CODE:

import pandas as pd

from sklearn.metrics import (precision\_score, recall\_score,f1\_score, accuracy\_score,mean\_squared\_error,mean\_absolute\_error, roc\_curve, classification\_report,auc)

testdata = pd.read\_csv('dnnres/dnn1predicted.txt', header=None)

traindata = pd.read\_csv('dnnres/expected.txt', header=None)

y\_train1 = traindata

y\_pred = testdata

accuracy = accuracy\_score(y\_train1, y\_pred)

recall = recall\_score(y\_train1, y\_pred , average="binary")

precision = precision\_score(y\_train1, y\_pred , average="binary")

f1 = f1\_score(y\_train1, y\_pred, average="binary")

print("DOS:")

print("\tAccuracy")

print("\t%.3f\n" %accuracy)

print("\tPrecision")

print("\t%.3f\n" %precision)

print("\tRecall")

print("\t%.3f\n" %recall)

print("\tF1 score")

print("\t%.3f" %f1)

print('\n')

testdata = pd.read\_csv('dnnres/dnn2predicted.txt', header=None)

traindata = pd.read\_csv('dnnres/expected.txt', header=None)

y\_train1 = traindata

y\_pred = testdata

accuracy = accuracy\_score(y\_train1, y\_pred)

recall = recall\_score(y\_train1, y\_pred , average="binary")

precision = precision\_score(y\_train1, y\_pred , average="binary")

f1 = f1\_score(y\_train1, y\_pred, average="binary")

print("U2R:")

print("\tAccuracy")

print("\t%.3f\n" %accuracy)

print("\tPrecision")

print("\t%.3f\n" %precision)

print("\tRecall")

print("\t%.3f\n" %recall)

print("\tF1 score")

print("\t%.3f" %f1)

print("\n" )

testdata = pd.read\_csv('dnnres/dnn3predicted.txt', header=None)

traindata = pd.read\_csv('dnnres/expected.txt', header=None)

y\_train1 = traindata

y\_pred = testdata

accuracy = accuracy\_score(y\_train1, y\_pred)

recall = recall\_score(y\_train1, y\_pred , average="binary")

precision = precision\_score(y\_train1, y\_pred , average="binary")

f1 = f1\_score(y\_train1, y\_pred, average="binary")

print("PROBE:")

print("\tAccuracy")

print("\t%.3f\n" %accuracy)

print("\tPrecision")

print("\t%.3f\n" %precision)

print("\tRecall")

print("\t%.3f\n" %recall)

print("\tF1 score")

print("\t%.3f" %f1)

print("\n" )

testdata = pd.read\_csv('dnnres/dnn4predicted.txt', header=None)

traindata = pd.read\_csv('dnnres/expected.txt', header=None)

y\_train1 = traindata

y\_pred = testdata

accuracy = accuracy\_score(y\_train1, y\_pred)

recall = recall\_score(y\_train1, y\_pred , average="binary")

precision = precision\_score(y\_train1, y\_pred , average="binary")

f1 = f1\_score(y\_train1, y\_pred, average="binary")

print("R2L:")

print("\tAccuracy")

print("\t%.3f\n" %accuracy)

print("\tPrecision")

print("\t%.3f\n" %precision)

print("\tRecall")

print("\t%.3f\n" %recall)

print("\tF1 score")

print("\t%.3f" %f1)

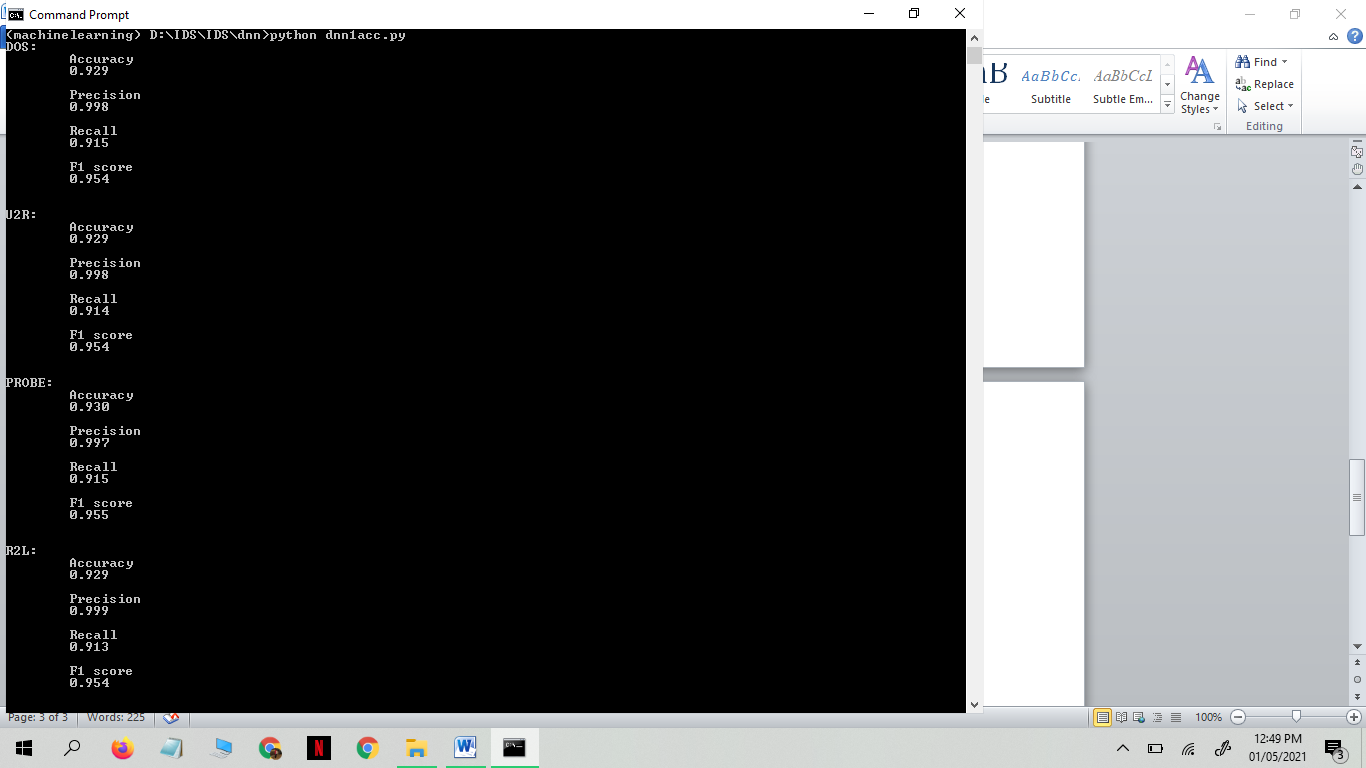
print("\n" )

testdata = pd.read\_csv('dnnres/dnn5predicted.txt', header=None)

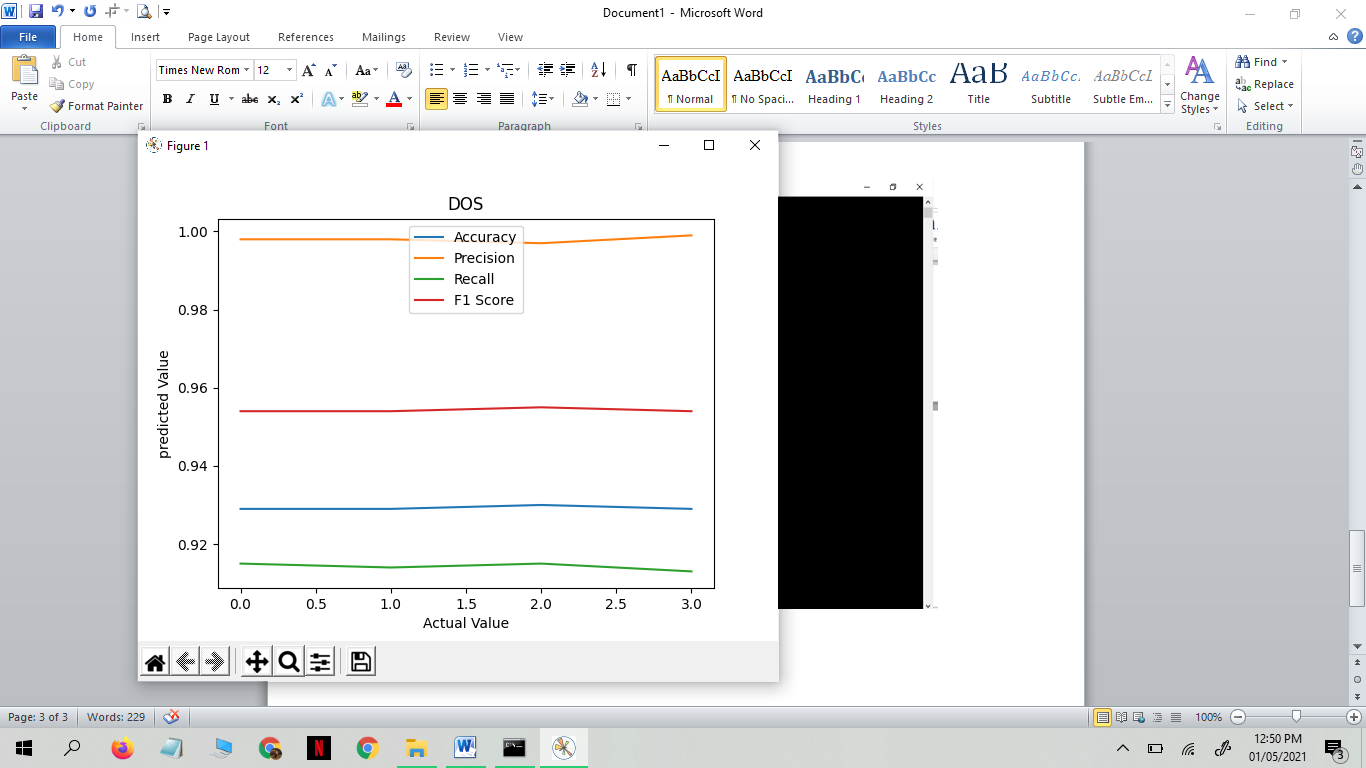
traindata = pd.read\_csv('dnnres/expected.txt', header=None)

OUTPUT:

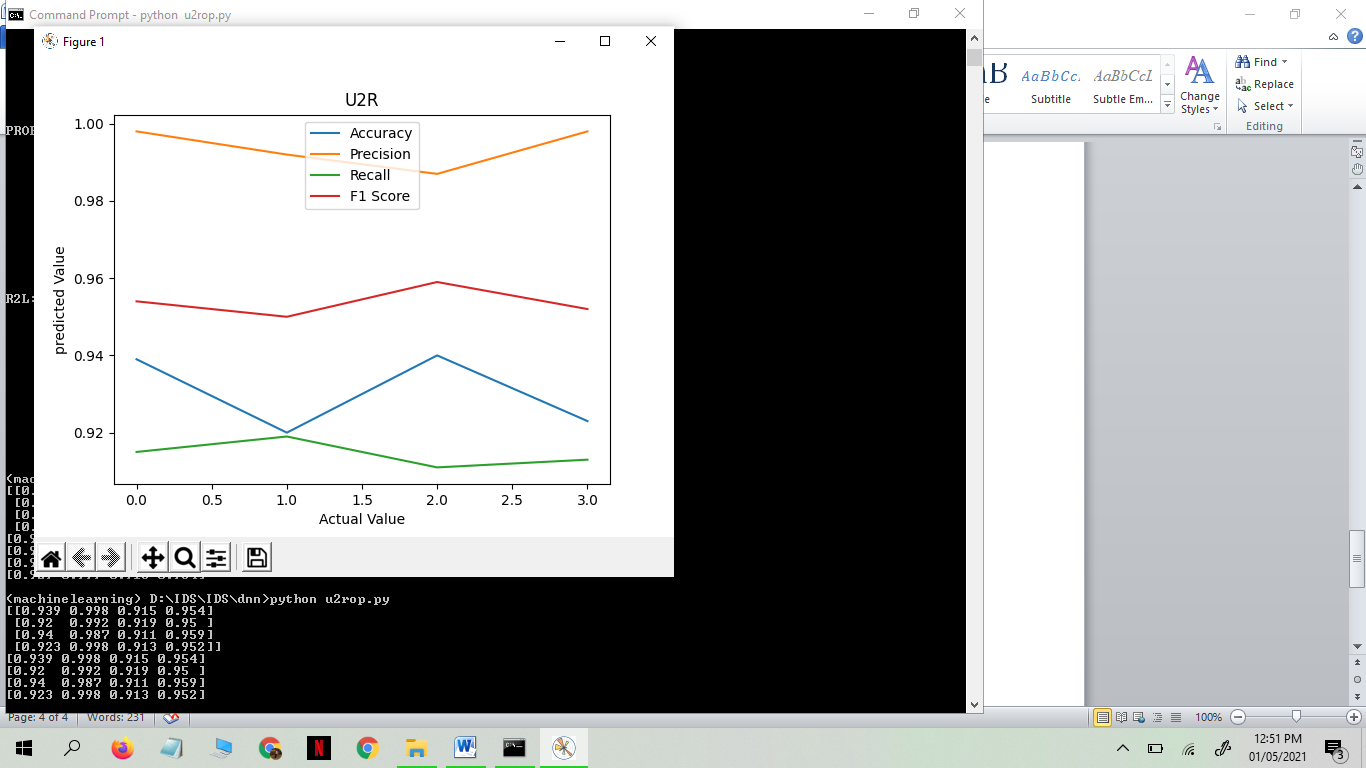
Cmd Output



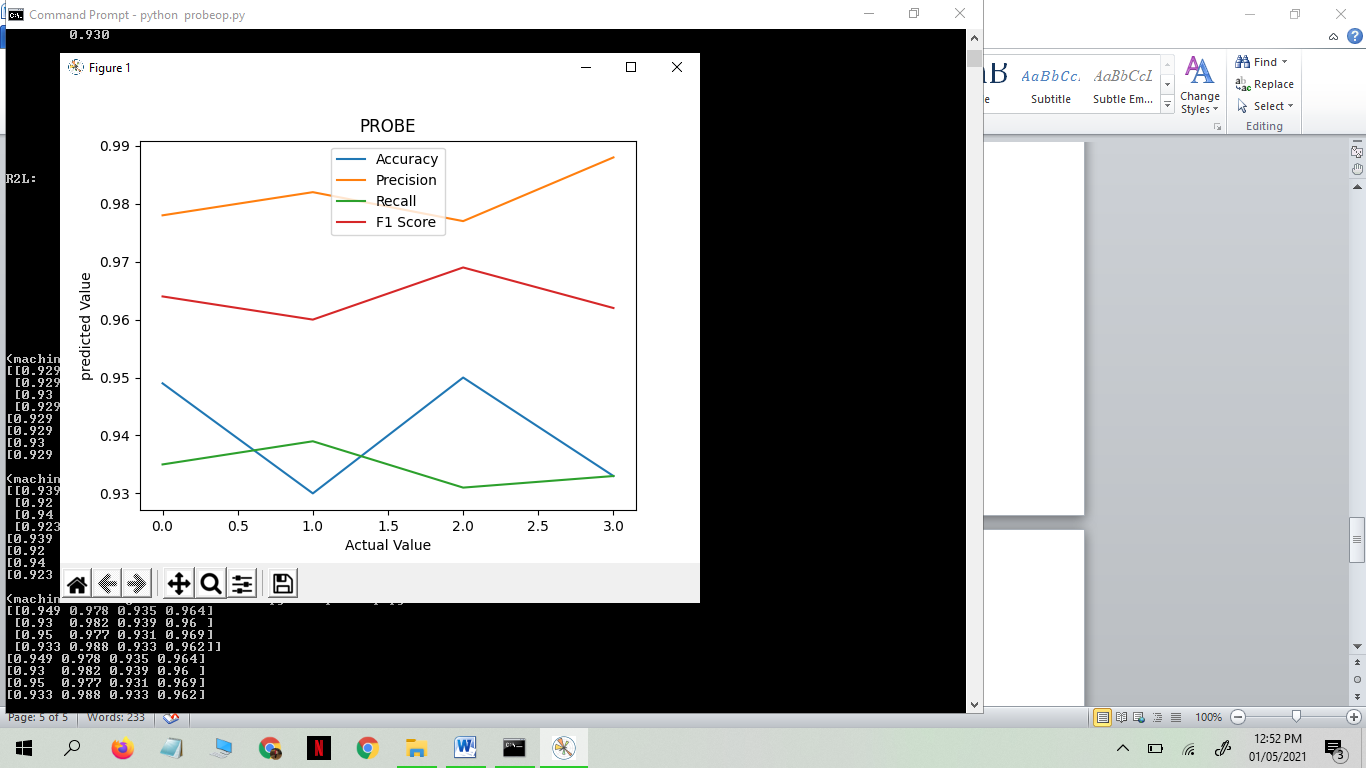
DOS Output:



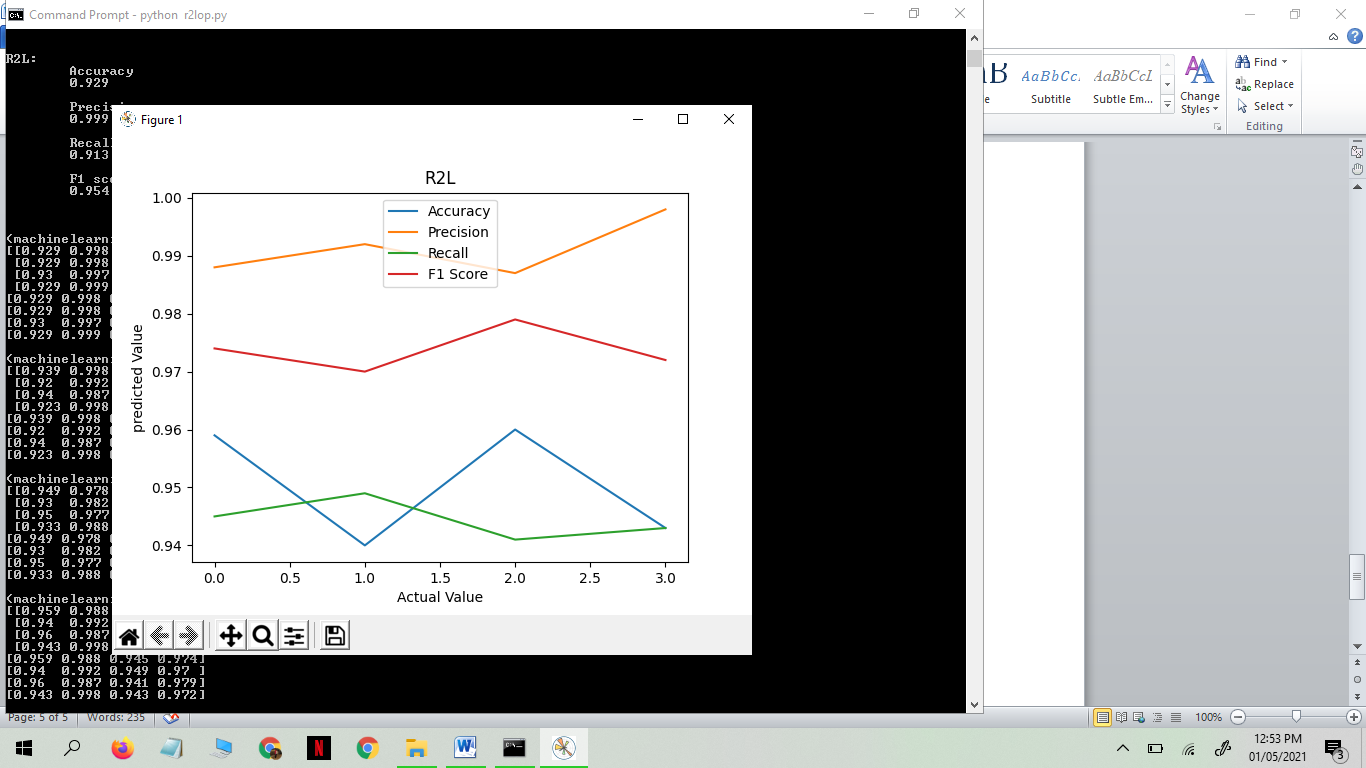
U2R Output:



PROBE Output:



R2L Output:



To conclude this and to suggest which one is best as per seeing using 4 of the algorithms, We conclude that DOS i.e Denial-of-Service-Attack is the best as it shows the consistent accuracy, precision, recall and F1 Score.