Automated Classifier in Response to Firewall Traffic

Jason McDonald

Problem Statement:

Growth in internet traffic has brought about the need for increased cybersecurity to protect networks and sensitive data against attacks from nefarious actors.

This case study approaches cybersecurity for a large client by creating the best, most advanced machine learning model which can take an action on a firewall traffic packet in real time. The client has asked that accuracy be reported as well as the level of performance, which this study considers as the time in nanoseconds that an action can be taken.

Provided Data and Evaluation:

The data provided by the client company is a csv file of internet traffic with a response of 'Action' that consists of 4 classes of actions that were taken on the traffic at the time of the request.

The features available are identified in table 1 below.

Table 1: Features in the Firewall Dataset

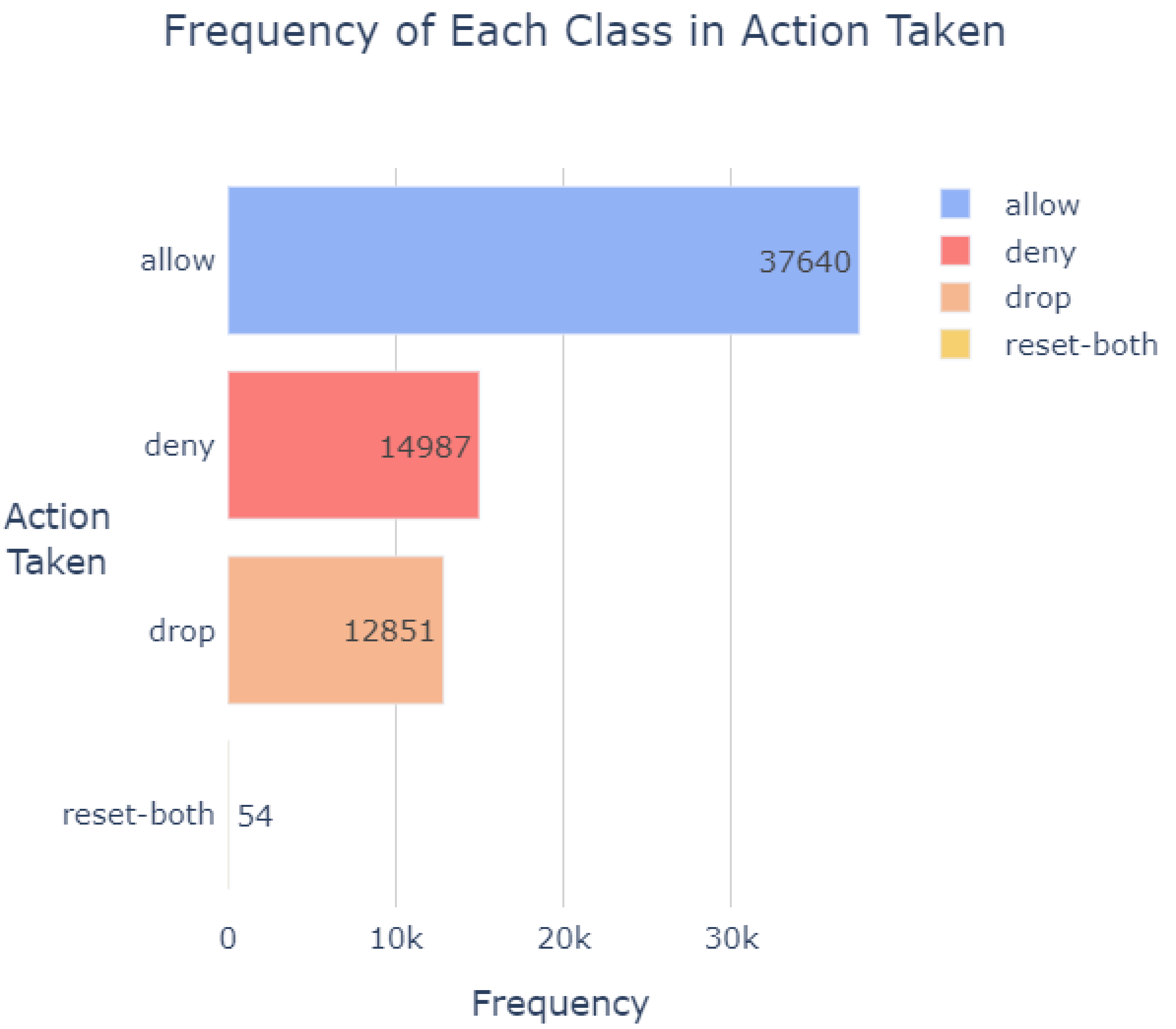
Total Features:	11
Total Rows:	65,532

Feature
Source Port
Destination Port
NAT Source Port
NAT Destination Port
Bytes
Bytes Sent
Bytes Received
Packets
Elapsed Time (sec)
pkts_sent (Packets Sent)
pkts_received (Packets Received)

Table 1: Features available in the firewall dataset

Missing Data: The data provided does not contain any missing values.

Response Variable: Analysis of the response variable, 'Action', indicates that there exists 4 classes of actions which may be taken in response to a request being sent to the firewall. The chart below shows the actions along with the frequency with which each occurs.



The dataset does have an imbalance among classes though for deny and drop, it is manageable without any additional amount of work. The study will explore under or over sampling of the classes if performance of the models dictate such.

For 'reset-both', the amount of examples of that class make it an extraordinarily difficult class to predict and the study expects that class to grossly underperform.

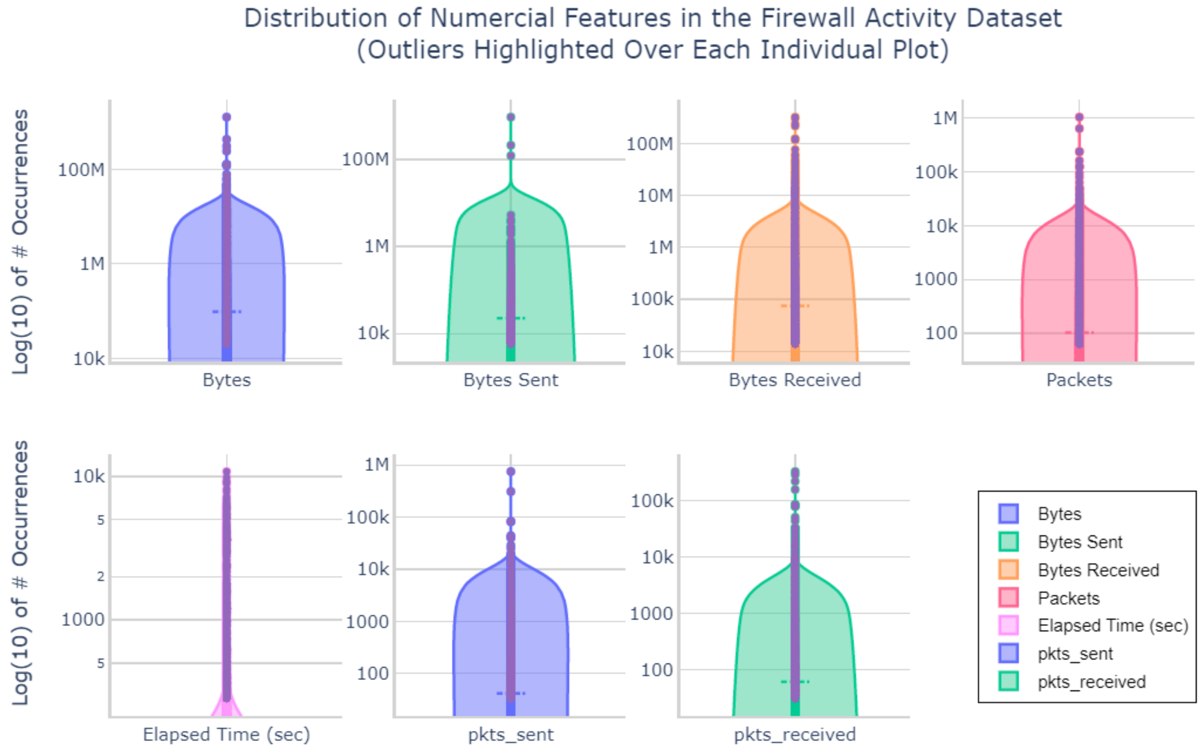
Given that fact and that resetting both connections effectively has the same result as dropping the connection, the study will simply eliminate the 'reset-both' action and replace it with the 'drop' action.

Action Required: The client will need to approve of this change prior to the model being released.

Features Provided for Classifying Actions on Firewall Traffic:

Numeric Features: Of the 11 features provided for prediction, 7 are numerical. These features include time in seconds, and for both bytes and packets, the number of each, plus the number sent and received of each.

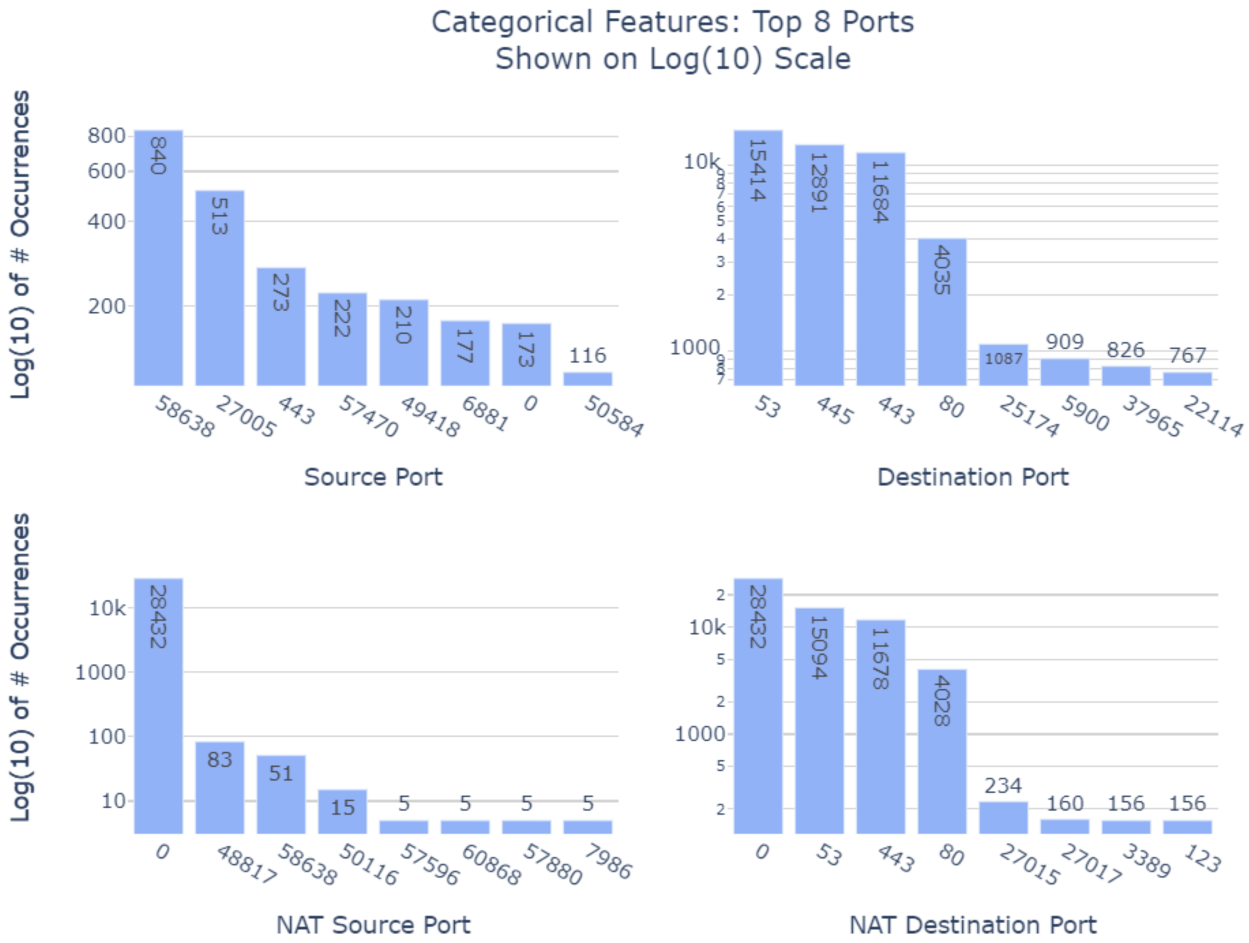
The distribution of each, with occurrences on a log scale to accent the right skewness of the distribution, with outliers above 1.5 times the IQR of 90% highlighted, are shown in the figure below.



Note that the Y axis has been transformed across a log(10) scale

Categorical Features: The remaining four features are categorical. These are the Source and Destination Ports both for the original request and for the NAT service.

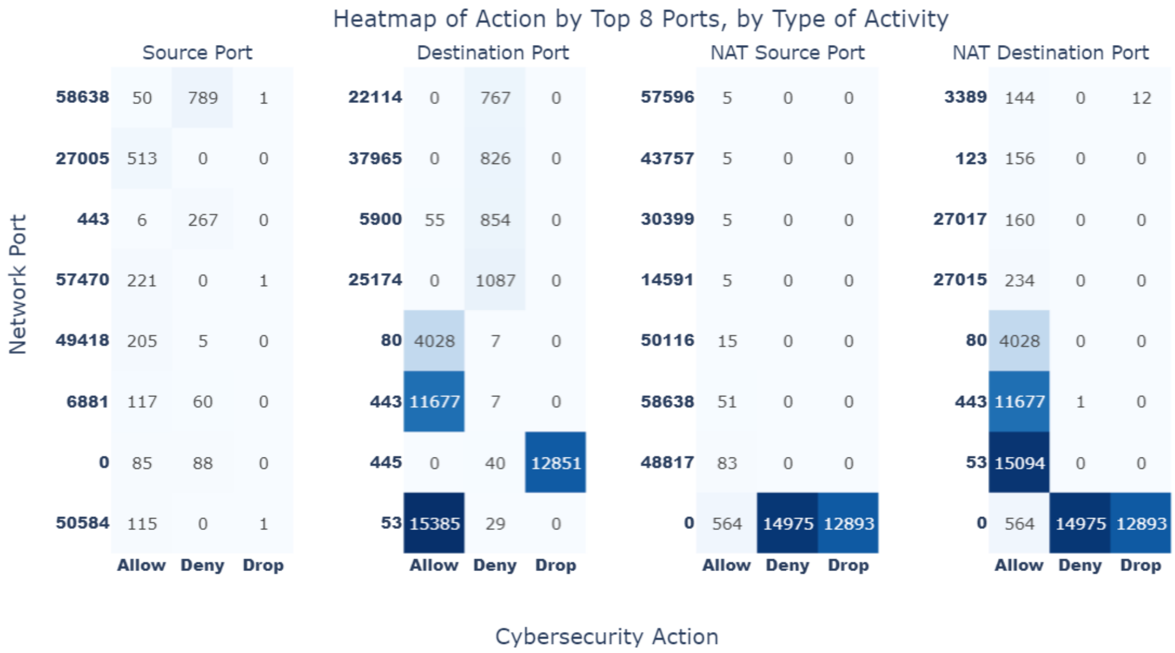
The top 8 ports for each categorical feature is displayed in the charts below. These are shown on a log(10) scale, with actual counts displayed on the chart, so as to allow for the smaller values to be shown. *It is important to understand that this causes the size of each bar to not be directly comparable.*



Note that the Y axis has been transformed across a log(10) scale

It is also valuable to examine the frequency with which each class occurs in a categorical feature. In this dataset, with such a high number of ports being used, this study will look only at the top 8 ports for each type of categorical feature.

The below chart displays in a heatmap format the count of action taken by port, for those top 8 in each feature category.



The study makes note that when the NAT port is 0 or not identified, there is a high likelihood that the action will be to deny or drop that packet. There are also common ports such as 80, 443, and 53 that appear to typically be allowed. Those ports align with typical HTTP traffic, SSL traffic, and Domain Name Service traffic.

Building Models to Classify for Action on Firewall Traffic:

In preparation for building models to classify the network traffic with an action to take, the data must be standardized. This will be done using the StandardScaler function in the ScitKit Learn library, by creating a pipeline for each model type that includes the StandardScaler along with the Classifier with its parameters.

The table below demonstrates the performance of each, along with the amount of time in nanoseconds for a single prediction to be made using that model.

ML Classifier Accuracy and Time in Nanoseconds Per Prediction

Model Type	Accuracy	Time (ns)
SVC - Linear Kernel	98.83%	944,200
SVC - Poly Kernel	98.92%	1,313,500
SVC - RBF Kernel	98.49%	466,500
SGD - Hinge Loss	98.54%	245,700
SGD - Log Loss	99.20%	194,400
SGD - Squared Hinge	85.92%	198,900
SGD - Modified Huber	97.64%	207,300

Table: ML Classifier Accuracy and Time in Nanoseconds Per Prediction contains a list of classifiers that the study used to determine an action to take on any given network request

Each model was fit with the class_weight parameter set to 'balanced' to compensate for the slight class imbalance between the 3 remaining classes. Each model then had its hyperparameters tuned by adjusting the penalty terms, the learning rate, and the regularization rate among other parameters.

It is important to note that the time measured and displayed above *does not* take into account the amount of time to load the model from disk as the model predict function was timed immediately after fitting. This time was excluded from the study as it is presumed to be similar in time among all models.

Conclusion:

This study analyzed the Support Vector Classifier and Stochastic Gradient Descent machine learning model types to develop a potential solution for predicting whether to Allow, Drop, or Deny a network packet traversing an IP network firewall.

The model with both the highest performance and fastest prediction time was the Stochastic Gradient Descent Learning Model with the Log Loss function. This model achieved a 99.2% accuracy on the held back test data set and was able to make a prediction in ***under .2 milliseconds***.

This model will be the best model to deploy into a production environment by being the most accurate and fastest model the study found.