CSC 180-01 Intelligent Systems

Modern Low Footprint Cyber Attack Detection

1. Problem Formulation

Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. This project aims to build a **network intrusion detector**, a predictive model capable of distinguishing between bad connections, called intrusions or attacks, and good normal connections.

Model this problem as a BINARY classification problem. Use the following models to detect bad connections (intrusions). Compare the recall, precision and F1-score of the models for attacks and normal connections, respectively. PLOT the confusion matrix and ROC curve for each model.

- Logistic Regression (scikit-learn)
- Support Vector Machine (scikit-learn)
- Fully-Connected Neural Networks (TensorFlow)
- Convolutional Neural Networks (TensorFlow)

2. Dataset

https://research.unsw.edu.au/projects/unsw-nb15-dataset

The UNSW-NB 15 dataset was created in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) which reflects **modern low foot print attacks**. UNSW-NB 15 dataset contains a hybrid of real modern normal activities and synthetic contemporary attack behaviors, as shown in Figure 1. This dataset has **nine types of attack categories**, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

The dataset has totally 49 features with the class label. The label for each record is either 0 if the record is normal and 1 if the record is attack.

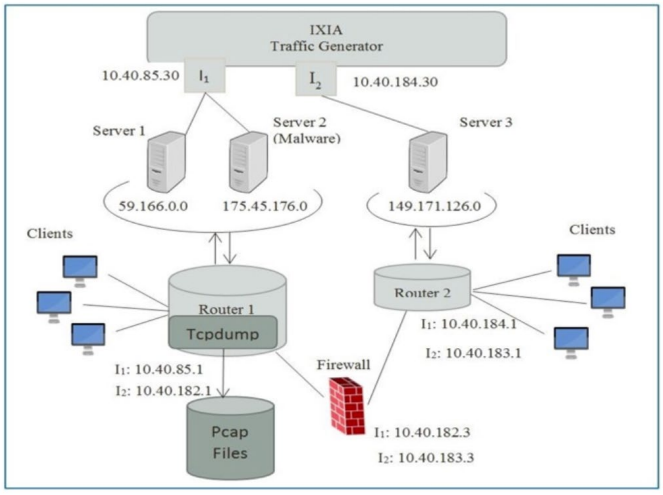


Figure 1: UNSW-NB15 Testbed

In this project, let's focus on **a subset of the UNSW-NB 15 dataset**, a partition configured as a training set and testing set, namely, UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv respectively, which can be downloaded from the following link:

(on canvas)

The number of records in the training set is 175,341 records and the testing set is 82,332 records from the different types, attack and normal.

Read the paper "UNSW NB15: A Comprehensive Data Set for Network" or go to the UNSW-NB15_features.csv file for detailed feature description.

3. Requirements

- Use training data to train your models and evaluate the model quality using test data
- Note that the categorical values in training data may not exactly match the categorical values in test data. **Remove all the records with categorical values that only appear in training or test data.**

  Hint: use function unique() https://favtutor.com/blogs/pandas-unique-values-in-column
- Drop any rows with missing values.
- Encode categorical features and normalize numeric features.

- You must use EarlyStopping and ModelCheckpoint when training neural networks using Tensorflow.

- Tune the following hyperparameters when training neural networks using Tensorflow to tabulate all the results on how they affect performance in your report. **Tabulate your findings.**
  - **Activation:** relu, sigmoid, tanh
  - **Layers and neuron counts**
  - **Optimizer:** adam and sgd

4. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

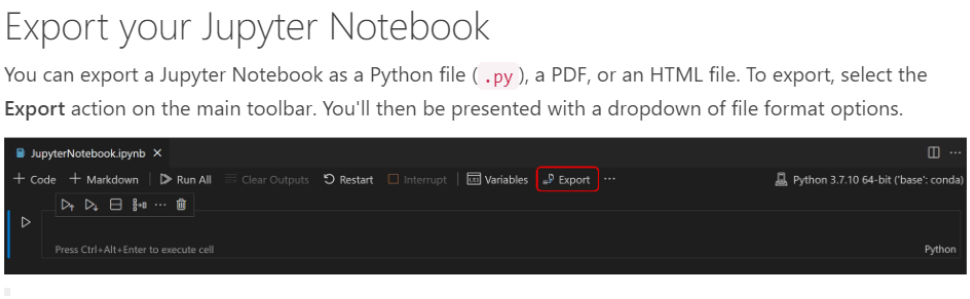
Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

5. Teaming

Students must work in teams of at most 4 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserves the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

6. Deliverables

(1) The HTML version of your notebook that includes all your source code.



5 pts will be deducted for the incorrect file format.

(2) **Your report in PDF format,** with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement
- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

(3) A **separate text file** named "additional.txt", which describes the additional features you implemented.

NO late submissions will be accepted.

7. Additional Features

(1) Can you model this intrusion detection problem as **a multi-class classification problem** so that we can detect the type of each intrusion? How good such predictive model can be in terms of detecting each specific attack?

(2) To build a multi-class classifier, can you create **a more balanced dataset** to train your model so that you model will not be biased to the more frequent classes? Perform downsampling or oversampling.

https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf

(3) Among all the features, can you identify the most important features (this is so called **feature importance analysis**) and train models only on those important features, e.g., top-10 most important features? Hint: use logistic regression to identify the coefficient for each feature.

(4) Another dataset for you to play with about **IoT applications**

https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php