Project Proposal

1 Topic/Title

Intrusion Detector Learning

2 Team Members

Daniel Silva
Jacob Tarnow
Simon Kwong

3 Problem

Adapted from: http://kdd.ics.uci.edu/databases/kddcup99/task.html
The main problem is to find a way to easily distinguish normal network connections from network intrusions/attacks. We will approach the learning task via supervised learning where we will build a classification model to help solve this problem. The data holds various parameters that have already been labeled as: TCP, HTTP, normal, bad, etc., which we will be using for training. Then we will test on this data. The target variable, Y, will be to predict whether a specific connection is good or bad. Y can take on different values that discern it from normal or bad, especially in respect to the types of attacks.

3.1 Definition

Software to detect network intrusions and protect a computer network from unauthorized users and insiders. This intrusion detector learning task is to build a predictive classification model capable of distinguishing between intrusions or attacks and normal connections.

3.2 Setting

In 1998 the DARPA Intrusion Detection Evaluation Program surveyed and evaluated intrusion detection simulated in a military network environment. Because the majority of people uses the internet, it is important to prevent the user's privacy and personal information from leaking out and to prevent unauthorized users from gaining access to this information. Within the last decade, many instances of cyber-attacks have been documented. One of the more notable attacks was an attack against a Ukraine power grid. This caused more than 200,000 residents and local businesses to experience a blackout, essentially costing banks and businesses to lose millions.

3.3 Type of Learning

We will approach network intrusion detection with supervised learning, more specifically, classification. In our data set, each tcp data packet falls into five categories: a normal connection, a denial-of-service (DOS), an unauthorized access from a remote machine (R2L), an unauthorized access to local superuser or root privileges (U2R), and probing or scanning. We can classify these categories into two main categories of either a “good” normal connection, or a “bad” intrusion connection. For every new tcp connection coming in, we want to predict and classify whether or not the network access is an intrusion or a normal connection. It is also important to note that there are specific attacks distributed in the test data set, but not in the training data. Because some intrusion experts believe that most novel attacks are variants of known attacks, having unknown attacks is more realistic.

4 Approaches

4.1 Models

As we have stated, we will be describing the predictors as either good or bad network connections. These predictors can be a binary representation, thus we believe that a logistic regression model would be the best choice. The response, Y, for this model will then lie between 0 and 1.

4.2 Features

Gathered from http://kdd.ics.uci.edu/databases/kddcup99/task.html, the features outlined by Stolfo, defined features that help in distinguishing normal connections from bad connection, i.e. attacks. They categorized the features into the follow: same host, same service, time-based traffic, host-based traffic, and content features. Same host features examine only connections in the past two seconds. These features have the same destination host as the current connections. Same service features examine connections that have the same service as the current connection in the past two seconds. Both of these together, same host and same service, are defined as time-based traffic. Host-based traffic on the other hand, involves sorting connection records by destination host. Thus, focusing on the same host instead of a specific time window. Finally, content features are added features that help in determining other predictors that may add to certain behaviors in the data.

5 Dataset

The dataset for this project has been supplied via KDD Cup 1999 Data. The raw training data is about 4GB of compressed binary. The TCP dump data contains seven weeks of network traffic. The data was processed into roughly five million connection records.

5.1 Data Source

The dataset has been supplied via KDD Cup 1999 Data.
Information and Computer Science, University of California, Irvine. Last modified on October 28, 1999.
Source: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

5.2 Data Size

The data files are separated via the full data set (18M records, 743 MB Uncompressed), a 10 percent subset (2.1M records, 75 MB Uncompressed), a 10 percent unlabeled subset (1.4M records, 45 MB uncompressed), an unlabeled test data (11.2M records, 430 MB Uncompressed), a 10 percent subset of the unlabeled test data (1.4M records, 45 MB Uncompressed), and test data with the corrected labels after a typo was found.

6 Softwares and Tools

Technologies: R, R Studio, Java, JavaScript

7 Timeline

Note: All of the dates are a rough estimate. Timeline will be updated based on workflow and how long each step takes.

Week 1 (Oct. 2 - Oct. 8) : Develop Project Proposal
Week 2 (Oct 9 - Oct 15) : Data Processing
Week 3 (Oct. 16 - Oct. 22) : Feature Selection
Week 4 (Oct. 23 - Oct. 29) : Feature Creation
Week 5 (Oct. 30 - Nov. 5) : Run Models and A/B Testing
Week 6 (Nov. 6 - Nov. 12) : QA on Models and update as necessary
Week 7 (Nov. 13 - Nov. 19) : Run tests on results to determine significance
Week 8 (Nov. 20 - Nov. 26) : Plot all comparisons and determine correct way to portray results
Week 9 (Nov. 27 - Dec. 3) : Clean up any errors and write up all challenges/results from project
Week 10 (Dec. 4 - Dec. 10) : Final Presentation