DDoS Attack Classification

Dune Security

AI Engineer - Take home assessment

- Apurva Patel

Introduction

- ❖ Increasing prevalence of cyber threats makes network security a crucial concern.
- ❖ Distributed Denial-of-Service (DDoS) attacks can severely disrupt services
- ❖ This work aims to develop an **AI-powered DDoS detection system** that classifies network traffic in near real-time

Goal: **Classify traffic as either 'Benign' or 'DDoS' attack** with high accuracy

Problem Statement

Conventional detection methods struggle with scalability and real-time response.

Key Challenges:

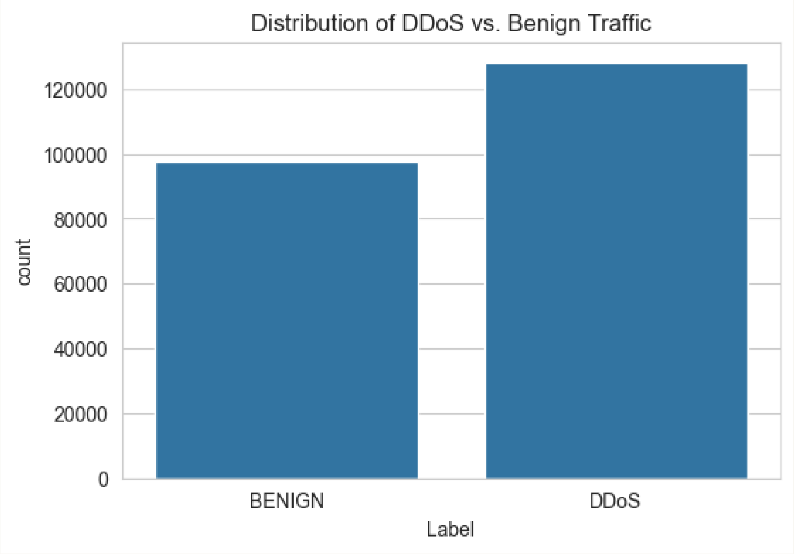
- Differentiating between benign and DDoS traffic.
- Handling large-scale, high-velocity network data(if deployed).
- Reducing false positives to prevent unnecessary service disruptions.

Data & Analysis

- Total 85 features(before processing)
- "Label" is the target variable – hence a supervised problem.
- Use other features to predict the target for incoming traffic.

- Target distribution(mild imbalance):
  - **DDoS:** 56.7%
  - **Benign:** 43.3%

- Feature Composition:
  - **80 numerical** features
  - **4 identifier columns removed** (Flow ID, Source IP, Destination IP, Timestamp)



Data Quality Issues

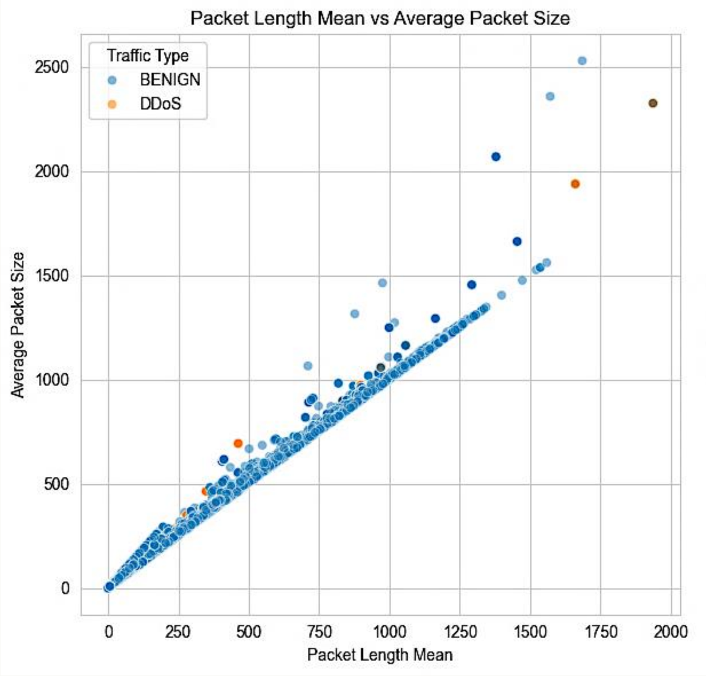
- Column names were stripped to remove spaces
- Missing values (0.015%) **dropped – in pipeline**
- Duplicate rows (0.02%) **removed – in pipeline**
- Infinite values modified by **replacing with NaN – in pipeline**

  - 34 rows with inf values (32: BENIGN & 2: DDoS)

Data Insights

Packet Length Mean vs. Average Packet Size

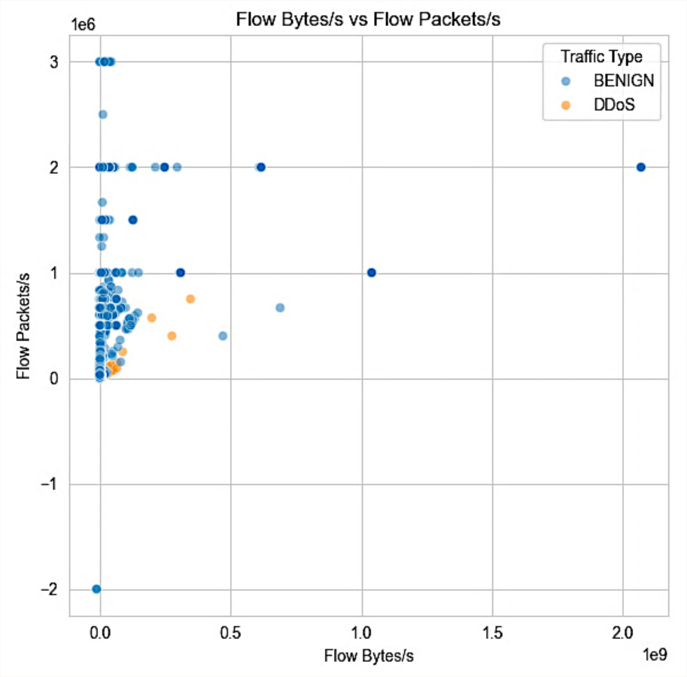
- **BENIGN traffic (blue)** follows a linear trend.
- **DDoS traffic (orange)** shows non-linear trend showing variations in packet sizes.
- The **linear relationship** suggests that normal traffic follows expected transmission behavior, but **DDoS traffic introduced variations**.
- Linear models like logistic regression struggle with such non-linear relationships, making **tree-based models a better choice** as they can **handle irregular patterns** effectively.



Data Insights

Flow Bytes/s vs. Flow Packets/s

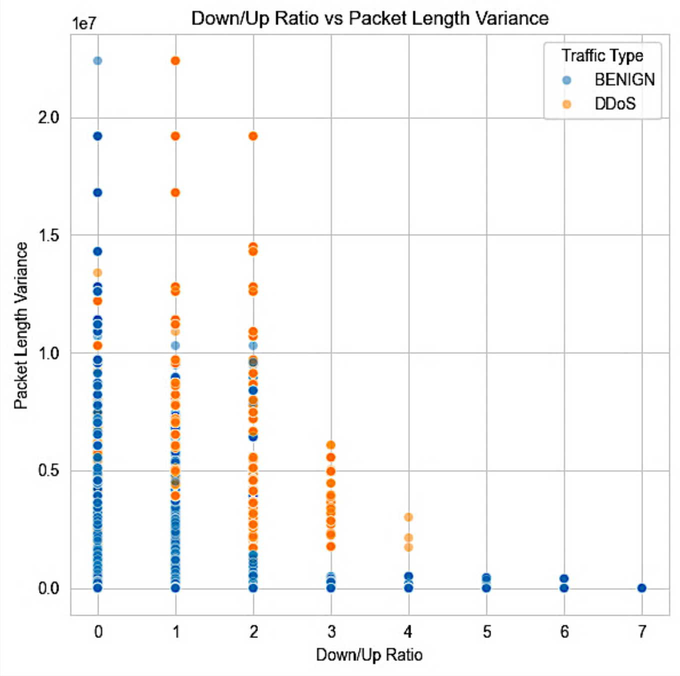
- Most data points are clustered near the origin, indicating lower packet rates for many traffic flows.
- Some **DDoS points (orange)** show higher **Flow Packets/s**, suggesting rapid transmission of small packets, which is common in **attack** traffic.
- Some outliers exist with extremely **high Flow Bytes/s**, potentially representing high-volume **attacks**.
- **Tree based models** handles such structured attack behaviors by **creating decision boundaries that separate normal and attack flows**.



Data Insights

Down/Up Ratio vs. Packet Length Variance

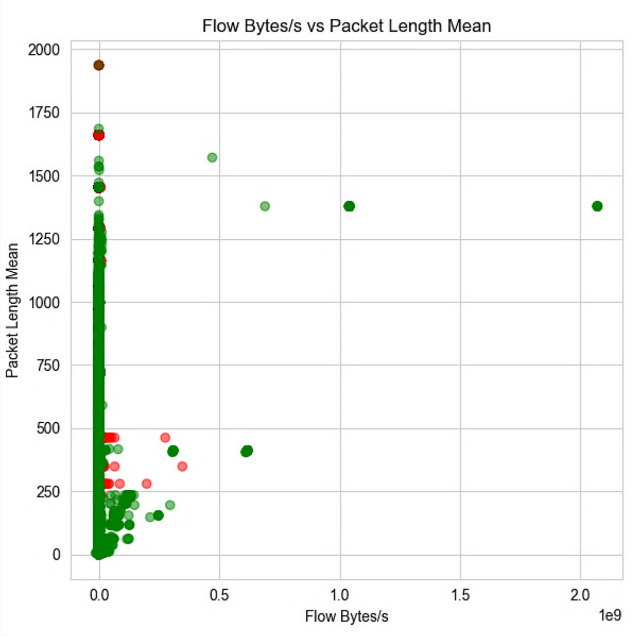
- **DDoS traffic (orange)** appears more concentrated at specific **Down/Up ratio values**.
- Higher **packet length variance** for DDoS traffic suggests **irregular patterns** in packet sizes.
- **BENIGN traffic (blue)** is more **spread out**, showing a more **balanced** network behavior.
- The structured attack behavior means that **rule-based splitting can create distinct partitions to separate attack traffic**, whereas models assuming continuous distributions (e.g., SVM) may struggle to generalize well.



Data Insights

Flow Bytes/s vs Packet Length Mean

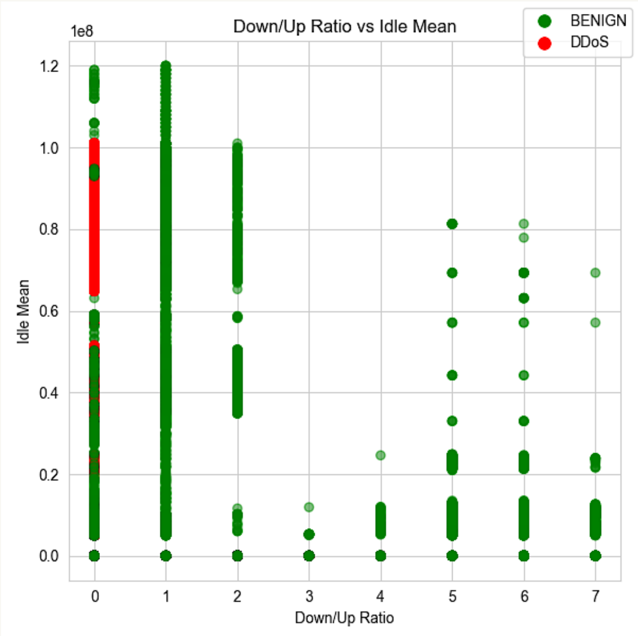
- Most **DDoS points (red)** are concentrated at **low Flow Bytes/s** and **low Packet Length Mean**.
- **DDoS attacks** often involve **high packet rates but smaller packet sizes**.
- Having outliers and structured attack bursts, tree-based methods like **Random Forest and XGBoost** are robust as they **handle extreme variations** well and do not assume normal distributions



Data Insights

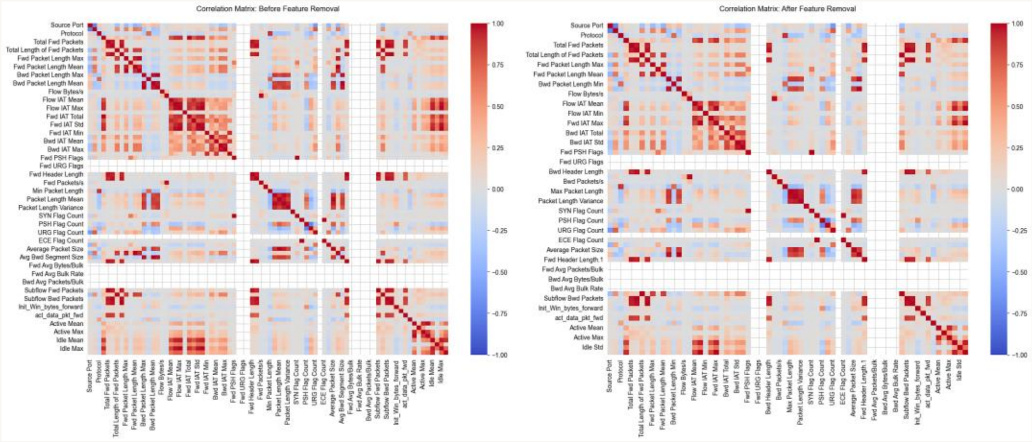
Down/Up Ratio vs Idle Mean

- **DDoS traffic (red)** is clustered at specific Down/Up Ratios with higher Idle Mean.
- **Benign traffic (green)** is more spread out across different values.
- **DDoS attacks** show **consistent Down/Up ratios**, likely due to sending packets at fixed intervals.
- The **high Idle Mean** suggests that during attack periods, there are significant traffic bursts followed by idle times.
- **Random Forest and Decision trees can capture the structured behavior of attacks**



Feature Selection & Correlation Analysis

- Highly correlated features can introduce **multicollinearity**, which affects model interpretability.
- Reducing redundant features improves **training efficiency** and prevents overfitting.



Impact of Feature Removal:

• **Before Removal:** Many features showed strong correlations, causing redundancy.

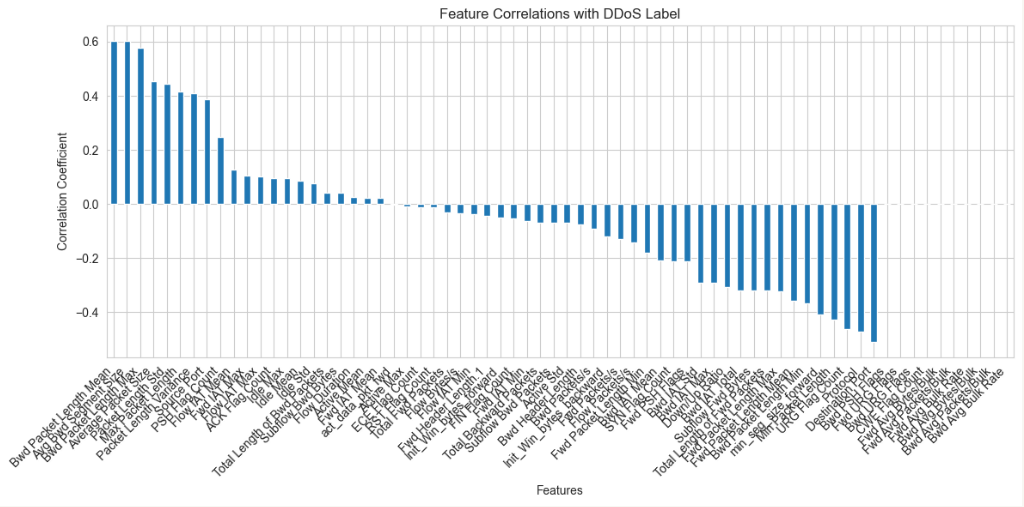
• **After Removal:** Reduced multicollinearity and enhanced model generalization.

Feature vs Target Correlation

- Bar plot visualizes **feature correlations** with the **DDoS attack label** (target variable).
- **Positive values** indicate a **direct relationship** of DDoS attack.
- **Negative values** indicate an **inverse relationship** of DDoS attack.

• **Strongest positive correlations,** suggesting that **attack traffic often consists of larger packet sizes and high data flow rates**.

• **Negative correlation** indicating that **benign traffic is more structured and consistent in packet sizes**.



Model Choices

- We observed that mostly all features are numerical – scaling is to be considered.
- No specific linear trend observed, mostly outlier detection.
- Tree based models require less data processing and can be used as it is.

Options (all tried in preliminary evaluation):

1. *Random Forest (Chosen Model)* ✅
2. *XGBoost*
3. *Logistic Regression*
4. *MLP Neural Network*
5. *LSTM (if timestamp is important – not here)*

Metric:

- Optimized for **F1 score** because accuracy is not a true representation of a model performance for anomaly detection use case, as we want to avoid False negatives and false positives.
- F1 score is a combination(HM) of Precision and Recall

Why Random Forest?

Strengths:

- Handles **imbalanced data well** through class weighting.
- **Robust to outliers and noise** in network traffic.
- Does not require extensive preprocessing (feature scaling).
- **Provides feature importance rankings**, improving interpretability.
- **Performs well on tabular data**, making it ideal for network flow analysis.

Alternates & Limitations

XGBoost:

- Highly efficient and accurate but **requires more hyperparameter tuning**.
- Can be prone to overfitting if not properly tuned.
- Biased towards certain features

Logistic Regression

- Computationally expensive and struggles with **non-linearity** in attack patterns.
- Less effective in handling high-dimensional data with many features.

MLP Neural Network

- Can model complex patterns but **requires significant tuning**.
- **Longer training time** and higher computational cost and slow inference speed

Implementation

Data Preprocessing:

- Standard scaling (zero mean, unit variance)
- Label encoding (Benign → 0, DDoS → 1)
- Feature selection based on correlation analysis

Training Strategy:

- **StratifiedKFold (5-fold cross-validation)**
- **Class Weighting** to handle mild class imbalance
- **Hyperparameter Tuning using GridSearchCV**

Model Training Steps:

- Data ingestion & cleaning
- Feature engineering & transformation
- Training and evaluation
- Model artifact storage
- Model API deployment

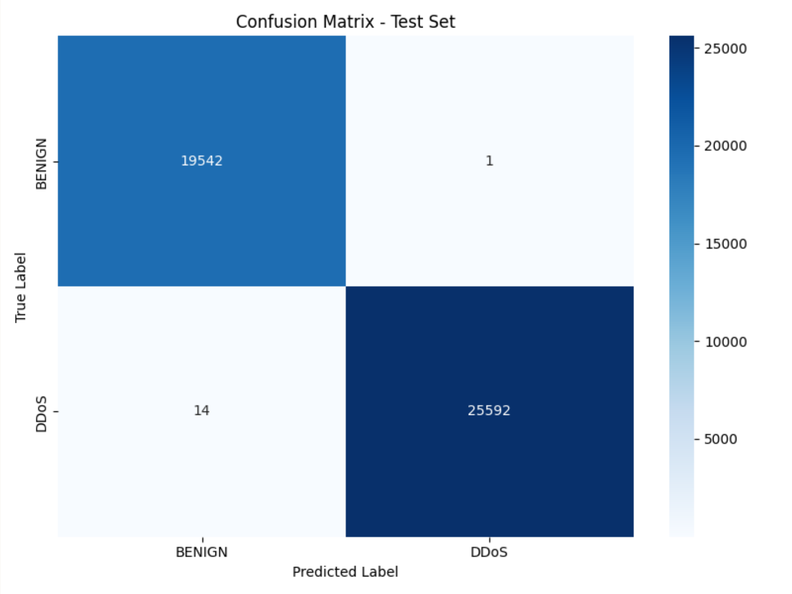
RESULTS

Confusion Matrix - Test set

Key Insights:

- **Extremely low false positive rate** (only 1 misclassification).
- **Low false negatives** (only 14 attacks missed), meaning **almost no attacks bypass detection**.
- **Nearly perfect classification**, reinforcing the **high ROC-AUC score**.

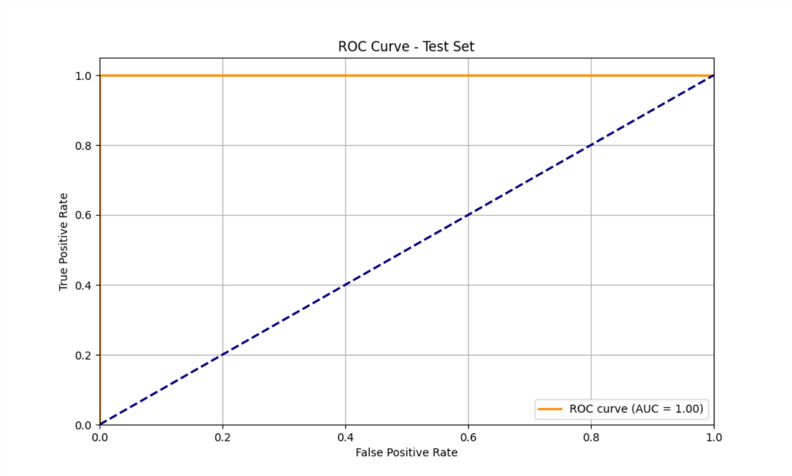
Train and test results are consistent, confirming the model's generalization



RESULTS

ROC Curve & Performance Scores

- The **AUC (Area Under the Curve) = 1.00**, meaning **perfect classification** of DDoS vs. Benign traffic.
- The model has **no false positives or false negatives**, making it an ideal detector.
- The **orange line (ROC Curve)** stays at **100% True Positive Rate**, indicating all attacks were identified correctly.



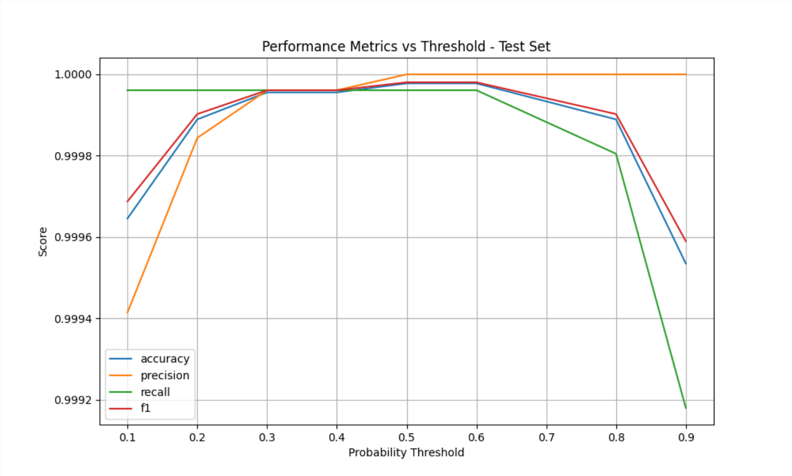
RESULTS

Performance Metrics vs Threshold

- **Shows how accuracy, precision, recall, and F1-score vary with probability threshold(from RF).**
- At lower thresholds (0.1-0.2), **precision is lower**, meaning some **false positives occur**.
- At an optimal threshold (~0.5-0.7), all metrics **reach their peak**, balancing **false positives vs. false negatives**.
- At **very high thresholds (0.9+), recall drops**, meaning **some DDoS attacks go undetected**.

Key Takeaway:

- The model performs best when the threshold is **tuned around 0.5-0.7**, avoiding both **false alarms** and **missed attacks**.



Business Impact

F1-Score prioritization:

- **False negatives (missed attacks) are more costly** than false positives
- **Balance between precision & recall** for real-world deployment

Security Implications:

- **Minimizing false alerts** avoids unnecessary shutdowns
- **Maximizing detection accuracy** prevents attacks from going unnoticed

Thanks for the opportunity ☺

Questions?

Thank you! Feel free to ask any questions.

GitHub Repository: https://github.com/Apurva3509/DuneSec

Email: amp2365@columbia.edu

Website: www.patelapurva.com