Assignment #3: Exploratory Data Analysis

INTRUSION DETECTOR LEARNING

Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections.

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset.

Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Attacks (for this assignment) fall into one category: DOS: denial-of-service, e.g. syn flood.

DERIVED FEATURES

Stolfo *et al.* defined higher-level features that help in distinguishing normal connections from attacks. There are several categories of derived features.

The “same host” features examine only the connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.

The similar “same service” features examine only the connections in the past two seconds that have the same service as the current connection.

"Same host" and "same service" features are together called time-based traffic features of the connection records.

Some probing attacks scan the hosts (or ports) using a much larger time interval than two seconds, for example once per minute. Therefore, connection records were also sorted by destination host, and

features were constructed using a window of 100 connections to the same host instead of a time window. This yields a set of so-called host-based traffic features.

Unlike most of the DOS and probing attacks, there appear to be no sequential patterns that are frequent in records of R2L and U2R attacks. This is because the DOS and probing attacks involve many connections to some host(s) in a very short period of time, but the R2L and U2R attacks are embedded in the data portions of packets, and normally involve only a single connection.

Useful algorithms for mining the unstructured data portions of packets automatically are an open research question. Stolfo *et al.* used domain knowledge to add features that look for suspicious behavior in the data portions, such as the number of failed login attempts. These features are called "content" features.

A complete listing of the set of features defined for the connection records is given in the two tables below.

duration	length (number of seconds) of the connection
protocol_type	type of the protocol, e.g. tcp, udp, etc.
service	network service on the destination, e.g., http, telnet, etc.
src_bytes	number of data bytes from source to destination
dst_bytes	number of data bytes from destination to source
flag	normal or error status of the connection
land	1 if connection is from/to the same host/port; 0 otherwise
wrong_fragment	number of ``wrong'' fragments
urgent	number of urgent packets

Table 1: Basic features of individual TCP connections.

hot	number of ``hot'' indicators
num_failed_logins	number of failed login attempts
logged_in	1 if successfully logged in; 0 otherwise
num_compromised	number of ``compromised'' conditions
root_shell	1 if root shell is obtained; 0 otherwise
su_attempted	1 if ``su root'' command attempted; 0 otherwise
num_root	number of ``root'' accesses
num_file_creations	number of file creation operations
num_shells	number of shell prompts
num_access_files	number of operations on access control files
num_outbound_cmds	number of outbound commands in an ftp session
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise
is_guest_login	1 if the login is a ``guest''login; 0 otherwise

Table 2: Content features within a connection suggested by domain knowledge.

count number of connections to the same host as the current connection in the past two seconds

Note: The following features refer to these same-host connections.

serror_rate % of connections that have ``SYN'' errors

rerror_rate % of connections that have ``REJ'' errors

same_srv_rate % of connections to the same service

diff_srv_rate % of connections to different services

srv_count number of connections to the same service as the current connection in the past two seconds

Note: The following features refer to these same-service connections.

srv_serror_rate % of connections that have ``SYN'' errors

srv_rerror_rate % of connections that have ``REJ'' errors

srv_diff_host_rate % of connections to different hosts

Table 3: Traffic features computed using a two-second time window.

Data Exploration

Create a data quality report with the format that we discussed in the lecture.

- For the numeric attributes, provide the following:
    - A table with one row per attribute and columns including Count, % missing, Cardinality, Minimum, 1st Quartile, Mean, Median, 3rd Quartile, Maximum, Standard Deviation
    - A histogram for each attribute
    - A Box and Whisker plot for each attribute that has a non-zero IQR to identify potential outliers
    - A matrix with the correlation coefficients
- For the categorical attributes, provide the following in a table:
    - A table with one row per attribute and columns including Count, % missing, Cardinality, Mode, and the Mode frequency
    - A bar chart for each attribute.

In your analysis, answer the following questions:

- Are any of the attributes useless for analysis? How can you tell?
- Why did we only consider some of the numeric attributes to identify potential outliers using the rule of thumb we discussed in class. Is there a potential problem with this approach for some attributes in this dataset?
- Which attributes have suspected outliers or outliers, based on the Box and Whisker plots?
- Looking at the bar charts for the categorical variables, do some values for attributes appear to be very unusual?
- Identify attributes with a large number of zeros, based on the interquartile range from the table of numeric attributes. Looking at the histograms, what do you notice about these attributes?
- We will be seeking to distinguish normal and attack behavior. Do you think that we should eliminate rows with unusual values?
- Why are some of the correlation coefficients set to NaN?
- Which pairs of attributes have a correlation coefficient greater than or equal to 0.98? Why might these attributes be correlated?
- Which pairs of attributes have a correlation coefficient less than -0.80? Why might these attributes be negatively correlated?
- Are there any other interesting observations you can make about the data?

In addition to the pdf report, please submit a file with your python source code.