Derived from ROC curve using TPR vs FPR</i>

3.6 Conclusion

In this chapter we describe how to build a Machine Learning based Intrusion Detection System (IDS) with a focus on structured data preparation, with a thorough testing of model evaluation. First, we began with the selection of the TII-SSRC-23 dataset which is a realistic and labeled dataset that enables the model to be trained effectively when it is given normal and malicious network traffic.

In fact, data preprocessing was an absolutely essential step - missing value imputation, feature scaling and categorical encoding to make sure that the data was tuned for supervised learning models. To make these transformations consistent and efficient in training, we implemented such a standardized pipeline for automation.

Then feature selection was conducted through correlation and importance based methods for model interpretability and reduce dimensionality after preprocessing. Multiple machine learning classifiers such as Random Forest, Gradient Boosting, XGboost, Logistic Regression, SVM and KNN were later implemented and evaluated using performance metrics such as Accuracy, F1 Score, Precision, Recall, and ROC AUC.

The methodology presented in this chapter gives some framework for the implementation of the system, which will be explored in the following chapter. It shows a systematized and scalable way towards making an IDS for real time network threat detection with better precision and flexibility.

Chapter 4 Experimental setup and Implementation

4.1 Introduction

The design of the Machine Learning-Based Intrusion Detection System (ML-IDS) receives thorough explanation through descriptions of its selected tools and technologies and methodologies. The research uses an overview of its programming languages library systems and computational environment for the initial section. A discussion is provided about the preprocessing pipeline which consists of data cleaning and feature encoding together with standardization to prepare the dataset for machine learning examination. The analysis employs Random Forest together with Decision Tree and Logistic Regression alongside Support Vector Machine (SVM) and Gradient Boosting to compare their results based on classification performance. The model accuracy receives optimization through hyperparameter tuning before researchers analyze the model performance through accuracy and F1-score metrics evaluation. This chapter establishes the necessary foundation which will guide the complete results overview in the following chapter.

4.2 Tools and Technologies Used

The deployment of the Machine Learning-Based Intrusion Detection System (IDS) requires different programming languages alongside machine learning frameworks and data processing tools. The IDS implementation depends on the following set of tools and technologies:

Table 3 Tools and Technologies Used

Category	Tools/Technologies Used	Purpose
Programming Language	Python	Used for machine learning model implementation and data processing.
Development Environment	Jupyter Notebook (Kaggle)	Executes and visualizes machine learning experiments.
Libraries Used	NumPy & Pandas	Handles data preprocessing, manipulation, and analysis.
	Scikit-learn	Implements machine learning models, feature selection, and evaluation.

	Matplotlib & Seaborn	Used for data visualization and feature importance analysis.
Computational Environment	Kaggle Cloud (GPU Support)	Provides high-performance computing for model training and evaluation.

4.3 Data Preprocessing

The first necessary step prior to applying machine learning analyzes is data preprocessing to convert raw data into a form suitable for training intrusion detection systems. The model performance improves because preprocessing methods strengthen its detection abilities for network traffic patterns.

Model accuracy suffers from incomplete data which needs handling as the very first processing step. The dataset contains missing entries which are fixed by three processes including imputation through statistical methods with simultaneous removal of entries to protect predictive integrity. StandardScaler() is used for feature scaling through which numerical data is transformed into standardized values to obtain uniform distribution. By scaling network traffic data one can stop values with large numbers from taking control over the machine learning process because features exhibit different magnitude levels.

The conversion of non-numerical Protocol Type values into numerical forms occurs through the application of LabelEncoder(). After the transformation process machine learning models succeed in processing categorical information through their algorithms.

Data visualization after cleaning

This figure 4 depicts the cleaned network flow features distribution. The distribution data reveals Protocol 1 stands as the most common category while the data separates into three protocols. A significant number of network flows tend to have shorter durations based on the "Flow Duration" distribution shape. Network flow data shows that a majority of packets exist within one of the two Total Fwd Packet or Total Bwd Packet distributions while extensive packet quantities remain scarce.

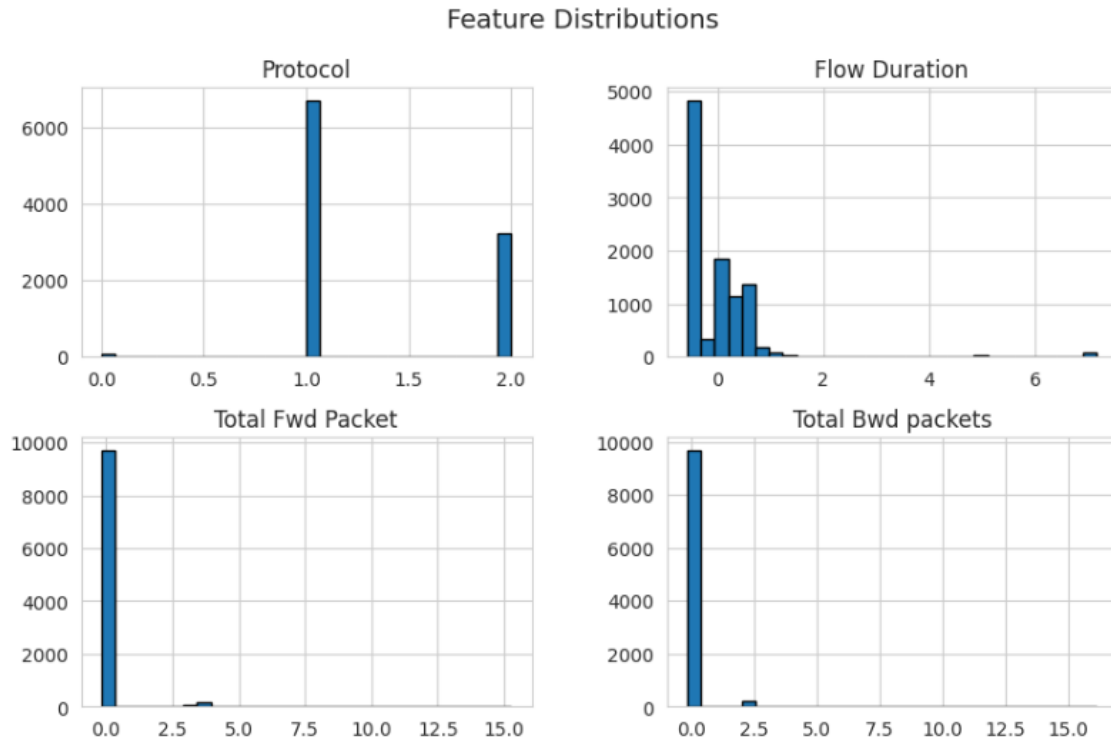


Figure 4 Feature Distribution

The box plot in figure 5 points to outliers in "Flow Duration," "Total Fwd Packet" and "Total Bwd Packet" because these variables show unequal distribution of data points.

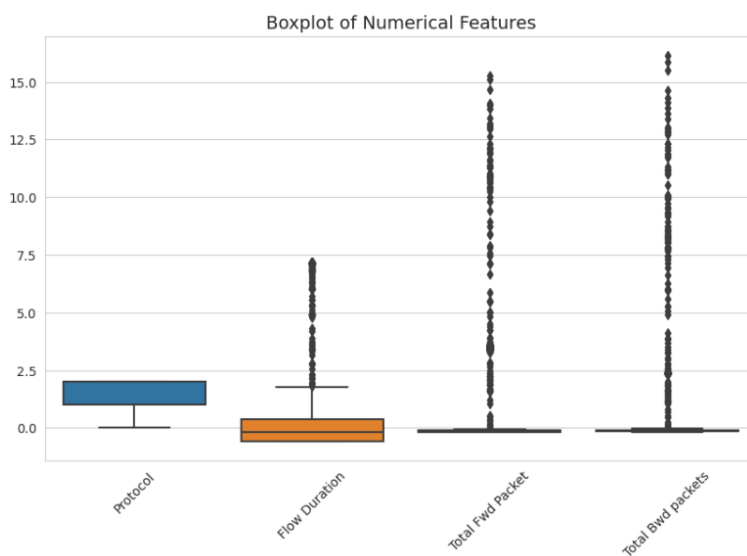


Figure 5 Boxplot of Numerical Features

Figure 6 in the pairplot visualizes the feature relationships by displaying different clustering patterns for different protocol types.

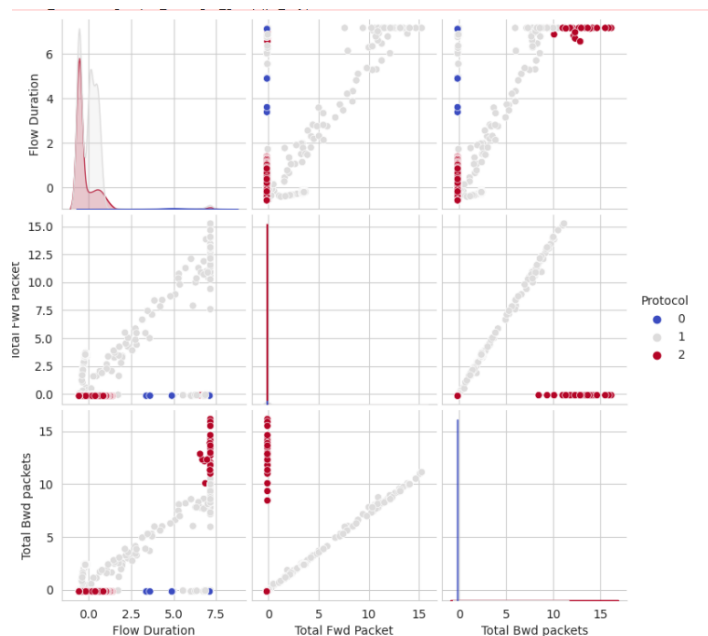


Figure 6 the feature relationships by displaying different clustering patterns

The illustration in figure 8 depicts how data cleaning modified the protocol type frequencies into categories 0, 1, and 2. Protocol 1 dominates the dataset occurrence followed by Protocol 2 with Protocol 0 being noted only rarely.

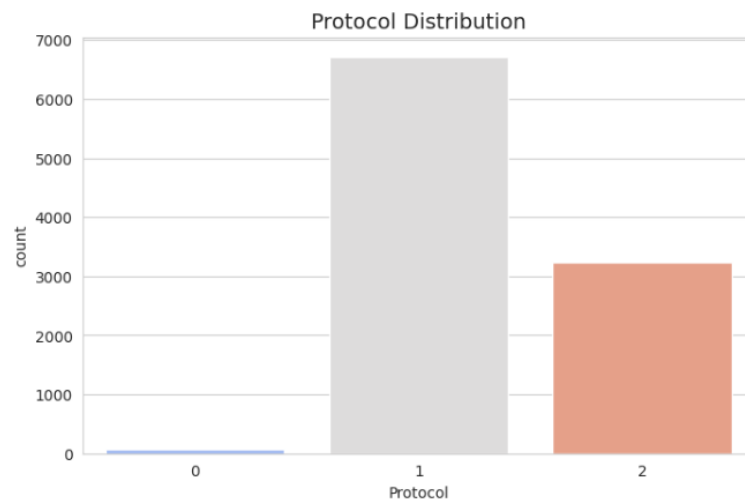
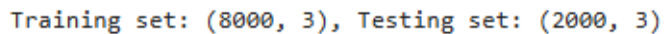


Figure 7 Protocol Distribution

4.4 Data Splitting:

A separation of data occurs through `train_test_split` from `sklearn` where 80% serves training needs while 20% exists for testing objectives with target variable "Protocol" as the stratification base. The trained dataset consists of 8,000 samples divided into 2,000 testing samples.



Training set: (8000, 3), Testing set: (2000, 3)

Figure 8 Data splitting

4.4.1 Low-Level System Workflow

Figure 9 shows the low level design of the implemented IDS system, which gives a picture of the technical operations processed by the machine learning pipeline. Data ingestion and raw packet log acquisition is the first step and then follow the process of step by step data cleaning handling the nulls and duplicates. Feature scaling using `StandardSclae` and correlation based feature selection are then done, followed by a few subsequent stages of label encoding. The system then runs different ML algorithms, figures out the right hyperparameters with the help of `GridSearchCV`, and measures evaluation using Accuracy, F1 Score, Precision, Recall and the ROC AUC. The Python implementation follows the practical logic as described by this diagram.

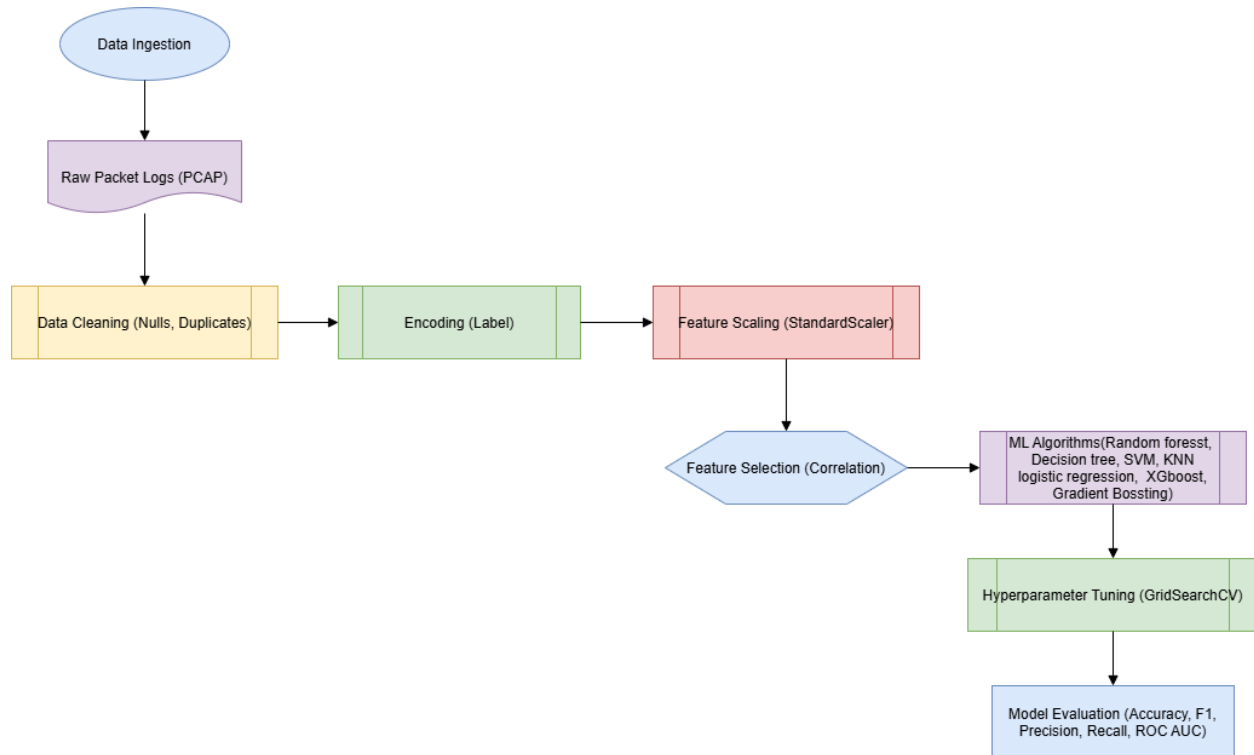


Figure 9 Low-Level System design

4.5 Machine Learning Model Implementation for Intrusion Detection

Additionally, employing this ML IDS entails multiple classification algorithms, testing and evaluation to determine the most suitable model to differentiate normal and malicious network traffic. All classifiers were implemented in a standardized pipeline, preprocessing, feature selection and training, thus making sure that they were both efficient and reproducible.

Among the tested models, the Random Forest Classifier is a strong baseline, for it is based on an ensemble learning, robust to overfitting and it exceeded the tested models in accuracy. As ensemble methods were more stable and less prone to overfitting when compared to Decision Tree Classifier, an ensemble method was selected.

Binary classification cases showed quite good results for Logistic Regression, while for complex and non linear traffic, Logistic Regression doesn't perform so well. The classification potential of the Support Vector Machine (SVM) was shown to be good however as datasets grew, it became computationally expensive.

Sequential learning to minimize error across iterations leads to a better detection accuracy and F1 scores by the Gradient Boosting Classifier. Finally, K-Nearest Neighbors (KNN), and XGBoost models were tested out and found XGBoost to be slightly better than all at many metrics.

Table below also provides the summarized performances of each classifier with five standard metrics including Accuracy, F1 Score, Precision, Recall and ROC AUC.

Table 4 Model Performance Comparison

Model	Accuracy	F1 Score	Precision	Recall	ROC AUC
Random Forest	0.9720	0.9486	0.9721	0.9720	0.9953
Decision Tree	0.9710	0.9778	0.9710	0.9710	0.9827
Logistic Regression	0.6710	0.3791	0.7761	0.6710	0.9186
Support Vector Machine	0.7095	0.5416	0.7879	0.7095	0.8882
Gradient Boosting	0.9810	0.9556	0.9815	0.9810	0.9949
K-Nearest Neighbors	0.9780	0.9285	0.9785	0.9780	0.9663
XGBoost	0.9820	0.9563	0.9825	0.9820	0.9966

Rebuilding this comparison, it is obvious that XGBoost had the hardest performance across most pertain metric such as Accuracy, Precision, Recall and ROC AUC therefore making it the best overall model provided for IDS. Gradient Boosting and Random Forest also showed excellent run but the key point is that they are also reliable for ensemble learning for the network intrusion detection.

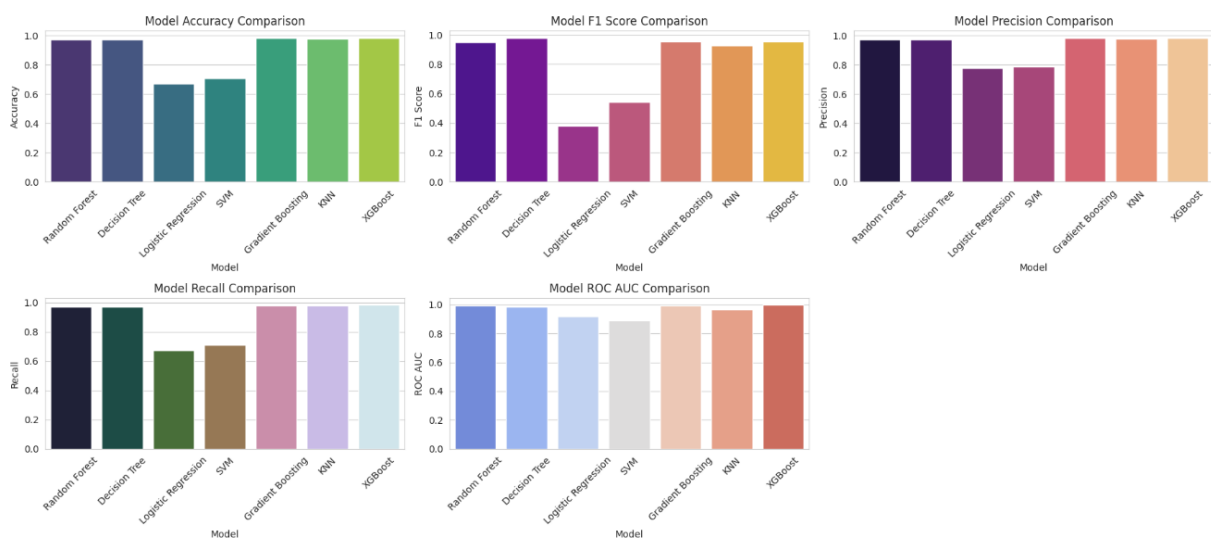


Figure 10 Visual Comparison of Machine Learning Model Performance on IDS Classification

The following figure 10 shows the relative performance of seven machine learning classifiers, namely Random Forest, Decision Tree, Logistic Regression, SVM, Gradient Boosting, KNN, and XGBoost, on five metrics for evaluating the performance of ML models, namely Accuracy, F1 Score, Precision, Recall, ROC AUC. Visualized bar plots illustrate that most of them achieved better result measured by Accuracy and Recall in Accuracy and Recall, with ensemble models like XGBoost and Gradient Boosting being ranking on top. However, Logistic Regression and SVM were relatively inefficient in terms of complex traffic patterns handling. XGBoost also demonstrates very high performance consistency, which means it is very reliable and generally applicable, thus it is indeed the most reliable model for intrusion detection in this study.

4.6 Hyperparameter tuning

Finally, the two top performing classifiers were hyper parameter tuned for further improvement in detector performance. GridSearchCV was used to tune the process, as it automates exhaustive parameter search through cross validation. It enabled better generalization on unseen data through a selection of optimal parameters for classification.

The best results were obtained for Random Forest using `n_estimators=50`, `max_depth=10` and `min_samples_split=10`. These settings made it easy for the model to achieve both high predictive performance and prevent overfitting. Likewise, for Gradient Boosting, best configuration was `n_estimators=100` (converges gradually, but with high accuracy) and `max_depth=5` and `learning_rate=0.01`.

Both the tuned models are reevaluated using the test dataset and the results are summarized in Table 4. Accuracy, F1 Score, Precision, Recall and ROC AUC, are the key performance metrics to validate the Intrusion Detection System as reliable and responsive.

Table 5 Tuned Model Performance Comparison

Model	Accuracy	F1 Score	Precision	Recall	ROC AUC
Random Forest (Tuned)	0.9810	0.9556	0.9816	0.9810	0.9970
Gradient Boosting (Tuned)	0.9815	0.9559	0.9820	0.9815	0.9780

However, both of the confusion matrices have strong classification capability on all three classes, with Gradient Boosting slightly miss classifying less instances than Random Forest. Although

Random Forest does not achieve the best, the ROC AUC is highest, indicating that Random Forest is the best algorithm which can accurately identify the positive vs. negative classes under various threshold values.

Overall, both models can accurately perform and Gradient Boosting just marginally has a better classification accuracy and F1 score, therefore it is the best model for this IDS implementation. Nevertheless, the higher AUC score given by Random Forest indicates that it is still a strong alternative, and in cases where ranking probabilities of malicious activity should be possible.

Further analysis of classification performance at the class level will be done below in figure 11 and 12 with the confusion matrices for both models.

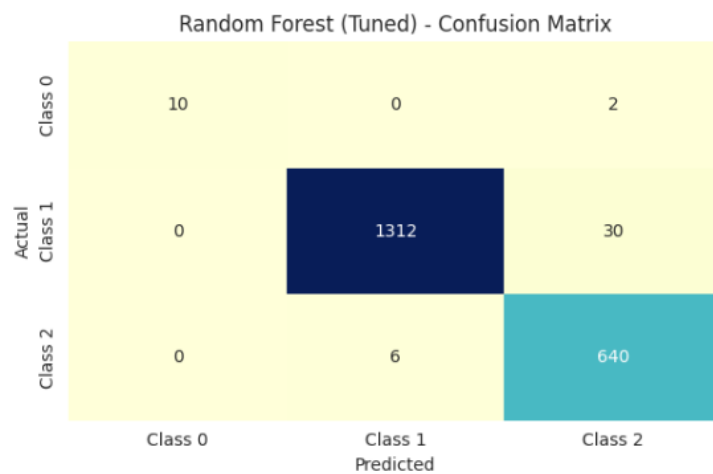


Figure 11 Confusion matrix of Random forest (tuned)

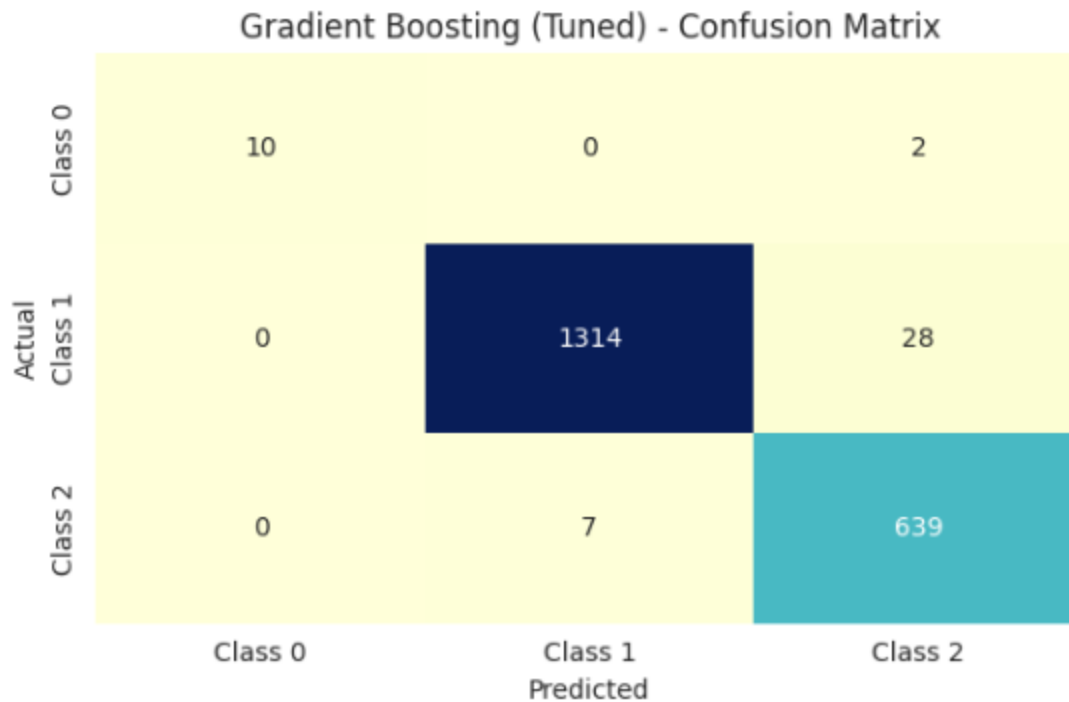


Figure 12 Confusion matrix of gradient boosting (tuned)

4.7 Model evaluation

An extensive evaluation was performed by using five standard classification metrics; namely, Accuracy, F1 Score, Precision, Recall, and ROC AUC to validate the effectiveness of the tuned machine learning models. They provide a comprehensive glimpse of how well the models are at detecting intrusions with high accuracy and reliability. As shown in Figure 13, the Random Forest (Tuned) and Gradient Boosting (Tuned) show exceptionally performance across all metrics. On Accuracy, F1 Score, Precision, and Recall, Gradient Boosting achieved slightly higher values so it could correctly classify both normal and malicious network traffic with almost no false positives and false negatives. Yet, Random Forest achieved better ROC AUCs than Gradient Boosting implies the higher class distinction ability of Random Forest than Gradient Boosting between various threshold settings. Because of this, it is especially helpful in environments where one critically needs to understand the ranking or probability of threats. Overall, both models turned out to be very good indeed, with Gradient Boosting just marginally better on overall classification, while Random Forest proved to be much more robust in probabilistic discrimination.

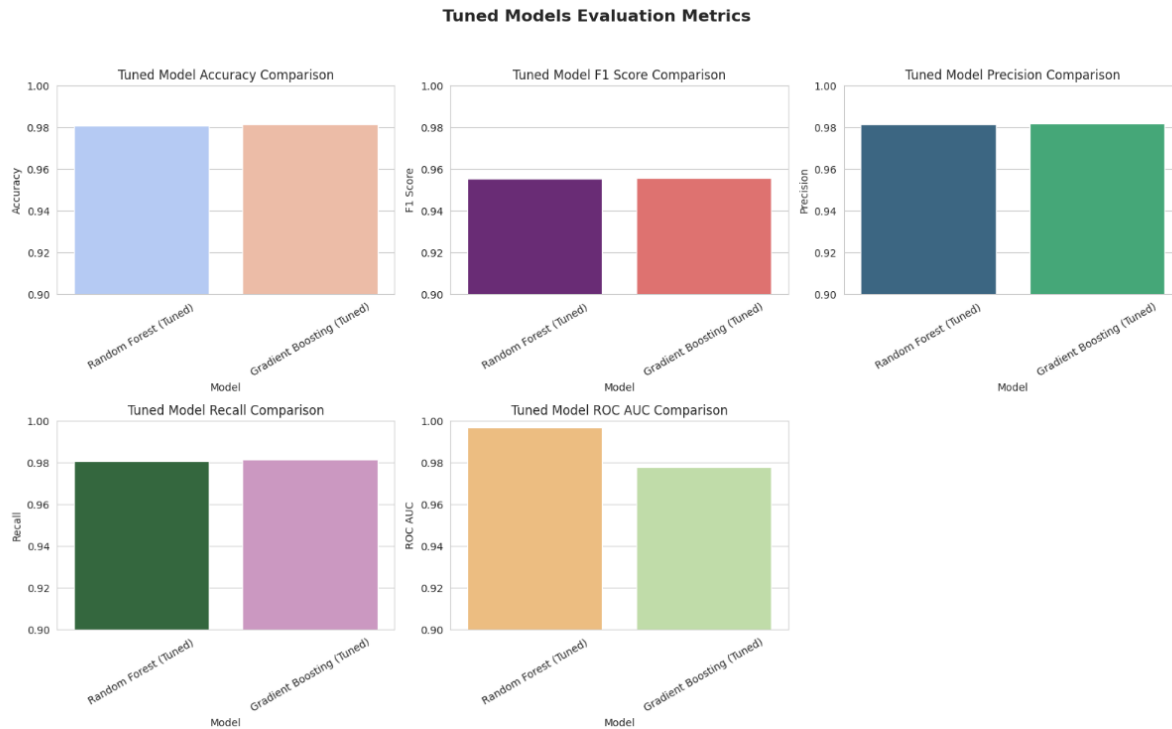


Figure 13 Tuned Models Evaluation Metrics

4.8 Conclusion

The experimental setup and implementation of the proposed ML-IDS using TII_SSRC_23 dataset was presented in this chapter. An approach that followed a structured approach including data preprocessing of cleaning, scaling and encoding as well as correlation based feature selection was taken to ensure model readiness. I implemented and evaluate several machine learning algorithms such as Random Forest, Gradient Boosting, XGboost, Logistic Regression, SVM, KNN. Ensemble based models such as Random Forest and Gradient Boosting proved to have the highest performance in terms of among these.

The two top models were hyperparameter tuned on GridSearchCV, and significantly, this increases the detection capability. Using these key metrics of Accuracy, F1 Score, Precision, Recall, and ROC AUC, the tuned models were evaluated and finally they were validated through confusion matrices and performance visualisations. In most metrics, Random Forest performed a bit stronger than Gradient Boosting, though it achieved the highest ROC AUC score, indicating powerful discrimination under spectrum of different thresholds.

Both models have results that confirm that both are effective for intrusion detection in an enterprise like network environment, although Gradient Boosting slightly outperforms in overall classification. Proficiently, the next chapter delineates the results and goes into more detail about insight with comparative information, limitations, and impacts on real-world deployment.

Chapter 5: Results and Discussion

5.1 Introduction

The results that are presented in this chapter are final results obtained from the intrusion detection using the tuned machine learning models. After hyper parameter tuning, a gradient boosting classifier and a random forest classifier are the best performing classifiers and thus evaluated. Performance metrics such as Accuracy, F1 Score, Precision, Recall, and ROC AUC were applied to these models on the TII-SSRC-23 dataset, in order to test these models on a wide range of evaluation metrics. This section compares the effectiveness in the classifying, based on the rule of anomaly checking, of these classifiers and discusses how these findings can be seen in the practice of real network security systems.

5.2 Results

The final experimental results showed that Random Forest (Tuned) and Gradient Boosting (Tuned) had also exhibited very high classification accuracy (Accuracy of 0.9815, F1 Score of 0.9559, Precision of 0.9820, and Recall of 0.9815) and Gradient Boosting slightly outperformed Random Forest. The portions of metrics that provide information about how well the model can classify both normal and malicious network flows while being as far as possible away from false positives and negatives.

The result shows that Random Forest achieved 0.9970, while Gradient Boosting achieved 0.9780 ROC AUC score, which means that Random Forest was better to classify the classes by different threshold values. In such high stakes environments, probabilistic ranking and threshold sensitivity are fundamental to security alert systems; and this matters greatly in the context of alert systems.

The higher value of confusion matrices also reflects the strong classification ability on all classes of both models with few misclassifications. Bar plots displaying the visual performance comparisons support these results as well as including all the time Gradient Boosting has a (consistently) edge over key classification metrics.

Overall, the result confirms the effectiveness of such machine learning based intrusion detection systems particularly the ensemble. In real time detection, Gradient Boosting turns out to be slightly

more accurate and balanced, while Random Forest is a powerful and interesting choice there, in particular if we require ranking or probabilistic outputs.

Feature Important analysis

In order to demonstrate the decision making steps of the intrusion detection models more, we did a feature importance analysis for the two best performing classifiers, Random Forest and Gradient Boosting. With this analysis, we identify which features made the most contribution to the classification decisions, thus allowing us to see which features of network traffic were most important to distinguishing between normal and malicious behavior.

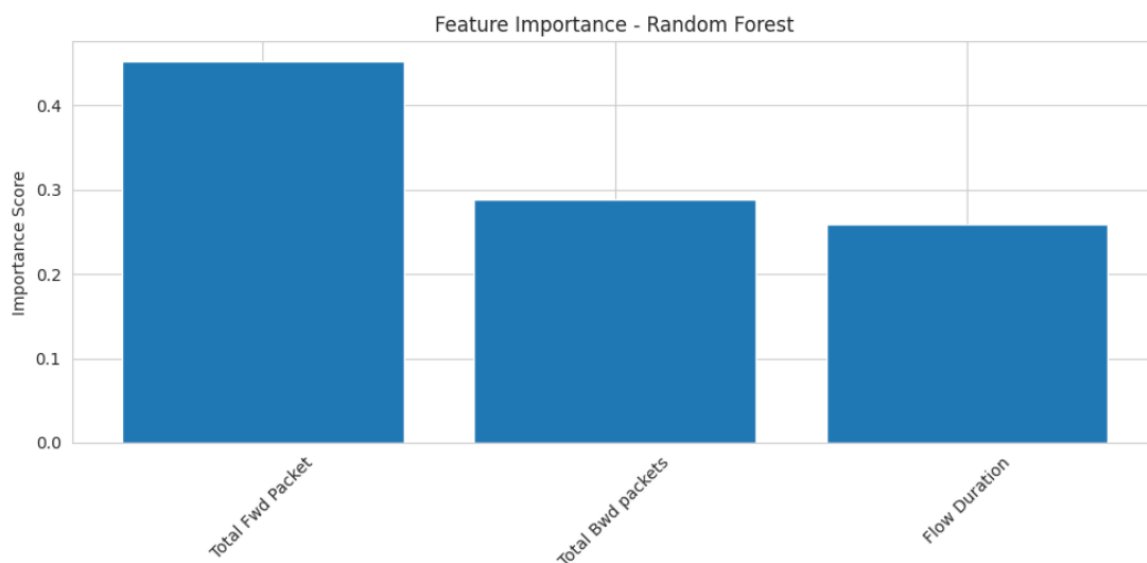


Figure 14 Feature important for Random forest

Figure 13 shows that also Random Forest model lists "Total Forward Packets" as the most important feature (almost half of the importance of the whole model). Then we have "Total Backward Packets" and "Flow Duration" which also have considerable weight in figuring out abnormal patterns. However, packet directionality is a high importance indication of important traffic volume between endpoints being a sign of potential intrusion.

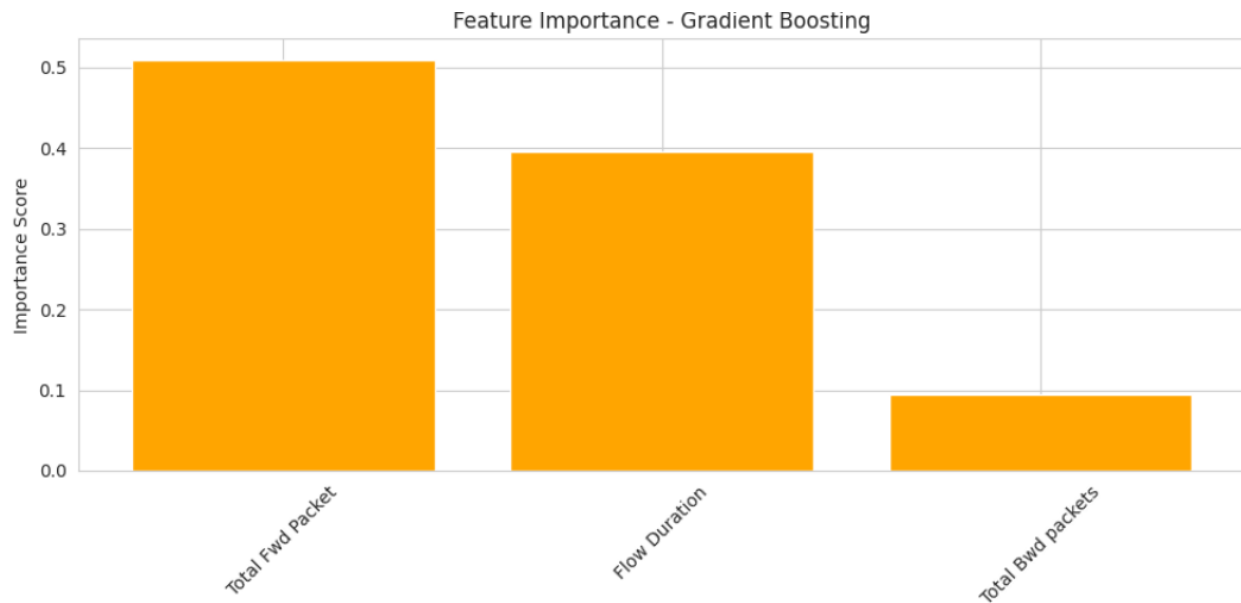


Figure 15 Feature importance for Gradient Bosting

Like before in Figure 14, the feature importance from Gradient Boosting model is shown. I can think of no other feature here that comes close to “Total Forward Packets”, which is the most dominant feature used by the model in making its decision, providing over 50% of the model’s decision points. Yet, while "Flow Duration" gets a distant second place compared to Random Forest, this indicates that running time of a network session matters more in terms of its impact on the performance of classification in Gradient Boosting’s iterative learning process. Although 'Total Backward Packets' is important, it contributes less than in Random Forest.

This comparative analysis confirms that ‘Total Forward Packets’ is indeed a key feature for both the models. The slight variations in the order of the ranking of 'Flow Duration' and 'Total Backward Packets' highlight the fact that the models are exploiting peculiarities of their own internal mechanism for learning and generalization by using specific patterns within the data.

The investigation of backward traffic movements stands as the second most vital aspect according to the measurement of Total Bwd Packets. Some cyberattacks cause unusual patterns in traffic flow since their communication patterns do not match the typical symmetry between sending and receiving data.

Network security relies heavily on time-related elements because Flow Duration represents the third vital feature in this framework. The ability to manipulate connection times allows attackers

to avoid capture so this metric provides vital information for discovering brute-force attacks, scanning and slow-rate Denial-of-Service events.

The analysis verifies packet-level characteristics serve as essential foundations for constructing a competent Machine Learning-Based Intrusion Detection System (IDS). These important features enable the Random Forest model to successfully identify normal traffic and attack patterns so it better detects threats in advance.

5.3 Discussion

The Random Forest model has proven to be the most influential indicators for intrusion detection based on the feature importance analysis, precisely Total Forward Packets, Total Backward Packets and Flow Duration. This result is in line with previous research focusing on the significance of packet level metrics in the attributes herausforderung of anomalous network behavior. In particular, this study adopts machine learning techniques to quantitatively analyze the relative contribution of each feature in the classification process and thus contributes to this understanding by providing a deeper analysis of this process

Table 6 provides a comparative overview of related research where core features and datasets and their performance in previous IDS research are provided. This is noteworthy because, in this study, it is shown that Total Forward Packets is the only most powerful feature for the high volume attacks As such as Denial of Service (DoS) or botnet traffic. Such observed traffic patterns line up with malicious activities, which tend to involve transmissions of abnormal volumes of data in one direction.

Table 6 Comparison of Feature Importance with Previous Research

Study	Model	Accuracy	Dataset	
(Herzalla, 2023)	XGBoost, Extra Trees, Deep SVDD	98.79% (XGBoost), 93.36% (Extra Trees), 97.84% (Deep SVDD)	TII-SSRC-23	Established baseline IDS performance; Identified

				critical IDS features
Our Work	Random Forest	98.10%	TII-SSRC-23	Emphasizes packet-level features, with Total Fwd Packets as the most critical; Effective for DoS and botnet detection
(Hassini, 2024)	Lightweight CNN1D	100% Accuracy Overfitting	TII-SSRC-23	The proposal introduces an efficient lightweight CNN1D model specifically for securing I-IoT security while CTGAN is used to balance data for enhanced performance.

Previous studies, especially those with such high accuracy, as deep learning or ensemble methods, are distinct in that they have transparent feature ranking and interpretability based on Random Forest. Unlike other CNN based approaches that are often used as black boxes, Random Forest is an interpretable and operationally feasible method, which is well-aligned for practical deployment in real time network security system.

The most important feature, by far, was always Total Forward Packets, particularly their use as a strong signal for attack behavior. The notion that packet level features, not only play an important

role, but can also be sufficient in the case of detection of larger scale attacks, is supported by this insight and extends the previous findings.

This derives a balanced balance approach—high accuracy, low overfitting risk and low interpretability—which brings the proposed method to become a practical and scalable solution in enterprise IDS deployment.

5.3.1 Limitations

Though the intrusion detection models based on the machine learning in this study had very strong performance, it is necessary to keep in mind a few limitations.

- **Evaluation:** The evaluation is limited to the TII-SSRC-23 dataset, a dataset of uncarved complexity and variability of various enterprise network environment. The model should be further improved to better understand what and when the cellular response to light occurs by incorporating multiple and more diverse datasets in future research.
- **Feature Importance:** It was found that Total Forward Packets is most important feature, but its importance may differ in other dataset and machine learning algorithms. Feature importance is model dependent, and different classifiers may use different traffic characteristics as basis for importances.
- **Though Random Forest's accuracy and robustness were high,** the computational complexity even in training with many trees was high. The feasibility of deploying this in a high speed, large scale network infrastructure that is also real time is further bounded by this limitation unless optimized or high performance computing resources can support it.
- **Vulnerability of Adversaries:** The models were trained in the clean unlabeled data and tested on a clean labeled one. In the practical setting, adversaries may try to hide from the detection by the application of obfuscation techniques or adversarial samples. It presents an open challenge to emphasizing the model's robustness with respect to such evasion strategies.
- **Current IDS Limits Encrypted Traffic:** The current IDS works on features obtained from plain text traffic. A portion of network communication is encrypted in real world, such as

HTTPS and VPN. Limitations exist in the detection of threats embedded within encrypted streams if the proposed system is integrated with some privacy preserving analysis techniques.

5.4 Conclusion

This chapter provided detailed analysis of our intrusion detection model assessment which included research comparison and evaluation of specific advantages. The research findings establish that Total Fwd Packets among packet-level characteristics serve as fundamental elements for recognizing network intrusions particularly DoS and botnet attacks. A Random Forest-based IDS model developed by our team strikes an excellent chord between precision and readability with its attribute significance analysis capabilities which make it applicable for real-world network defense solutions.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

Current and future power of cyberattacks persist throughout all sectors, so network security is still one of the top priorities for organizations. Intrusion Detection Systems such as the standard

signature detection or rule based systems have historically found a growing number of such previously unseen threats such as zero-day attacks, polymorphic malware, denial of service (DoS) attacks, etc. However, the currently available conventional systems are inflexible and cannot detect dynamic attack vectors in the real time. To solve this problem, we seek for more intelligent and flexible detection mechanisms.

The presented research helped build a Machine Learning based IDS framework on the TII-SSRC-23 dataset using the dataset with the focus of building a model that can be used for general purpose enterprise networks rather than domain specific environment like IoT or cloud. A number of supervised ML algorithms were used to implement and evaluate the system. After running it through extensive evaluation and hyperparameter tuning, Random Forest and Gradient Boosting became the best performing models among them.

For this research, one of the most robust, interpretable, and effective models for a wide variety of features was the Random Forest. It does a fantastic job of detecting different types of intrusion and displays feature importance which is crucial to designing practical IDS implementation in enterprise environment. Compared to those deep learning models which are computationally expensive and not transparent, Random Forest is a good trade off between accuracy and scalability and is extremely suitable with general purpose systems for which user behavior is diverse and there is mixed traffic protocol.

The study showed high detection performance, on accuracy and F1 Score Gradient Boosting was very close to Random Forest, while the Random Forest was better at ROC AUC, i.e. at classifying observations as positives or negatives across thresholds. The two models had very little false positives, and provided strong cluster class accuracy, specifically identifying packet based anomalies of DoS and botnet attack. The reliability and adaptability of the ML based IDS to handle the network intrusion detection problems with complex network intrusion patterns are confirmed by this.

6.2 Future Work

This intrusion detection system requires future development to increase model performance capabilities for practical deployment needs. The detection accuracy of the system will improve across different traffic patterns by adding diverse network environments to the dataset. The identification of complex attack behaviors would be enhanced by applying transformers together with recurrent neural networks (RNNs) as deep learning models. The system must receive essential optimization to ensure its deployment supports high-speed network traffic functions with no time delays. Research efforts should focus on defending against adversarial attacks since attackers keep developing detection-evading methods. Secure intrusion detection for encrypted traffic becomes possible through homomorphic encryption which offers both network confidentiality and security during detection.

References

- Abdallah, M. A. (2021). A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs. *In Proceedings of the 16th International Conference on Availability, Reliability and Security (pp. 1-7)*.
- Ahmad, Z. S. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150.
- Ahmad, Z. S. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. . *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150.
- Ahmed, U. N. (2025). Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. *Scientific Reports*, 15(1), 1726.
- Alamin Talukder, M. F. (2022). A Dependable Hybrid Machine Learning Model for Network Intrusion Detection. *arXiv e-prints*.
- Albulayhi, K. A.-H. (2022). IoT intrusion detection using machine learning with a novel high performing feature selection method. *Applied Sciences*, 12(10), 5015.

- Almotairi, A. A. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering*, 12(1), 2321381.
- Awotunde, J. B. (2021). Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wireless communications and mobile computing*, 2021(1), 7154587.
- Awotunde, J. B. (2021). Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wireless communications and mobile computing*, 2021(1), 7154587.
- Aygul, K. M. (2024). Benchmark of machine learning algorithms on transient stability prediction in renewable rich power grids under cyber-attacks. *Internet of Things* 25.
- Chua, T. H. (2022). Evaluation of machine learning algorithms in network-based intrusion detection system. . *arXiv preprint*.
- Hassini, K. &. (2024). Enhancing Industrial-IoT Cybersecurity Through Generative Models and Convolutional Neural Networks. In *International Conference On Big Data and Internet of Things* (pp. 543-558). Cham: Springer Nature Switzerland.
- Heidari, A. &. (2024). Internet of Things intrusion detection systems: a comprehensive review and future directions. . *Cluster Computing*, 26(6), 3753-3780.
- Herzalla, D. L. (2023). TII-SSRC-23 dataset: typological exploration of diverse traffic patterns for intrusion detection. *IEEE Access*, 11, 118577-118594.
- Jose, J. &. (2023). Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. . *International Journal of Electrical and Computer Engineering (IJECE)*, 13(1), 1134-1141.
- Kale, R. L. (2022). A hybrid deep learning anomaly detection framework for intrusion detection. . In *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*.

- Karthikeyan, M. D. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. *Scientific Reports* 14, no. 1.
- Lee, S.-W. M. (2021). Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *Journal of Network and Computer Applications* 187.
- Lefoane, M. G. (2021). Machine learning for botnet detection: An optimized feature selection approach. In *Proceedings of the 5th International Conference on Future Networks and Distributed Systems*.
- Lin, Y. D. (2022). Machine learning with variational autoencoder for imbalanced datasets in intrusion detection. *IEEE Access*, 10, 15247-15260.
- Mohale, V. Z. (2025). A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity. *Frontiers in Artificial Intelligence*, 8, 1526221.
- Samunnisa, K. K. (2022). Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. *Measurement: Sensors*, 25, 100612.
- Sarker, I. H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. *Annals of Data Science* 10, no. 6.
- Sun, Z. G. (2024). Optimized machine learning enabled intrusion detection 2 system for internet of medical things. *Franklin Open* 6.
- Umer, M. A. (2022). Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, 38, 100516.
- Xiao, X. X. (2024). A comprehensive analysis of website fingerprinting defenses on Tor. *Computers & Security* 136.
- Yin, Y. J.-J. (2021). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. . *Journal of Big data*, 10(1), 15.
- Yue, C. W. (2021). An ensemble intrusion detection method for train ethernet consist network based on CNN and RNN. *IEEE Access*, 9, 59527-59539.

