**Dataset Information & Exploration**

**Dataset name**: IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018)

**Link**: https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv

**Features**: 80

**File Structure:** Data is divided into various files based on date. Each individual file is unbalanced, and it is up to the notebook creator to divide up the dataset into a balanced form for higher-quality predictions.

**Background information on dataset**: This dataset was originally created by the University of New Brunswick for analysing DDoS data. You can find the full dataset [here](https://www.unb.ca/cic/datasets/ids-2018.html). This dataset was sourced fully from 2018, and will not be updated in the future, however, new versions of the dataset will be available at the link above. The dataset itself was based on logs of the university's servers, which found various DoS attacks throughout the publicly available period. When writing machine learning notebooks for this data, note that the **Label** column is arguably the most important portion of data, as it determines if the packets sent are malicious or not. Reference the below Column Structures heading for more information about this and more columns.

**Column Structure**: In total, there are eighty columns within this dataset, each of which corresponds to an entry in the IDS logging system that the University of New Brunswick has in place. Since their system classifies traffic as both forward and backward, there are columns for both. The most important columns within this dataset are listed below.

* Dst Port (Destination port)
* Protocol (Protocol used during transmission)
* Flow Duration (Duration of Connection)
* Tot Fwd Pkts (Total forward packets)
* Tot Bwd Pkts (Total backward packets)
* Label (Label)

All the features above will be kept the same for continuity throughout all development of the models. These were chosen due to

**Tips for Dataset**: Due to the large amount of data encompassed in the CSV files alone, you may want to pre-sort the data within Python, which will both speed up the actual processing of data in either PyTorch or TensorFlow.

**Neural Networks Architectures** to be Used:

* CNN (1D)
* RNN (LSTM)

**Machine Learning Algorithms** to be used:

* Decision Tree
* Support Vector Models
* Naïve Bayes
* Logistics Regression

The table below shows the type of data that can be found in the dataset:

|  |  |
| --- | --- |
| Type of Data | Description |
| Sequential Data | Each data point in the dataset is a sequence of network packets transmitted over a period of time, making it sequential. |
| Categorical Data | The dataset includes categorical features such as protocol types, service types, and attack types. |
| Numerical Data | The dataset also includes numerical features such as duration of the network connection and number of packets transmitted. |
| Imbalanced Data | The dataset is imbalanced, with a small number of attack instances compared to the number of benign instances. |
| Noisy Data | Some instances in the dataset are labelled as attacks, but they may not be real attacks or may be mislabelled. |

## Data Pre-processing Plan

Now that we understand the data it is appropriate to devise a plan to pre-process it for model analysis. The step that will be taken are as follows:

1. **Data Cleaning**: Check for missing values, remove duplicates, and fix any errors in the data.
2. **Data Transformation**: Convert categorical variables to numerical variables through one-hot encoding or ordinal encoding.
3. **Data Scaling**: Scale the data to ensure all variables are on the same scale. You could use techniques such as standardization or normalization.
4. **Feature Selection**: Identify the most important features that contribute to the model's accuracy. You could use techniques such as feature importance, correlation matrix, or principal component analysis (PCA). We have already chosen what we deem the most important as the dataset contains over 80 features.
5. **Data Splitting**: Split the data into training, validation and testing sets to evaluate the performance of the model. You could use techniques such as k-fold cross-validation or stratified sampling.
6. **Data Augmentation**: If the dataset is imbalanced, consider techniques such as oversampling, under sampling, or SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes.