DoS

Amish Regmi, Eman Abdelfattah

School of Theoretical & Applied Science

Ramapo College of New Jersey

Mahwah, NJ

aregmi@ramapo.edu , [eabdelfa@ramapo.edu](mailto:eabdelfa@ramapo.edu)

***Abstract*—This paper presents ……**

***Index Terms – ;***

# Introduction

Introduction goes here.

This paper is organized as follows: section II presents the related work ….. In section III, ……….. Section IV ……... Finally, section V offers the conclusion.

# Related Work

related work goes here.

Ray investigated the effects of architecture on the performance of intrusion detection systems (IDSs). An equation was formed to find the optimal number of hidden neurons in a multi-layer feed forward neural network (MLFFNN) IDS. This equation can be used to determine the number of hidden neurons to eliminate the lengthy trail and error calculations in case of MLFFNN.

Guo *et al.*

# Dataset description

# experimental result and analysis

This paper applied different techniques of classification and analyzed the CICID2017 dataset in numerous ways. Different performance measures were calculated and compared, including Precision, Recall, and F1 scores. The training time for the classifiers were recorded, and all the calculations were done in an i5-8th generation machine with 8 GB of memory. The precision is calculated by dividing the number of true positive (TP) instances over the sum of the number of true positive and false positive (FP) instances. The recall is calculated by dividing the number of true positive instances over the sum of the number of true positive and false negative (FN) instances. The equation for calculating the F1 score is as follows:

F1 score = 2TP/ (2TP+FN+FP)

The classifier can get a high F1 score only if both recall and precision are high.

Scholastic Gradient Descent (SGD), also known as incremental gradient decent classifier, has advantages of handling very large datasets and dealing with training instances independently. SGD classifier demonstrated average performance before scaling the values, since the features have considerable differences between the minimum and maximum values. The training time for the classifier was under 20 seconds. The precision score, recall score, and f1 score before scaling were around 85.7% before scaling. The confusion matrix before scaling is shown in Figure 1.

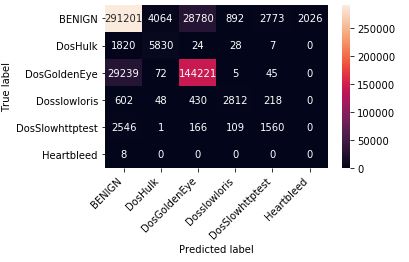


Figure 1. SGD confusion matrix before scaling

After scaling the values, however, the performance of the classifier was good, with precision and recall scores over 97%. The time taken for training for the SGD classifier was a little more than 8 seconds. The confusion matrix of the classifier after scaling is shown in Figure 2.

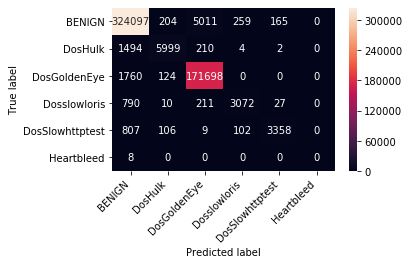


Figure 2. SGD Confusion matrix after scaling

Random Forests (RF) classifier works by training over many Decision Trees on random subsets of the features, and then averages out their predictions [2]. The performance of RF classifier was excellent, with near to perfect accuracy scores. Even without scaling the data, the precision, recall, and F1 scores of the classifier exceeded 99.9%. The time taken for Random Forest classifier was 25.16 seconds. The confusion matrix of the classifier before scaling is shown in Figure 3.

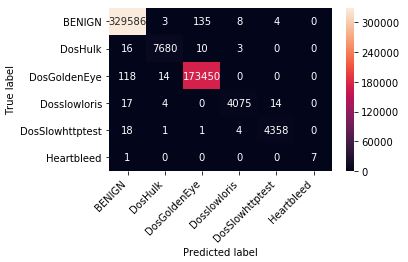


Figure 3. RF confusion matrix before scaling.

The RF classifier didn’t show a significant difference in its confusion matrix after the values were scaled, as demonstrated by Figure 4.

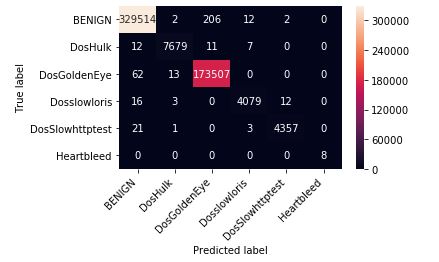


Figure 4. RF confusion matrix after scaling.

A Support Vector Machine (SVM) is a powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification and regression. SVM is well suited for classification of complex but small or medium sized datasets [3]. The results of using a linear kernel for SVM demonstrated excellent performance after scaling, with a prediction score of over 99%. The training time for the classifier was extremely high, exceeding 2120 seconds. The confusion matrix obtained from the classifier after scaling the values is shown in Figure 5.

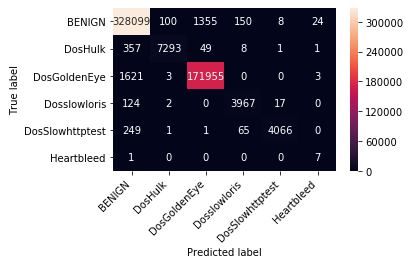


Figure 5. SGD confusion matrix after scaling.

K-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In k-NN classification, the output is a class membership. k-NN classifier demonstrated poor performance, with an accuracy score of just above 50%. The training time for the classifier was a little more than 1800 seconds.

Naive Bayes (NB) classifier is a probabilistic classifier based on applying Bayes’ theorem with strong independence assumptions between the features. A Gaussian Naïve Bayes (GNB) is a special type of NB algorithm specifically used when the features have continuous values. It’s also used when all features are following a normal distribution. The classifier showed poor accuracy with a precision, recall, and F1 score of just over 50%. The training time for the classifier was very short, just over 5 seconds. The confusion matrix of the classifier before scaling the values is shown in Figure 6.

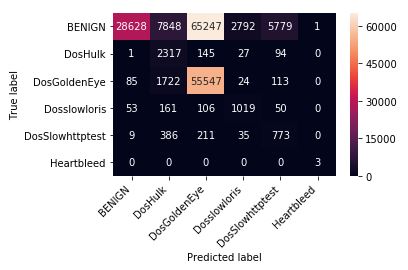


Figure 6. NB confusion matrix before scaling.

The behavior of the classifier was strange after scaling it, with it’s output only in the “Benign” class. The precision, recall, and F1 scores remained in a similar range. The confusion matrix after scaling is displayed in Figure 7.

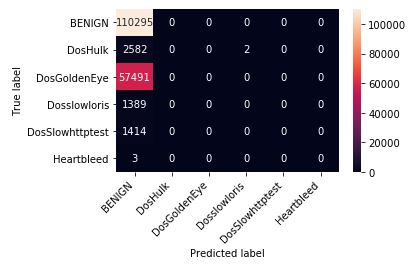


Figure 7. NB confusion matrix after scaling.

Decision Tree (DT) builds a tree by determining features in the dataset to split the data on, splitting the data, and recursively generating new nodes by using subsets of data created. The training time for the classifier exceeded 35 seconds. The classifier displayed average performance, with an accuracy score of just over 60%. The confusion matrix of the classifier after scaling the values has been displayed in Figure 8.

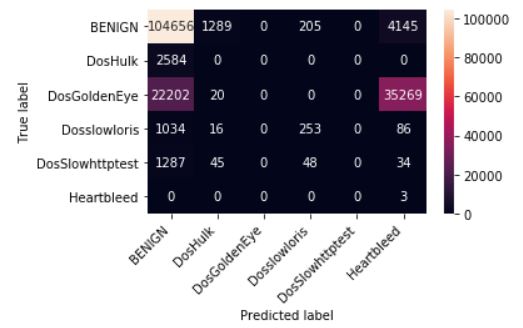


Figure 8. DT confusion matrix after scaling

The precision recall and F1 scores of the classifiers have been compared in the table in Figure 8.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SGD | RF | SVM | DT | GNB |
| Precision | 0.978 | 0.999 | 0.992 | 0.99 | 0.932 |
| Recall | 0.978 | 0.999 | 0.992 | 0.99 | 0.932 |
| F1 | 0.978 | 0.999 | 0.992 | 0.99 | 0.932 |

Figure 8. Comparison of performance parameters of different classifiers after scaling the data

Due to the uneven of distribution of target dataset, the performance scores of GNB came out better than expected, since all the predictions belonged to the target with the greatest number of samples in the dataset i.e. BEINGN. As shown in Figure 8, Random Forest outperformed other classifiers.

# Conclusion and future work

Conclusion goes here.

# References

[a] L. Ray, "Determining the Number of Hidden Neurons in a Multi Layer Feed Forward Neural Network," *Journal of Information Security Research,* vol. 4, no. 2, pp. 63-70, 2013.

.