Unsupervised Anomaly Detection Based on Minimum Spanning Tree Approximated Distance Measures and Its Application to Hydropower Turbine

Welcome to the online companion (Code and Datasets) of our paper. There are three subfolders located under the root folder along with this README file. The contents of these three subfolders are explained below:

Datasets:

This subfolder stores all the datasets required for generating the results in the paper. 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 ('yes' indicates an anomaly and '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. This dataset just like the benchmark datasets is available in .csv format.

Codes:

This subfolder contains all the scripts and functions to generate results in the paper. The primary scripting language used is R. Additionally to generate one figure a Matlab (.m file) script will be used. 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 Code folder, we have several R scripts and one Matlab script. Their purposes are explained below:

Resultgeneration.r: This R script will guide the user on how to perform anomaly detection, in general, using our LoMST approach under both the best K and practical K setting. It will generate the number of true detections and their indices using an example dataset. The users are welcome to replace it with their own dataset. However, keep in mind that if you know the anomaly information beforehand, use 'LoMST.r' function, otherwise refer to 'hydroLoMST.r' function. The anomaly information, if available, needed to be supplied in a separate column ('yes' for an anomaly, 'no' for a normal observation).
LoMST_BestK.r: This R script will generate true detection counts for all 20 benchmark datasets under the best K scenario for our LoMST approach. Note that for the other 13 competing approaches their best K results can be accessed from https://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/.
LoMST_PracticaltK.r: This R script will generate true detection counts for all 20 benchmark datasets under the practical K scenario for our LoMST approach. It also generates the average rank performance of competing approaches across 20 datasets. Note that for the other 13 competing approaches their results using the same K value (practical K) can also be accessed from https://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/.
hydro.r: This R script will generate the anomaly detection outcome for the hydropower plant for our LoMST approach and competing approaches (LOF and SOD)
PracticalKselection.r: This R script will guide the users on the practical K selection process through two example datasets (Cardiotocography and Glass).
AllRank.r: This R script will generate the rank outcome of our approach for all 20 benchmark datasets under both best K and practical K setting.
BestKRankAverage.r: It will generate the average rank performance of competing approaches across 20 datasets under the best K setting.
friedmanj67.r: It will first perform the Friedman test on the detection results generated from both best k and practical k setting. Then it will generate the p-value table for both settings.
friedmanplot.m: This only Matlab script will be used to generate the post hoc multiple comparison results.

Apart from these R scripts we have several R functions that are required to compile the above scripts, they are briefly mentioned below:

BestK.r: It will be called to return the best K value for any dataset.
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.
hydroLoMST.r: This function will be called to generate the anomaly scores for the hydropower dataset for a specified K value. It can also be called to generate anomaly scores for any dataset whose anomaly information is not known.

Figures:

This small subfolder mainly contains the figures which are generated running the scripts described above.

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

Reproducing the results in the paper:

For the convenience of the user of this online companion, in the following table, we summarize how to reproduce different tables and figures used in the paper.

Which Results to Reproduce	Data File	Code File	Output
Figure 6	cardioto.csv and glass.csv	PracticalKSelection.r	Two Plots in Figure 6 generated as .SVG files
Table II	Best k Original counts.csv (from Table IV)	BestKRankAverage.r	The last row of Table II (Average ranks of the competing approaches for best K)
Table III	All 20 datasets (.csv) and Practical k Original counts.csv	LoMST_PracticalK.r	The last row of Table III (Average ranks of the competing approaches for practical K)
Table IV	All 20 datasets (.csv)	LoMST_BestK.r	The first column (Our approach's result for best K) of Table IV
Table VII	Best k Original counts.csv and Practical k Original counts.csv	friedmanj67.r	Two columns of Table VII
Table VIII	Best k Original counts.csv and Practical k Original counts.csv	AllRank.r	Last two columns of Table VIII
Figure 7	formatlab.csv	Friedmanplot.m	Figure 7 generated in Matlab
Table IX	Hydropower.csv	Hydro.r	All columns of Table IX

[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