CSC730: Assignment 5
ADBench – Installing and taking it for a test drive

Kristophor Ray Jensen
Electrical Engineering and Computer Science
South Dakota School of Mines and Technology
Rapid City, United States
0009-0001-7344-349X

I. Introduction

For assignment 5 we are tasked to learn about ADBench, a so-called anomaly detection toolset. This toolset is a compilation of 57 datasets and 30 anomaly detection algorithms. These algorithms can be classified as supervised, semi-supervised, and unsupervised. ADBench was presented in a paper by Han et al. in 2022 and at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022). The paper is titled "AD-Bench: A Benchmark for Anomaly Detection in High-Dimensional Data" [1].

The assignement objectives are listed in the next section. One of the common themes of this assignment is the re-utilization of the 'skewed_MNIST' dataset. In this report we will discuss the installation of ADBench, the application of two algorithms to the 'skewed_MNIST' dataset, and the characterization of the results.

II. Homework Objectives

During class we were provided a set of requirements for the homework. The requirements were clarified in [2] a as follows:

- Install ADBench
- Verify the installation is working
- Use 'skewed_MNIST' dataset
- Treat the two sparsest classes as anomalies
- Select two algorithms from the ADBench library
- Apply the selected algorithms to the dataset
- Characterize the results
- Comment code
- Write a report
- Submit the code and report

III. Methodology

A. Installing ADBench

Installing ADBench was a difficult and time consuming task. The authors did not document the versions of the libraries they used during their paper. This also leads to a potential source of error in reproducing results.

The first attempt to install ADBench was on a Windows 10 machine using a fresh python 3.10 environment. The first calls to RunPipeline failed due to a deprecation error within scikit-learn originating from the parallel processing libary. Upon further research it was noticed that the authors requirements file did not fix the versions of the libraries. This was the first source of error. The version os scikit-learn that was installed by default was produced after the paper was published.

When this error was realized, a search of the Github pull requests began to locate the timeline and potential versions used by the authors. After several iterations of downgrading the libraries, the installation was successful. The final requirements file is located in the repository.

B. Running ADBench

When the installation issues were resolved and confirmation that the ADBench tool was running properly, the next step was to load the 'skewed_MNIST' dataset. The dataset was loaded and the two sparsest classes were identified. The two sparsest classes were then treated as anomalies. To treat these sparsest classes as anomalies, the labels were changed to 1 for the two sparsest classes and 0 for the rest of the classes.

The chosen models were multi-layer perception (MLP) and CatBoost (CatB). These two models were arbitrarily chosen. The models were then applied to the dataset and the results were characterized. The dataset was split using the $train\_test\_split$ from the $model\_selection$ class of scikit-learn. The specification for this split was that 20% of the data would be reserved for testing. Results characterization includes descriptive statistics, plotting the classes in a high-bin count histogram, then calculating confusion matrices data and accuracy based on verious thresholds. The results are discussed in the next section.

IV. Results

The skewed data set contains 12244 datapoints each containing 784 dimensions of data. Review table I for the counts of the split dataset. The results of the MLP model and the CatB model are shown in Figures 1 and 2 respectively. The same dataset was used for both models.

After the models were fit to the data, the results were characterized. The first step in characterizing the results was to generate descriptive statistics. The statistics for the MLP model are shown in Table II and the statistics for the CatB model are shown in Table III.

Table I
DATA SET COUNTS

Data Set	*Normal 0*	*Anomaly 1*
Training	9769	26
Test	2441	8

Table II
MLP MODEL DESCRIPTIVE STATISTICS

Statistic	*Normal*	*Anomaly*
Mean	0.0011367	0.516422
Standard Deviation	0.0326209	0.482698
Minimum	0.0000000	0.0000000
Maximum	1.0000000	1.0000000

Table III
CATB MODEL DESCRIPTIVE STATISTICS

Statistic	*Normal*	*Anomaly*
Mean	0.000452	0.365747
Standard Deviation	0.004549	0.317829
Minimum	0.000000	0.000976
Maximum	0.160315	0.782518

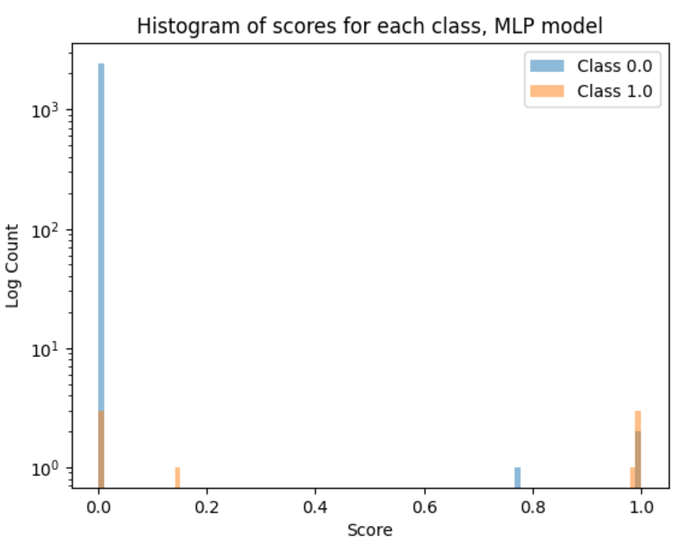

Figure 1. Multi-layer perception model results.

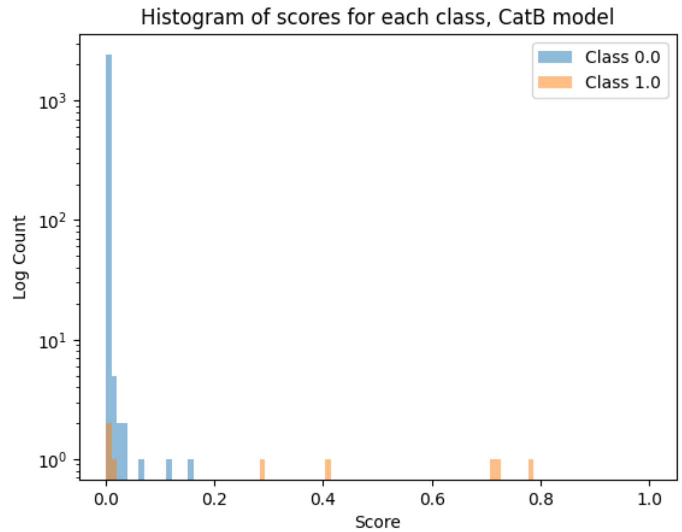

Figure 2. CatBoost model results.

After reviewing the descriptive statistics and the histogram plots of the results there appears to be a seperation between the anomalous data and the normal data. To further characterize the results, confusion matrices were generated for the MLP model and the CatB model. The confusion matrices are shown in Table IV and Table V respectively. Also, the accuracy of the models were calculated. Due to the large number of non-anomalous data points, the accuracy is not a good measure of the model's performance. A better measure of the model's performance is the precision and recall [3]. The precision and recall for each model with varying thresholds are shown with their confusion matrix data.

Table IV
MLP MODEL CONFUSION MATRIX

Predicted	0.5σ *N*	*A*	1.0σ *N*	*A*	2.0σ *N*	*A*	3.0σ *N*	*A*
Actual Normal	2438	3	2438	3	2438	3	2438	3
Actual Anomaly	3	5	3	5	3	5	3	5
Accuracy	0.9975		0.9975		0.9975		0.9975	
Threshold	0.0175		0.0338		0.0664		0.0990	
Precision	0.625		0.625		0.625		0.625	
Recall	0.625		0.625		0.625		0.625	

V. CONCLUSION

Assignment 5 was a time consuming task due to the installation issues with ADBench. The authors did not document the versions of the libraries they used during their paper. Withstanding the installation issues, the results of the MLP model and the CatB model show that the models are able to distinguish between the normal data and the anomalous data. However, the precision and recall of these simple and untuned

Table V
CatB model confusion matrix

Predicted	0.5σ ***N***	***A***	1.0σ ***N***	***A***	2.0σ ***N***	***A***	3.0σ ***N***	***A***
Actual Normal	2394	47	2415	26	2428	13	2431	10
Actual Anomaly	1	7	1	7	2	6	2	6
Accuracy	0.9804		0.9890		0.9938		0.9951	
Threshold	0.0027		0.0050		0.0096		0.0141	
Precision	0.130		0.212		0.316		0.375	
Recall	0.875		0.875		0.875		0.875	

models are not ideal. This was a good excercise in understanding anomaly detection in general and the capabilities compiled by the Authors within ADBench.

References

[1] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," in *Neural Information Processing Systems (NeurIPS)*, 2022.

[2] R. Loveland, "Assignment_5.pdf," From SDSMT D2L Website, 2024.

[3] W. Contributors, *Precision and recall*, https://en.wikipedia.org/wiki/Precision_and_recall, 20 Feb. 2024.