# Neighborhood Structure Assisted Non-negative Matrix Factorization and its Application in Unsupervised Point-wise Anomaly Detection

Welcome to the online companion of our paper. It contains the codes and datasets required to generate the results reported in the paper. The description of these datasets and individual code files are explained below:

## Datasets:

The paper uses two groups of anomaly detection datasets, referred to as:

**Benchmark datasets:** We used 20 benchmark datasets for the performance evaluation. Several versions of these data sets are stored in the online repository of [1] (https://www.dbs.ifi.lmu.de/research/outlier-evaluation). These versions mainly differ in terms of the preprocessing steps used and the proportion of anomalies compared to the normal observations. Table I in the paper summarizes the basic characteristics of these 20 data sets used in our study. All of these 20 datasets are stored in .csv format. The last columns of these .csv files indicate the anomaly information (1 or 'yes' indicates an anomaly and 0 or 'no' indicates a normal observation).

**Hydropower dataset:** We also used an industry dataset for anomaly detection performance evaluation. It came from a hydropower plant located in Europe. The hydropower data is time-stamped (a total of 7 months' worth of data) and divided into different functional areas (turbines, generators, bearings, etc.). After preprocessing by ourselves and combining data across functional areas, there are around 9200 observations (rows in a table) and 222 attribute variables (columns in a table). The first row contains the headers for each column. Each row has a time-stamp assigned to it and attributes are primarily temperatures, vibrations, pressure, harmonic values, active power, and so on. This hydropower data set was studied in a preliminary effort [2], which presents additional details of the data preprocessing step.

## Codes:

Codes & Datasets subfolder contains all the scripts and functions to generate results in the paper. The scripting language used includes Matlab, Python and R. In Python, we used several libraries that need to be installed and imported during the execution of python (.py) codes. The required libraries are listed below:

- SciPy 1.5.4
- Tensorflow 1.5.0
- numpy
- multiprocessing
- h5Py
- math
- time
- sys
- random

Similarly in R, we used several packages apart from the basic packages and they are required to be installed and loaded before running any of the R code (.r) files. The required packages are listed below:

- dbscan
- dplyr
- fossil
- PCCMR
- matrixStats
- HighDimOut

Under the Codes & Datasets folder, we have several Matlab scripts & functions, a couple of R scripts & functions, several Ipython notebooks and two Python scripts. The main code files are explained below:

**DSGD_CNMF_public.py:** This Python script will implement the offline NS-NMF algorithm. It will generate the low rank matrices using the original data matrix and the MST similarity matrix generated by mstsim.r. It implements a parallel block algorithm for faster and parallel processing.
**GNMF_Multi1.m:** This Matlab function will implement the GNMF algorithm.
**symnmf_newton.m:** This Matlab function will implement the SNMF algorithm.
**OnlineNSNMFV3.m:** This Matlab function will implement the online NS-NMF algorithm.
**OnlineNSNMFhydro.m:** This Matlab function will implement the online NS-NMF algorithm for the hydropower plant data.
**Table5.m:** This Matlab script will generate anomaly detection performance for the 5 algorithms across 20 benchmark datasets using all of the above scripts and functions.
**Table7.m:** This Matlab script will generate anomaly detection performance for the 5 algorithms on hydropower dataset.
**LoMST_PracticalK.r:** This R script will implement the LoMST algorithm on all 20 datasets.
**'datasetname'euclidMDSLE.ipynb:** These IPython notebooks will generate the anomaly detection performance following the Euclidean distance regularized autoencoder model in Table 9.
**DAGMM.ipynb:** This IPython notebook will generate the anomaly detection performance following the DAGMM approach for each of the 20 datasets used in Table 9.
**RankAverage.r:** It will generate the average rank performance of competing approaches across 20 datasets.
**friedman.r:** It will first perform the Friedman test on the detection results generated by competing approaches. Then it will generate the p-value table reported in Table 4 in the paper.
**Figure2.m:** This Matlab script will be used to generate the post hoc multiple comparison figure.
**ExampleRun_WDBC.m:** This Matlab function will simulate the anomaly detection process of all 5 competing approaches using an example dataset (WDBC.csv)

Apart from these main scripts we have several other helper functions that are required for compiling the above scripts or generate data files used in those scripts, they are briefly mentioned below:

**mstsim.r:** This R script will generate the similarity/distance matrix using minimum spanning tree. The output will be saved as a .csv file which will be used as an input to the DSGD_CNMF_public.py file.
**Mat2BlockPublic.py:** This Python function will be called from DSGD_CNMF_public.py to divide the data files into blocks.
**LoMST.r:** It is the core LoMST algorithm function. The function will be called to return the number of true detection and anomaly indices for a specified K value for any dataset whose anomaly information is known to the user.

**scale_dist3_knn.m:** This Matlab function generates the k-nearest neighbors' distance for a data matrix, utilized in both GNMF and SNMF algorithm.
**newW_new.m:** This Matlab function generates temporal LoMST weights for online NS-NMF approach.
**NSNMF_Update_U.m:** This Matlab function provides update of one of the low-rank matrices.
**NSNMF_Update_V.m:** This Matlab function provides update of one of the low-rank matrices.

**Machine Specification**: Intel Core i7(7700HQ@2.80 GHz), 16GB Ram; Windows 10
**Software Version:** Python 3.6; R version 3.6.2; MATLAB version 2019b

## Steps to generate anomaly detection outcome for an example data file (WDBC.csv):

- Generate MST distance matrix using mstsim.r file, it will be saved as edgesWDBC.csv.
- Run DSGD_CNMF_public.py to generate low-rank matrices. They will be saved as .mat files.
- Run ExampleRun_WDBC.m to generate detection outcome for all 5 approaches.

## Reproducing the results in the paper:

For the convenience of the users, in the following table, we summarize how to reproduce different tables and figures used in the paper. Before running any codes, set your working directory to “.\Codes & Datasets”.

| Which Results to Reproduce | Data File | Script File | Output |
|---|---|---|---|
| Table 3 | papercounts.csv | RankAverage.r | This R file only produces the last row of Table 3 (mean relative ranks of the competing approaches based on their detection outcome in Table 5); other rows of Table 3 are manually produced using data in Table 5 |
| Figure 2 | ranks.csv (generated from friedman.r) | Figure2.m | Post-hoc comparison figure |
| Table 4 | papercounts.csv | friedman.r | p-value table and Friedman test summary |
| Table 5 | All 20 datasets (.csv) | Table5.m | All columns of Table 5 |
| Table 6 | All 20 datasets (.csv) | LoMST_PracticalK.r | Column 3 of Table 6 |
| Table 7 | hydro.csv and hydropower.mat | Table7.m | All columns of Table 7 |

| | | | |
|---|---|---|---|
| Table 9 | All 20 datasets (.csv) | ‘datasetname’euclidMDSLE.ipynb* (Column 3)<br>DAGMM.ipynb (Column 4)<br>***Replace ‘datasetname’ with the respective data file name** | Column 3 and 4 of Table 9 |

[1] G. O. Campos et al., “On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study,” Data Mining Knowl. Discovery, vol. 30, no. 4, pp. 891–927, 2016.

[2] I. Ahmed, A. Dagnino, A. Bongiovi, and Y. Ding, “Outlier detection for hydropower generation plant,” in Proc. 14th IEEE Int. Conf. Automat. Sci. Eng. (CASE), Aug. 2018.

****Thank you for using this online companion. If you have any questions on implementing our algorithm, please feel free to send an email at imtiazavi@tamu.edu or imtiaz_avi@yahoo.com**