Android Malware Detection Project
Intermediate Report

Ritesh Sharma
U110289

Algorithms Used:
The project is an attempt to accurately classify the Android Malware data set. The project considers *ID3*, *Normal Perceptron* and *Averaged Perceptron* for the Intermediate phase reporting.

Accuracies:
The first classifier used on the data set was ID3 algorithm. This algorithm performed very poorly with the original F-score of only 39%, this was mainly because no values were discretized and no depth were set. To combat this, I discretized the values of each features into 4 buckets and set depth to 8, which brought me to F-score of 75%. The depth of 8 was chosen after performing a cross validation with different depth values and calculating the F-score to choose best depth.

The second classifier used was normal Perceptron Algorithm, this algorithm also performed poorly, with an original accuracy score of 40%. This was mainly because, inability of the algorithm to optimize over vectors containing continuous valued features. This was taken care of by implementing an array processing algorithm which maps continuous valued feature to binary feature. This mapping is described below where i is the nth element of the feature array:

f(i) : R -> [0,1] = 1 if i > 0 ; 0 otherwise.

The Normal Perceptron was then converted to Averaged Perceptron algorithm by adding hyperparameters. Using threshold mapping technique described above, the algorithm produced initial F-score of 77%. Then the attempt was made to maximize the accuracy of the hyperparameter r, mu, and the epoch count e. The learning rate(r) was tested with 20 values ranging from 0.0001 to 2.0, the margin parameter mu was tested with 20 values ranging from 0 to 12. Below are the table for these tests:

Avg Perceptron Cross Validation Chart
Learning Rate(r) vs mu

r/mu	**0.01**	**0.05**	**0.1**	**0.5**	**0.7**	**1.0**	**2.0**
0	0.734	0.712	0.753	0.763	0.745	0.753	0.723
0.1	0.717	0.722	0.712	0.746	0.734	0.732	0.734
1	0.720	0.727	0.727	0.732	0.754	0.712	0.728
1.5	0.719	0.723	0.728	0.731	0.726	0.734	0.754

2	0.738	0.705	0.716	0.742	0.728	0.732	0.728
2.5	0.747	0.703	0.725	0.754	0.717	0.765	0.732
3	0.755	0.705	0.767	0.737	0.752	0.745	0.712
4	0.742	0.701	0.713	0.735	0.727	0.712	0.743
5	0.713	0.721	0.717	0.736	0.726	0.734	0.745
7	0.742	**0.781**	0.727	0.728	0.738	0.743	0.745
9	0.752	0.718	0.747	0.747	0.710	0.723	0.728
12	0.771	0.758	0.728	0.737	0.716	0.756	0.765

After performing the cross-validation, the table above shows that with the learning rate(r) of 0.05 and the margin parameter(mu) of 7 produced the max F-score 0f 78%.

The ID3 algorithm for my case performed well with the hyperparameter set to the best using cross validation, but even after performing cross-validation and choosing the best hyper-parameter, the max F-score the algorithm was able to attain was 75%.
Performing Average Perceptron yielded the maximum F score, this was also yielded after the best hyper-parameter was chooses for the Average Perceptron.

Plan:
Since it is challenging to optimize data set using linear techniques. Now the project will shift its focus to the use boosting and SVM algorithms in an attempt to combat this challenge.
For the rest of the semester I plan on implementing Boosting, SVM and other neural network techniques that will be covered in class.
The final F score will be based on the most optimized version of the algorithms taught in class or are yet to be taught.

Submissions:
For my 7 submission on Kaggle:
AvgPercep.csv (0.78142) was for the Average Perceptron with mu = 7 and r = 0.05.
evaOut.csv (0.38034) was for ID3 without any hyper parameter and max depth.
evaOut.csv (0.74509) was for ID3 with best hyperparameter.

Other files are just small tweaks because the F-score that I get does not exactly match when I submit the file.