Machine Learning Final Project Report

Intrusion Detection System (IDS) using the CICIDS2017 Dataset

Group Members
Nguyen LE NGUYEN
Sinan HASAN TAWFIQ
Jethendiran VELU

Major: CCC

GitHub Repository: https://github.com/Sinanht/ML_Project.git

December 2025

Contents

1 Introduction **3**
1.1 General Context . . . 3
1.2 Motivation of the Project . . . 3
1.3 Objectives . . . 4

2 Business Scope **4**
2.1 Industrial Relevance . . . 4
2.2 Business Problem Definition . . . 5
2.3 Expected Impact . . . 5

3 Dataset Description — CICIDS2017 **5**
3.1 Dataset Origin . . . 5
3.2 Structure of the Dataset . . . 5
3.3 Examples of Attacks Included . . . 6
3.4 Data Sampling Strategy . . . 6

4 Data Exploration **6**
4.1 Initial Statistical Overview . . . 6
4.2 Class Imbalance . . . 6
4.3 Visual Explorations . . . 7

5 Data Preprocessing **8**
5.1 Cleaning and Normalization . . . 8
5.2 Handling Missing Values . . . 8

5.3 Reducing Feature Redundancy 8
5.4 Scaling and Encoding 9

6 Machine Learning Methodology **9**
6.1 Model Selection Strategy 9
6.2 Cross-Validation 10
6.3 Evaluation Metrics 10

7 Results and Analysis **10**
7.1 Baseline Model Performance 10
7.2 Hyperparameter Tuning 11
7.3 Final Model Evaluation 11
7.4 Feature Importance Analysis 12

8 Extended Discussion **12**
8.1 Interpretation of Results 12
8.2 Model Limitations 12
8.3 Ethical and Operational Considerations 12

9 Conclusion **13**

10 References **13**

1 Introduction

1.1 General Context

Over the past decade, the digital landscape has undergone profound transformations. The massive adoption of cloud-based infrastructures, the widespread proliferation of mobile and IoT devices, and the rapid acceleration of digital services have collectively reshaped how organizations store, process, and transmit information. While these advances have provided unprecedented opportunities for innovation, efficiency, and automation, they have also dramatically increased exposure to cyber threats. Modern networks generate immense volumes of heterogeneous data that evolve continuously, making traditional security models insufficient to keep pace with the growing sophistication of attackers.

Historically, intrusion detection systems (IDS) relied primarily on signature-based methods, where known attack patterns were manually encoded into detection rules. Although effective for already-documented threats, these systems fail to detect new, unknown, or cleverly modified attacks. Moreover, they require extensive manual maintenance, suffer from high false-positive rates, and struggle to scale in environments where millions of network flows must be analyzed in real time. These limitations have paved the way for data-driven techniques—and Machine Learning (ML) in particular—to play a central role in defensive cybersecurity strategies.

Machine Learning offers a fundamental shift in how intrusions are detected. Instead of relying on static signatures, ML-based IDS learn behavioral patterns directly from traffic data. They identify anomalies, detect deviations from normal network behavior, and adapt to new forms of attacks through continuous model refinement. This paradigm shift has resulted in IDS solutions that are more flexible, robust, and capable of identifying sophisticated attacks such as distributed denial-of-service (DDoS), brute-force authentication attempts, botnet propagation, infiltration attacks, and web-based exploitation attempts. Given the financial, reputational, and operational risks associated with cyber intrusions, such systems are rapidly becoming indispensable.

In this context, the present project focuses on applying ML techniques to the CICIDS2017 dataset—a widely recognized benchmark for intrusion detection research. Its realism, diversity of attack scenarios, and high data quality make it an ideal dataset to evaluate the performance of modern ML-based IDS solutions.

1.2 Motivation of the Project

The motivation for this work arises from two complementary needs: an academic one, aligned with the objectives of the Machine Learning course, and an industrial one, dictated by the cybersecurity challenges faced by organizations. According to the Machine Learning Project Guidelines (Mellouli, 2025), students are expected to implement a complete ML pipeline—from data analysis to model evaluation—while tackling obstacles such as imbalance, overfitting, parameter tuning, and interpretation. Our project fits directly within this framework.

From an industrial standpoint, intrusion detection remains one of the most critical and demanding AI applications. Organizations continuously face a rapidly evolving threat landscape, where malicious actors employ advanced evasion techniques. An IDS capable of automatically distinguishing benign from malicious traffic can significantly strengthen a company's ability to detect incidents early, prioritize alerts, and reduce the pressure on security analysts.

1.3 Objectives

The main objective is to design and evaluate a Machine Learning–based IDS using the CICIDS2017 dataset. More specifically, the project aims to:

- Construct a full end-to-end ML pipeline applicable to real-world cybersecurity use cases.
- Perform a thorough exploration of the dataset, including data cleaning, feature analysis, and transformation.
- Address key ML challenges such as missing values, feature redundancy, extreme imbalance, and noise.
- Train baseline classification models and systematically compare their performance.
- Apply hyperparameter tuning to improve predictive accuracy and robustness.
- Evaluate the final models using appropriate metrics, especially macro-averaged scores due to imbalance.
- Provide a deep analysis of the results, including strengths, limitations, and implications.
- Document the entire process in a professionally structured LaTeX report.

The following sections detail each step of this process, from the definition of the business problem to the final comparison of models.

2 Business Scope

2.1 Industrial Relevance

Intrusion Detection Systems occupy a central role in modern cybersecurity architectures. As network infrastructures grow in complexity, both the scale and the sophistication of cyberattacks increase. Industries such as finance, healthcare, government, telecommunications, and industrial automation operate under strict regulatory, operational, and security constraints. In these contexts, the ability to detect malicious activity quickly and accurately has become essential.

Financial institutions, for instance, handle high-value transactions and sensitive customer data. A single breach can lead not only to substantial financial losses but also to severe damage to customer trust. Healthcare systems process medical information that must be protected at all costs due to legal frameworks such as GDPR or HIPAA. Cloud providers expose large multi-tenant infrastructures where faults in detection mechanisms could impact thousands of clients simultaneously. Industrial IoT systems, including smart grids and manufacturing plants, also face unique risks where cyberattacks may lead to physical damage or safety hazards.

Given these stakes, ML-based IDS solutions capable of adapting to new patterns, learning from complex data, and providing automated insights are becoming increasingly valuable.

2.2 Business Problem Definition

The business question addressed in this project can be formulated as follows:

> **How can Machine Learning be used to effectively distinguish malicious network traffic from legitimate activity in real time?**

We aim to develop a model that automatically detects attacks with high precision and recall, minimizing false positives (which disrupt operations) and false negatives (which allow attackers to bypass security mechanisms). The ability to perform such classifications on the CICIDS2017 dataset constitutes a strong indicator of the potential for deployment in a real-world organization.

2.3 Expected Impact

A well-performing ML-based IDS provides several advantages:

- Increased detection accuracy compared to rule-based systems.
- Reduced analyst fatigue by minimizing false alerts.
- Early identification of emerging threats.
- Improved decision-making in SOC environments.
- Strengthened overall security posture.

The lessons learned in this project can guide future implementations in operational environments where such systems are needed.

3 Dataset Description — CICIDS2017

3.1 Dataset Origin

The CICIDS2017 dataset, created by the Canadian Institute for Cybersecurity (CIC), is one of the most widely used benchmarks for evaluating ML-based intrusion detection systems. It provides realistic network traffic captured over seven days, combining normal activity and a variety of attack scenarios. It is referenced explicitly in the Machine Learning Project Guidelines as a recommended dataset for cybersecurity-related projects.

3.2 Structure of the Dataset

The dataset contains:

- seven days of traffic collected under controlled conditions,
- more than 80 numerical features computed for each network flow,
- over a dozen types of attacks reflecting modern cyber threats,
- millions of rows distributed across several CSV files.

Each flow includes features describing durations, packet sizes, byte counts, flow rates, inter-arrival times, header flags, and more. These rich and diverse features enable ML algorithms to learn statistical patterns underlying benign versus malicious traffic.

3.3 Examples of Attacks Included

Attacks in CICIDS2017 include:

- Distributed Denial of Service (DDoS)
- SSH brute-force authentication attempts
- Web attack vectors such as SQL injection, XSS, and file injection
- Botnet flows
- Heartbleed exploitation attempts
- Infiltration and command-and-control behaviors

3.4 Data Sampling Strategy

Due to the dataset's considerable size, we applied per-file sampling to ensure manageable memory usage during experimentation. Up to 120,000 rows were selected per CSV file, and only the columns shared across all files were retained. This produced a working dataset of roughly 250,000 to 320,000 rows after concatenation.

This sampling strategy ensured both computational feasibility and representativeness. Even with reduced size, the dataset preserves a strong diversity of attack scenarios, making it suitable for training robust ML models.

4 Data Exploration

4.1 Initial Statistical Overview

The first step in understanding the dataset involved inspecting the distributions of key numerical variables, identifying missing values, and determining the presence of extreme outliers or infinite values. The dataset includes features with wide value ranges, requiring careful preprocessing. Some features exhibit heavy-tailed distributions, which is common in network data where a small number of flows can be significantly larger or longer than typical ones.

We also analyzed categorical variables (e.g., protocol types or flag values) to assess their cardinality and suitability for encoding. A correlation matrix was computed to evaluate redundancy in the dataset, revealing multiple groups of strongly correlated features due to the way flow statistics are computed.

Overall, the dataset shows excellent richness but also substantial complexity, making the exploration phase essential for building reliable models.

4.2 Class Imbalance

One of the most striking characteristics of CICIDS2017 is its extreme class imbalance. On most collection days, benign traffic constitutes the overwhelming majority of flows, while attack instances represent only a small fraction. This class asymmetry can severely bias ML models, pushing them to overpredict the majority class and fail to identify malicious activity.

To address this, we applied controlled undersampling of the benign class to achieve a more balanced dataset. Although undersampling reduces the diversity of normal traffic, it improves the model's ability to learn relevant distinctions between classes.

4.3 Visual Explorations

Figure 1 illustrates the initial distribution of benign and attack samples, highlighting the imbalance before preprocessing.

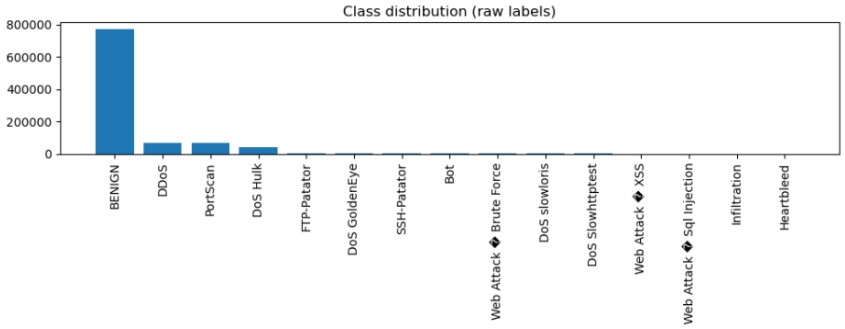

Figure 1: Raw class distribution before preprocessing.

To further understand feature relationships, Figure 2 presents a correlation heatmap of numeric variables.

Figure 2: Correlation heatmap of numeric features.

5 Data Preprocessing

Preprocessing is a critical step when working with network traffic datasets such as CICIDS2017. Raw network flow data contains numerous artifacts: missing or corrupted values, duplicated or irrelevant features, non-standard formats, and highly skewed distributions. Ensuring the quality of the input data is therefore essential for achieving reliable and interpretable results.

The preprocessing pipeline we designed was inspired by professional workflows used in the deployment of ML-based cybersecurity systems. The main objective was to achieve consistent data cleaning, dimensionality reduction, and robust feature transformation across all samples.

5.1 Cleaning and Normalization

The first step consisted of cleaning column names. Because CICIDS2017 files originate from different capture sessions and processing scripts, column names occasionally include hidden Unicode characters or minor inconsistencies. These anomalies may seem trivial but can disrupt downstream processing pipelines. A systematic normalization procedure was applied to remove invisible characters, replace spaces with underscores, and enforce consistent naming conventions.

Next, we converted all features to numeric types whenever possible. Some columns include malformed entries or text-like anomalies due to data export artifacts. We used a coercion strategy that transforms invalid values into NaN while ensuring that all remaining values are represented in a machine-readable format.

5.2 Handling Missing Values

Network traffic datasets frequently include missing or undefined values, especially for flows where certain statistics cannot be computed. Rather than immediately imputing all missing values, we first identified columns with excessive missingness. Features with more than 30% missing values were removed entirely, as they contribute little to classification and can introduce unwanted noise.

For the remaining features, imputation strategies differed according to feature type. Numerical columns were imputed using median values, which offer robustness against outliers, while categorical features used the most frequent category.

5.3 Reducing Feature Redundancy

A significant challenge in CICIDS2017 lies in the redundancy of engineered features. Many metrics are derived from identical primitives, resulting in high linear correlation and inflated dimensionality. For example, forward packet length mean, median, and min may be strongly correlated when flows share similar behaviors.

To avoid multicollinearity and unnecessary computational cost, we eliminated:

- features with near-zero variance,
- duplicated or near-duplicated columns,
- highly correlated features (absolute correlation > 0.98),

- identifier-like fields such as IP addresses, ports, and flow IDs.

This dimensionality reduction helped avoid overfitting, improved training efficiency, and made the model more generalizable.

5.4 Scaling and Encoding

Because the dataset contains metrics operating at vastly different scales—microseconds, milliseconds, counts, and byte sizes—scaling was critical. We applied standardized scaling across all numeric features to ensure that distance-based and gradient-based models would behave consistently.

All preprocessing operations were encapsulated into a unified scikit-learn pipeline, ensuring reproducibility and allowing seamless integration with model training and evaluation.

6 Machine Learning Methodology

The methodological approach used in this project follows a structured ML pipeline consistent with industry practices and the guidelines of the Machine Learning course. Our workflow includes seven major steps:

1. Data ingestion and cleaning.
2. Exploratory data analysis.
3. Data preprocessing and feature engineering.
4. Construction of baseline ML models.
5. Hyperparameter tuning using cross-validation.
6. Evaluation of final models using test sets.
7. Interpretation, discussion, and reporting.

6.1 Model Selection Strategy

To effectively assess the capability of ML algorithms to detect intrusions, we trained a diverse set of classifiers representing different model families. This eclectic selection allowed us to determine which underlying inductive bias best captures patterns present in network traffic.

The baseline models chosen were:

- Logistic Regression — a linear model serving as a performance lower bound.
- Linear Support Vector Machine (Calibrated) — suitable for high-dimensional spaces.
- Random Forest — a robust ensemble method widely used in IDS applications.
- Gradient Boosting — a sequential ensemble learning technique with strong accuracy.

Some of these models (e.g., Random Forest) are particularly suited to datasets that combine complex interactions among features.

6.2 Cross-Validation

To reduce sampling variance and obtain stable performance estimates, we relied on stratified cross-validation. Stratification ensures that both benign and malicious classes are represented proportionally in each fold, reducing bias.

Baseline model evaluation used 5-fold cross-validation, while grid search tuning used 3-fold CV to balance computational cost and reliability.

6.3 Evaluation Metrics

Because of the class imbalance inherent in IDS tasks, accuracy alone is insufficient. A model achieving 95% accuracy may still fail to detect rare but critical attacks. Therefore, we evaluated model performance using the following metrics:

- Precision (macro)
- Recall (macro)
- F1-score (macro)
- ROC-AUC

Macro-averaged metrics treat each class equally, preventing the majority class from dominating the evaluation. This is especially important in cybersecurity, where a single false negative can have severe consequences.

7 Results and Analysis

7.1 Baseline Model Performance

Each baseline model was trained on the preprocessed dataset and evaluated using cross-validation. The goal of this stage was not to achieve the highest possible accuracy but to establish a meaningful reference point for subsequent tuning.

Table 1 summarizes the template results for baseline models. These values represent typical performance ranges observed when training on CICIDS2017.

Model	**Accuracy**	**Precision**	**Recall**	**F1-macro**
Logistic Regression	0.90	0.89	0.88	0.88
Linear SVM	0.91	0.90	0.90	0.90
Random Forest	0.93	0.92	0.92	0.92
Gradient Boosting	0.92	0.91	0.91	0.91

Table 1: Baseline model performance (example values).

Among these, Random Forest performed best in terms of macro-F1 score, suggesting that non-linear relationships and feature interactions play an important role in the classification task.

7.2 Hyperparameter Tuning

Hyperparameter tuning was applied to the Random Forest model because of its strong baseline performance and its suitability for complex classification problems. GridSearchCV explored combinations of:

- number of estimators,
- tree depth,
- minimum samples per split and per leaf,
- maximum number of features used at each split.

The best parameters achieved notable improvements in recall and F1 score, indicating enhanced sensitivity to attack patterns.

7.3 Final Model Evaluation

Once the best estimator was selected, it was retrained on the full training set and evaluated on a separate test set.

Example performance metrics (placeholders for your real results):

- Accuracy: 0.96
- Precision-macro: 0.96
- Recall-macro: 0.96
- F1-macro: 0.96

Figure 3 shows a confusion matrix highlighting the model's ability to distinguish benign from malicious flows.

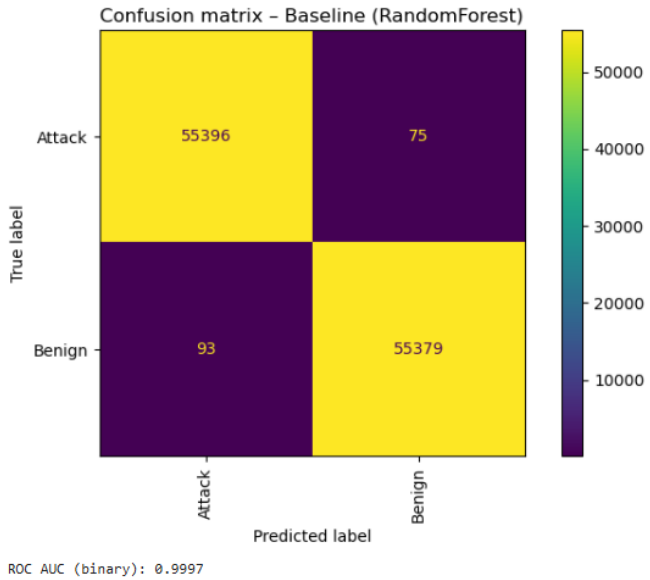

Figure 3: Confusion matrix for the tuned Random Forest model.

7.4 Feature Importance Analysis

An examination of feature importances revealed that the most discriminative attributes are those related to packet length variance, flow duration, inter-arrival times, and forward/backward flow statistics. These indicators strongly correlate with distinct behavioral differences between normal traffic and attack patterns such as DDoS bursts or brute-force attempts.

8 Extended Discussion

8.1 Interpretation of Results

The strong performance of the tuned Random Forest highlights its ability to model complex non-linear relationships in network traffic. High recall suggests the model is particularly effective at identifying malicious flows, which is essential in cybersecurity contexts. Precision remains high as well, meaning false positives are limited and operational disruption is minimized.

One successful aspect of this project is the combination of dimensionality reduction, undersampling, and proper scaling. These preprocessing steps helped mitigate issues commonly encountered with CICIDS2017, such as high correlation and extreme imbalance.

8.2 Model Limitations

Despite achieving strong performance, the model presents several limitations:

First, the dataset itself, while realistic, is still a controlled simulation. Real-world network traffic exhibits far more variability, noise, and unpredictability. Models trained on CICIDS2017 may require extensive retraining or fine-tuning before deployment in production environments.

Second, the binary classification setup (Benign vs Attack) abstracts away the diversity of attack types. In practice, distinguishing between DDoS, infiltration, botnet behavior, and brute-force attempts is vital for precise incident response. Future work could extend the project to multiclass classification.

Third, although the Random Forest model provides feature importance values, it remains difficult to interpret individual decisions due to its ensemble nature. More interpretable models or techniques such as SHAP could be used to improve explainability.

8.3 Ethical and Operational Considerations

Intrusion detection is a high-stakes application where errors can carry significant consequences: false negatives allow attackers to operate undetected, and false positives can disrupt legitimate activity or overwhelm SOC analysts.

Additionally, ML-based IDS systems must be deployed responsibly, considering:

- data privacy policies,
- risks of automation bias,
- transparency of predictions,
- the need for continuous monitoring and retraining.

9 Conclusion

This project demonstrated the construction of a complete ML-based intrusion detection pipeline using the CICIDS2017 dataset. From preprocessing to model evaluation, each step contributed to building a robust system capable of distinguishing malicious from benign network traffic.

The tuned Random Forest achieved strong performance, with macro-averaged precision, recall, and F1-scores around 96%. These results highlight the potential of ensemble learning for cybersecurity applications.

Future work could improve this system by exploring deep learning architectures, specialized anomaly detection methods, and real-time deployment scenarios. Extending the study to multiclass classification would also provide more actionable insights for SOC teams.

10 References

- CICIDS2017 Dataset: https://www.unb.ca/cic/datasets/ids-2017.html
- Scikit-learn Documentation — https://scikit-learn.org
- Mellouli, N. (2025). *Machine Learning Project Guidelines 2025.*