# CSC 5240 – Introduction to Artificial Intelligence Fall 2017
# Project Assignment: Computer Security

**DUE: Friday December 1st by 8:00 AM (notice the time!)**

**SUBMISSION: ALL materials through the iLearn class-website dropbox**

**PRESENTATION: Friday December 1st (class time)**

## Overview

As computers and the Internet become increasingly popular, malicious activities in the cyberspace have increased significantly. Intrusion detection is an area of computer security that focuses on detecting these attacks reliably. Intrusion detection systems (IDS) usually have a knowledge base containing rules that characterize attacks. Building such knowledge base manually can be time consuming. Machine learning can help build such knowledge base in a more efficient manner. In order to detect attacks, we need to differentiate between instances of normal and attack behavior. Based on previous instances of normal and attack behavior, a machine learning algorithm can gain the knowledge on how to differentiate between the two types of behavior and represent the knowledge in a form than can be used to predict if current instances are malicious or not.

## Objectives

This project aims to investigate machine learning techniques for detecting attacks/intrusions. More specifically, the objectives are:

- machine learning can be achieved from historical data (experience)
- machine learning algorithms can be applied to computer security
- understanding the learning task of trying to detect attacks
- understanding a decision-tree learning algorithm
- a better understanding of search and knowledge representation
- evaluation of machine learning algorithms

## Project Description

As computers and the Internet become increasingly popular, malicious activities in the cyberspace have increased significantly. Intrusion detection is an area of computer security that focuses on detecting these attacks reliably. Intrusion detection systems (IDS) usually have a knowledge base containing rules that characterize attacks. Building such knowledge base manually can be time consuming. Machine learning can help build such knowledge base in a more efficient manner.

In order to detect attacks, we need to differentiate between instances of normal and attack behavior. Based on previous instances of normal and attack behavior, a machine learning algorithm can gain the knowledge on how to differentiate between the two types of behavior and represent the knowledge in a form than can be used to predict if current instances are malicious or not.

For this project, implement the decision-tree learning algorithm (Russell and Norvig, Chapter 18, Figure 18.5) and evaluate the accuracy of the algorithm on the two provided training and test sets (described below).

1. Input to your program:
    - file name of the attribute description,
    - file name of the training set, and
    - file name of the test set.
2. Output from your program:
    - the tree using pre-order traversal with more indentation for nodes at deeper levels,
    - accuracy of the tree on the training set, and
    - accuracy of the tree on the (unseen) test set.

**Restaurant Data Set in the Book**
The Restaurant data set in Figure 18.3 on page 656 is the training set. No test set for this data set. Your implementation should reproduce the tree in Figure 18.6. Files for the data set:

- Attribute description: restaurant-attr.txt
- Training set: restaurant-train.txt (13 records)

**IDS Data Set**
The IDS data set contains records of network activities that are normal or part of a denial of service (DOS) attack(s) called Neptune (aka SYN-flood). Neptune tries to make many "half" connections to a server. Due to limited resources, a server usually has a maximum number of connections that it can handle. Many malicious "half" connections can prevent legitimate connections to be made. That is, the server might be filled with useless "half" connections, and cannot accept legitimate connections and provide the intended service (hence "denial of service"). The provided data set is adapted from the much larger KDD Cup Data set (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html). All values in the data set have been converted into discrete values.

Files for the data set:

- Attribute description: ids-attr.txt
- Training set: ids-train.txt (800 records)
- Test set: ids-test.txt (200 records)

**Submission**
For this assignment, you must submit the following:

1. (30 points) Source code of your program
2. (10 points) Executable of your program (runnable in either Windows or Unix)
3. (20 points) Output from running your program with BOTH provided data sets.

4. (30 points) Report (minimum 2 pages) that includes a discussion of your experiences creating decision-tree learning software, and in general, with the decision-tree learner in terms of the inputs, outputs, and performance.

This brings the possible number of points to 90.

In addition, you will have 10-15 minutes to present your project to the class for a total of 20 points, bringing the maximum total of points possible to 110.

Have fun!