INFO6105 Final Project Repport

Hongting Chen

1. Problem：

As the internet continues to grow in both scale and complexity, cyberattacks have been rising rapidly. Traditional intrusion detection systems that rely on predefined rules or manually set thresholds often struggle to identify new attacks,and they also tend to generate a high number of false alarms. My goal is to use machine learning algorithms to help identify attack behavior.

In the dataset, normal traffic is labeled as 0 and attacks are labeled as 1. I will use machine learning algorithms to learn the features and determine whether a network flow is normal or an attack.

2. Data：

Dataset name: cybersecurity_intrusion_data.csv From Kaggle

Url:https://www.kaggle.com/datasets/dnkumars/cybersecurity-intrusion-detection-dataset

9537 records

Numeric Features: network_packet_size, login_attempts, session_duration, ip_repuattion_score, failed_logins.

Categorical Features: protocol_type, encryption_used, browser_type, unusual_time_access.

No missing values in dataset. All numerical features show very low linear correlation with the attack label, indicating that attack behavior is not determined by simple linear relationships. The number of normal and attack samples is slightly imbalanced. Some features contain many outliers and show strong right-skewed distributions. I chose to keep these values without applying transformations, as such extreme behaviors may represent important indicators of attacks rather than noise.

70% for training and 30% for testing

3. Model：

The selected models: Logistic Regression, Random Forest and XGBoost

None of the three models appears to be overfitting, as the training and testing accuracies are relatively close. Logistic Regression performs the worst among the three models. It has the lowest accuracy and F1-score, and it also produces the highest numbers of both false positives (FP) and false negatives (FN). Random Forest achieves the highest accuracy and F1-score, and it produces 0 false positives, the lowest among all models. Its false-negative count is the second lowest of the three models.

XGBoost shows performance similar to Random Forest, with accuracy and F1-score only slightly lower. It generates 26 false positives—higher than Random Forest but still very low—and it has the lowest false-negative count, at 295.

4. Limitations:

This project did not include hyperparameter tuning and used only basic feature engineering based on the raw attributes without constructing higher-level features.

5. Future Works:

Future work could focus on obtaining a larger and more realistic dataset that better reflects the diversity of real-world network traffic. Feature engineering could also be improved by introducing additional behavioral or derived features that may capture more subtle attack patterns. In addition, stronger machine learning models could be explored, and hyperparameter tuning could be applied to optimize the performance of the existing models.