Hands-On Session

PROBLEM STATEMENT

HolesInMyFirewall Inc has hired you to help detect and protect them from unauthorized users accessing their computer network through network intrusion attacks. They have been attacked numerous times in the past, and recognise that they need to find ways of monitoring and stopping such attacks.

You explained to them that 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. Data about this sequence of packet transfers is logged within TCP logs, and under your guidance they have painstakingly collected their TCP logs and labelled the data with different attack types but realised that this was too slow and laborious for them to do on a regular basis. Also, such labelling was prone to error and did not give them a path forward to protecting their network

Learning the “signatures” of such attacks from a TCP dump is an interesting machine learning challenge. While each connection has been labelled as either ‘normal’ or a specific attack type, you recognise that individual attack types feature in small numbers in the data, and so you start by building a classifier capable of distinguishing between “bad” connections (intrusions or attacks), and “good” (normal) connections.

You then are tasked with extending your work with learning specific attack signatures.

THE DATA SET

The data used in this session has 494020 rows with 42 columns. The table below provides the data dictionary.

Column name	Description	Type
Duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	symbolic
service	network service on the destination, e.g., http, telnet, etc.	symbolic
flag	normal or error status of the connection	symbolic
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
land	1 if connection is from/to the same host/port; 0 otherwise	symbolic
wrong_fragment	number of “wrong” fragments	continuous

urgent	number of urgent packets	continuous
hot	number of “hot” indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	symbolic
num_compromised	number of “compromised” conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	continuous
su_attempted	1 if “su root” command attempted; 0 otherwise	continuous
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_host_login	1 if the login belongs to the “host” list; 0 otherwise	symbolic
is_guest_login	1 if the login is a “guest” login; 0 otherwise	symbolic
count	number of connections to the same host as the current connection in the past two seconds	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
serror_rate	% of connections that have “SYN” errors	continuous
srv_serror_rate	% of connections that have “SYN” errors	continuous
rerror_rate	% of connections that have “REJ” errors	continuous
srv_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_diff_host_rate	% of connections to different hosts	continuous
dst_host_count	among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose source IP address is also the same to that of the current connection	continuous

dst_host_srv_count	among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose service type is also the same to that of the current connection	continuous
dst_host_same_srv_rate	%age of past 100 connections whose destination IP address is same as the current connection, the number of connection whose service type is same as that of current connection	continuous
dst_host_diff_srv_rate	%age of past 100 connections whose destination IP address is different to that of the current connection, the number of connection whose service type is same as that of current connection	continuous
dst_host_same_srv_port_rate	% of connections whose source port is the same to that of the current connection in Dst host count feature	continuous
dst_host_srv_diff_host_rate	% of connections whose source port is different from the current connection in Dst host count feature	continuous
dst_host_serror_rate	% of connections that have “SYN” errors in Dst host count feature	continuous
dst_host_srv_serror_rate	% of connections that “SYN” errors in Dst host srv count feature	continuous
dst_host_rerror_rate	% of connections that have “REJ” errors in Dst host count feature	continuous
dst_host_srv_rerror_rate	% of connections that “SYN” errors in Dst host srv count feature	continuous
label	‘normal’ vs. attack types	symbolic

APPROACH

In the hands-on session, you will learn how to:

- Explore the dataset and understand the interaction between features and labels
- Build a binary classifier, and distinguish a normal connection from an intrusion
    - You will use the following machine learning algorithms:
        - Naive Bayes
        - Logistic Regression
        - Support Vector Machines
        - Decision trees
        - Random Forests

    - Gradient boosted decision trees
- Compare different algorithms using K-fold cross validation and metrics like accuracy, precision and recall
- Perform hyper-parameter tuning of the algorithms using grid-search cross validation
- Engineer features i.e. combining features in interesting ways to improve the accuracy
- Build multi-class classifiers
- Balance class labels and improve their performance

The session is concluded by bringing all the parts together and building a machine learning pipeline which can be explored further and re-used in other classification problems.