MLCS – fall 2024
Assignment 3

Instructor: Hans P. Reiser

Submission Deadline: Friday, Sept 27, 2024 – 23:59

1 Network intrusion detection – decision trees

In this assignment, we will be using a different (slightly more recent) dataset of labelled network data, to be found at `https://www.unb.ca/cic/datasets/ids-2017.html`, i.e. the “CIC-IDS2017” datset. I suggest to use the pre-processed network traffic features (in the file `MachineLearningCSV.zip`)

You should aggregate the labels into four groups as follows: Benign (as it is), DoS (all labels containing the word “DoS” or “DDos”), Scan (PortScan), Exploit (all remaining labels).

1.1 Data preprocessing and splitting (2pt)

Your objective is to examine the given dataset, determine the necessary preprocessing steps, and create a Python function that accomplishes the following tasks:

- Read all CSV files located in a specified folder (provided as the first argument to the function)
- Apply the preprocessing steps you consider appropriate to the data.
- Split the processed data into two distinct Pandas data frames: one for the training set and one for the testing set.
- Return these two Pandas data frames containing the labeled datasets.

```python
def get_dataset(path_to_files, splitmode):
    ...
    return d_train, d_test
```

If splitmode is a floating point number $0 < x < 1$, divide the entire dataset (aggregated from all CSV files) into a randomized subset, with a fraction of 'x' records allocated to the training set and the remaining records assigned to the test set.

Otherwise, split the dataset such that all data from Monday–Wednesday forms the training dataset, and all the data from Thursday/Friday forms the test dataset.

1.2 Train and test a decision tree classifier (2pt)

Implement Python code (using scikit-learn) that trains and tests a decision tree classifier. Perform experiments with two different flavours of the dataset: (1) with a 60%:40% random split (i.e. $x = 0.6$) of the full dataset, and (2) with the dataset split by days of the week.

1.3 Discuss the results you observed (2pt)

In your report, include a confusion matrix (with the four aggregated traffic classes Benign, DoS, Scan, Exploit) for the two datasets, and try to explain (one paragraph / half a page maximum) differences in performance you might observe between the two datasets.

1.4 Random Forest classifier (2pt)

Replace your decision tree classifier with a random forest classifier.

Repeat the experiments of the previous item, and present the new confusion matrix (for both datasets), with a (short) explanation how you would interpret the observation.

1.5 Explainability (2pt)

Have a look at the decision trees/random forests generated by the previous steps. can you identify some main features that were used as discriminating node labels (this is easier for simple decision trees – for random forests you have to look up approaches for feature importance calculation)? Can you explain why these features might be useful features for detecting specific attack classes?