Elaboration for Online Processing

- **Data Generator:** Merge and read from multiple data files and write them to TSDB at a fixed rate
  - Including the input files for normal and faulty situations
    - Assume that they are F0, F1, F2 and their sizes are size(F0), size(F1), size(F2)
    - Let Tsize = size(N) + size(F1) + size(F2)
  - Divide the data by input window size and we will refer to each as an input block
    - Let IW be the input window size (each block is of size W)
    - You need to choose a reasonable IW such that the data is sufficient for fault recognition
  - Repeat in a loop the following, until offT % of the data are retrieved
    - offT: offline training; onlT: online training
    - Read a block of data from one of the input files
    - The input file is selected randomly with probability = size(Fi) / Tsize
    - Write the retrieved data to TSDB-offT
      - Add labels to the data before writing (label the fault type)
      - Data should be written at a fixed rate SR (sensor rate)
  - Repeat in a loop the following, until all data in all data files are retrieved
    - Do the same as above, but write the retrieved data to TSDB-onlT
      - If some data files are exhausted, but others are not, then just skip the exhausted ones when they are selected

- **Offline Training**
  - Read the data from TSDB-offT and feed to the learning system
    - You can start training after all the offline training data are in TSDB or in sync with the Data Generator
  - Obtain a Training Model and pass it to the Online Testing and Training unit

- **Online Testing and Training**
  - Define a processing window size PW, we refer to each as a data segment
    - Generally speaking, we should have PW < IW
  - Read a data segment from TSDB-onlT and process the segment
    - Need to setup a sync mechanism to make sure that the complete window of data is available
      - Explore what the TSDB has offered for easy sync
      - Some TSDBs may support the continuous queries, which can be used for this purpose
      - In the worst case, sync the sleep time in Data Generator and Online Testing and Training
- Feed it to the Trained Model to determine whether there is a fault and if so, the fault type
  - Without the label
- Feed it to the Trained Model to perform additional training
  - With the label added by Data Generator

- Send the output for graphing

- **Online data Graphing**
  - **Measurement data graphing**
    - Use the graphing utilities provided by the TSDB to graph each measurement
  - **Fault data graphing (accurate labels)**
    - Use the graphing utilities provided by the TSDB to graph the system fault label
    - The label generated by the Data Generator
    - Set normal = 0, different fault types with different positive integers
    - Graph the system condition and observe faults
  - **Fault data graphing (diagnosis results)**
    - Obtain the data from your Online Testing unit
    - Use the TSDB graphing libraries to plot the fault diagnosis results
      - With the same fault value settings
      - Layout the two fault data series next to each other to allow visual comparisons
  - **Fault diagnosis result reporting**
    - Output the fault diagnosis analysis accuracy in various metrics in a text window