

This document is adapted from the paper *Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project* by Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodrromidis, and Philip K. Chan.

## 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 fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow" attacks;
- probing: surveillance and other probing, e.g., port scanning.

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the "signature" of known attacks can be sufficient to catch novel variants. The datasets contain a total of 24 [training attack types](#), with an additional 14 types in the test data only.

## 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 three tables below. The data schema of the contest dataset is available in [machine-readable form](#).

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

Table 1: Basic features of individual TCP connections.

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

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

| <i>feature name</i> | <i>description</i>  | <i>type</i> |
|---------------------|---|-------------|
| count               | number of connections to the same host as the current connection in the past two seconds    | continuous  |
|                     | <i>Note: The following features refer to these same-host connections.</i>                   |             |
| error_rate          | % of connections that have ``SYN" errors  | continuous  |
| rerror_rate         | % of connections that have ``REJ" errors  | continuous  |
| same_srv_rate       | % of connections to the same service  | continuous  |
| diff_srv_rate       | % of connections to different services  | continuous  |
| srv_count           | number of connections to the same service as the current connection in the past two seconds | continuous  |
|                     | <i>Note: The following features refer to these same-service connections.</i>                |             |
| srv_error_rate      | % of connections that have ``SYN" errors  | continuous  |
| srv_rerror_rate     | % of connections that have ``REJ" errors  | continuous  |
| srv_diff_host_rate  | % of connections to different hosts   | continuous  |

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