Technical Report: DDoS Attack Detection Using ML

Apurva Patel

February 27, 2025

Abstract

This report presents a comprehensive technical analysis of a DDoS detection system using machine learning. The system achieves 99.99% accuracy using Random Forest classification, with an average prediction time of 5.3ms per sample. We discuss the data exploration process, modeling choices, implementation details, and performance evaluation.

Github: https://github.com/Apurva3509/DuneSec

1 Data Exploration & Analysis

We are provided with a dataset of network traffic flows collected using CICFlowMeterV3. The dataset contains both benign traffic and traffic from DDoS attacks. Our task is to develop a solution that distinguishes between these two types of traffic i.e. Binary classification.

The dataset consists of network traffic flows with 84 features and 1 target variable ("Label"). Key characteristics include:

1.0.1 Feature Composition

- **Numerical Features (80):**
  - Flow statistics (duration, bytes/packets)
  - Protocol-specific metrics
  - Network behavior indicators
- **Identifier Columns (4):**
  - Flow ID, Source IP, Destination IP & Timestamp
  - Source IP & Destination IP introduced a lot of bias, model depended very highly on them.

1.0.2 Class Distribution Analysis

- DDoS Traffic: 56.71%
- Benign Traffic: 43.29%

'DDoS' is the anomaly and we have to make a model to predict it accurately to avoid loss of data and not affect the important data from 'BENIGN' traffic.

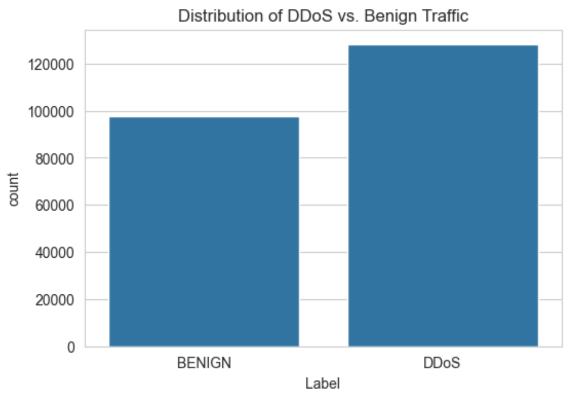


Figure 1: Distribution of Target variable

This represents a mild class imbalance, not severe enough to require aggressive resampling techniques but addressed through balanced class weights in the model.

1.1 Data Quality Assessment

Although the given data was fairly organized, there were some issues I faced during processing mentioned as below.

Issue Type	Percentage	Resolution Strategy
Missing Values	0.015%	Dropped (minimal impact)
Duplicate Rows	0.02%	Removed to prevent data leakage
Infinite Values	Variable	Replaced with NaN
Invalid Column names	Fixed	Stripped before making dataframe

Table 1: Data Quality Issues and Resolution

1.2 Data Analysis and Model Selection

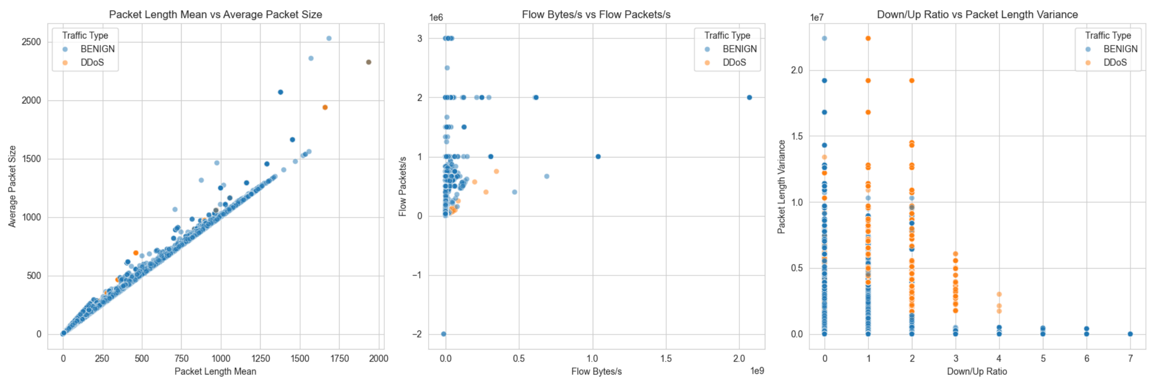


Figure 2: Feature Distributions for DDoS and Benign Traffic

Packet Length Mean vs. Average Packet Size

- Strong **linear correlation** exists between packet length mean and average packet size.
- **BENIGN traffic (blue)** follows a predictable linear trend.
- **DDoS traffic (orange)** deviates from this trend, showing **non-linear variations**, indicating packet size anomalies.
- **Implication:** Linear models like logistic regression struggle with such **non-linear relationships**, making tree-based models a better choice as they can handle **irregular patterns** effectively.

Flow Bytes/s vs. Flow Packets/s

- Most data points cluster near the origin, indicating **low packet rates** for typical network flows.
- **DDoS traffic (orange)** shows **high Flow Packets/s**, suggesting rapid packet bursts, a key trait of attack patterns.
- Some outliers have extremely high **Flow Bytes/s**, potentially indicating **high-volume attacks**.
- **Implication:** Conventional parametric models might struggle with these **sharp spikes in distributions**, but **Tree based moedls handles such structured attack behaviors by creating decision boundaries that separate normal and attack flows**.

Down/Up Ratio vs. Packet Length Variance

- **DDoS traffic (orange)** clusters at specific Down/Up ratio values, indicating **consistent attack patterns**.
- Packet Length Variance is **higher for DDoS**, meaning attack packets are often **irregular in size**.
- **BENIGN traffic (blue)** is more dispersed, reflecting natural variations in communication.
- **Implication:** The structured attack behavior means that **decision based splitting can create distinct partitions to separate attack traffic**, whereas models assuming continuous distributions (e.g., SVM) may struggle to generalize well.

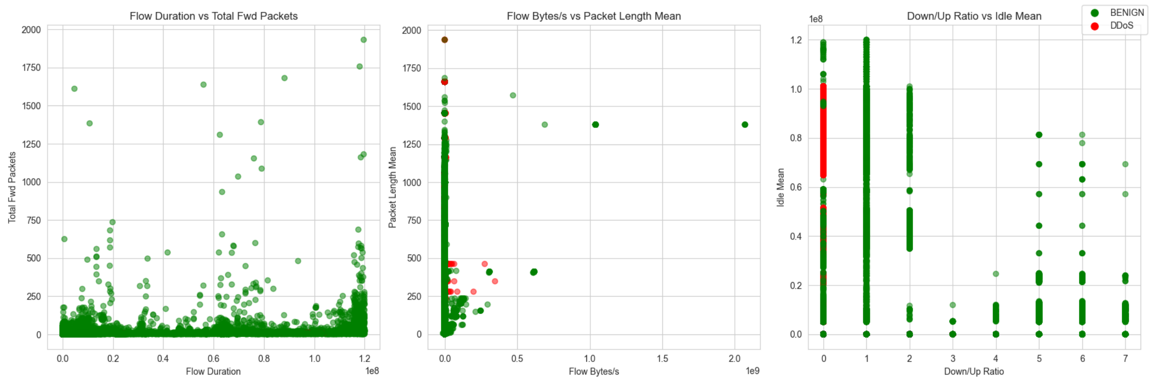


Figure 3: Flow Behavior Analysis of DDoS and Benign Traffic

Flow Bytes/s vs. Packet Length Mean

- **DDoS traffic (red)** is concentrated at **low Flow Bytes/s and low Packet Length Mean**, reinforcing attack bursts.
- Some extreme outliers exist with **high Packet Length Mean**, indicating rare but impactful attacks.
- **DDoS attacks favor small but high-frequency packets**, disrupting services.
- **Implication:** having these **outliers and structured attack bursts**, tree-based methods like **Random Forest and XGBoost** are robust as they **handle extreme variations well** and do not assume normal distributions.

Down/Up Ratio vs Idle Mean

- **DDoS traffic (red)** is clustered at specific Down/Up Ratios with **higher Idle Mean**.
- **Benign traffic (green)** is more widely spread, reflecting varied usage patterns.

- **DDoS attacks send packets at fixed intervals**, leading to **consistent Down/Up ratios**.
- High Idle Mean suggests that during attacks, there are **periodic bursts followed by idle times**.
- **Implication: Random Forest and Decision trees** can capture the **structured behavior of attacks** efficiently, while models assuming continuous distributions (e.g., Naïve Bayes) would fail to model this periodicity.

1.3 Feature Distribution and Its Impact on Model Selection

Understanding the **statistical properties** of network traffic is crucial in selecting the most effective machine learning model.

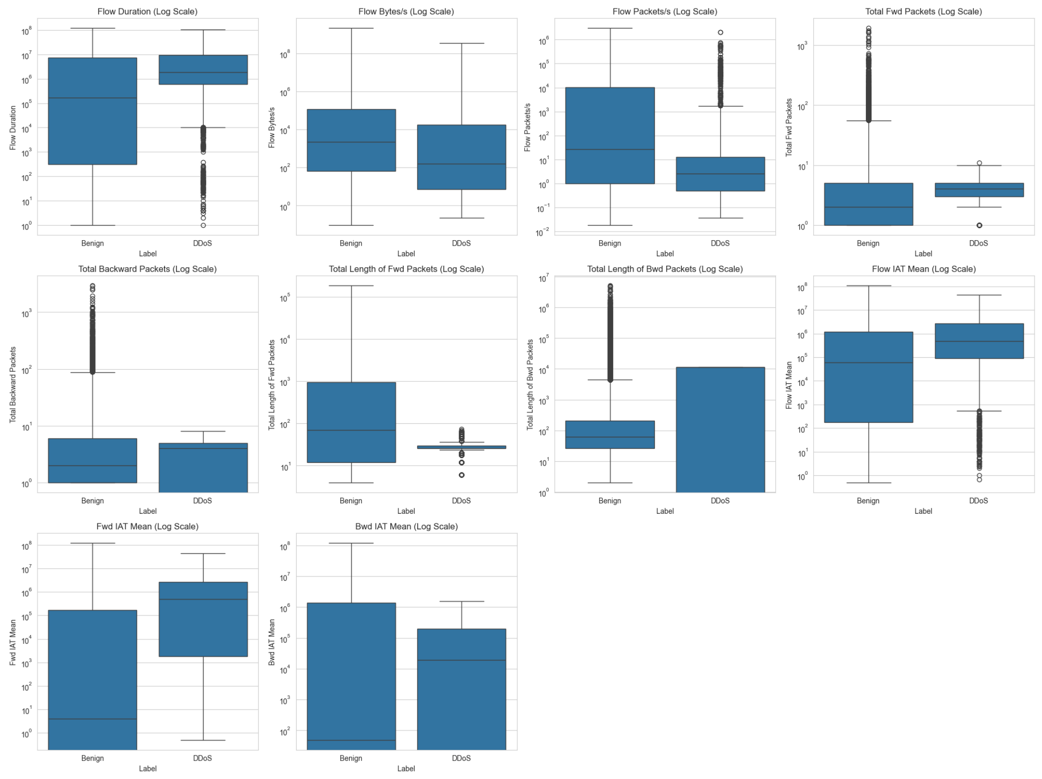


Figure 4: Boxplots of Network Features (Log Scale) Comparing Benign and DDoS

Non-Linear Decision Boundaries and Variability

- Benign traffic exhibits **high variability** in Flow Duration, Packet Length, and Inter-Arrival Times, while DDoS traffic follows **structured, repetitive patterns**.
- A model must capture both structured attack patterns and the diverse nature of benign traffic. **Tree-based models** (like Random Forest) excel in handling **heterogeneous distributions**, making them more effective than linear models.

Feature Importance and Redundancy Handling

- Flow Bytes/s and Packets/s reveal **distinct separation**, with DDoS traffic having significantly **higher packet transmission rates**.
- Some features may be redundant due to high correlation. **Random Forest automatically assigns lower importance to less relevant features**, reducing overfitting without requiring extensive manual feature selection.

Outliers and Robustness

- Extreme outliers exist in benign traffic, suggesting **rare high-volume legitimate communications**.
- Unlike linear models, **Random Forest is resilient to outliers** since decision trees partition data based on thresholds rather than assuming a fixed distribution.

Interpretability and Efficiency

- Benign traffic is **more spread out**, while DDoS attacks **cluster around specific values** in multiple features.
- Random Forest provides **feature importance rankings**, helping refine model performance by identifying **key predictors**.
- Unlike deep learning models, **Random Forest offers low inference time**, making it practical for **real-time detection**.

1.4 Feature(s) collinearity

All the numerical features were extracted from the given data and correlation was found among them to give a better understanding of the modeling strategies.

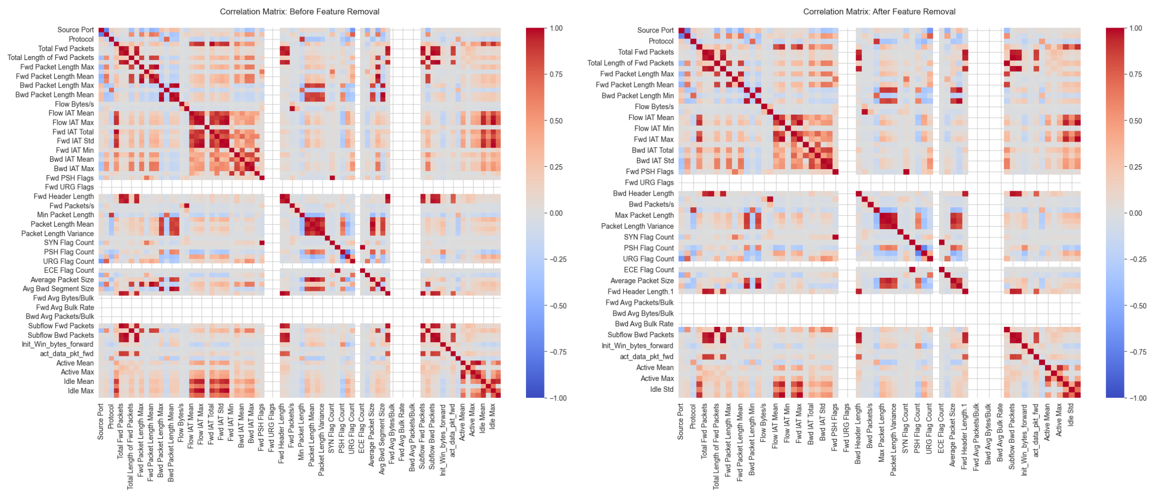


Figure 5: Feature correlation before and after processing

Note: There is some white space in the correlation plot because those values were NaN.

Highly correlated features can introduce multicollinearity, which affects model interpretability. Reducing redundant features improves training efficiency and prevents overfitting. the threshold for collinearity initially was set to 0.85.

Impact of Feature Removal:

1. Before Removal: Many features showed strong correlations, causing redundancy.
2. After Removal: Reduced multicollinearity and enhanced model generalization for unseen data.

2 Model Development Strategy

2.1 Data Split Strategy

The data splitting strategy was carefully designed considering the nature of DDoS attacks:

- **Training Set (80%):**
  - Used for model training and startified cross-validation
  - Sufficient size for feature importance analysis
- **Test Set (20%):**
  - Completely held out for final evaluation

2.2 Algorithm Selection Rationale

Random Forest was chosen as the primary classifier after comparative analysis:

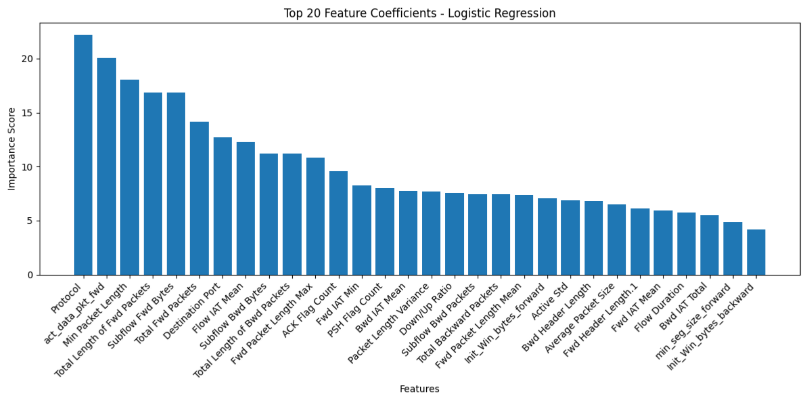
Advantage	Technical Justification
Handling Non-linearity	Effectively captures complex relationships in network traffic patterns
Feature Importance	Provides interpretable importance scores for network metrics
Parallelization	Enables efficient training and prediction through parallel processing
Robustness	Handles outliers and noisy features common in network data
No Feature Scaling Required	Can process raw network metrics without normalization
Balanced Feature selection	Balances the class weight to account for feature imabalance

Table 2: Random Forest Selection Rationale

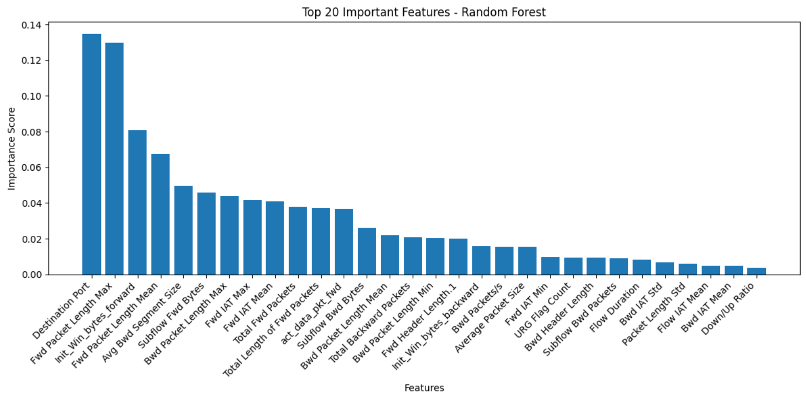
Feature Importance for multiple models

I tried 4 models, Random Forest, XGBoost, Logistic regression and 2-layer MLP to understand how they use features and how we can select the best modle choice for production.

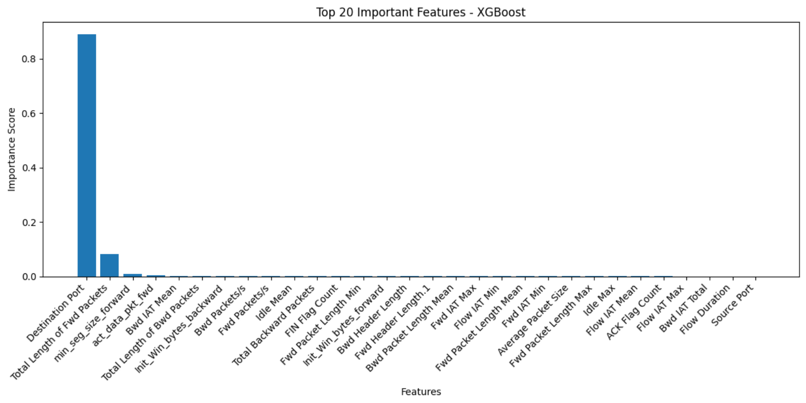
2.3 Justification for Choosing Random Forest



(a) Feature Importance - Logistic Regression



(b) Feature Importance - Random Forest



(c) Feature Importance - XGBoost

Figure 6: Comparison of Feature Importance across Models

Limitations of Logistic Regression

- Incapable of Handling Non-linearity
- Feature Independence Assumption - assumes independent features, but in network flow data, features are highly correlated.
- Lower Predictive Power: Logistic regression assigns importance based on linear weights, making it ineffective for capturing complex traffic behaviors.

Issues with XGBoost

- Over-reliance on a Few Features: Indicates potential overfitting or feature bias.
- Hyperparameter Sensitivity: Requires extensive tuning to prevent overfitting making it less practical for real-time deployment.
- Higher Computational Cost

Based on these factors, Random Forest is the most suitable model for DDoS detection, offering the best balance between accuracy, robustness, and efficiency.

2.4 Hyperparameter Optimization

Grid search cross-validation was employed with the following configuration:

```
param_grid = {
    "n_estimators": [50, 100],
    "max_depth": [None, 10, 20],
    "class_weight": ["balanced"],
    "min_samples_split": [2, 5]
}
```

2.4.1 Cross-Validation Strategy

- **Method:** StratifiedKFold
- **Folds:** 5
- **Scoring:** F1-Score (balanced metric for imbalanced data)
- **Stratification:** Maintains class distribution across folds

2.5 API for model inference

I have also created an API endpoint using FastAPI() to make inference from the trained model. More on this in ReadMe.md file in Github.

3 Performance Analysis

3.1 Comparative Model Performance

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	1.0000	1.0000	0.9999	1.0000	1.0000
XGBoost	0.9989	0.9990	0.9999	0.9999	1.0000
Logistic Regression	0.9993	0.9997	0.9992	0.9994	0.9998
MLP Neural Network	0.9940	0.9890	0.9980	0.9980	1.0000

Table 3: Comprehensive Model Performance Comparison

3.2 Operational Performance

- **Average Prediction Time:** 5.3ms per sample
- **Batch Processing:** 1024 samples simultaneously
- **Memory Footprint:** 50MB (model size in RAM)
- **Throughput:** 188 predictions/second

3.3 Feature Importance

This was the importance of each feature from the processed data which was obtained from trained random forest model. Wecan clearly see it is not bias towards any feature like XGBoost.

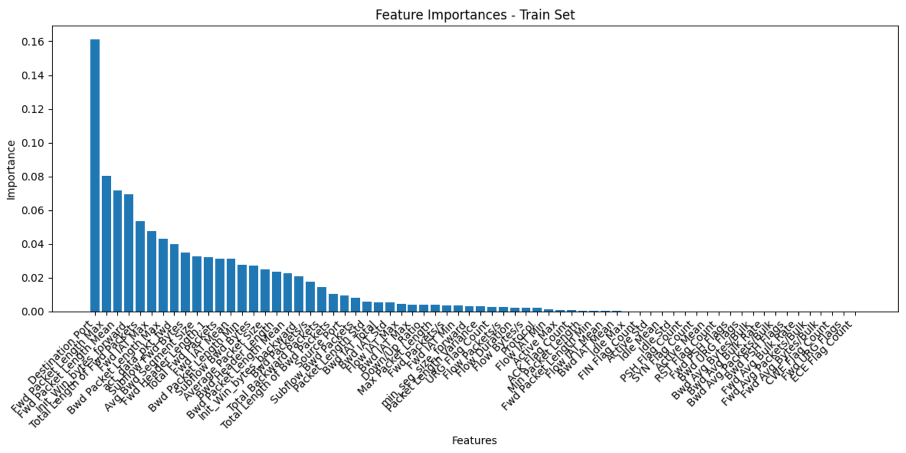


Figure 7: Feature Importance Analysis

The feature importance analysis reveals key network traffic characteristics that contribute most to DDoS detection.

3.4 Performance Across Thresholds

I also varied threhsolds(by default 0.5 in sklearn) for random forest and observed some interesting things for variable scale.
Shows the trade-offs between metrics:

- Higher threshold → Higher precision, lower recall
- Lower threshold → Higher recall, lower precision

For DDoS detection:

- To minimize false positives, choose a higher threshold
- To minimize false negatives, choose a lower threshold

The intersection point of precision and recall curves often provides a good balance.

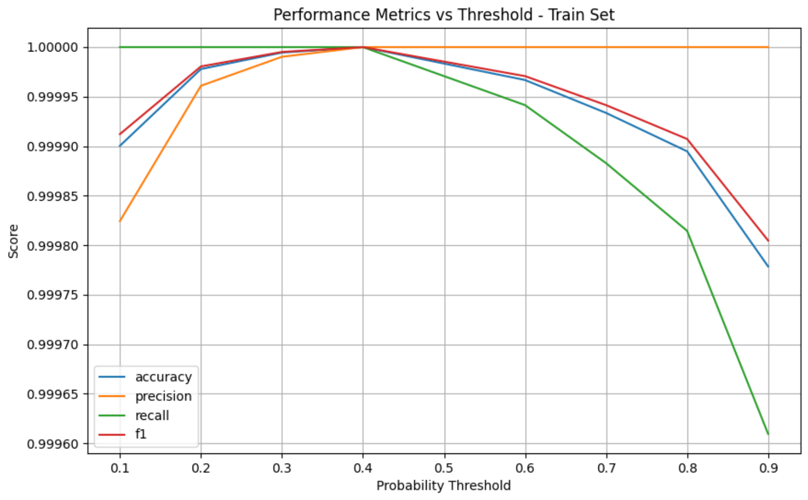


Figure 8: Performance Metrics vs Classification Threshold

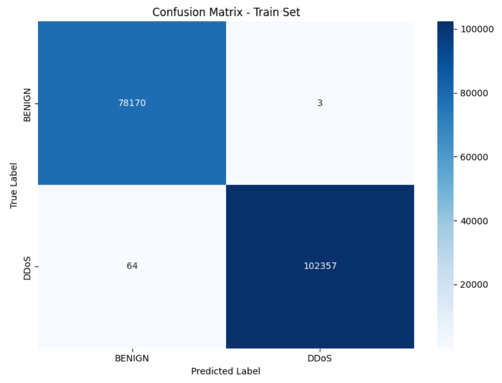
The threshold analysis demonstrates robust performance across different decision boundaries, with optimal performance around 0.5 threshold.

4 Results & Evaluation

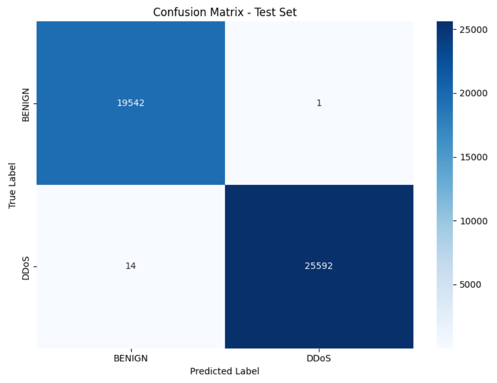
4.1 Confusion Matrix Analysis

The confusion matrices show minimal misclassifications, with:

- True Negatives (Benign correctly classified)
- True Positives (DDoS correctly classified)
- False Positives and False Negatives negligible.



(a) Training Set Confusion Matrix



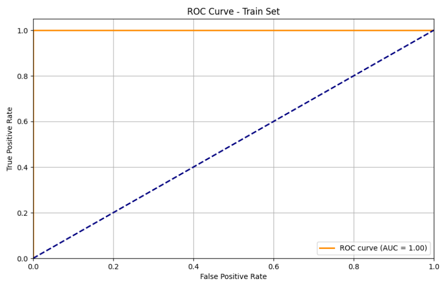
(b) Test Set Confusion Matrix

Figure 9: Confusion Matrices for Training and Test Sets

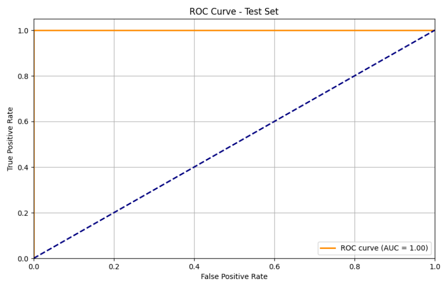
1. High Accuracy: The model correctly classifies almost all samples in both train and test sets.
2. High Recall: Ensures that nearly all DDoS attacks are detected.
3. Low False Positives: Prevents unnecessary disruptions in normal network operations.
4. Minimal Overfitting: Train and test results are consistent, confirming the model's generalization.

4.2 ROC Curve Analysis

The ROC curves demonstrate near-perfect classification performance on both training and test sets, with AUC scores of 1.0000.



(a) Training Set ROC



(b) Test Set ROC

Figure 10: ROC Curves for Training and Test Sets

ROC Curve Analysis:

1. Both training and test sets show an AUC score of 1.00, meaning perfect classification.
2. The true positive rate (TPR) is maximized while the false positive rate (FPR) remains at 0, leading to a flat ROC curve at the top.
3. This indicates that the model does not struggle with any borderline cases.

4.3 Prediction Distribution Analysis

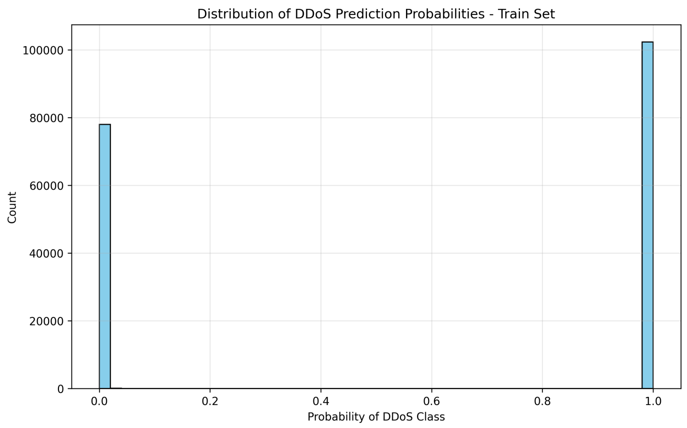


Figure 11: Distribution of Model Predictions on Training Set

The prediction distribution shows clear separation between classes, indicating strong discriminative power of our model.

5 Deployment Considerations

5.1 Prediction Time Analysis

Average prediction time of 5.3ms(as per local runs via temrinal) per sample enables real-time detection capabilities. The system can process approximately 188 predictions per second on standard hardware.

5.2 Model Serving Architecture

- **API Framework:** FastAPI
- **Serialization:** joblib for model artifacts
- **Versioning:** MLflow for experiment tracking
- **Monitoring:** Custom logging for prediction metrics

6 Business Impact

F1-Score Prioritization

- False negatives (missed attacks) are more costly than false positives, as an undetected DDoS attack can disrupt services.
- A well-balanced precision-recall tradeoff ensures high detection rates while minimizing disruptions to legitimate(Benign) network traffic.

Security Implications

- Minimizing false alerts reduces the likelihood of unnecessary security responses, avoiding service interruptions or wasted resources.
- Maximizing detection accuracy ensures that real threats are identified and mitigated in real-time and maintaining continuity.

7 Future Improvements

Potential enhancements include:

- Feature engineering optimization based on importance analysis.
- Model compression for faster inference.
- Implementation of drift detection for model monitoring.
- Cloud deployment for scalable predictions.