Network Intrusion Detection using ML Techniques

PART-A (Datasets)

Q1. What are the different objectives of generating benchmark datasets for intrusion detection and explain each objective in no more than two sentences? Refer this

Q2. What are the drawbacks of the existing **KDDCup'99** dataset that led to the formation of its refined version **NSL-KDD**? Why are KDPCUP'99 and NSL-KDD datasets considered unreliable to validate novel intrusion detection algorithms of late?
Refer: https://ieeexplore.ieee.org/document/8586840

Q3. Sketch a table specifying the different properties of the following datasets
KDD CUP'99, NSL-KDD, CICIDS 2017, CICIDS 2018, UNSW-NB15

1. Properties- Year of public availability of dataset, Number of features, Number of different class labels, Names of different types of attacks.

PART-B (Anomaly Detection)

For the following experiments, we will use the **CICIDS2017 and UNSW-NB15 partial datasets.**

CICIDS2017:
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv,Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv and Friday-WorkingHours-Morning.pcap_ISCX.csv

UNSW-NB15:
UNSW-NB15_4.csv, UNSW-NB15_3.csv

Required **CSV** files can be downloaded from here

This is a process of sanitizing our data. We need to ensure that our data is free from ambiguities, outliers, and biases. The first step towards realizing it would be to remove source IP, destination IP, Source port, Destination port, and Timestamp from CSV files which may add bias to the detection algorithm. Now concatenate each individual CSV file of corresponding datasets to form a unified single dataset per each CICIDS2017 and UNSW-NB15. Replace all the attack labels with label '1' and benign/no-attack sample labels with label '0'.

At this point, we translated the dataset which is suitable for anomaly detection. Replace all the categorical feature values with label encoding (don't use one-hot encoding), which inherently doesn't increase the dimensionality of the data. Drop the columns which have more than 25%

NaN or NULL values, and replace NaN/NULL values in other columns with the average(or most repetitive) value of the corresponding column. Next, remove data duplication by eliminating all redundant rows and also multi-label rows from each dataset. Eventually, normalize the data using a min-max scaler

Feature Selection (Subtask-2)

This is an important component to select the most representative features on which the predictor variable has a higher dependency. In other words, it also reduces the computation time of the learning algorithm. We use the following feature selection technique(s)

Univariate feature selection: use the SelectKBest method with chi-square as score function. Try with at least three different 'k'-values (Refer here)

At this juncture, we will have three variants of a refined version of datasets corresponding to different 'k'-value per each CICIDS and UNSW-NB15 (In total we will have six datasets).

We also offer additional Bonus points for those extending the feature selection strategy to Principal Component Analysis (PCA) with a total variance of the components of at least 90%

This completes the feature engineering task. Now split each dataset into Train, ~~validation~~, and Test subsets (80:20)

Experiment with different models (Subtask-3)

Modeling is typically a relationship assumption between the dependent and predictor variable(s). We approximate this relationship with the help of function estimators ranging from simple linear to complex non-linear models. In this sub-task, we will try to model anomaly detection with the following models

Gaussian Naive Bayes, Logistic Regression, SVM (Linear kernel), Decision Trees (ID3, ~~C4.5~~), Random Forest, XGBoost (Refer), Voting classifier (Refer, please try with Guassian naive byes, logistic regression, SVM with linear kernel methods), AdaBoost Classifier (you are free to choose the parameter), MLP classifier (with one hidden layer of size 100).

Create the confusion matrix for each model (with the test data), sketch a single table containing accuracy, precision, recall, f1-score, and running time(s) comparison among all the models
Plot the AUC-ROC curve for all the models

Dealing with Class Imbalance in Network Traces(Subtask-4)

Class imbalance is an inherent problem in anomaly detection/novelty detection. There are various ways to handle this issue i.e., data and algorithmic driven. In this subtask, we focus on data-driven approaches (a.k.a resampling techniques). We take the help of a well-known class imbalance library for our experiments.

Now for this task, we take the dataset after processing the subtask-2, and we will apply the following resampling techniques (free to choose the parameter)

1. Random OverSampling(ROS)
2. Random underSampling(RUS)
3. Synthetic Minority Oversampling TEchnique (SMOTE)

Now your latest datasets are partly balanced (class imbalance ratio will be reduced), **repeat the subtask-3 for these balanced datasets.**

Deliverables

A tar ball with:

1. A report describing your answers for PART-A of the assignment
2. For PART-B, you need to submit the python-notebook containing
    a. All the plots, confusion matrix, and comparison tables
    b. It should be executable on Google's Colab environment
3. Credit Statement (1-pager): share an accurate and detailed description of each of the group member's contributions to the assignment in terms of coding, plots, report writing, bug fixes, analysis, etc.