PROJECT WORK

ON

"A MULTI-PHASE TREE-BASED INTRUSION DETECTION SYSTEM USING ENSEMBLE LEARNING AND SMOTE"

FOR THE AWARD OF BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE

By

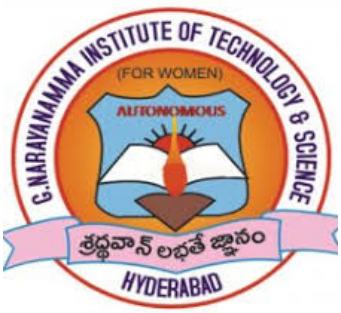
RAVURI SAI SRILEKHA

At

DEFENCE ELECTRONICS RESEARCH LABORATORY, HYDERABAD-05

EXTERNAL GUIDE

Mr. SK.SAHOO Sc.F



G. NARAYANAMMA INSTITUTE OF TECHNOLOGY AND SCIENCE

Shaikpet, Hyderabad, Telangana 500104

2025-2026

This is to certify that **RAVURI SAI SRILEKHA of G. NARAYANAMMA INSTITUTE OF TECHNOLOGY AND SCIENCE, HYDERABAD**, has undergone project training from June to July 2025 at the Defence Electronics Research Laboratory (DLRL), Hyderabad – 500005.

The project "**A Multi-Phase Tree-Based Intrusion Detection System Using Ensemble Learning and SMOTE**" is a record of the bonafide work undertaken by her towards partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. She has completed the assigned task satisfactorily.

SK. Sahoo	CS. Krishna Kumar	K Vikram
Sc 'F'	Sc 'G'	Sc 'F'
Guide	Wing Head	Wing Head HRD
		DLRL, Hyderabad

DECLARATION

I hereby declare that the results embodied in this dissertation titled "**A Multi-Phase Tree-Based Intrusion Detection System Using Ensemble Learning and SMOTE**" have been carried out by me during the year 2025–2026 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering from G. Narayanamma Institute of Technology and Science, Hyderabad.

I further declare that this work has not been submitted to any other university or organization for the award of any other degree

RAVURI SAI SRILEKHA

ACKNOWLEDGEMENT

This is an acknowledgement of the intensive drive and technical competence of many individuals who have contributed to the success of my project.

I am grateful to **Sri K. Murali, Sc- 'H' and Director, DLRL**, Hyderabad, and Sri K Vikram, Scientist 'F', Wing Head HRD, and the Members of the HRD for granting me permission to undertake practical training and project development at DLRL.

I am immensely thankful to **Sri C.S. Krishna Kumar, Scientist 'G', Wing Head of EW SW Wing, DLRL**, for providing me this opportunity and access to the necessary facilities during my training.

I am deeply obliged to my guide, **Sri S.K. Sahoo, Scientist 'F', of the SW-CC Division, DLRL**, for his invaluable suggestions, technical insight, and consistent guidance throughout the duration of the project.

Finally, I express my sincere thanks to **Dr. A. Sharada, Head of Department, Computer Science and Engineering, GNITS**, for her support and encouragement.

DLRL PROFILE

DEFENCE ELECTRONICS RESEARCH LABORATORY (D.L.R.L) was established in the year 1962 under the aegis of Defence Research and Development Organization (DRDO), Ministry of Defence, to meet the current and future needs of tri services Army, Navy and Air force equipping them with Electronics Warfare Systems.

DLRL has been entrusted with the primary responsibility of the design and development of Electronic Warfare Systems covering both Communication and RADAR Frequency bands.

DLRL consists of large number of dedicated technical and scientific manpower adequately supported by sophisticated hardware and software development facilities. Computers and dedicated Workstations are extensively used for, design and development of sub-systems. Main software required for various types of applications is developed in-house. The quality assurance group is responsible for quality assurance of software developed for Electronic Warfare Systems.

DLRL has number of supporting and technology groups to help the completion of the projects on time and to achieve a quality product. Some of the supporting and technology groups are Printed Circuited Board Group, Antenna Group, Microwave and Millimeter Wave Components Group, Mechanical Engineering Group, LAN, Human Resource Development Group etc. apart from work centers who carryout system design and development activities. Long-Term self-reliance in Technologies / Systems has been driving principle in its entire development endeavor to make the nation self-reliant and independent.

In house Printed Circuit Board facilities provide faster realization of the digital hardware. Multi-layer Printed Circuit Board fabrication facilities are available to cater for a high precision and denser packaging.

The Antenna Group is responsible for design and development of wide variety of antennas covering a broad electromagnetic spectrum (HF to Millimeter Frequencies). The Group also develops RADOMES, which meet stringent environmental conditions for the EW equipment to suit the platform.

The MMW Group is involved in the design and development of MMW Sub-systems and also various Microwave Components like Solid State Amplifier, Switches, Couplers and Filters using the latest state-of-the–art technology.

The Hybrid Microwave Integrated Circuit Group provides custom-made microwave components and super components in the microwave frequency region using both thin film and thick film technology.

In the Mechanical Engineering Group, the required hardware for EW Systems is designed and developed and the major tasks involved include Structural and Thermal Engineering.

The Technical Information Center, the place of knowledge bank is well equipped with maintained libraries, books, journals, processing etc. Latest Technologies in the electronic warfare around the globe are catalogued and easily accessible.

The Techniques Division of ECM wing is one such work center where design and development of subsystems required for ECM applications are undertaken. ESM Work Centers design and development of DF Rx, Rx Proc etc. for various ESM Systems using state art-of the technology by employing various techniques to suit the system requirements by the end users.

All the subsystems are designed and developed using microwave, and processor/DSP based Digital hardware in realizing the real time activities in Electronic Warfare.

Most of the work centers are connected through DLRL LAN (Local LAN) for faster information flow and multi point access of information critical to the development activities. Information about TIC, stores and general administration can be downloaded easily.

The Human Resource Division play a vital role in conducting various CEP courses, organizing service and technical seminars to upgrade the knowledge of scientists in the laboratory.

DLRL has been awarded ISO 9001:2015 certification for Design and Development of Electronic System of assured quality for Defence Services; utilizing

advanced and cost effective technologies & systems on time. DLRL shall comply with the requirements of quality. Management System with a focus on its continual improvement.

ABSTRACT

With the rapid increase in cyber attacks and evolving threat patterns, ensuring secure and reliable network infrastructure has become a pressing challenge. Intrusion Detection Systems (IDS) serve as a critical defense mechanism by monitoring network traffic and identifying suspicious behavior. This project proposes a tree-based machine learning approach for intrusion detection using the CICIDS2017 dataset, which contains realistic and diverse attack scenarios.

The work is structured into three phases. In Phase 1, baseline models: Decision Tree, Random Forest, and XGBoost — were trained using the full feature set. In Phase 2, feature selection was performed by averaging feature importances from all three models, followed by SMOTE oversampling to address class imbalance, particularly rare attacks like Infiltration. Phase 3 involved building an ensemble stacking classifier combining the three base learners with a logistic regression meta-learner to boost performance.

The system was evaluated based on accuracy, precision, recall, F1-score, and time efficiency. Results showed significant performance improvement after feature selection and oversampling, with the stacked ensemble achieving the highest accuracy of **99.72%**. Time analysis revealed Random Forest had the highest training cost, while Decision Tree was fastest in prediction. Among all, XGBoost offered a balanced trade-off between accuracy and efficiency, making it an ideal candidate for real-time IDS deployment.

This project demonstrates how classical machine learning models, when systematically enhanced, can offer robust and interpretable solutions for modern network intrusion detection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Background

In today's digitally connected world, organizations face constant cyber threats that range from simple intrusions to complex and coordinated attacks. As technology advances, so do the techniques used by malicious actors, making cybersecurity a top priority across government, defense, and private sectors.

Intrusion Detection Systems (IDS) play an essential role in protecting networks by monitoring traffic and identifying unusual or harmful activities. Traditional rule-based IDS work well for known threats, but often struggle to detect new or evolving attack patterns. Their static nature makes them less effective in dynamic environments with growing volumes of traffic and diverse threat behaviors.

Machine Learning (ML) has emerged as a promising solution to these challenges. ML-powered IDS can learn from data, recognize complex patterns, and adapt over time to new types of intrusions. They improve detection accuracy and offer more flexible responses to unknown threats.

However, developing a reliable ML-based IDS comes with its own set of challenges:

- Many cybersecurity datasets suffer from class imbalance, where rare attacks are underrepresented and often misclassified.
- High-dimensional data with noisy or irrelevant features can affect model performance.
- Some algorithms require considerable computational resources, which may not be ideal for real-time detection systems.

As the threat landscape continues to evolve, it becomes crucial to design intelligent, scalable, and adaptive IDS solutions that can keep up with modern security needs.

1.2 Problem Statement

Despite significant progress in the field of **cyber security**, detecting **unknown or sophisticated network intrusions** remains a major challenge. Traditional intrusion detection systems that rely on static rule-based approaches often fall short when it comes to identifying **new or evolving attack patterns**. These systems struggle to adapt to the growing scale, diversity, and complexity of modern network traffic, which can lead to delayed or missed detection of malicious activities.

Another critical limitation arises from the nature of intrusion detection datasets. Many publicly available datasets suffer from **severe class imbalance**, where rare but dangerous attacks such as infiltration or data exfiltration are vastly underrepresented. This imbalance results in **machine learning models** that become biased toward the majority classes like normal or common attack types, leading to poor generalization and high **false negative rates** for **minority attacks**.

This project addresses these limitations by developing a scalable and intelligent intrusion detection system using **classical machine learning models such as decision tree, random forest, and XGBoost**. The objective is to build a system that not only performs well on common attacks but is also capable of identifying rare and complex threats by **integrating feature selection techniques, data resampling with SMOTE, and a powerful ensemble learning strategy.**

1.3 Proposed System

The proposed intrusion detection system adopts a modular and structured learning pipeline to improve the accuracy, robustness, and generalizability of attack detection. It is composed of three systematic phases, each designed to enhance model performance at a different level.

In the **first phase**, the system trains multiple classical machine learning models—Decision Tree, Random Forest, and XGBoost—on the raw features of the CICIDS2017 dataset. This phase helps establish baseline performance and evaluate how each individual model responds to multiclass network intrusion data.

The **second phase** aims to improve detection accuracy and generalization. Feature importance scores are used to identify and retain the most influential attributes from the dataset, thereby reducing noise and computational overhead. To address the problem of imbalanced class distribution—especially rare attack classes like Infiltration—the Synthetic Minority Oversampling Technique (SMOTE) is applied. This generates synthetic samples for underrepresented classes and ensures the models are trained on a more balanced dataset.

The **third phase** involves ensemble learning. Here, the individually trained models from the earlier phases are combined using a stacking strategy. A Logistic Regression model is employed as the meta-classifier to aggregate the outputs of the base learners (Decision Tree, Random Forest, XGBoost). This ensemble approach leverages the strengths of each base model and minimizes their individual weaknesses, resulting in a more stable and accurate prediction system.

The system evaluates each model using precision, recall, F1-score, accuracy, confusion matrices, and time-based efficiency. This multilayered design ensures adaptability, scalability, and strong detection capability, even under evolving threat landscapes.

1.4 Methodology Overview

The methodology adopted in this project is designed to iteratively enhance the performance of the intrusion detection system through incremental refinement in each phase. The entire process is centered around data preprocessing, model training, and ensemble integration.

1.4.1 Phase 1: Model Training

The project begins with cleaning and preprocessing the CICIDS2017 dataset, followed by training three tree-based classifiers: Decision Tree, Random Forest, and XGBoost. These models are evaluated on accuracy, precision, recall, and F1-score to assess their baseline effectiveness.

1.4.2 Phase 2: Feature Selection and Oversampling

In this phase, **feature selection is performed** by averaging feature importance scores obtained from all three models. The top-ranked features are retained to reduce dimensionality. Subsequently, **SMOTE is applied to balance the training data** by generating synthetic instances for minority classes, particularly Infiltration. Each model is retrained using this enhanced feature space and balanced dataset to improve detection accuracy for all classes.

1.4.3 Phase 3: Stacked Ensemble Construction:

The final phase involves **constructing a stacked ensemble model** where the predictions of the three individual classifiers are combined using Logistic Regression as a meta-learner. This ensemble model provides greater generalization capability and improving the system's ability to correctly classify both frequent and infrequent attack types.

Throughout all phases, model performance is evaluated not only on predictive metrics but also on computational efficiency, including training and inference time. This structured methodology ensures that the final system is well-optimized for both accuracy and operational deployment.

1.5 Hardware and Software Requirements

- **Processor**: Intel Core i5 / i7 (or equivalent)
- **RAM**: Minimum 8 GB (16 GB recommended)
- **Operating System**: Windows 10 or Ubuntu 20.04 LTS
- **Python Version**: 3.10 and above
- **Libraries Used**: NumPy, Pandas, Scikit-learn, XGBoost, Imbalanced-learn, Seaborn, Matplotlib, Joblib
- **Development Tools**: Jupyter Notebook, Visual Studio Code

1.6 Objectives

The primary objectives of this project are:

- To develop an Intrusion Detection System using classical Machine Learning algorithms.
- To evaluate and compare the performance of Decision Tree, Random Forest, and XGBoost classifiers.
- To handle class imbalance using SMOTE and improve model generalization via feature selection.
- To build a stacked ensemble model that improves detection performance.
- To analyze model efficiency in terms of both accuracy and execution time.

1.7 System Architecture

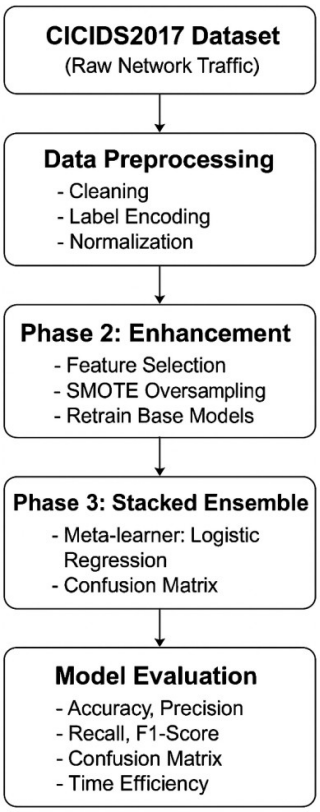


Figure 1.7.1: System Architecture of proposed IDS

The above architecture summarizes the Intrusion Detection System (IDS) pipeline. It begins with the collection of raw network traffic data from the CICIDS2017 dataset. This data is then preprocessed through cleaning, normalization, and label encoding to make it suitable for model training.

- In **Phase 1**, three classical machine learning models—Decision Tree, Random Forest, and XGBoost—are individually trained and evaluated to establish baseline performance.
- **Phase 2** enhances the model's learning by performing feature selection using average importance from base learners, followed by oversampling minority attack classes using SMOTE to handle class imbalance.
- **Phase 3** combines the strengths of the individual models through a stacked ensemble approach, using Logistic Regression as a meta-classifier to make the final prediction.

This end-to-end pipeline outputs whether incoming traffic is **benign or represents one of several attack types**, enabling more accurate and automated network intrusion detection.

1.8 Organization of the project

This project has been divided into various chapters.
Chapter 1 introduces the background of cybersecurity and intrusion detection systems, outlines the motivation for using classical machine learning techniques, and describes the limitations of traditional approaches. It also presents the problem statement, objectives, and a summary of the proposed multi-phase detection strategy.
Chapter 2 describes the CICIDS2017 dataset used for training and evaluation. It includes details about feature types, attack categories, and preprocessing methods such as label encoding and handling class imbalance.
Chapter 3 explains the methodology in depth, breaking the system into three phases: baseline model training, feature selection combined with SMOTE oversampling, and ensemble stacking. This chapter also includes system architecture and hardware/software requirements.

Chapter 4 discusses the evaluation of each model, including detailed performance metrics, confusion matrices, and time efficiency analysis. It compares individual classifiers and the final ensemble model, justifying the effectiveness of stacking. **Chapter 5** presents the conclusions drawn from the study, summarizes key outcomes, and reflects on how the objectives were achieved. It highlights the system's relevance and potential in real-world deployment scenarios. **Chapter 6** outlines the limitations of the current approach and proposes future directions. Suggestions include integrating deep learning models, real-time deployment strategies, explainable AI techniques, and validation on additional datasets to enhance scalability and robustness.

2. DATASET DESCRIPTION

2.1 CICIDS 2017 Dataset

The **CICIDS2017** dataset, developed by the Canadian Institute for Cybersecurity (CIC), is a widely used benchmark dataset for evaluating intrusion detection systems. It simulates realistic network traffic and contains both **benign activities** and a wide variety of **attack types**, collected over five days using a setup that mimics real-world enterprise environments.

The dataset includes **flow-based features** extracted using CICFlowMeter, totaling **approximately 80 features** per network flow. These features encompass **basic, time-based, content-based, and statistical** characteristics of network traffic.

The types of attacks included are:

- **DoS (Denial of Service)** – Flooding services to make them unavailable.
- **PortScan** – Scanning for open ports to find vulnerabilities.
- **Brute Force** – Repeated login attempts (FTP/SSH).
- **Web Attack** – Attacks like XSS, SQL Injection, etc.
- **Bot** – Infected hosts behaving maliciously under remote control.
- **Infiltration** – Subtle attacks bypassing firewalls.
- **Benign** – Normal user behavior.

Due to the large size of the full dataset (~16 GB), a **representative subset** was used, ensuring a manageable size while retaining all major attack categories. This allowed for efficient model training and evaluation.

2.2 Pre-Processing and Label Encoding

Before feeding the data into machine learning models, several **pre-processing steps** were applied:

- **Missing Value Handling:** All rows with missing or infinite values were removed to ensure model stability and consistency.
- **Data Type Conversion:** All features were converted to numerical format; categorical or non-numeric values were dropped or encoded.
- **Label Encoding:** The 'Label' column, representing the attack class, was **label-encoded** into numeric values using LabelEncoder from sklearn. This allowed classification models to interpret class labels effectively.
- **Feature-Target Split:** The dataset was divided into:
    - X: Features (excluding the label)
    - y: Encoded target labels
- **Train-Test Split:** An **80-20 split** was performed with stratification to preserve class distribution across training and test datasets.

Additionally, **class imbalance** was observed (especially for rare classes like Infiltration), which was later addressed in the second phase using **SMOTE oversampling**.

3. PROJECT METHODOLOGY

This project was executed in **three structured phases** to systematically develop and optimize a Machine Learning-based Intrusion Detection System. Each phase built upon the previous, refining the model performance and robustness.

3.1 Phase 1: Model Training

In the first phase, the preprocessed CICIDS2017 dataset was used to train three classical tree-based machine learning models:

- **Decision Tree Classifier**
- **Random Forest Classifier**
- **XGBoost Classifier**

The objective was to establish a **baseline performance** for each algorithm using raw features from the dataset. Key steps included:

- Feature-target split and stratified train-test partition (80:20)
- Model training using default/scikit-learn parameters
- Performance evaluation using:
    - **Accuracy**
    - **Precision**
    - **Recall**
    - **F1-score**
    - **Confusion Matrix**

This phase demonstrated strong baseline results, but also revealed **class imbalance issues** and **redundant features** in the dataset, motivating further refinement in Phase 2.

3.2 Phase 2: Feature Selection and SMOTE Oversampling

To improve the model's generalization, especially for minority attack classes, the second phase introduced **feature selection** and **oversampling**:

➢ Feature Selection

- Feature importance scores were calculated using:
  - Decision Tree
  - Random Forest
  - XGBoost
- The **average importance** across these models was computed.
- The **top features** contributing to **90% of cumulative importance** were retained.

This dimensionality reduction helped eliminate noisy, less-informative features and accelerated training without compromising performance.

➢ SMOTE Oversampling

- The **Synthetic Minority Over-sampling Technique (SMOTE)** was applied only to **underrepresented classes** (e.g., Infiltration).
- This artificially increased the number of samples in rare classes by generating synthetic examples, helping the models better learn patterns associated with such attacks.

The same three models were retrained using the reduced feature set and balanced training data. **Performance gains** were observed across most metrics.

3.3 Phase 3: Ensemble Learning

In the final phase, an **ensemble model** was constructed using a **Stacking Classifier**:

- **Base learners:** Decision Tree, Random Forest, XGBoost (from Phase 2)
- **Meta-learner:** Logistic Regression

Stacking combines the strengths of individual models, allowing the meta-model to learn from their predictions and reduce individual biases. This resulted in:

- **Highest overall accuracy and F1-score**
- **Improved robustness**
- Better generalization on test data compared to standalone models

The final model was evaluated using both classification metrics and **time efficiency analysis** (training/prediction time), validating its suitability for real-time deployment scenarios.

Compared to bagging and boosting, which respectively focus on variance and bias reduction, stacking leverages model diversity to capture complementary decision patterns. By combining heterogeneous learners through a meta-model, it produces more balanced predictions—particularly valuable in complex, multi-class intrusion scenarios where individual classifiers may struggle with minority class detection.

4. MODEL EVALUATION AND RESULTS

After completing the three-phased machine learning pipeline, the models were thoroughly evaluated to measure their effectiveness in intrusion detection. The evaluation involved key classification metrics as well as timing analysis to ensure the practicality of the approach.

4.1 Performance Comparison

Each model's performance was assessed using the following metrics:

- **Accuracy**: Overall correctness of the model
- **Precision**: Correctly identified attacks vs. total predicted attacks
- **Recall**: Ability to detect all true attacks
- **F1-Score**: Harmonic mean of precision and recall

Model	Accuracy	Precision	Recall	F1-score	Phase
Decision Tree	0.995588	0.995592	0.995588	0.995585	Phase 1
Random Forest	0.99497	0.994959	0.99497	0.994958	Phase 1
XGBoost	0.994176	0.994167	0.994176	0.994158	Phase 1
Decision Tree	0.9955	0.995504	0.9955	0.995497	Phase 2
Random Forest	0.996647	0.996646	0.996647	0.996628	Phase 2
XGBoost	0.994529	0.994516	0.994529	0.994513	Phase 2

Table 4.1.1: Phase 1 vs Phase 2 – Performance Metrics Comparison

The results showed:

- **Phase 2 models outperformed Phase 1**, especially Random Forest which showed a significant gain in accuracy and F1-score after feature selection and SMOTE.

- Decision Tree remained consistent, while XGBoost showed moderate improvement.

This confirms the benefit of addressing class imbalance and redundant features before model training.

4.2 Ensemble vs Individual Models

To enhance prediction reliability and generalization, a **Stacked Ensemble** was constructed using Phase 2's Decision Tree, Random Forest, and XGBoost models as base learners and Logistic Regression as a meta-learner.

Key observations:

- The **Ensemble model achieved the highest accuracy (99.72%)**, outperforming all standalone models.
- It delivered the **best balance of precision and recall**, resulting in a superior F1-score.
- It effectively handled rare attack classes better than any single classifier.

This demonstrates that combining the strengths of multiple classifiers through stacking leads to a more robust IDS.

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	0.9955	0.995504	0.9955	0.995497
Random Forest	0.996647	0.996646	0.996647	0.996628
XGBoost	0.994529	0.994516	0.994529	0.994513
Stacked Ensemble	0.997265	0.997264	0.997265	0.997247

Table 4.2.1: Phase 2 Models vs Ensemble – Accuracy, Precision, Recall, F1-Score

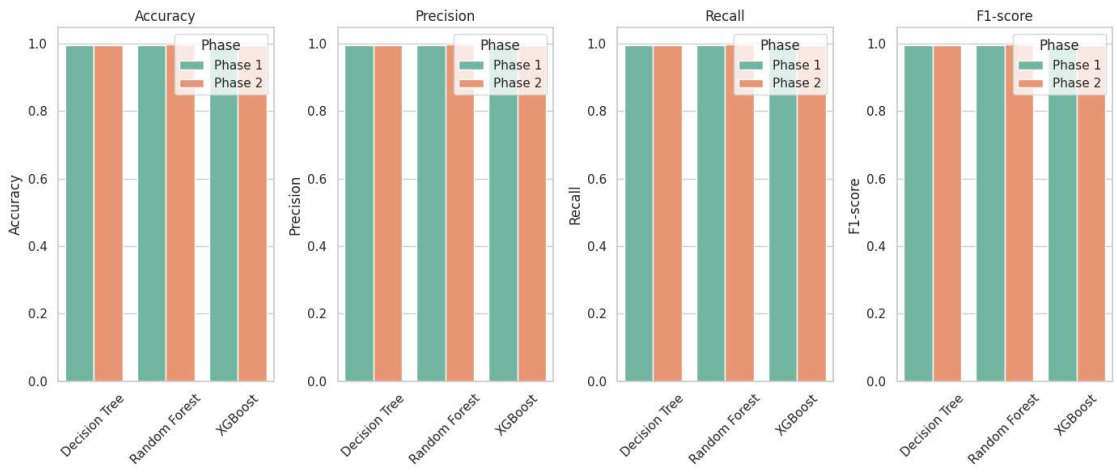


Figure 4.2.2: Comparison of model accuracy, precision, recall, and F1-score between Phase 2 models.

4.3 Confusion Matrices

For each model, confusion matrices were generated to visualize performance across all attack categories. Insights:

- **Phase 1 models** showed high accuracy for majority classes but struggled with underrepresented classes like Infiltration.
- **Phase 2 models** improved classification for minority classes after SMOTE was applied.
- The **Ensemble model** achieved the most balanced classification across all classes with minimal misclassifications.

These matrices highlight the importance of handling class imbalance for security-critical datasets.

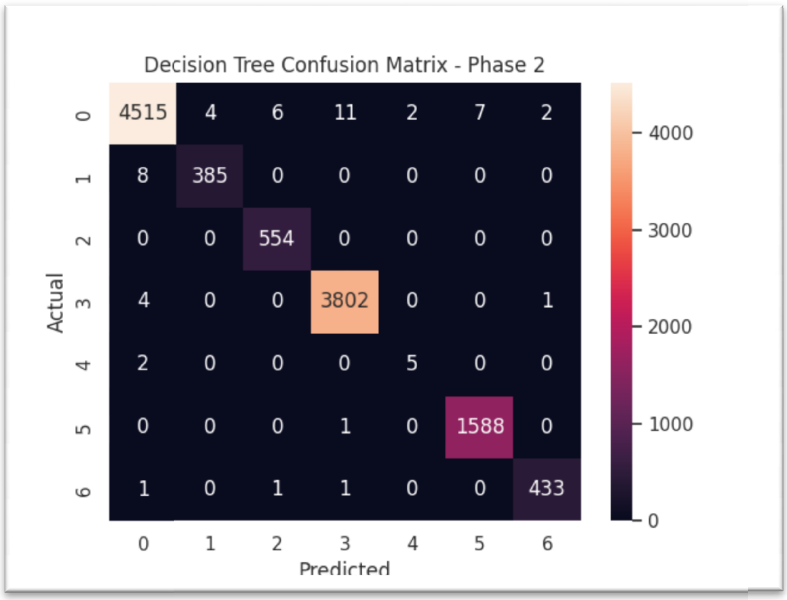


Figure 4.3.1: Confusion Matrix – Decision Tree (Phase 2)

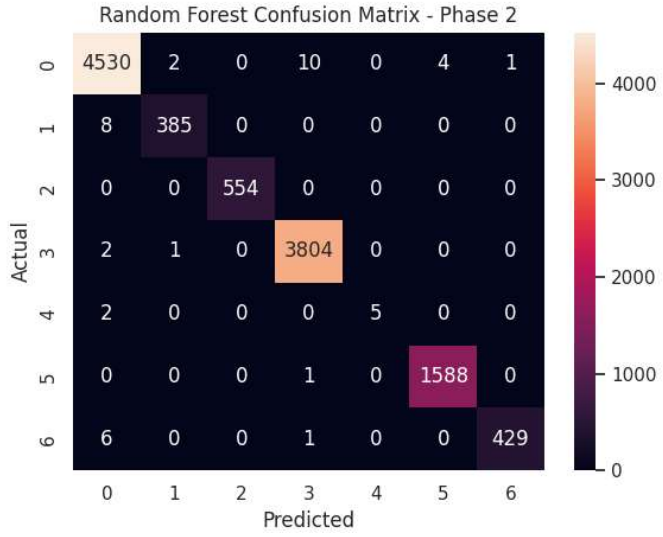


Figure 4.3.2: Confusion Matrix – Random Forest (Phase 2)

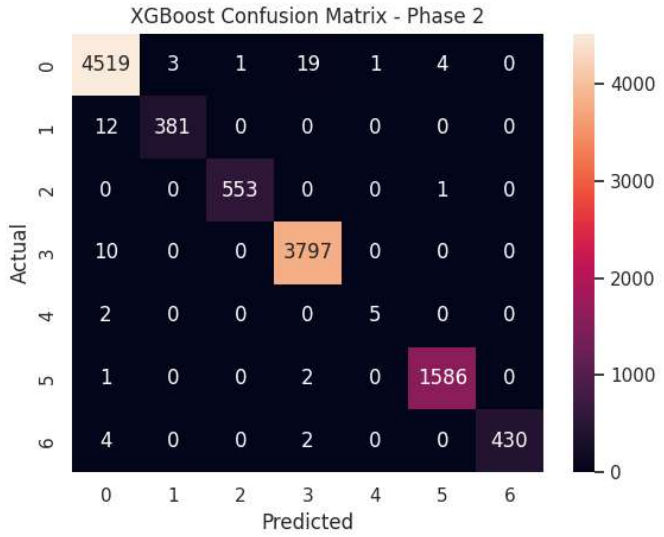


Figure 4.3.3: Confusion Matrix – XGBoost (Phase 2)

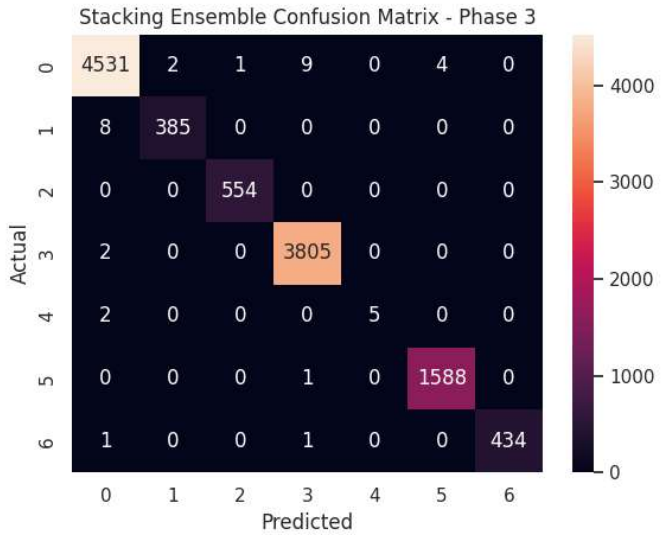


Figure 4.3.4: Confusion Matrix – Stacked Ensemble (Phase 3)

4.4 Time Efficiency Analysis

In real-world applications, **model speed** is crucial for timely threat detection. Training and prediction times were recorded:

Model	Accuracy	Precision	Recall	F1-score	Training Time (s)	Prediction Time (s)
Decision Tree	0.9955	0.995504	0.9955	0.995497	0.655972	0.001668
Random Forest	0.996647	0.996646	0.996647	0.996628	5.564972	0.070642
XGBoost	0.994529	0.994516	0.994529	0.994513	0.467644	0.013188
Stacked Ensemble	0.997265	0.997264	0.997265	0.997247	34.48137	0.125811

Table 4.4.1: Time Efficiency of Each Model (Training & Prediction)

While the **Stacked Ensemble** had the highest computational cost, its superior detection performance justifies the trade-off for many high-security environments. For low-resource deployments, **XGBoost offers the best balance** between speed and accuracy.

It is also important to consider the real-world implications of false positives and false negatives in IDS deployment. A high false positive rate can overwhelm security analysts with alerts, while false negatives may allow critical attacks to go unnoticed. The ensemble model's balanced performance across all classes suggests improved reliability in distinguishing between benign and malicious traffic—even for rare attack types.

All performance metrics were evaluated on an isolated test set to avoid data leakage, ensuring the integrity of results and model generalizability.

5. CONCLUSION AND FUTURE SCOPE

In this project, a robust Intrusion Detection System (IDS) was developed using classical machine learning techniques on the CICIDS2017 dataset. The system followed a structured three-phase methodology — beginning with baseline model training, progressing through feature selection and SMOTE-based oversampling, and culminating in a stacked ensemble classifier.

Among the individual models, Random Forest and Decision Tree classifiers demonstrated high accuracy. However, the Stacked Ensemble model outperformed all standalone models, achieving an accuracy of **99.72%**, along with improved precision, recall, and F1-score. The inclusion of feature selection and SMOTE significantly enhanced performance, especially for underrepresented attack classes.

Additionally, time efficiency analysis showed that while Random Forest and the Stacked Ensemble had longer training times, the prediction latency remained within acceptable bounds, making the models viable for near real-time detection scenarios.

The proposed system demonstrates how structured machine learning workflows—when optimized for feature relevance and class balance—can achieve high performance in cybersecurity tasks. Its modular design makes it adaptable to evolving threats and suitable for integration into enterprise-level network security pipelines.

➢ Limitations of the Study

While the proposed Intrusion Detection System using classical machine learning demonstrates strong performance, there are a few limitations to consider:

1. **Dataset Representation**: The CICIDS2017 dataset, though comprehensive, only reflects network traffic patterns and attack behaviors of 2017. Modern threats may differ, making real-world generalization a challenge without periodic retraining on updated datasets.
2. **Feature Dependency**: The results significantly rely on effective feature engineering and the selection of optimal features. Any noise or irrelevant attributes in the dataset can adversely impact performance.

3. **Computational Overhead**: The ensemble model, particularly the stacking technique, introduces added computational complexity during both training and prediction phases. This could be a concern for real-time or resource-constrained environments.

➢ Future Scope

1. **Deep Learning Approaches**: The current work can be extended by incorporating deep learning models such as LSTM, GRU, or CNN for improved temporal pattern recognition in network traffic.
2. **Real-Time Deployment**: The system can be integrated into a real-time network monitoring environment using a user interface, REST APIs, or Docker-based deployment.
3. **Feature Engineering**: Further work can include automated or domain-specific feature extraction to capture hidden patterns in raw network data.
4. **Multi-dataset Evaluation**: Testing across other benchmark datasets like NSL-KDD, UNSW-NB15 can validate the generalizability of the model.
5. **Explainability**: Integrating explainable AI (XAI) techniques can help make model decisions more interpretable to cybersecurity experts.
6. **Interdisciplinary Integration**: The model architecture can be adapted to work alongside hardware-based IDS solutions or deployed in federated learning environments where data privacy is critical. This opens opportunities for collaboration across systems engineering, embedded design, and secure distributed computing.

6. REFERENCES

[1] H. Liu, Z. Lu, F. Song, H. Wu, and X. Chen, "A Tree-Based Stacking Ensemble Method for Detecting Anomalous Network Traffic," *Applied Sciences*, vol. 10, no. 7, p. 2338, 2020. DOI: 10.3390/app10072338.

[2] Canadian Institute for Cybersecurity, "CICIDS2017 Dataset," [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[4] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[6] M. Al-Omari et al., "An Intelligent Tree-Based Intrusion Detection Model for Cyber Security," *Journal of Network and Systems Management*, vol. 29, article no. 20, 2021. DOI: 10.1007/s10922-021-09591-y.

[7] R. Panigrahi et al., "A Consolidated Decision Tree-Based Intrusion Detection System for Binary and Multiclass Imbalanced Datasets," *Mathematics*, vol. 9, no. 7, p. 751, 2021. DOI: 10.3390/math9070751.

[8] A. Alabdulatif, "A Novel Ensemble of Deep Learning Approach for Cybersecurity Intrusion Detection with Explainable Artificial Intelligence," *Applied Sciences*, vol. 15, no. 14, p. 7984, 2025. DOI: 10.3390/app15147984.

[9] T. Wu et al., "Intrusion Detection System Combined Enhanced Random Forest with SMOTE Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, article no. 39, 2022. DOI: 10.1186/s13634-022-00871-6.

Glossary

- **Intrusion Detection System (IDS)** – A security mechanism that monitors network traffic to detect suspicious or malicious activities.
- **CICIDS2017** – A comprehensive intrusion detection dataset created by the Canadian Institute for Cyber security, simulating real-world attack scenarios.
- **SMOTE (Synthetic Minority Over-sampling Technique)** – A method used to generate synthetic data points for minority classes to address class imbalance.
- **Feature Selection** – The process of selecting the most important input variables to improve model performance and reduce noise.
- **Decision Tree** – A tree-based classification model that splits data based on feature thresholds to make decisions.
- **Random Forest** – An ensemble method that builds multiple decision trees and averages their results for better accuracy and reduced overfitting.
- **XGBoost** – A powerful and optimized gradient boosting algorithm that improves classification performance.
- **Stacked Ensemble / Stacking** – A method where multiple models are trained, and their predictions are combined using a meta-model for final predictions.
- **Meta-Learner** – The model in a stacking ensemble that learns from the outputs of base learners to make final decisions.
- **Precision** – The proportion of true positive predictions out of all predicted positives.
- **Recall** – The proportion of true positive predictions out of all actual positives.
- **F1-Score** – The harmonic mean of precision and recall; a balanced measure of a model's accuracy.
- **Confusion Matrix** – A performance measurement tool that shows the correct and incorrect classifications of each class.
- **Train-Test Split** – A technique to divide data into training and testing sets to validate model performance.
- **Label Encoding** – A method to convert categorical text data into numerical form for machine learning algorithms.