# Multiple Kernel Anomaly Detection for FOQA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Notices:

Copyright © 2010, 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. All Rights Reserved.

Disclaimers

No Warranty: THE SUBJECT SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THE SUBJECT SOFTWARE WILL CONFORM TO SPECIFICATIONS, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE SUBJECT SOFTWARE WILL BE ERROR FREE, OR ANY WARRANTY THAT DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE SUBJECT SOFTWARE. THIS AGREEMENT DOES NOT, IN ANY MANNER, CONSTITUTE AN ENDORSEMENT BY GOVERNMENT AGENCY OR ANY PRIOR RECIPIENT OF ANY RESULTS, RESULTING DESIGNS, HARDWARE, SOFTWARE PRODUCTS OR ANY OTHER APPLICATIONS RESULTING FROM USE OF THE SUBJECT SOFTWARE. FURTHER, GOVERNMENT AGENCY DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THIRD-PARTY SOFTWARE, IF PRESENT IN THE ORIGINAL SOFTWARE, AND DISTRIBUTES IT "AS IS."

Waiver and Indemnity: RECIPIENT AGREES TO WAIVE ANY AND ALL CLAIMS AGAINST THE UNITED STATES GOVERNMENT, ITS CONTRACTORS AND SUBCONTRACTORS, AS WELL AS ANY PRIOR RECIPIENT. IF RECIPIENT'S USE OF THE SUBJECT SOFTWARE RESULTS IN ANY LIABILITIES, DEMANDS, DAMAGES, EXPENSES OR LOSSES ARISING FROM SUCH USE, INCLUDING ANY DAMAGES FROM PRODUCTS BASED ON, OR RESULTING FROM, RECIPIENT'S USE OF THE SUBJECT SOFTWARE, RECIPIENT SHALL INDEMNIFY AND HOLD HARMLESS THE UNITED STATES GOVERNMENT, ITS CONTRACTORS AND SUBCONTRACTORS, AS WELL AS ANY PRIOR RECIPIENT, TO THE EXTENT PERMITTED BY LAW. RECIPIENT'S SOLE REMEDY FOR ANY SUCH MATTER SHALL BE THE IMMEDIATE, UNILATERAL TERMINATION OF THIS AGREEMENT.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Dependencies:

* Red Hat 6.
* Python 3.7:
  + Recommended Anaconda 3.7 or higher (<https://www.continuum.io/downloads>). This has all the packages and is a self-contained installation.
  + If using the standard python 3.7 installation on Red Hat 6 the following additional packages are needed:
    - Numpy 1.15.4 or higher.
    - Scipy 1.2.0 or higher.
    - Scikit-learn 0.20.3 or higher.
    - Matplotlib 3.0.2 or higher
    - progress 1.4 or higher

## Setup and Configuration:

The code assumes that the data has been exported into CSV files and gzipped with proper 1 line header.

1. Create a txt file list of flights with absolute path for each airport/runway/fleet type set.
2. Edit the appropriate fields in the config.json file. You will need to create separate config.json files for each airport/runway/fleet type. Note each field should be a separate line so that the shell scripts can parse appropriately.
   * “name: Unique run ID name to be appended to files and directories.
   * “filelist”: absolute path to the filelist created above.
   * “working\_dir”: path to the directory where the landing phase of the flights will be saved as pickle dumps. The normalization statistics and the ordered list of flights in the SVMlight file are saved here as well. This may be a subset of the original list due to flights not passing data quality checks for partitioning.
   * “svmlight\_file”: absolute path to where the SVMlight file will be stored.
   * “MKAD\_folder”: path to the models, logs, and output report file.
   * “important\_params”: this is a python dictionary with the following keys:
     + “alt”: The barometric altitude parameter.
     + “td\_indicator”: Parameter that indicates weight on wheels. Used to determine when the flight is airborne or not.
     + ground\_speed”: Ground speed parameter. Used to integrate distance from touchdown.
   * “params”: this is a python dictionary with the following keys:
     + “continuous”: path to the continuous parameter list. One line per parameter.
     + “discrete”: path to the discrete parameter list. One line per parameter.
   * “alphabet”: hyper parameter for the SAX algorithm. This is the quantization value that scales the data between 1 and A (max is 20 but typically 10)
   * “window\_size”: hyper parameter for SAX algorithm. Non-overlapping window size to down sample the data (typically 30 seconds).
   * “starting\_alt”: AGL altitude that the preprocessing step uses to star the partition of the landing phase of the flight (typically 6000 ft).
   * “cluster\_eps” : Setting in DBSCAN clustering of parameter contributions. (typically 0.07)
   * “save\_kernel” : Whether to save the kernel as a pickle file in the working\_dir. The kernel can be loaded using existing\_kernel.
   * “use\_existing\_kernel” : If the kernel has been saved from a previous run it will search in the working\_directory. If found it will skip computing the kernel.

## Run Code:

1. Run:
   * **PyMKAD\_Release]$ python PythonCode/preprocess\_files\_multiprocess.py config.json [number\_of\_processesors (default=1)]**
2. Run:
   * **PyMKAD\_Release]$ python PythonCode/run\_mkad.py config.json [number\_of\_processesors (default=1)]**
3. Run:
   * **PyMKAD\_Release]$ python PythonCode/visualization.py config.json [number\_of\_processesors (default=1)]**

## Output:

The output will be a CSV file called: anomalous\_flights\_contributions\_${NAME}.csv. ${NAME} is the “name” field in the config.json file. The rows are the flights (file names) that were identified as anomalous by the algorithm. The 2nd column is the cluster ID. Flights that have similar contributing factors will cluster together. The remaining columns are the weighted contributions for each of the continuous parameters. These weights will sum to 1. High weights indicate the parameter is significantly unusual for this flight. Visualization will produce 2 pdf files for each anomalous flight: one for the continuous parameters and another for the discrete. Red parameter names indicated higher contributions. These files will be located in: ${MKAD\_Folder}/figs/.