**Project 1: Advanced Intrusion Detection System using Machine Learning (CICIDS2017)**

**Executive Summary**

This project applies **Machine Learning (ML), Decision Trees, and Neural Networks** to develop an **Intrusion Detection System (IDS)** capable of identifying cyber threats such as **DDoS, Port Scans, and Web Attacks**. Using the **CICIDS2017** dataset (2.8M+ records, 79 features), this study tackles cybersecurity challenges with a **data-driven approach**.

**Objective**

* **Build a scalable, automated IDS** for real-time cyber threat detection.
* Overcome traditional systems' limitations in handling **high-volume, complex network traffic**.
* Apply **Decision Trees & Neural Networks** to detect **anomalies & rare attack patterns**.

**Key Achievements**

* **97% Accuracy** with Decision Tree model for common attacks (SYN Flag Count).
* **Neural Network outperformed** for rare attacks like **SQL Injection & Web Attacks**.
* **Data Optimization:** Chunked processing (10,000 rows) prevented memory crashes.
* **Feature Engineering:** Addressed missing values, outliers, and class imbalances.

**Dataset Overview (CICIDS2017)**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Size | Features | Attack Types |
| CICIDS2017 | 2.8M+ record | 79 | DDoS, PortScan, Botnet, SQL Injection, Brute Force |

* **Key Features:** Flow Duration, Packet Length, Flag Counts, Flow Bytes/s
* **Challenges:** Data imbalance, large size (causing crashes), missing values, and high dimensionality.

**Methodology**

**1. Data Processing & Cleaning**

* ✔**Chunked Processing (10K rows at a time) to prevent system crashes**
* ✔**Imputation (Median for Flow Bytes/s, IQR for outlier removal)**
* ✔**Feature selection using correlation analysis & heatmaps**
* ✔**Handling class imbalance with SMOTE (Synthetic Minority Oversampling Technique)**

**2. Model Development**

**a. Decision Tree Model**

* **Achieved 97% accuracy** for common attacks.
* Identified important features: **SYN Flag Count, Flow Duration, Fwd Packets**.
* **Strength:** High interpretability, fast performance.
* **Weakness:** Struggled with rare attacks.
* **Performed better for rare attack detection (SQL Injection, Web Attacks).**
* **Architecture:** 2 hidden layers, 128 neurons, dropout layers for overfitting control.
* **Weakness:** Computationally expensive, struggled with class imbalance.

**b. Neural Network Model (Optimized)**

* **Initial Accuracy:** 0.196 → **Optimized Accuracy:** 0.324
* **Performed better for rare attack detection (SQL Injection, Web Attacks).**
* **Architecture:** Increased neurons per layer, added dropout layers for overfitting control.
* **Batch Size Adjustments:** Used **64 batch size** to improve stability.
* **Epochs Increased:** Ran for **50 epochs** to refine learning.
* **Weakness:** Computationally expensive, still requires class balance improvements.

**📈Data Visualizations**

**1️⃣1. Correlation Heatmap: Identifying Key Features**

Highlights correlation between **Flow Duration, Packet Length, and SYN Flag Count**.

**Key Insight:** High correlation allows feature reduction to improve model efficiency.

A screenshot of a graph

AI-generated content may be incorrect.

**2️⃣2. Class Distribution (Bar Chart)**

**Class Imbalance:** BENIGN class dominates (~75%), rare attacks are underrepresented.

**Solution:** SMOTE used to balance dataset and improve model recall.

A screen shot of a graph

AI-generated content may be incorrect.

**3. 3️⃣Decision Tree Feature Importance**

**Top Features:** Flow Duration, SYN Flag Count, Total Fwd Packets.

**Insight:** These features are highly predictive for intrusion detection.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Business Impact**

**Key Objectives**

**1. Build a Scalable, Automated IDS for Real-Time Cyber Threat Detection**

**Impact:**

**Enhances Security Operations**: Automates detection of cyber threats, reducing the need for manual monitoring.

**Faster Response to Threats:** Identifies attacks in real-time, allowing Security Operations Centers (SOCs) to mitigate risks immediately.

**Prevents Data Breaches:** Early detection stops attackers before they compromise sensitive data.

**2. Overcome Traditional Systems' Limitations in Handling High-Volume, Complex Network Traffic**

**Impact:**

**Handles Big Data Efficiently**: My model optimizes processing large-scale network traffic without crashes.

**Improves Threat Visibility:** ML-based IDS can detect advanced persistent threats (APTs) and evolving cyber threats that traditional rule-based systems miss.

**Reduces False Positives**: Unlike traditional IDS, my ML models learn from network behavior, preventing unnecessary alerts.

**3. Apply Decision Trees & Neural Networks to Detect Anomalies & Rare Attack Patterns**

**Impact:**

**Reduces Security Gaps:** Many companies fail to detect low-frequency but high-risk threats (SQL Injection, Web Attacks).

**AI-Powered Threat Detection**: My system adapts to new attack patterns automatically.

**Enhances Network Resilience:** Helps CISOs & SOC teams focus on high-risk alerts instead of analyzing millions of logs manually.

**📌Final Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Strengths | Weaknesses |
| Decision Tree | **97%** | Fast, interpretable, effective for common attacks | Poor recall for rare attacks |
| Neural Network | **32% (Optimized)** | Effective for rare attacks (SQL Injection, Web Attacks) | Computationally expensive, requires better class balancing |

**Best Approach:** Hybrid Model (combining Decision Tree, Neural Network, and Gradient Boosting) to re-train the models and optimize them better.

**References**

Datacamp. (2024, June 27). *Decision Tree Classification in Python Tutorial.* <https://www.datacamp.com/tutorial/decision-tree-classification-python>.

Verdence, C. (2020, November 24). *How to Create a Simple Neural Network in Python.* Better Programming. Medium. <https://betterprogramming.pub/how-to-create-a-simple-neural-network-in-python-dbf17f729fe6>.

ScikitLearn. (2024). *Neural Network Models (Supervised).* [*https://scikit-learn.org/stable/modules/neural\_networks\_supervised.html*](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)*.*

Moffitt, C. (2017, February 6). *Guide to Encode Categorical values in Python.* Practical Business Python. <https://pbpython.com/categorical-encoding.html>.

Aslansefat, K. (2022). *CICIDS2017 SafeML.* Kaggle. <https://www.kaggle.com/code/kooaslansefat/cicids2017-safeml>.

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A. & Foozy, C. F. M. (2018). *Benchmarking of Machine Learning for Anomaly-based Intrusion Detection Systems in the CICID2017 Dataset*. IEEE Xplore. <https://ieeexplore-ieee-org.ezproxy.umgc.edu/document/9345704>.

Strgenix, L. D. (2022). *CIC-IDS2017-00-cleaning.* Kaggle. <https://www.kaggle.com/code/dhoogla/cic-ids2017-00-cleaning>.

UNB. *Canada’s Research Leader in Cybersecurity*. <https://www.unb.ca/cic/research/index.html>.

Rodriguez, M., Alesance, A., Mehavilla, L., & Garcia J. (2022, Nov 30). *Evaluation of Machine Learning Techniques for Traffic Flow-based Intrusion Detection*. PMC PubMed Central. National Library of Medicine. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9740321/>.

Lalduhsaka, R., Bora, N. (2022, Oct). *Anomaly-based Intrusion Detection Using Machine Learning: an Ensemble Approach.* ResearchGate. <https://www.researchgate.net/publication/364437088_Anomaly-Based_Intrusion_Detection_Using_Machine_Learning_An_Ensemble_Approach>.

Wilkinson, L. & Friendly, M. (2009, May). *The History of the Cluster Heatmap*. American Statistical Association. <https://www.cs.uic.edu/~wilkinson/Publications/heatmap.pdf>.

Indratmo, Howorko, L. Boedianto, J. M., Daniel B. (2018). *The Efficacity of Stacked Bar Charts in Supporting Single-attribute Comparisons.* Science Direct. <https://www.sciencedirect.com/science/article/pii/S2468502X18300287>.

Kotsiantis, S. B. (2011). *Decision trees: A recent overview*. Artificial Intelligence Review*, 54*(4), 2415–2453. <https://moscow.sci-hub.se/1859/e4793845940124a40551f7ce3e502b6b/kotsiantis2011.pdf>.

Schmidhuber, J. (2014, October 8). Deep learning in neural networks: An overview. *Neural Networks, 140*, 87–117. <https://arxiv.org/pdf/1404.7828>.