Anomaly and Intrusion Detection Using Machine Learning

Team Details

Team Mentor: *Aishwarya Sarda*

Team Members:
- ❑ Suvrat Sanson (*Team Leader*)
- ❑ Sarthak
- ❑ Amrendra Singh
- ❑ Prapti Rana

Problem Definition

- **Technology Problem**:
  - Detection of anomalies and intrusions in network traffic using machine learning.
- **Business Problem**:
  - Preventing cyberattacks and ensuring network security by identifying malicious activities in real-time.
- **Importance:**
  - Rising frequency of cyberattacks leading to financial and reputational losses.
  - Existing systems struggle with false positives and scalability.
- **Value Additions:**
  - A robust machine learning solution for accurate anomaly detection.
  - Reduced false positives to minimize alert fatigue.
  - Scalable solution for dynamic network environments.

Suggested Solution and EDA

greatlearning

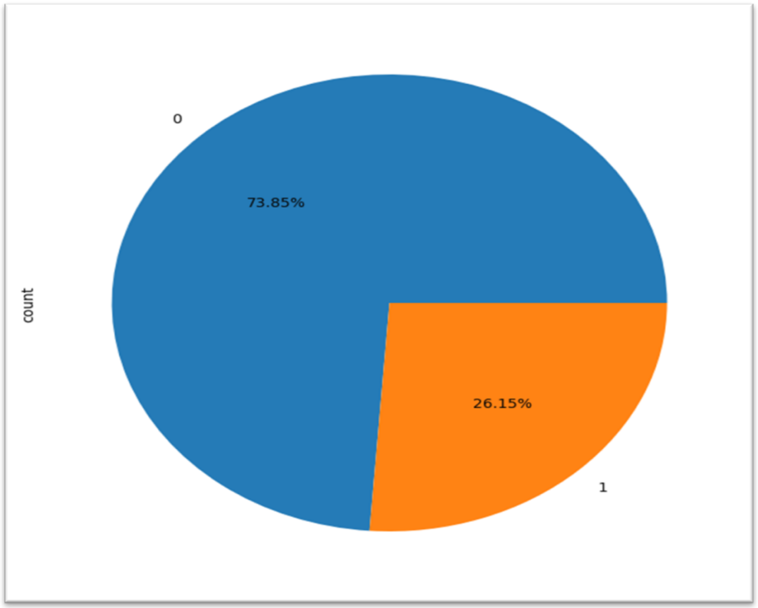
- **Solution:**
  - Machine learning models were employed to build a robust system capable of detecting anomalies and intrusions in network traffic. These models analyze patterns in data to identify malicious activities with high accuracy and reliability.
- **Dataset:**
  - UNSW-NB15 dataset with labeled normal and anomalous network traffic data.
    - **Number of Rows**: 6,72,436
    - **Number of Columns**: 49

Key Details:

- **Feature Types**:
  - Categorical features like **protocol**, **service**, **state**.
  - Numerical features like **duration**, **packet length** and **bytes transferred**.
- **Target Labels**:
  - **0**: Normal traffic.
  - **1**: Malicious traffic (various types of attacks)
- **Attack Categories**:
  - Includes different types of attacks like **DoS**, **Exploits**, **Reconnaissance**, **Backdoor**, **Shellcode**, **Worms**, etc.

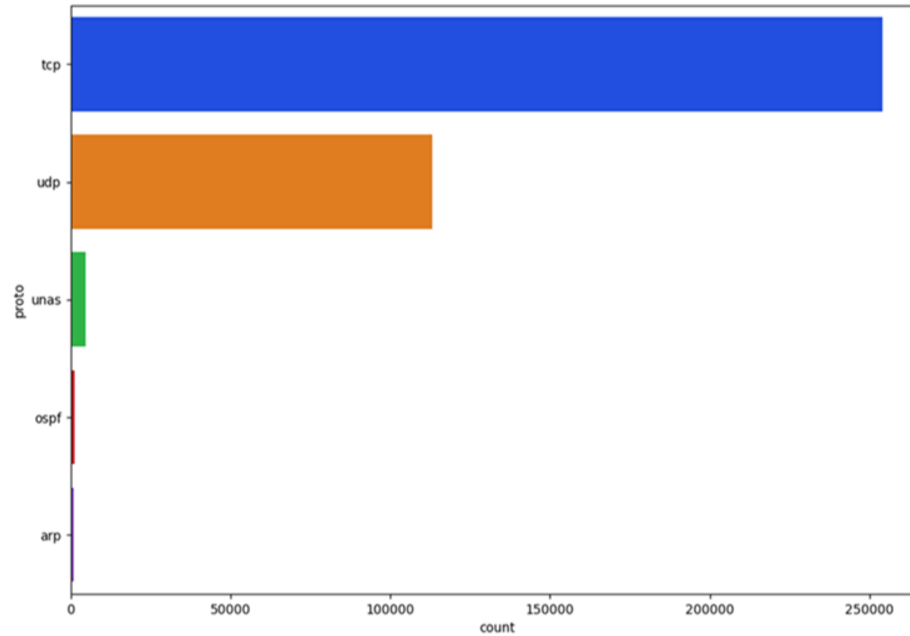
Exploratory data analytics

Class Distribution

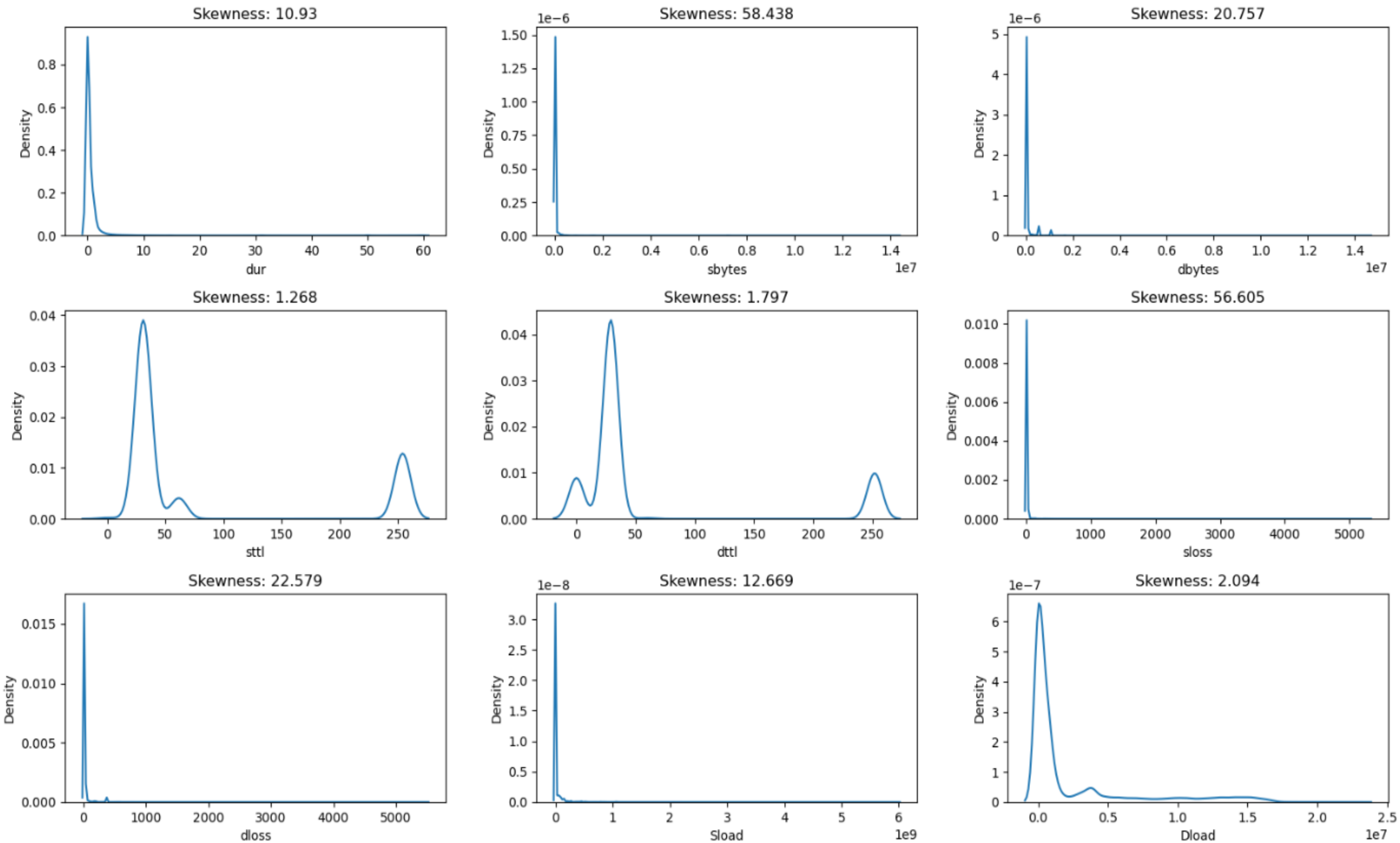


- The Label column in the dataset indicates a class distribution of 73.85% normal (benign) network traffic and 26.15% attack (malicious) traffic, demonstrating a moderate imbalance favoring benign instances.

Insights for the column proto

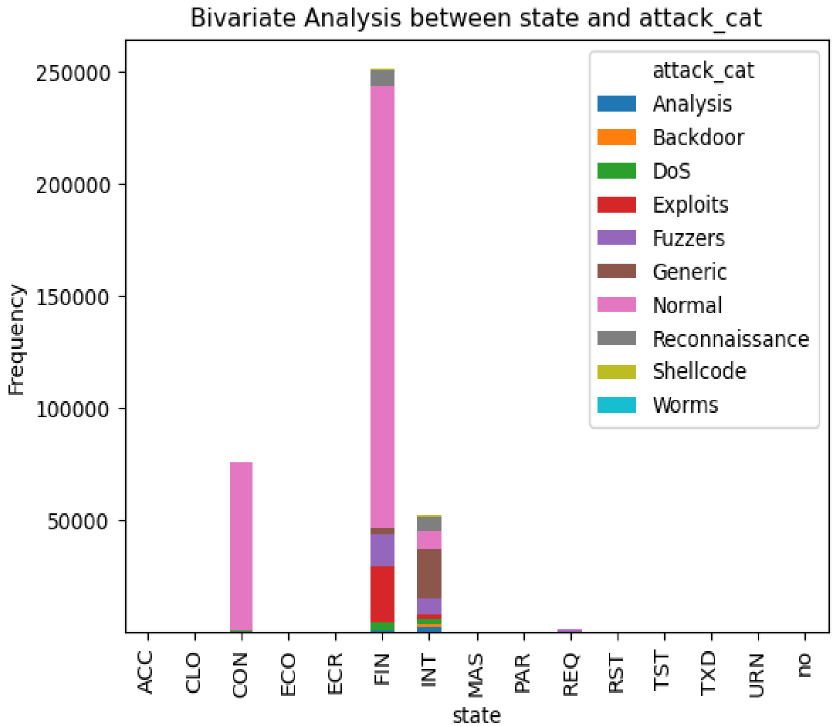


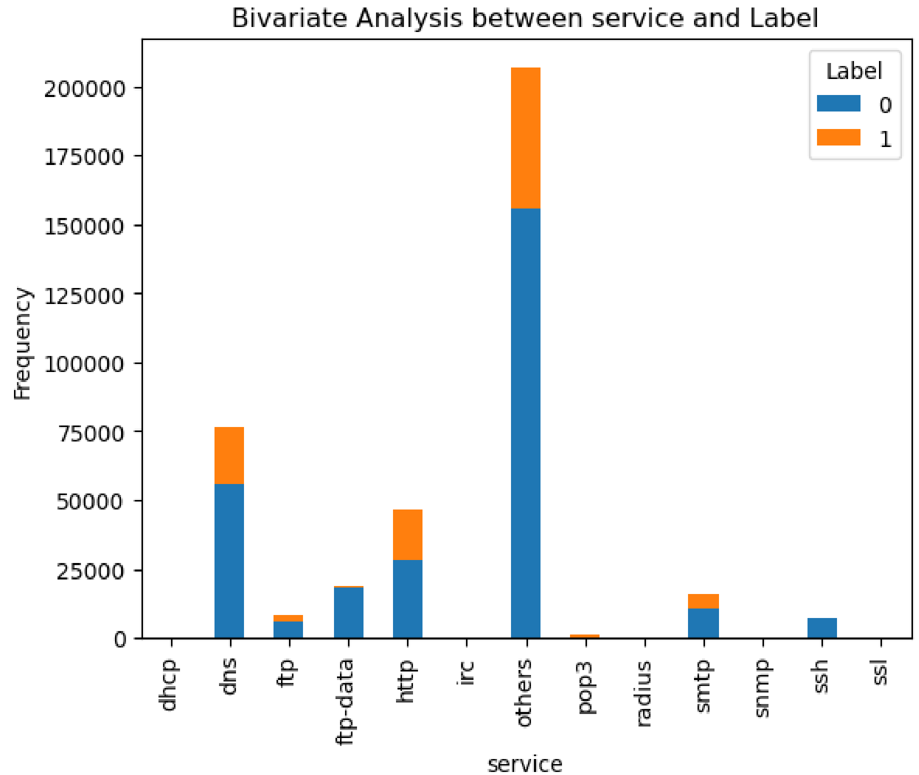
- **TCP**: The most prevalent protocol, essential for establishing connections, but prone to SYN flood attacks.
- **UDP**: Frequently used for fast, connectionless communication; high volumes may indicate regular usage or potential DoS attacks.
- **UNAS** : Uncommon; increased traffic may suggest anomalies or suspicious activity.



Feature Distribution:

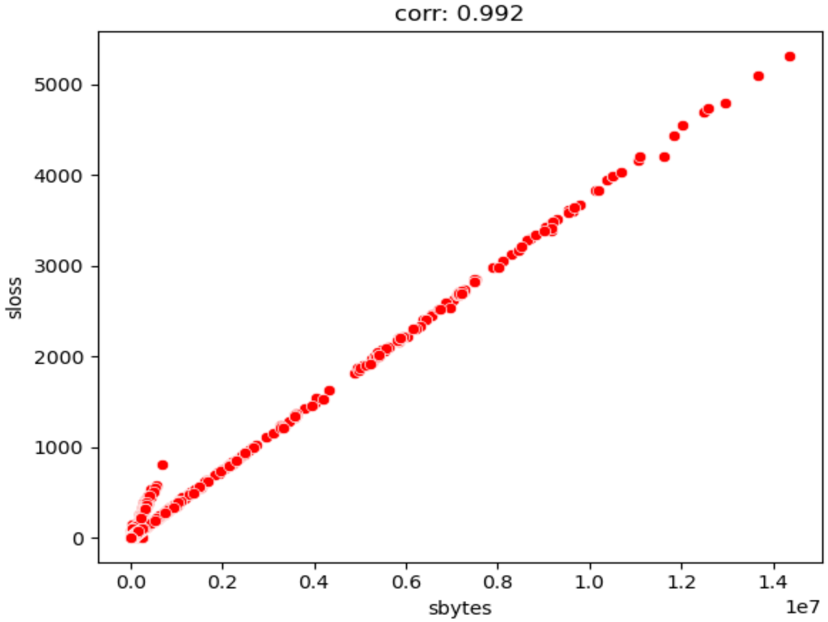
- Majority features show right-skewed distributions.
- Normal traffic: short, simple, low-volume patterns.
- Outliers suggest potential anomalies/attacks.
- Stable baseline with distinct deviation patterns.

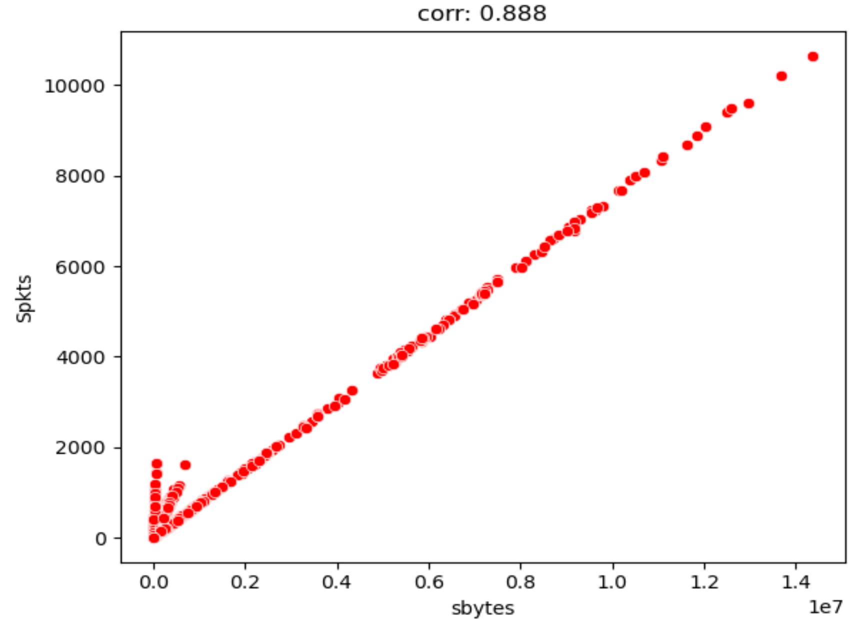




- The **FIN state** is mostly benign with occasional Exploits and Fuzzers. The **INT state** shows diverse attacks, indicating malicious activity in interrupted connections. The **CON state** contains normal traffic and Worms, suggesting malware. Overall, FIN is benign, while INT and CON are key for detecting intrusions.

- The **Others category** has the highest frequency, with both normal and attack instances. **DNS, HTTP,** and **FTP** show significant frequencies, often targeted by attacks. Low-frequency services like **DHCP**, **POP3, and SSH** are mostly normal. Overall, attacks are concentrated in common services, while the Others category includes both normal and attack traffic.





- **sbytes vs. sloss:** There's a near-perfect positive correlation **(0.992)**, meaning that as source bytes (sbytes) increase, source loss (sloss) also increases almost proportionally.
- **sbytes vs. Spkts:** A strong positive correlation **(0.888)** shows that as source bytes (sbytes) increase, the source packet count (Spkts) also tends to rise, though with some variability.
- **Summary:** Higher sbytes generally results in more sloss and Spkts, indicating that increased data transfer is associated with more packet losses and a higher packet cont.

Challenges expected/addressed

- **Data Imbalance**: The dataset contained a significantly higher number of normal traffic instances compared to malicious ones, which could lead to biased predictions. This was addressed by applying SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes.
- **Duplicate Rows**: The dataset had many duplicate entries that could skew model training. We removed these duplicates to ensure data integrity.
- **Missing Values**: There were several missing values in the dataset, which were imputed using appropriate techniques to ensure consistency.
- **Outliers**: High outliers in numerical features were present, potentially affecting model performance. These were treated using the IQR (Interquartile Range) method to ensure accurate predictions.
- **Data Preprocessing**: Categorical features were encoded to numerical values to make them suitable for machine learning models. Data cleaning and standardization were performed to improve model accuracy.

Solution Architecture (Technical and Functional)

Technical Workflow

Preprocessing Pipeline:

Handle missing values.

Treat outliers using –CAPPING.

Apply Smote for class balancing.

Model Training:

Train and evaluate models using cross-validation.

Tune hyper parameters for best results.

Functional Flow

Input: Raw network traffic data.

Processing: Feature scaling, encoding, and Class balancing.

Output: Predicted label (malicious).

Algorithms considered with pros and cons

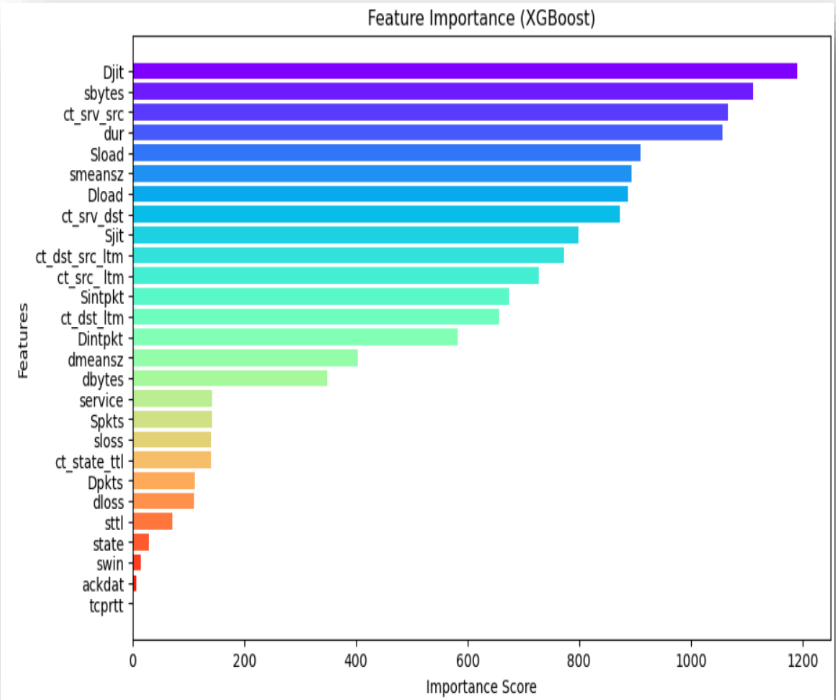
Model	Pros	Cons
Logistic Regression	- Simple and interpretable - Efficient on small datasets	- Limited to linear relationships - Lower recall and F1-score
Decision Tree	- Improved performance - Reduced overfitting	- Can still overfit with complex data - High variance without tuning
Bagging Classifier	- Reduces overfitting - Increases stability and accuracy	- Computationally intensive - Harder to interpret
Random Forest	- Handles missing values - Reduces overfitting - High accuracy	- Slower for large datasets - Less interpretable
AdaBoost	- Focuses on hard-to-predict instances - High recall and F1-score	- Sensitive to noisy data - Requires proper tuning
Gradient Boosting	- Excellent performance - High recall and precision	- Computationally expensive - Prone to overfitting without tuning
XG Boost	- Best overall performance - Optimized and scalable	- Complex to tune - Requires more computational resources

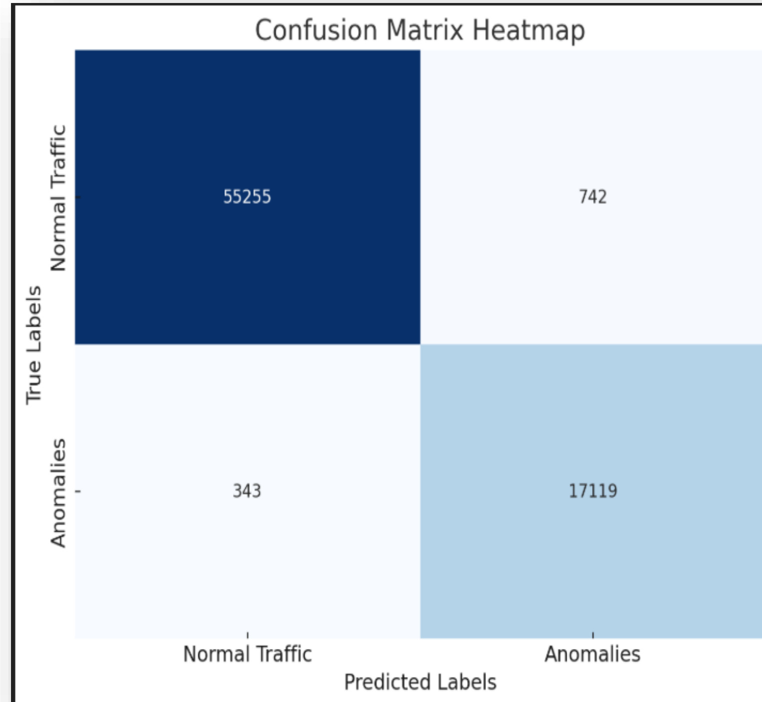
Results

Model	Accuracy	Recall	Precision	F1 Score	Cohen's Kappa
Logistic Regression	87.27%	89.24%	67.58%	76.92%	0.683
Decision Tree Classifier	98.01%	95.73%	95.91%	95.82%	0.945
Decision Tree (max-depth = 6,)	98.07%	99.98%	92.51%	96.10%	0.948
Bagging Classifier	98.39%	97.29%	95.99%	96.64%	0.956
Bagging (max-samples=6, ...)	96.48%	96.07%	89.83%	92.84%	0.905
Random Forest Classifier	98.47%	98.46%	95.23%	96.85%	0.958
Ada Boost Classifier	98.01%	99.50%	92.68%	95.97%	0.947
Gradient Boosting Classifier	98.09%	99.79%	92.74%	96.13%	0.949
XG Boost Classifier	**98.56%**	**98.86%**	**95.51%**	**97.02%**	**0.961**

The **XG BOOST** model ensures reliability with a **precision of 99%** for **normal traffic** and **96% for anomalies, minimizing false positives**

FINAL MODEL CHARTS





- Key features like **Djit, sbytes**, and **ct_srv_src** contribute 25% to feature importance.
- Traffic volume and timing dominate feature rankings, followed by connection state and service features.
- Feature importance gradually declines, reflecting a balanced and well-rounded model.

- The model **correctly classified 55,255 normal instances** and **17,119 anomalies**. However, it misclassified 343 as normal instances as anomalies and misclassify 742 as actual anomalies.

Follow-up potential capstone project problems

Enhance Recall for Anomalies: Improve the recall for anomalies (currently 98%) by refining imbalanced dataset handling techniques, like advanced SMOTE or cost-sensitive learning

Real-Time Detection: Design a real-time anomaly detection system optimized for low latency in high-traffic environments

Feature Optimization: Simplify the model by focusing on the top 15 features with importance scores above 100 to reduce complexity without sacrificing performance.

Multi-class Classification: Expand the model to classify specific attack types, improving actionable insights for network security.

Deployment Strategy: Develop a robust deployment pipeline to integrate the model into existing network monitoring systems for seamless anomaly detection.

CONCLUSION

- The XG Boost model achieved 99% accuracy and an AUC-ROC of 0.999, demonstrating excellent performance.
- Key features such as **Djit, sbytes**, and **ct_srv_src** contributed significantly (25%) to predictions, highlighting their importance in detecting traffic anomalies.
- The model achieves high reliability with 99% precision for normal traffic and 96% for anomalies, minimizing false positives while retaining the top 15 critical features for optimal balance and performance.
- This project establishes a robust anomaly detection framework, suitable for real-time applications and future multi-attack classification.

Thanks!