**Slide 1**

Hello, my name is Marvin Newlin and today I am going to be talking to you about Network Traffic Classification with Machine Learning.

**Slide 2**

In Overview, I am going to be discussing some background on the problem and present the research questions for this project. We will conduct some exploratory data analysis, discuss the experimental setup, discuss results, and present conclusions and future work.

**Slide 3**

For some background, Intrusion Detection Systems or IDSs classify network traffic. IDSs are responsible for analyzing large amounts of data, so naturally we are inclined to ask, “can we use Machine Learning to do this classification??”

The dataset we will be using in this project is the CICIDS2017 dataset, a network trace dataset developed in a test bed environment. This is a fairly large dataset so we will only be utilizing a subset of this dataset.

This subset contains only one type of malicious traffic and the dataset is labelled. This means that we will be performing supervised binary classification.

**Slide 4**

For this project, we have two Research Questions. First, can we successfully classify network traffic as benign or malicious using the 8 features from the table on the right?

Second, What features are most important for classifying traffic as benign or malicious?

Inspecting the table on the right, we see the 8 features we have chosen for this project. They are all continuous valued features and their inclusive ranges are shown on the right side of the column.

**Slide 5**

The dataset we are utilizing for the project has 170,000 observations. The approximate class distributions are 99% benign traffic and 1% malicious traffic.

Inspecting the pairwise plots of the features doesn’t tell us a whole lot but by selecting 5% of the data to scatterplot we can make a little bit more sense of the how the data is distributed.

In the plots on the right hand side, we see pairs of features plotted with the benign observations as blue plus marks and the malicious traffic as red triangle markers.

The main takeaway from these plots is that there does not appear to be any clear separation between the classes and there is heavy overlap between the benign and malicious traffic. This indicates to us that higher flexibility models may perform better on this dataset.

**Slide 6**

To handle the imbalanced data distributions, we will be utilizing the balanced accuracy score from the scikit learn package as our main performance indicator. Many of the models from scikit learn such as Logistic Regression include a parameter for class weights called ‘balanced’ which weights each class according to the inverse of the class frequency. This means that the minority class will be weighted higher in the model than the majority class.

With the dataset, we have partitioned the data into a non-test and test set with the non-test set containing 2/3 of the data and the test set containing 1/3 of the data. We chose this particular split due to the large amount of data available. Due to the wide range of feature values displayed in the earlier table, we scale the data into the [0,1] interval using the scikit learn standard scaler. Since we have a large amount of benign traffic and we are interested in detecting malicious traffic we use the label conventions of the benign class being labelled as 0 and malicious labelled as 1.

**Slide 7**

Next is our experimental setup. For model fitting, we will be fitting 3 models: Logistic Regression (LR), K-Nearest Neighbors (KNN), and Balanced Random Forest, a Random Forest classifier designed for imbalanced classes.

We tune the hyperparameters for each model utilizing the standard 10-fold cross validation on the entire non-test set. With respect to hyperparameter tuning, for logistic regression we will be tuning the cost parameter which is the inverse of regularization strength. For KNN we will be tuning the k. For Balanced Random Forest we will be examining the number of trees, number of features, and the depth.

For model evaluation, once we have tuned all of the hyperparameters, we will compare the performance of each model based on the balanced accuracy score and select the highest performing model.

For test set evaluation we will utilize the balanced accuracy score, precision, recall, and F1 score.

Additionally, we will perform feature selection on the best performing model.

**Slide 8**

For logistic regression, we utilize an L1 penalty, the saga solver, max iterations of 1000, and explore Cost values between [0.0001 and 0.0025]. These specific values were chosen because they were the only values of C for which the model would converge.

In the leftmost plot, we see each of the 10 cost values plotted against the balanced accuracy of the model. As we can see from the black ‘x’, the highest balanced accuracy occurs at cost=0.0006333.

For KNN we examine k values from 1 to 40 and plot the results of the balanced accuracy score for each k. As we can see, the highest balanced accuracy occurs with k=36.

**Slide 9**

For Balanced Random Forest we tune 3 hyperparameters. First we tune the number of estimators and the tune max\_features and max\_depth with that number of trees. The amount of time required to tune all 3 together was too great so we tune features and depth separately.

As we can see from the rightmost plot, the highest balanced accuracy occurs at num\_trees=600.

The leftmost plot is a contour plot of max\_features vs max\_depth vs balanced accuracy score. The orange ‘x’ indicated the values of max\_features and max\_depth that maximize balanced accuracy.

The final balanced random forest model uses 600 trees, 2 features, and a depth of 19. Additionally, the imbalanced learn aspect allows us to specify a ratio of majority to minority so we specify it as 1.0 and set the replacement parameter to true to sample from the minority class with replacement.

**Slide 10**

With each of our models tuned, we compare their performances with the balanced accuracy score. The plot on the right illustrates the balanced accuracy performance for each of the 10 cross validation folds. As we can see, the green line representing Balanced Random Forest has the highest balanced accuracy score so we select it as the best model.

**Slide 11**

With the best model selected, we plot the precision recall (PR) curve for the model. The PR curve is similar to the ROC curve, but utilizes the precision metric instead of false positive rate. This curve has been shown to be more informative for imbalanced datasets than the ROC curve. Like the ROC curve though, the goal is to maximize the area under the curve and get it as close to 1 as possible. For our model we have an AUC of 0.785 and we see that the model does suffer some on the precision side.

The final decision we need to make before predicting on the test set is the probability threshold p for which we will base our class decisions on. To make this decision, we examine the point on the PR curve that balances precision and recall, which is the point where the F1 score is maximized. This point is indicated by the red ‘x’ on the plot on the right and corresponds to a probability threshold of p=0.89

**Slide 12**

With our probability threshold chosen, we fit the model on the entire nontest set and predict class probabilities and classes based on our chosen threshold. The chart on the left shows the confusion matrix for the test set results.

Inspecting the metrics, we see that we have a 95.1% balanced accuracy, 90.7% recall, 75.1% precision and an F1 score of 0.82. As we can see from the confusion matrix, we are fairly accurate with our predictions with a higher false negative rate than false positive rate. However, the overall performance is very good for our purposes.

**Slide 13**

To examine the features that are most important to the model we use the feature importances attribute and we see that the top 4 features for predicting the class of the traffic are fwd\_packet\_length\_mean, flow\_bytes/s, flow duration, and fwd packets/s.

**Slide 14**

In conclusion, to answer Research Question 1, we are able to accurately classify at 95% malicious network traffic with the Balanced Random Forest model.

To answer Research Question 2, we saw that the 4 most important features in the classification are fwd\_packet\_length\_mean, flow\_bytes/s, flow duration, and fwd packets/s.

In future work, we would perform multi class classification and include more features since the dataset contains 80 features. Additionally, another avenue would be to use the full dataset which contains about 10 times the amount of data that our subset has.

**Slide 15**

These are my references

**Slide 16**

Thank you for watching!