# Updated Model Performance Report

**Preprocessing Pipeline:**

1. Non-ASCII characters are removed
2. URL's are removed
3. Duplicate tweets are removed
4. Empty tweets are removed

**Classification Pipeline:**

1. Load Json object containing tweet text and relevancy label

2. Convert Json object into two numpy arrays, one containing tweet text and one containing label

3. Run arrays through preprocessing pipeline

4. Run text array's through BERT to get array of tweet encodings

5. Run encodings through a final classification layer to get final relevant/irrelevant classifications

**Train Set Description:**

This is the original dataset we used in our previous report. As a refresher, the breakdown of tweets by relevancy category is as follows (for this report, tweets classified as 'maybe relevant' were also considered relevant):

Total Tweets: 3982

Relevant Tweets: 1174

Irrelevant Tweets: 2808

**Test Set Description:**

Tweets taken from completely different sources. This time, several tweets that contained keywords like "vulnerability" or "exploit" but in non-cybersecurity related terms were purposefully included in order to gauge how well the model learned the correct context. The breakdown by relevancy category is as follows:

Total Tweets:756

Relevant Tweets:231

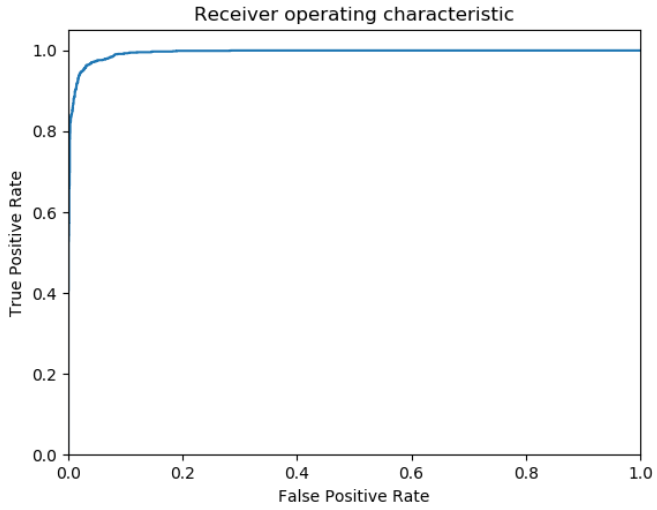
Irrelevant Tweets:525

We now report the performance of three different final classification layers. We tested logistic regression, support vector machine, and single layer neural network

**Logistic Regression:**

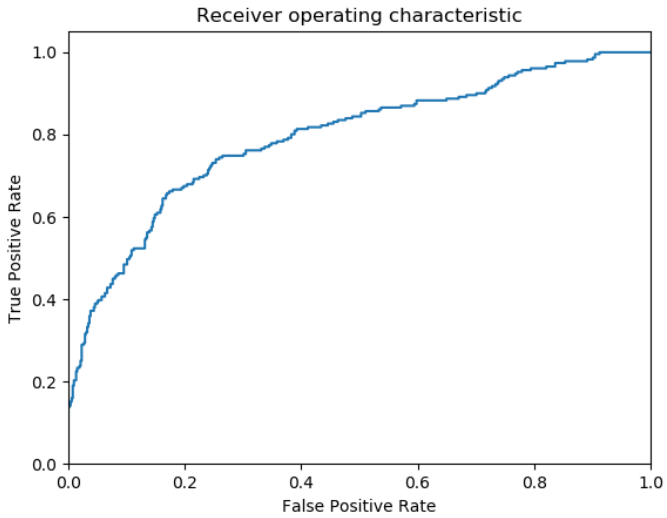
Training Set:

- Accuracy: 96.84%
- AUC: 0.9946
- ROC Curve:




Testing Set:

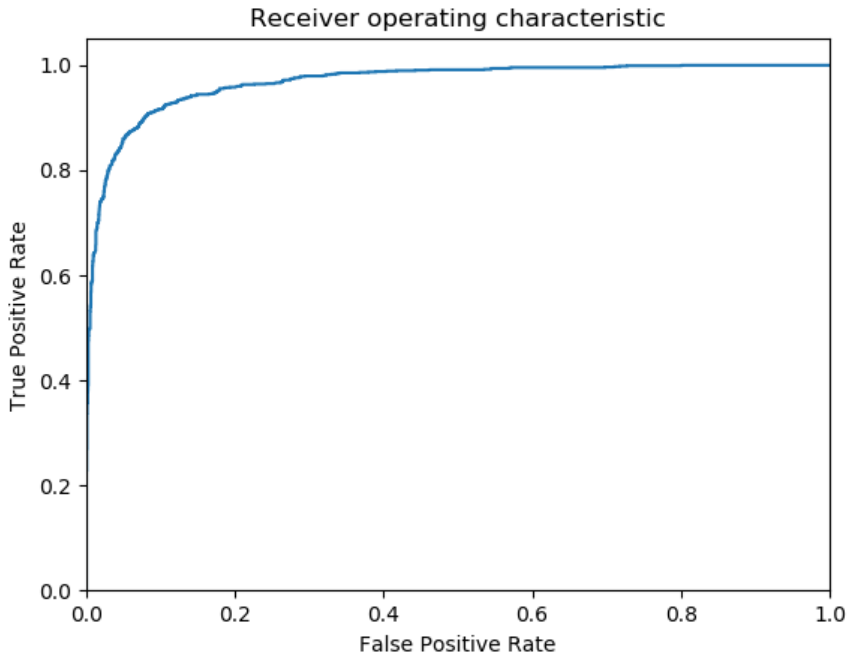
- Accuracy: 76.98%
- AUC: 0.7935
- ROC Curve:



**Support Vector Machine:**

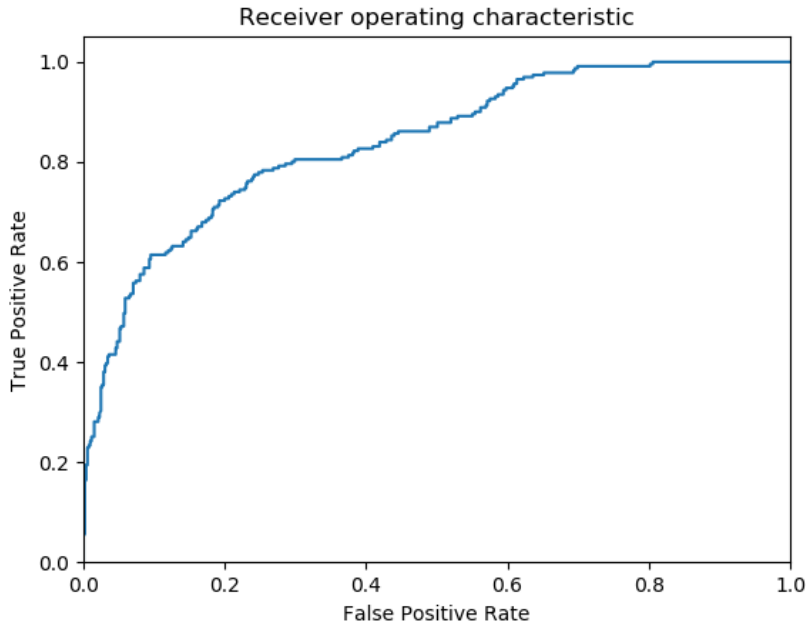
Training Set:

- Accuracy: 92.13%
- AUC: 0.9681
- ROC Curve:

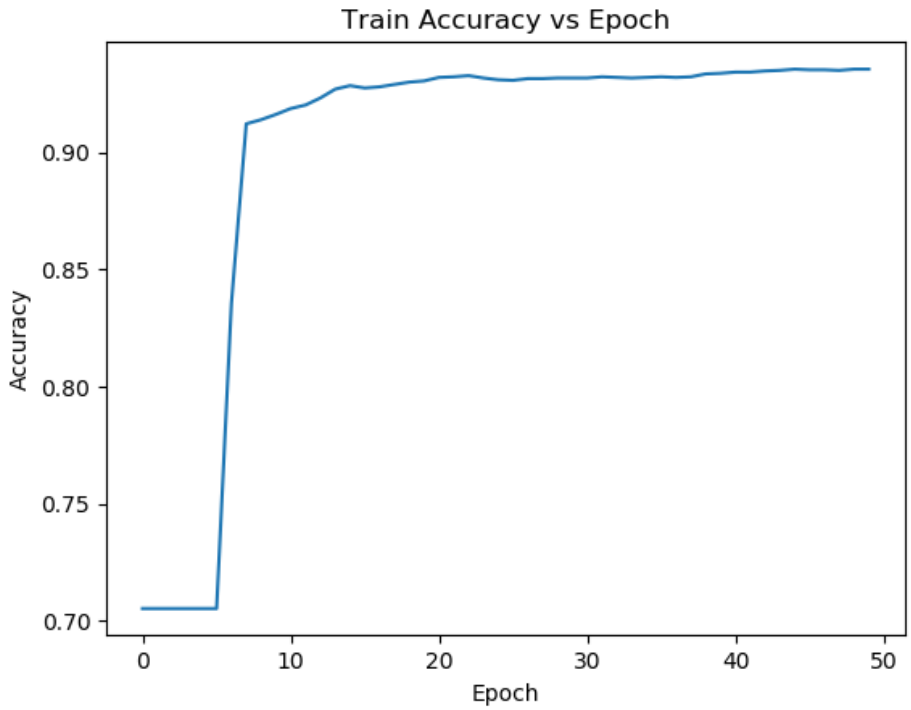


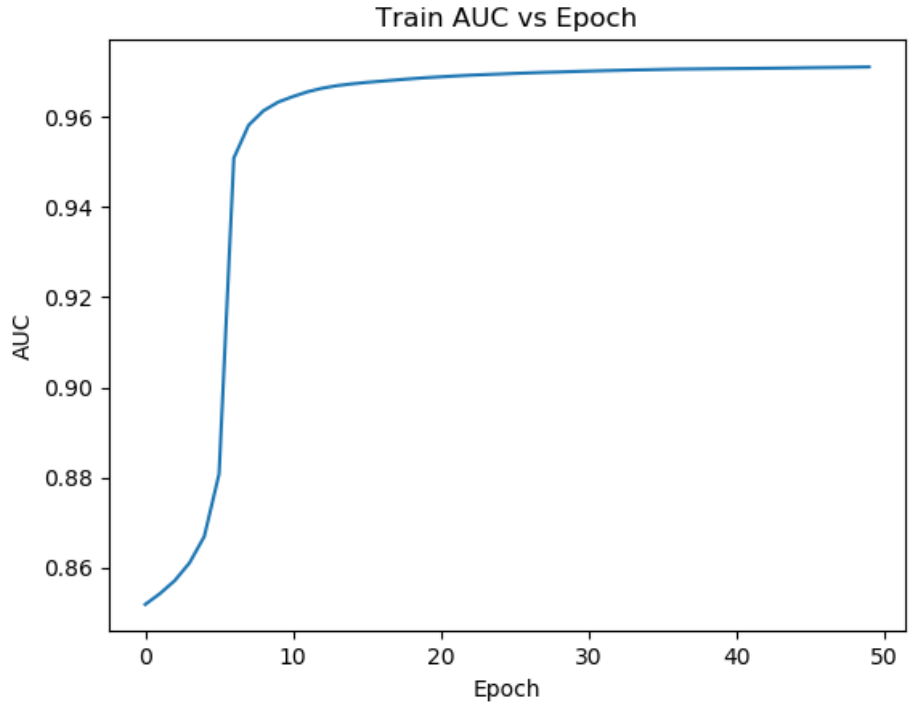

Testing Set:

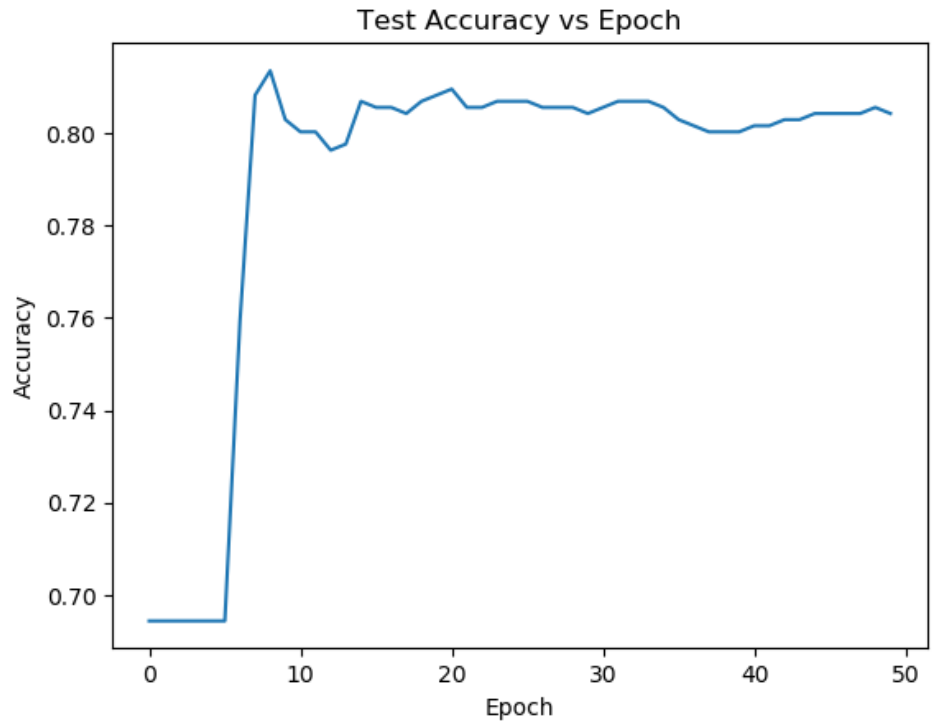
- Accuracy: %
- AUC: 0.8391
- ROC Curve:

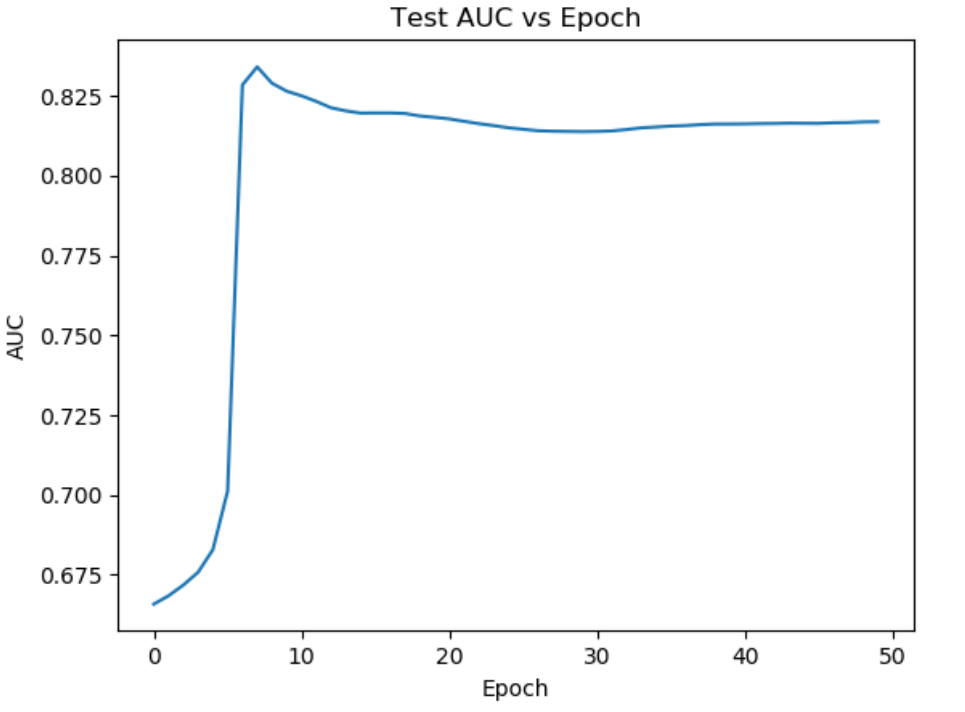


**Single Neural Layer:**



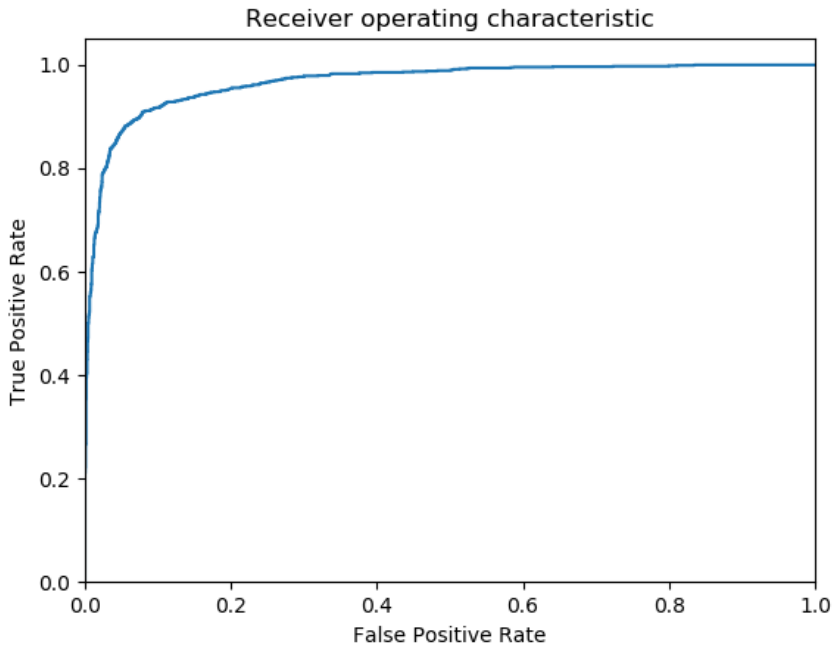



We see that test accuracy and AUC are highest around epoch 9, after which test performance slightly drops and training performance increases, signaling that the model began to overfit after epoch 9. Accordingly, we reported metrics for the model after 9 epochs of training.

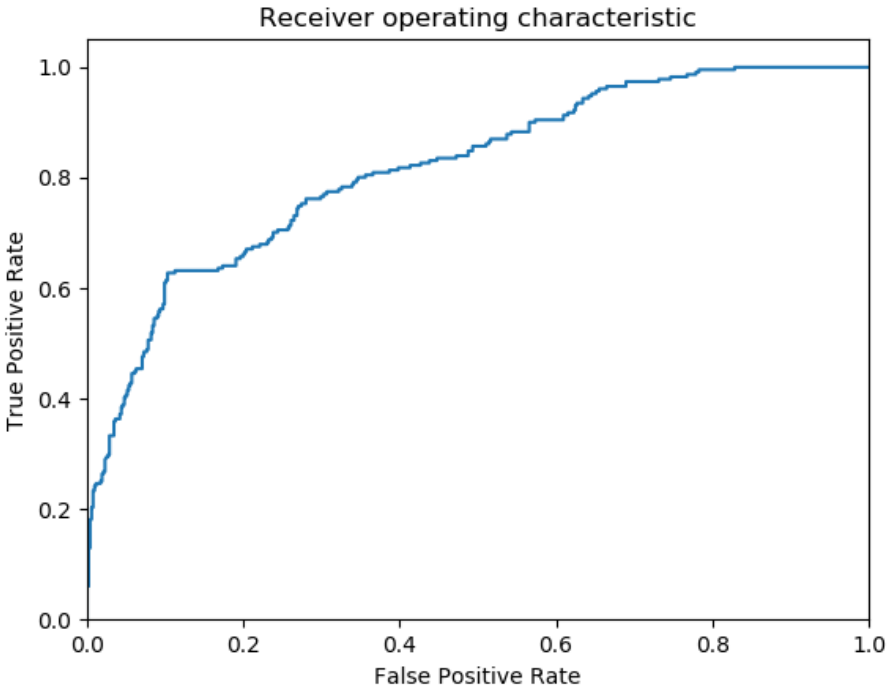
Training Set:

- Accuracy: 91.41%%
- AUC: 0.9613
- ROC Curve:




Testing Set:

- Accuracy: 81.35%
- AUC: 0.8290
- ROC Curve:



Analysis of Performance on words like “Vulnerability” and “Exploit”

Tweets Containing “vulnerability”:

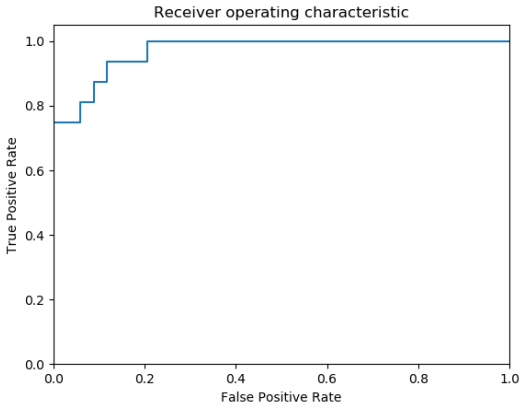
Relevant: 16

Irrelevant: 34

Accuracy: 88%

AUC: 0.9706

ROC:




Tweets Containing “exploit”:

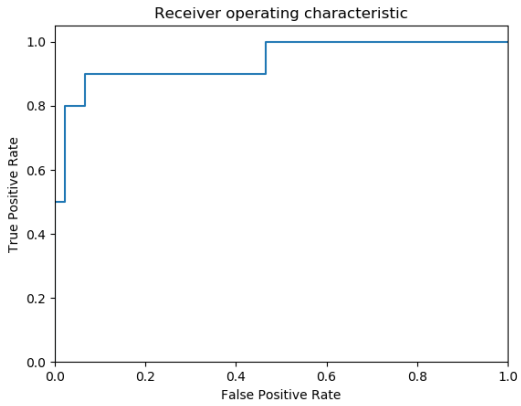
Relevant: 10

Irrelevant: 45

Accuracy: 94.55%

AUC: 0.94

ROC:




While ROC and AUC values were provided for these “mini test sets” for the sake of completion they are not as informative. Inspecting the actual tweets and their predictions is a better way to gauge how well the model performs at filtering out tweets that use words like “vulnerability and exploit” in a non-cybersecurity related context. Examining these predictions shows that the model is quite adept at picking out when such keywords are used in cybersecurity contexts. The files containing these prediction results are named “vulnerability_predictions.txt” and “exploit_predictions.txt” accordingly.