IST.718.M001.SPRING25.Big Data Analytics

Project Report

Intrusion Detection System

Team Members:

Nithish Kumar Senthil Kumar

Subhiksha Murugesan

Rishi Manohar Manoharan

Kunal Ahirrao

Project Overview

The rise of cyber threats in modern digital infrastructures demands the development of highly accurate Intrusion Detection Systems (IDS). This project aims to address this demand through the application of supervised machine learning models to detect and classify network intrusions. Utilizing the comprehensive CSE-CIC-IDS2018 dataset, we engineered a multiclass classification model capable of identifying benign traffic and differentiating it from malicious activities such as Denial of Service (DoS), Distributed Denial of Service (DDoS), and Brute Force attacks. Given the magnitude of the dataset, which contained approximately 10 million flow records, extensive efforts in data preprocessing, memory optimization, feature engineering, model selection, and evaluation were essential for successful project execution.

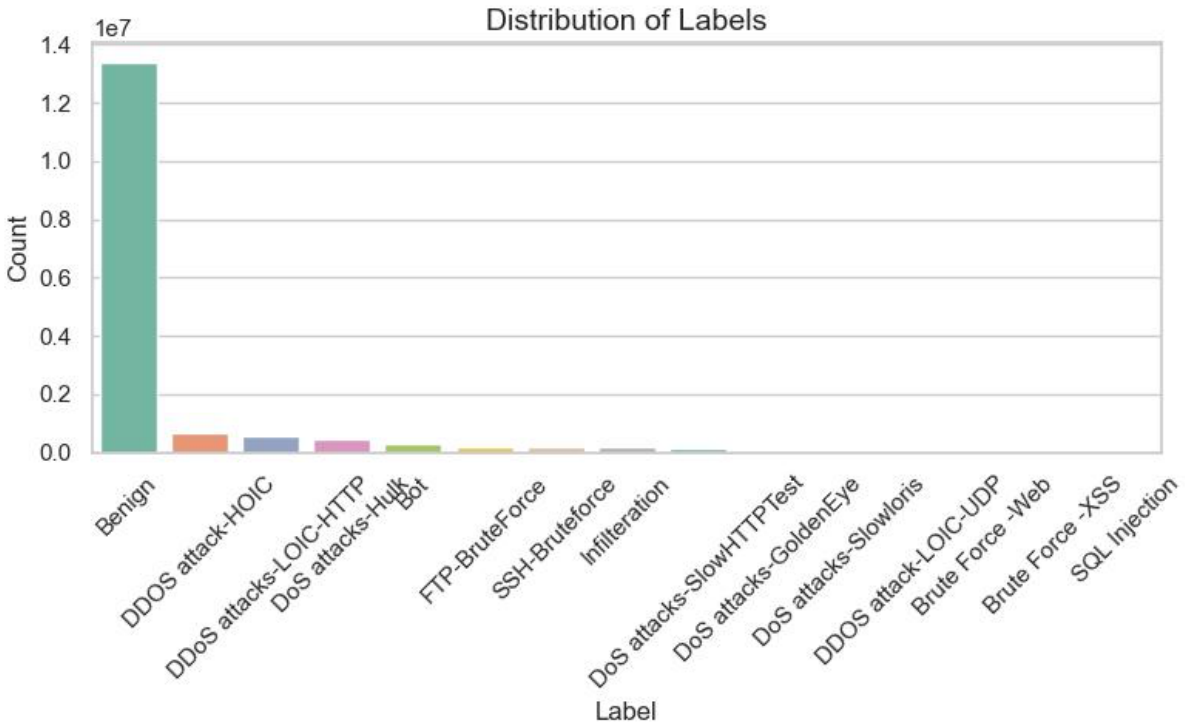
Project Goals

- Construct a supervised machine learning pipeline tailored for multiclass traffic classification tasks.
- Mitigate the computational and memory challenges inherent in handling massive datasets within cloud-based environments like Google Colab.
- Employ feature selection and dimensionality reduction techniques to enhance model interpretability, reduce overfitting, and optimize training times.

- Address significant class imbalances through strategic sampling methods such as random undersampling.
- Systematically benchmark multiple machine learning algorithms to identify the optimal predictive model for IDS deployment.

Data Exploration and Preprocessing

Dataset Description: The CSE-CIC-IDS2018 dataset is an industry-standard benchmark designed to emulate corporate network environments. It contains around 81 features per flow, encompassing packet-level statistics, byte counts, inter-arrival timings, and protocol-specific flags. The dataset covers diverse attack categories including Brute Force SSH, DoS GoldenEye, DDoS LOIC, Botnet behaviors, and HTTP flood assaults. Each record provides granular insights into the session-level interactions between networked devices.



Key Data Issues: Data exploration revealed several inconsistencies and quality challenges that needed resolution prior to modeling:

- Mixed data types, where numeric fields such as ports and packet sizes were erroneously stored as strings.
- The presence of "Infinity" and "NaN" values encoded as text, leading to parsing errors during loading.
- Severe memory inefficiency, with default float64 and int64 data types causing load failures in environments with limited RAM.

Cleaning and Optimization: To address these challenges, the following data engineering interventions were employed:

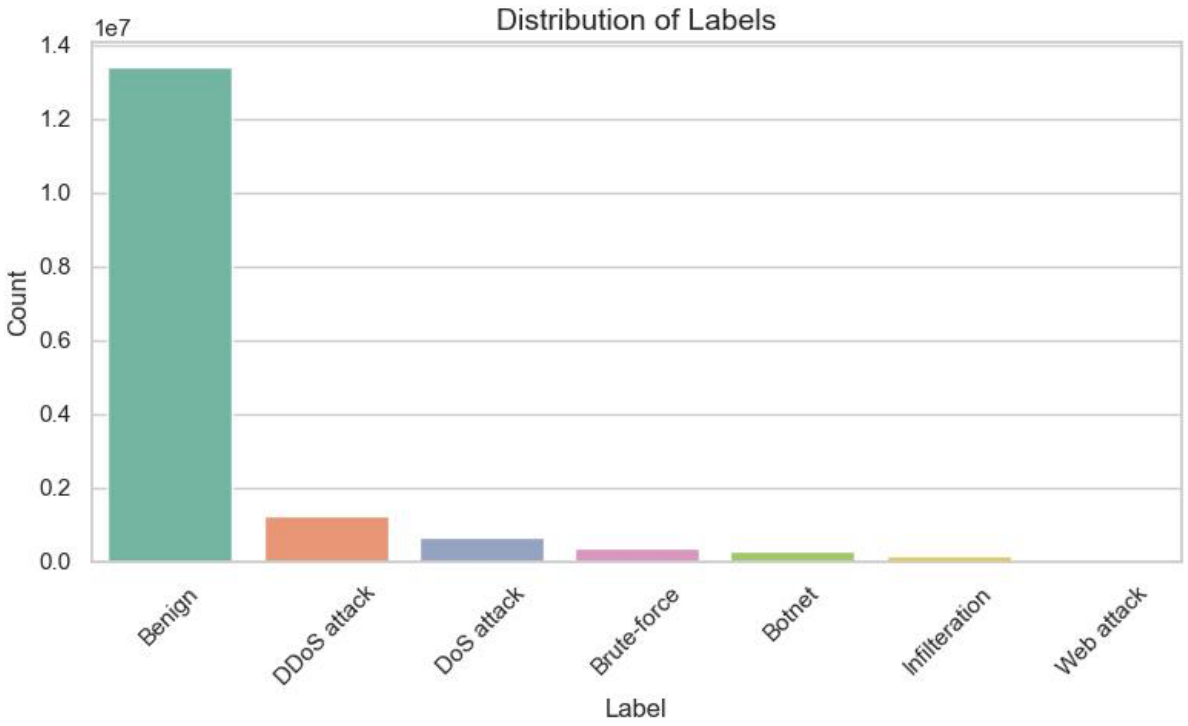
- Column-wise explicit data type casting to compact representations like uint8 for discrete variables and float32 for continuous variables.
- Systematic replacement of invalid string placeholders with appropriate numerical surrogates.
- Aggregation of ten CSV files into a singular, validated DataFrame, followed by exportation into the Parquet format, resulting in a substantial reduction in memory overhead and load times.

Attack Class Harmonization and Balancing

Initial inspection revealed the presence of more than a dozen distinct attack types, many of which were underrepresented and unsuitable for effective machine learning modeling. To create a coherent classification schema:

- Semantically similar attack types were consolidated into broader categories.
- Classes comprising less than 2% of the dataset were omitted to prevent data sparsity.
- A Random Undersampling approach was applied to achieve class balance, resulting in a harmonized dataset with approximately 400,000 samples per class, totaling 1.6 million records.

This process ensured that the classifier would not be biased toward the majority class (Benign traffic) and could generalize better to minority classes.



Feature Selection

Robust feature selection was crucial to enhance model generalization and reduce training time:

- Features exhibiting near-zero variance were eliminated using a Variance Threshold filter.
- Redundant features identified via domain knowledge and statistical correlation were pruned.
- Highly correlated feature pairs ($\rho > 0.9$, Spearman correlation) were analyzed, and one feature from each pair was discarded to avoid multicollinearity.
- After these steps, the resultant feature set preserved maximum information content while significantly decreasing dimensionality.

The final feature set provided a rich yet manageable foundation for effective machine learning model training.

Modeling Pipeline

Data Partitioning and Scaling:

- The dataset was partitioned into training (80%) and testing (20%) subsets, using stratified sampling to preserve class distribution.
- Feature scaling was conducted using MinMaxScaler to normalize all input attributes within a [0, 1] range, thereby preventing scale dominance issues during model training.

Machine Learning Algorithms Evaluated:

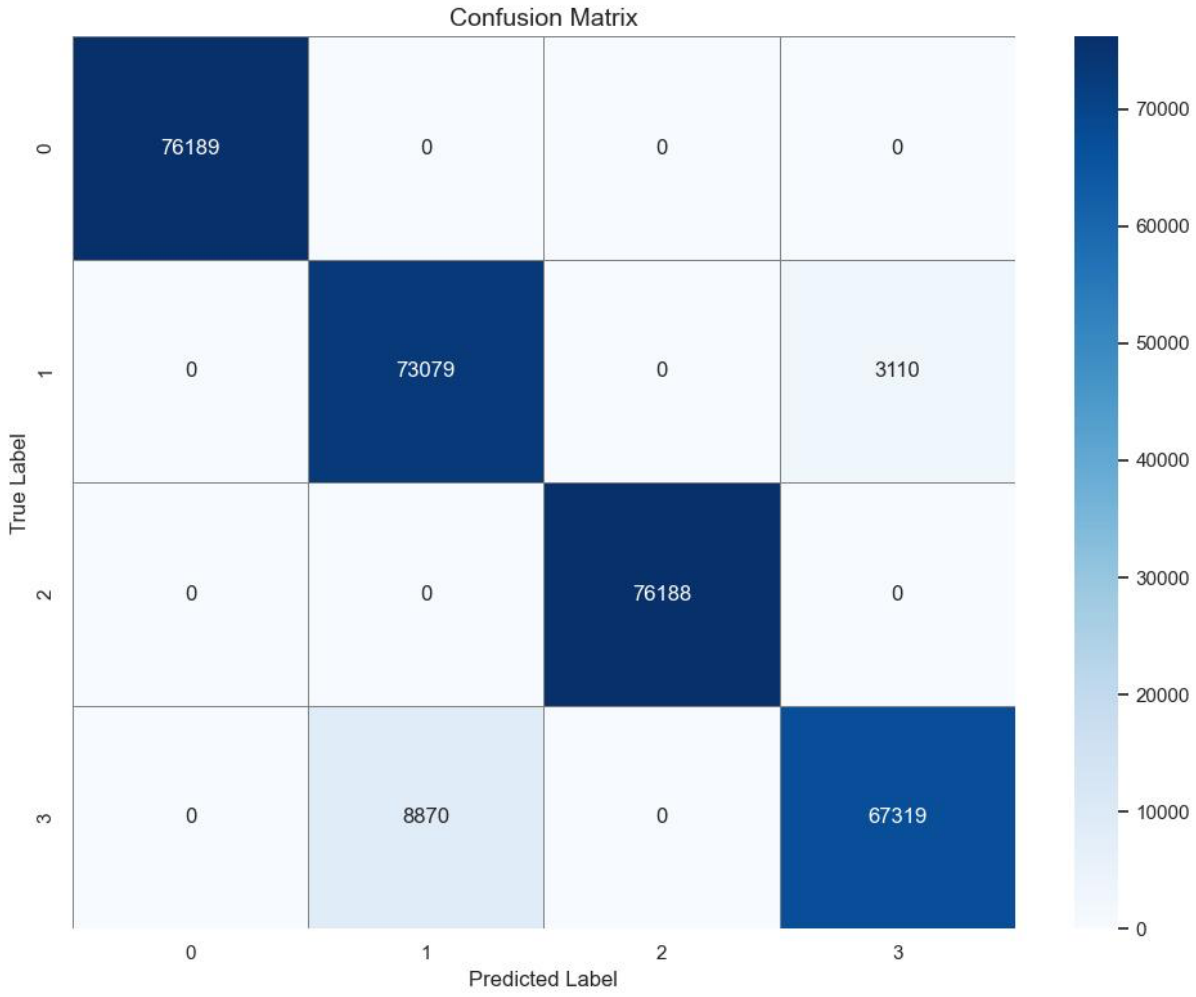
- **Decision Tree Classifier:** Served as a baseline model. While intuitive and easy to interpret, it exhibited a high tendency to overfit the training data.
- **Random Forest Classifier:** This ensemble technique reduced overfitting compared to single trees and offered better generalization but was resource-intensive.
- **Extreme Gradient Boosting (XGBoost):** The most sophisticated model tested, XGBoost leveraged regularized boosting to enhance predictive performance while maintaining computational efficiency through parallelized execution.

Evaluation Metrics:

- Overall Accuracy
- Class-specific Precision, Recall, and F1-Scores (macro-averaged to account for class imbalance)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76189
1	0.89	0.96	0.92	76189
2	1.00	1.00	1.00	76188
3	0.96	0.88	0.92	76189
accuracy			0.96	304755
macro avg	0.96	0.96	0.96	304755
weighted avg	0.96	0.96	0.96	304755

- Area Under the Receiver Operating Characteristic Curve (ROC-AUC)
- Confusion Matrix Analysis to identify false positives and false negatives



Final Model and Performance

After comprehensive experimentation, the XGBoost classifier was selected as the final model:

- **Accuracy:** Achieved an impressive 96% on the holdout test set.
- **Macro-averaged F1-Score:** High F1-scores across all classes indicated strong class-wise performance, balancing precision and recall effectively.
- **Confusion Matrix Insights:** Notably, there were zero false positives for the benign class, which is critical in operational IDS deployments where false alarms can overwhelm security analysts.
- **ROC-AUC:** High area under the curve values for each class further validated model robustness.

The model demonstrated exceptional capability in correctly identifying intrusion attempts while minimizing the incidence of false positives.

Challenges Faced

Throughout the project lifecycle, several technical obstacles emerged, necessitating innovative problem-solving strategies:

- **Memory and Scalability Constraints:** Addressed through data type optimization, downsampling strategies, and the use of efficient data formats.
- **Class Imbalance:** Resolved through Random Undersampling, ensuring equitable class representation without introducing synthetic data.
- **Data Noisiness:** Extensive preprocessing, including handling of outliers and normalization of data distributions, was crucial.
- **Feature Engineering Complexity:** Required iterative domain-informed feature selection to isolate the most predictive attributes and mitigate dimensionality-induced model degradation.

Final Thoughts and Future Work

This project successfully delivered a scalable, highly accurate machine learning solution for network intrusion detection. The experience underscored the critical role of meticulous data preparation, thoughtful model selection, and comprehensive evaluation metrics in building effective IDS models.

Future research directions and enhancements include:

- **Binary Classification:** Simplifying the problem space to binary outcomes (benign vs. malicious) to explore improvements in sensitivity and specificity.
- **Deep Learning Integration:** Experimentation with CNNs and LSTMs to capture spatial and temporal dependencies within network flows.
- **Real-time IDS Deployment:** Integrating the model into live network environments with online learning capabilities to adapt dynamically to evolving threats.
- **Ensemble Model Approaches:** Combining multiple classifiers via stacking or blending to potentially exceed the performance ceiling achieved by single models.

Works Cited

"CSE-CIC-IDS2018 Dataset." Canadian Institute for Cybersecurity, University of New Brunswick, 2018, https://www.unb.ca/cic/datasets/ids-2018.html. Accessed 27 Apr. 2025.

Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

Sommer, Robin, and Vern Paxson. "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection." 2010 IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 305–316.

Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. "A Deep Learning Approach to Network Intrusion Detection." IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, 2018, pp. 41–50.