Configuration Manual: Network Intrusion Detection System using UNSW-NB15

1. Introduction

Modern computer networks are continuously exposed to a wide range of cyber threats such as denial-of-service attacks, reconnaissance activities, malware injections, and unauthorized access attempts. To ensure secure and reliable communication, intrusion detection systems (IDS) are essential components of network security infrastructure. An IDS monitors network traffic and identifies malicious behavior by analyzing packet-level and flow-level features.

Traditional rule-based and signature-based intrusion detection approaches struggle to detect novel and evolving attack patterns. Machine learning–based intrusion detection systems offer improved adaptability by learning discriminative patterns from historical traffic data. This project focuses on building a machine learning–driven network intrusion detection system using the UNSW-NB15 dataset. Multiple classification models are trained and evaluated to distinguish between normal and attack traffic, with an additional emphasis on model performance, computational efficiency, and explainability.

2. Research Question

Primary Research Question

How effectively can machine learning models detect and classify network intrusions in the UNSW-NB15 dataset, and does a hybrid ensemble approach improve detection performance compared to individual classifiers?

Explanation of the Research Question

Network intrusion detection involves analyzing complex traffic patterns influenced by:
• diverse attack behaviors,
• high-dimensional network features,
• severe class imbalance between normal and attack traffic.

This study investigates two main aspects:

(A) Classification Performance
How accurately do individual machine learning models identify normal and malicious traffic across different attack categories?

(B) Ensemble Effectiveness and Interpretability
Does combining multiple classifiers into a hybrid ensemble enhance robustness and reduce misclassification, while maintaining explainability through feature attribution techniques?

3. Environment Setup

Correct environment configuration ensures reproducible training and evaluation of intrusion detection models.

3.1 Recommended Hardware

Component	Recommended
CPU	Quad-core or higher
RAM	Minimum 8 GB, recommended 16 GB
GPU	Optional (not mandatory for classical ML models)
Storage	~5–8 GB for dataset and results

3.2 Software Requirements

Operating System:
• Windows 10/11
• Linux (Ubuntu recommended)
• macOS

3.3 Python Version

Python 3.6.x (compatible with all used libraries).

3.4 Required Libraries

Core libraries:
• numpy
• pandas
• matplotlib
• seaborn

Machine Learning:
• scikit-learn
• xgboost
• imbalanced-learn (SMOTE)

Explainability:
• lime

Utilities:
• time

3.5 Installation Commands

pip install numpy pandas matplotlib seaborn
pip install scikit-learn xgboost imbalanced-learn
pip install lime

4. Dataset Description

Primary Dataset: UNSW-NB15 Network Intrusion Dataset

The UNSW-NB15 dataset contains real and synthetic network traffic generated using the IXIA PerfectStorm tool. Network packets were captured using tcpdump and processed using Argus and Bro-IDS tools to extract flow-based features.

Key characteristics:
• 49 extracted features per record
• Binary class label (normal / attack)
• Multiple attack categories including DoS, Exploits, Reconnaissance, and others
• Predefined training and testing splits provided by the dataset

The dataset used in this project is sourced directly from Kaggle and loaded in CSV format.

5. Data Preprocessing

The dataset is cleaned by removing irrelevant and duplicate records, handling missing values, and encoding categorical features into numerical form. Features and class labels are separated, and SMOTE is applied to the training data to address class imbalance.

6. Exploratory Data Analysis

Exploratory analysis is performed to examine the distribution of normal and attack traffic. Visualizations reveal class imbalance and feature variations, supporting the need for oversampling and informed model selection.

7. Model Architectures

Multiple supervised machine learning models are implemented, including Random Forest, AdaBoost, Gradient Boosting, and XGBoost. A hybrid ensemble model combines these classifiers with XGBoost as a meta-classifier to improve detection performance.

8. Training Pipeline

The dataset is divided into 80% training and 20% testing subsets. Models are trained on balanced data and evaluated on unseen samples, with latency and throughput recorded for performance comparison.

9. Evaluation Metrics

Model performance is assessed using accuracy, confusion matrix, and precision, recall, and F1-score. Computational efficiency is evaluated through latency and throughput measurements.

10. Explainability and Analysis

LIME is employed to explain individual model predictions by identifying key contributing features. This enhances transparency and supports reliable interpretation of intrusion detection results.