Network Threat Detection using Machine Learning

EECE5644 : Introduction to Machine Learning and Pattern Recognition

Submitted By:
Akshata Kumble
Vidya Kalyandurg

Introduction

Network threats are growing more advanced, creating serious risks for both organizations and individuals. Protecting sensitive information and ensuring system security requires effective and timely threat detection.

This project harnesses the power of Machine Learning to build a threat detection system by examining network traffic and uncovering patterns that signal potential attacks. The system aims to strengthen defenses and proactively address risks before they escalate.

Overview

Objective:
Develop a threat detection system using machine learning techniques to enhance cyber security

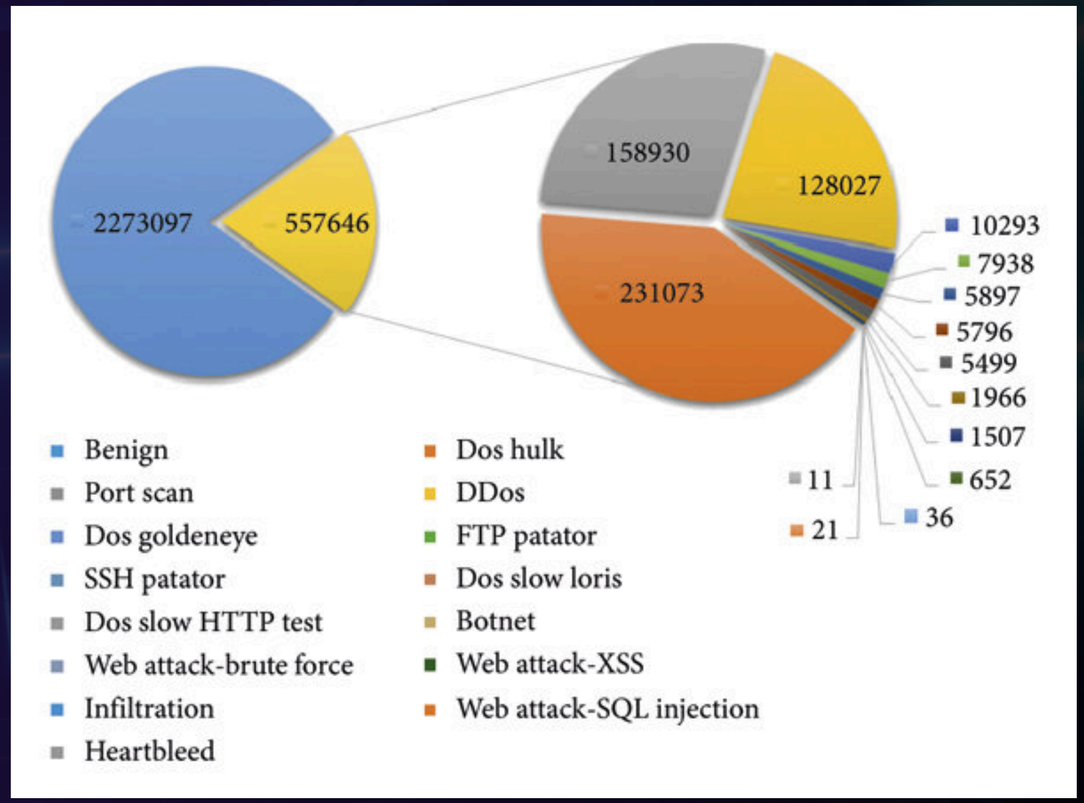
Dataset:
CIC-IDS2017, a comprehensive dataset for network intrusion detection.

Project Overview:
Step 1: Data Preprocessing
Clean and prepare data from the "CSVs" folder for analysis.
Step 2: Attack Data Filtering
Create separate files for 12 attack types for focused analysis.
Step 3: Feature Selection
Identify top features to feed into machine learning models.
Step 4: Machine Learning Evaluation
Test seven algorithms, analyze results, and generate visualizations.

Significance:
Enhances cyber security by automating the detection of various attack types with high accuracy.

Intrusion detection evaluation dataset (CIC-IDS2017)

CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).



Key Features of the Dataset:

- Diversity of Traffic: Captures a mix of benign and malicious activities, reflecting real-world scenarios.
- Attack Variety: Includes 12 different attack types such as Brute Force, DoS/DDoS, Heartbleed, Web Attacks, and Botnet.
- Complete Network Configuration: Simulates a full network environment with routers, firewalls, and multiple operating systems (Windows, Ubuntu, MacOS).
- Realistic Background Traffic: Generated using the B-Profile system to mimic human behavior across protocols like HTTP, FTP, SSH, and email.

Dataset Structure

- *Data Collection Period: July 3–7, 2017 (5 days).*
  - *Monday: Normal traffic only.*
  - *Tuesday to Friday: Mixture of attacks and normal activities.*
- *Data Format:*
  - *PCAP Files: Raw network traffic.*
  - *CSV Files: Labeled flows with over 80 features extracted using CICFlowMeter.*

Attack Timelines and Highlights:

- *Brute Force Attacks: FTP and SSH on July 4.*
- *DoS/DDoS Attacks: Slowloris, Hulk, GoldenEye, and others on July 5.*
- *Web Attacks: SQL Injection, XSS on July 6.*
- *Botnet and Port Scans: On July 7, including ARES botnet activity.*

Importance for Threat Detection:

- *Rich Feature Set: Enables the evaluation of machine learning models with diverse attributes.*
- *Labeling and Metadata: Includes timestamps, IP addresses, protocols, and attack types for precise analysis.*
- *Comprehensive Evaluation: Meets criteria for network topology, attack diversity, and protocol variety, making it ideal for developing robust AI-based detection systems.*

DATASET PREPROCESSING & ATTACK DIVISION

Data Preprocessing

Dataset is prepared by cleaning, normalizing, and encoding data.

- Key steps include handling missing values and infinities, normalizing numerical features, encoding categorical variables and concatenating the preprocessed data

Attack Division

The Objective of Attack Division is to separate attack types from benign traffic to create binary classification datasets.

Process:

- The data can be divided into 14 attack categories and one benign category.
- Data Handling:
  - Filter and combine attack data with corresponding benign instances.
  - Shuffle datasets for randomness.
  - Save each pair as a separate CSV file (e.g., DDoS_vs_BENIGN.csv).

Result:

- Created datasets for attacks like DDoS, DoS, PortScan, Web Attacks, etc.
- Ensured balanced data for accurate classification and testing.

FEATURE SELECTION

Feature Selection Process:

• A Random Forest Regressor is trained on the dataset, treating it as a binary classification problem:

• Label Encoding: The “Label” column is transformed into binary values, where non-benign (attack) traffic is labeled as 1 and benign traffic is labeled as 0.

• The feature importances are computed from the trained Random Forest model. These importances measure how much each feature contributes to the prediction.

Output and Visualization:

• The top 20 most important features for each attack type are printed along with their contribution percentages.

• These results are saved as CSV files for further use.

• Bar plots are created to visually display the relative importance of the top 20 features for each attack type.

Iterative Processing:

• The process is repeated for every attack type in the attack_types list.

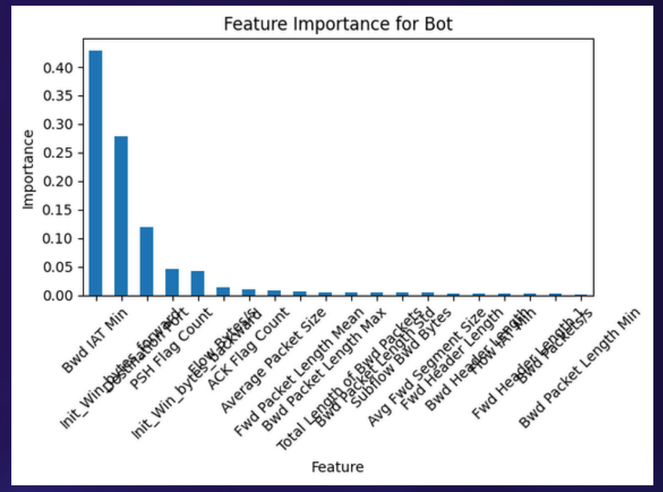
The feature importances reveal which features are most critical for distinguishing attack traffic from benign traffic.

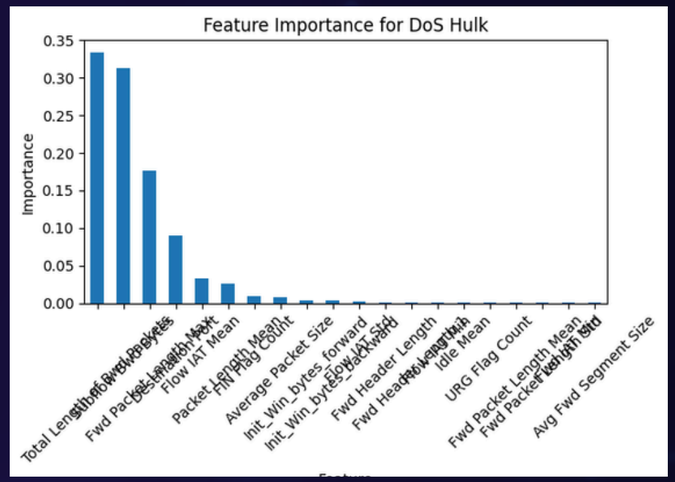
FEATURE SELECTION

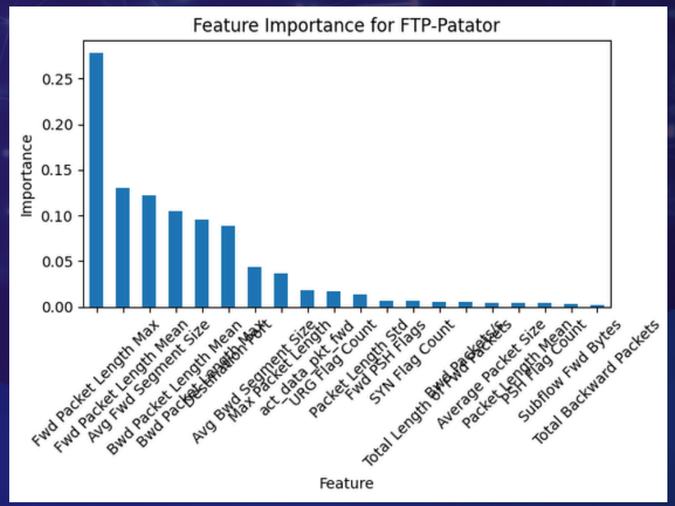
Overall Insights:

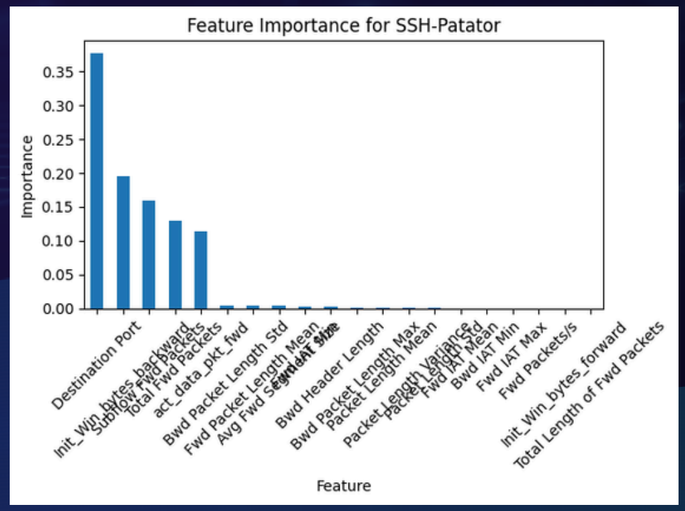
- Destination Port and Packet Length metrics frequently appear as significant, indicating they are critical for identifying most types of attacks.
- Flow-related features like Flow Packets/s and Subflow Fwd Packets highlight that monitoring packet dynamics is essential for detecting anomalies.
- Flag counts (PSH Flag Count, ACK Flag Count) play a significant role in DoS and infiltration attacks, suggesting that abnormal flag usage is a red flag (pun intended).

FEATURE SELECTION









Machine Learning Models

Goal: Evaluate the performance of three classifiers in detecting various cyber attacks.
Dataset: Compares different attack types against benign traffic using important features selected for each attack.
Models Evaluated: Gaussian Naive Bayes, QDA, and MLP.

Overview of Models

Gaussian Naive Bayes

- Works well when the assumption of independence roughly holds.

Quadratic Discriminant Analysis (QDA)

- Models each class with a separate Gaussian distribution and allows for different covariance matrices per class.
- Captures non-linear decision boundaries, unlike Naive Bayes.
- Sensitive to data imbalance and small sample sizes.

Multi-Layer Perceptron (MLP)

- A type of neural network with one or more hidden layers.
- Capable of learning complex, non-linear relationships.
- Requires more data and computational resources but can adapt to diverse scenarios.

Analysis and Insights

Naive Bayes

- *Performed well for simpler attack types like "Bot" and "SSH-Patator."*
- *Struggled with attacks like "DoS Hulk" due to feature dependencies.*

QDA

- *Strong performance in capturing non-linear patterns.*
- *Performed better than Naive Bayes in most cases but still lagged behind MLP.*

MLP

- *Outperformed other models in almost all cases, thanks to its ability to learn complex relationships.*
- *High accuracy across all attack types, including difficult ones like "DoS Hulk."*

	Attack Type	Naive Bayes Accuracy	QDA Accuracy	MLP Accuracy
0	Bot	0.998870	0.996972	0.999462
1	DDoS	0.974113	0.981548	0.998230
2	DoS GoldenEye	0.987561	0.984218	0.999013
3	DoS Hulk	0.413865	0.941248	0.960518
4	DoS Slowhttptest	0.992667	0.992856	0.998386
5	DoS slowloris	0.984961	0.985161	0.998242
6	FTP-Patator	0.904381	0.859582	0.998043
7	PortScan	0.983302	0.982643	0.993626
8	SSH-Patator	0.998276	0.998788	0.998032

THANK YOU!

Code Repository:
https://github.com/Vidya1811/EECE5644_Threat_Detection