

CSC 215-01 Artificial Intelligence (Fall 2019)

Mini-Project 1: Modern Low Footprint Cyber Attack Detection

Due at 5:30 pm, Wednesday, September 25, 2019

Demo Session: class time, Wednesday, September 25, 2019

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 accuracy, recall, precision and F1-score of ALL the models. PLOT the confusion matrix and ROC curve for each model.

- Logistic Regression
- Nearest Neighbor
- Support Vector Machine
- Fully-Connected Neural Networks

2. Dataset

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

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.

A few tools and twelve algorithms were used to **generate totally 49 features with the class label**. This data set is labelled as attack categories (i.e., attack_cat) and label for each record, **either 0 if the record is normal and 1 if the record is attack**.

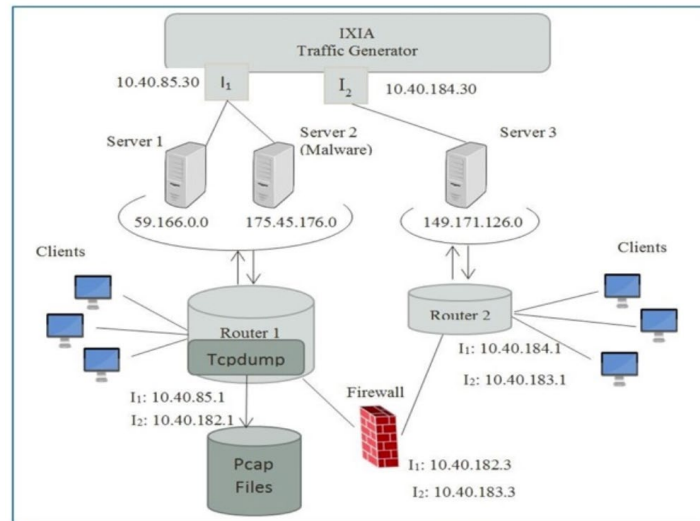


Figure 1: UNSW-NB15 Testbed

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:

<https://drive.google.com/open?id=1-jiDgzfbnTzyF5MZDGzictIKjnJQ39lj>

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

- Split data for training and testing. Use training data to train your models and evaluate the model quality using test data
- 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.
- Tuning the following hyperparameters when training neural networks using Tensorflow to see how they affect performance
 - **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.

Implementation	70 pts
Your report	20 pts
In-class defense	10 pts

5. Teaming

Students must work in teams of 2 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 reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

6. Deliverables

- (1) **All your source code** in Python Jupyter notebook.
- (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.

10 pts will be deducted for missing the section of task division and project reflection.

All the files must be submitted **by team leader** on Canvas before

5:30 pm, Wednesday, September 25, 2019

NO late submissions will be accepted.

7. In-class Demo:

Each team member must demo your work during the scheduled demo session. The following is how you should allocate your time:

- Model design (1 minute)
- Findings/results (2 minute)
- Task division (1 minute)
- Challenges encountered and what you have learned from the project (1 minutes)

Failure to show up in demo session will result in **zero** point for the project.

8. Think beyond the project

- (1) Among all the features, can you identify the important features (this is so called feature importance analysis) and train models only on those important features, e.g., top-10 features? What would be the benefits to do that?
- (2) 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 the predictive model can be in this case?
- (3) Can you create a more balanced dataset to train your model so that your model will not be biased to the more frequent classes?
- (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