# CSC 180-01 Intelligent Systems (Fall 2025)

## Project 2: Modern Low Footprint Cyber Attack Detection

**Due at 10:30 am, Monday, October 13, 2025**

**Demo: Monday, October 13, 2025**

Note that you must **print and fill in** your names on the Evaluation Form and bring the copy to your demo session to receive credit.

## 1. Problem Formulation

In this project, we will practice with **cybersecurity-based AI** such as intrusion detection systems, threat intelligence, malware detection, fraud detection, privacy-preservation, digital forensics, adversarial machine learning, and threat hunting. Software to detect network intrusions protects a computer network from unauthorized users. This project aims to build a **network intrusion detector**, a predictive model capable of distinguishing between bad, malicious 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 your best model in terms of the F1-score for intrusion.

- Fully-Connected Neural Networks
- Convolutional Neural Networks

## 2. 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 or 1 if the record is attack.**

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

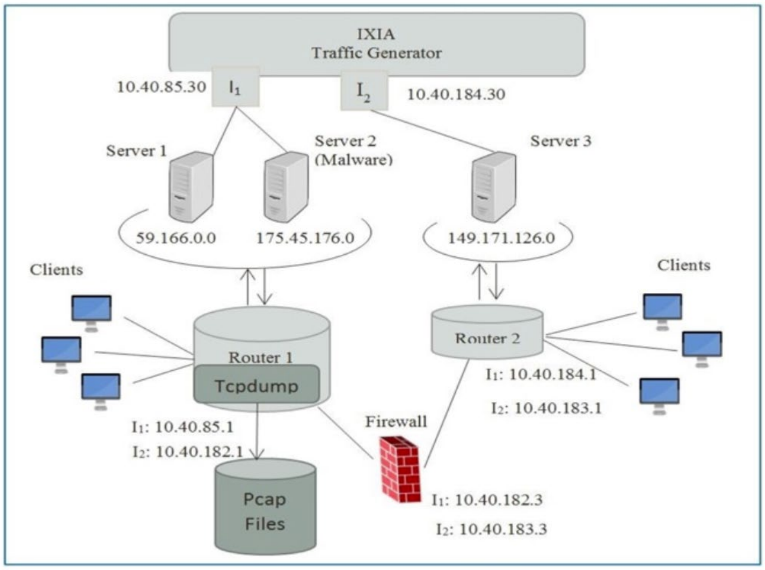


Figure 1: UNSW-NB15 Testbed

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

https://drive.google.com/drive/folders/1HN3UVEM7zKpW4a5YNJxPZUlYppQBJtob?usp=sharing

**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 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 pandas function *unique()* https://favtutor.com/blogs/pandas-unique-values-in-column, or python function *set()*, https://sparkbyexamples.com/pandas/create-set-from-series-in-pandas/

- 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 of each model on how they affect performance in your report. Also, save all the models you have tried as proof in your notebook.
  - **Activation:** relu, sigmoid, tanh
  - **Layers and neuron counts**
  - **Optimizer:** adam and sgd
  - **Kernel number and kernel size** (for CNN only)

## 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.

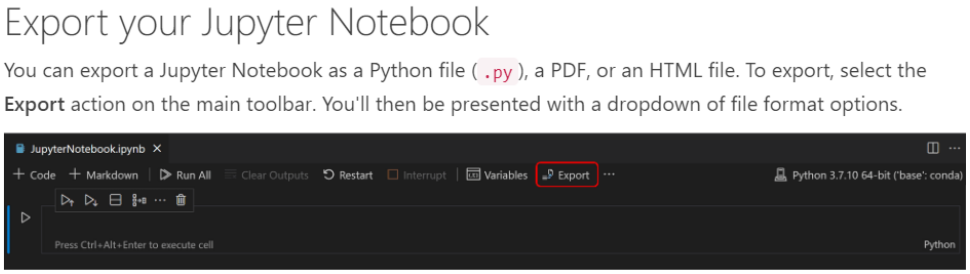
Use the evaluation form on Canvas as a checklist to make sure your work meets all the requirements.

## 5. Teaming

**Students may work independently or in a team of at most 3 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.



Export your Jupyter Notebook

You can export a Jupyter Notebook as a Python file (.py), a PDF, or an HTML file. To export, select the **Export** action on the main toolbar. You'll then be presented with a dropdown of file format options.

5 pts will be deducted for the incorrect file format.

**NOTE: Comment each notebook cell with author name and date.**

**(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.

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

**10:30 am, Monday, October 13, 2025**

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 specific intrusion? How good such a model can be in terms of detecting each specific attack? Which specific attacks have lowest F-1 scores, hardest to detect?

(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 input features, can you identify the most important features (this is called **feature importance analysis**) and train models only on those most important features, e.g., the top-10 or top-20 most important features? Compare the best model with all features and the best model with most important features only.

Hint: one option is to use logistic regression to identify the coefficient for each feature. **A higher coefficient indicates a higher importance.**

https://machinelearningmastery.com/calculate-feature-importance-with-python/

**(4)** Another related application for you to explore is **Internet of Things (IoT) Security.** Check the TON_IoT Datasets. The testbed was deployed using multiple virtual machines and hosts of windows, Linux and Kali operating systems to manage the interconnection between the three layers of IoT, Cloud and Edge/Fog systems

https://research.unsw.edu.au/projects/toniot-datasets

Go to the following link and you may choose one dataset/operating system to work with. Report your best model.

https://unsw-my.sharepoint.com/:f:/g/personal/z5025758_ad_unsw_edu_au/EvBTaetotpdGnW7rJQ8fCvYBh8063CNeY9W33MpRsarJaQ?e=yZlnxW