Project Report: Multi-Label Classification for Network Traffic Analysis

Introduction

In this project, we aimed to develop a robust solution for classifying network traffic into multiple labels based on various features extracted from network data. Multi-label classification is distinct from traditional single-label classification as each instance (data point) may be assigned multiple labels simultaneously, rather than being restricted to one label from a set of discrete categories.

Data Preprocessing

The dataset was initially preprocessed to handle missing values and transform categorical data into a format suitable for machine learning models. Key steps included:

1. Cleaning Data: We identified columns with a high percentage of missing values (over 60%) and removed these columns to improve model accuracy and efficiency. Additionally, placeholders (e.g., '-') were replaced with NaN to standardize missing data representation.
2. Feature Encoding:
    - One-Hot Encoding: Applied to categorical variables like 'proto' to transform them into a binary matrix, essential for handling categorical data in machine learning algorithms.
    - Label Encoding: Used for ordinal categorical data such as 'conn_state', converting them into a model-readable numerical format.
3. Data Splitting: The cleaned and encoded dataset was split into training and testing sets, ensuring that both sets were representative of the overall data distribution.

Multi-Label Strategy

To manage the multi-label aspect of our classification task, we employed a MultiLabelBinarizer. This tool transformed our target labels into a binary format for each label category, allowing us to apply traditional classification algorithms to multi-label data.

Model Selection and Training

We chose a variety of models to compare their performance in handling multi-label classification:

1. Random Forest: Known for its high accuracy and robustness, it handles overfitting well. We used a `MultiOutputClassifier` wrapper to adapt it for multi-label tasks.

2. Decision Tree: Offers a good baseline with clear interpretability. Like Random Forest, it was adapted for multi-label classification using `MultiOutputClassifier`.
3. Gradient Boosting Machine (GBM) and XGBoost: Advanced ensemble techniques known for their effectiveness in a variety of prediction tasks. Both models handle complex non-linear relationships well.

Each model was trained using the training dataset, and performance was assessed on the testing set.

Performance Metrics

We utilized several metrics to evaluate model performance, given the multi-label nature of our task:

- Accuracy: Measures the proportion of true results (both true positives and true negatives) among the total number of cases examined.
- F1 Score (Macro): Harmonic mean of precision and recall, calculated globally across all labels.
- ROC AUC (Average): Area under the ROC curve, a plot of true positive rate against false positive rate, averaged over all labels.
- Precision and Recall (Macro): These metrics provide insights into the accuracy of positive predictions and the proportion of actual positives identified, respectively.

Results

The models demonstrated robust performance across all metrics. The Random Forest classifier achieved the highest F1 and ROC AUC scores, indicating its effectiveness in handling the complexity and variability in the data. Decision Tree and XGBoost also performed commendably, showing their capability in capturing the relationships within the data. GBM, while slightly lower in F1 score, still maintained high precision and respectable ROC AUC values.

Visualization

Confusion matrices for each model were visualized to provide deeper insights into model performance across different classes. These matrices helped in identifying any biases or weaknesses in the models, particularly in terms of false positives and false negatives.

Conclusion

Through rigorous preprocessing, thoughtful model selection, and comprehensive evaluation, this project successfully applied multi-label classification techniques to network traffic data. The models developed can significantly aid in network security systems, allowing for precise and simultaneous detection of multiple types of network behaviors and threats. Future work could explore deeper integration of model predictions into real-time network security solutions or the application of more complex neural network architectures to enhance predictive performance.