VANDERLANDE

User Manual for Outlier Detection in Event Logs of Material Handling System

Ozge Koroglu

August 2019

1. APPLICATION ARCHITECTURE

The application follows the atchitecture that is presented in Figure Architecture. First we obtain the event log and segments from BPI reports with the help of the Performance Spectrum Miner (PSM) tool. Then we execute the clustering.py if we want to update the clusters for the new interval of date. To get the outliers and outlier patterns for each segment for the given day, we execute the Outlier_Detection_in_BHS.py pipeline. Then the results that were obtained from the Outlier_Detection_in_BHS.py pipeline is visualized in the performance spectrum miner.

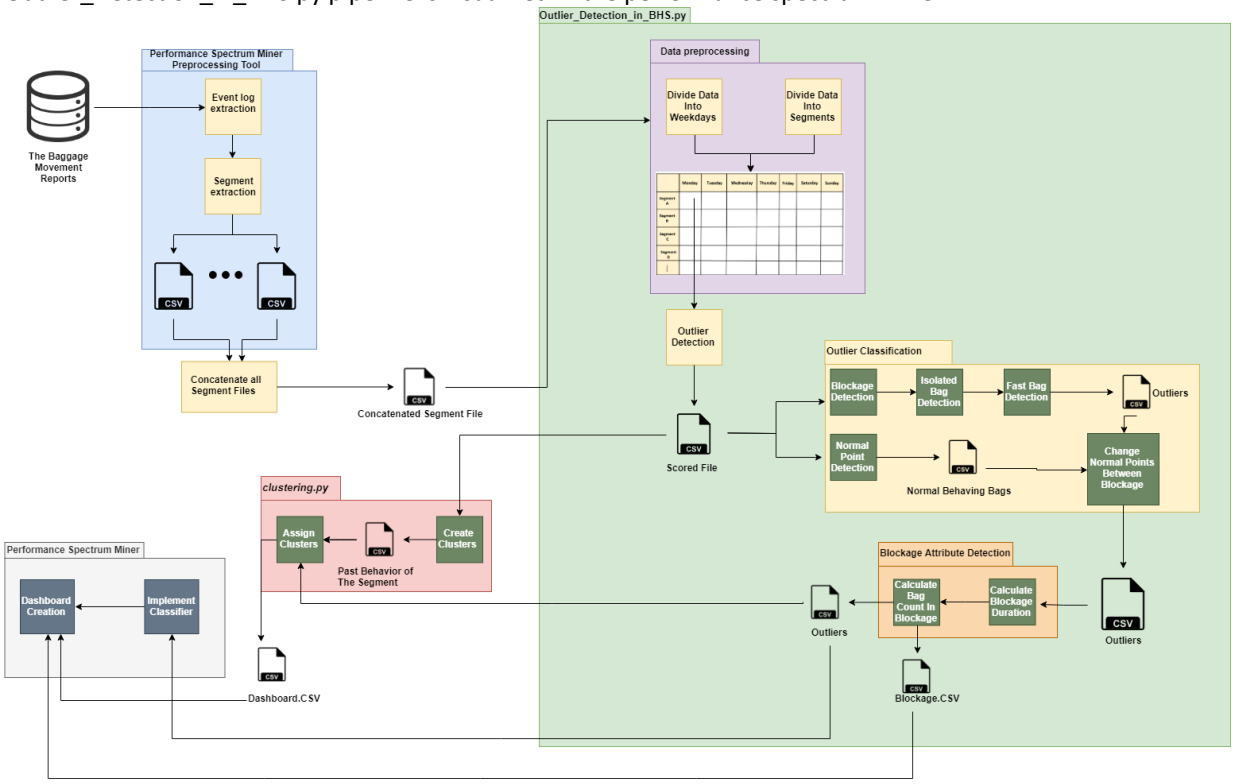


Figure 1 : Architecture

2. ENVIRONMENT SETUP

2.1. Install Anaconda

Start by accessing the Jupyter Project website. In the install tab, the most recommendable installation is described step by step.

We're using the latest version of Python 3 through the Anaconda distribution, so the first step is to download Anaconda and follow the step by step installation.

2.2. Managing Environments

After Anaconda is installed, we can open the terminal **Anaconda Prompt**. If you'd like to use the graphical user interface, execute anaconda-navigator. Using the terminal Anaconda Prompt is quite straightforward. We can create virtual environments and install packages on them. It is convenient to create a specific virtual environment for a concrete project, to isolate the project and avoid affecting other programs with your modifications in the environment.

2.2.1. Create a new environment

We start by creating a new environment. The command is `conda create -n env_name`

Also, to specify which version of Python to install in the environment, the command is `conda create -n outlier_detection python=3` This command will install the most recent version of Python 3.

2.2.2. Activate the new environment

The new environment has been created in the directory we were located. To activate the new environment, type `activate outlier_detection.`

If you wish to delete an environment, `conda env remove -n  outlier_detection.`

2.2.3. Install packages

Once we have created an environment and it is active, we can install all the packages needed for our software to run.

To install all the required packages at once, since it can take some time, we provide the file **requirements.txt** with the list of packages. Then executing the following command will install all the packages one after the other:

- For Windows:

```
FOR /F "delims=~" %f in (requirements.txt) DO conda install --yes "%f" || pip install "%f"
```

Hint: this command must be run from the folder where the **requirements.txt** file is located. Move to the given directory first (e.g. cd Desktop) or just add the path in the command, e.g. FOR /F "delims=~" %f in (Desktop/requirements.txt) DO conda install --yes "%f" || pip install "%f".

2.3. *Necessary instalments*

It is required to have installed:

- Java SDK > 1.8 64-bit → https://www.oracle.com/technetwork/java/javase/downloads/index.html
- Spark > version 2.4.3 → https://spark.apache.org/downloads.html
- Scala (sbt) → https://www.scala-sbt.org/download.html?_ga=2.123928130.935966207.1567084519-2133533685.1567084519

3. INSTALL THE PERFORMANCE SPECTRUM MINER

3.1. System Requirements

- Microsoft Windows 7 or higher. The PSM is *not tested* yet on other OS.
- 2 GB RAM minimum, 16 GB RAM recommended
- 2 GB hard disk space for caches recommended
- 1024x768 minimum screen resolution.

3.2. Prequisites

The PSM is implemented and tested with Java 8 and is not compatible with previous Java version (e.g. with Java 7).

1. Install the most recent JRE/JDK 1.8 64bit
2. Make sure that a correct installation of Java is configured: execute `java -version` in the command line. You should get a response like this:

```
java version "1.8.0_171"

Java(TM) SE Runtime Environment (build 1.8.0_171-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

3.3. Installation of PSM

1. Download the project as a ZIP file from https://github.com/ozgekoroglu/perf_spec
2. Unzip the folder
3. Open the file location in the command prompt with the following command:

   ```
   >cd file_path
   ```

4. Type the following commands respectively:

   ```
   >sbt

   >compile

   >package

   >exit
   ```

```
C:\Users\nlokor\Desktop\perf_spec-master>sbt
Java HotSpot(TM) 64-Bit Server VM warning: Ignoring option MaxPermSize=256m; support was removed in 8.0
[info] Loading global plugins from C:\Users\nlokor\.sbt\1.0\plugins
[info] Loading settings for project perf_spec-master-build from plugins.sbt ...
[info] Loading project definition from C:\Users\nlokor\Desktop\perf_spec-master\project
[info] Updating ProjectRef(uri("file:/C:/Users/nlokor/Desktop/perf_spec-master/project/"), "perf_spec-master-build")...
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies.
[warn] Run 'evicted' to see detailed eviction warnings
[info] Loading settings for project sim_ein from build.sbt ...
[info] Loading settings for project framework from build.sbt ...
[info] Loading settings for project ppm from build.sbt ...
[info] Loading settings for project classifiers_outlier from build.sbt ...
[info] Loading settings for project perf_spec from build.sbt ...
[info] Loading settings for project perf_spec-master from build.sbt ...
[info] Set current project to psm (in build file:/C:/Users/nlokor/Desktop/perf_spec-master/)
[info] sbt server started at local:sbt-server-e6cace6dabe05f8faf5f
```

```
sbt:psm> compile
[info] Updating ...
[info] Updating perf_spec...
[info] Done updating.
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies.
[warn] Run 'evicted' to see detailed eviction warnings
[info] Updating ppm...
[warn] Multiple dependencies with the same organization/name but different versions. To avoid conflict, pick one version
:
[warn]  * org.apache.commons:commons-collections4:(4.0, 4.1)
[info] Updating framework...
[info] Updating classifiers_outlier...
[info] Compiling 85 Scala sources and 24 Java sources to C:\Users\nlokor\Desktop\perf_spec-master\perf_spec\target\scala
-2.11\classes ...
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies.
[warn] Run 'evicted' to see detailed eviction warnings
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies.
[warn] Run 'evicted' to see detailed eviction warnings
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies.
[warn] Run 'evicted' to see detailed eviction warnings
[info] Updating sim_ein...
[warn] Multiple dependencies with the same organization/name but different versions. To avoid conflict, pick one version
```

```
[info] Done compiling.
[success] Total time: 65 s, completed 18-aug-2019 23:32:10
sbt:psm> package
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\target\scala-2.11\psm_2.11-1.1.0.jar ...
[info] Done packaging.
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\classifiers_outlier\target\scala-2.11\classifiers_outlier_2.1
-1.1.0.jar ...
[info] Done packaging.
[warn] Multiple main classes detected.  Run 'show discoveredMainClasses' to see the list
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\ppm\target\scala-2.11\ppm_2.11-1.1.0.jar ...
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\perf_spec\target\scala-2.11\perf_spec_2.11-1.1.0.jar ...
[info] Done packaging.
[info] Done packaging.
[warn] Multiple main classes detected.  Run 'show discoveredMainClasses' to see the list
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\framework\target\scala-2.11\framework_2.11-1.1.0.jar ...
[warn] Multiple main classes detected.  Run 'show discoveredMainClasses' to see the list
[info] Packaging C:\Users\nlokor\Desktop\perf_spec-master\sim_ein\target\scala-2.11\sim_ein_2.11-1.1.0.jar ...
[info] Done packaging.
[info] Done packaging.
```

```
sbt:psm> exit
[info] shutting down server
```

5. Open Intellij. Inside the Intellij, File->Settings-> Build, Execution,Deployment -> Build Tools -> sbt and specify the custom sbt launcher.

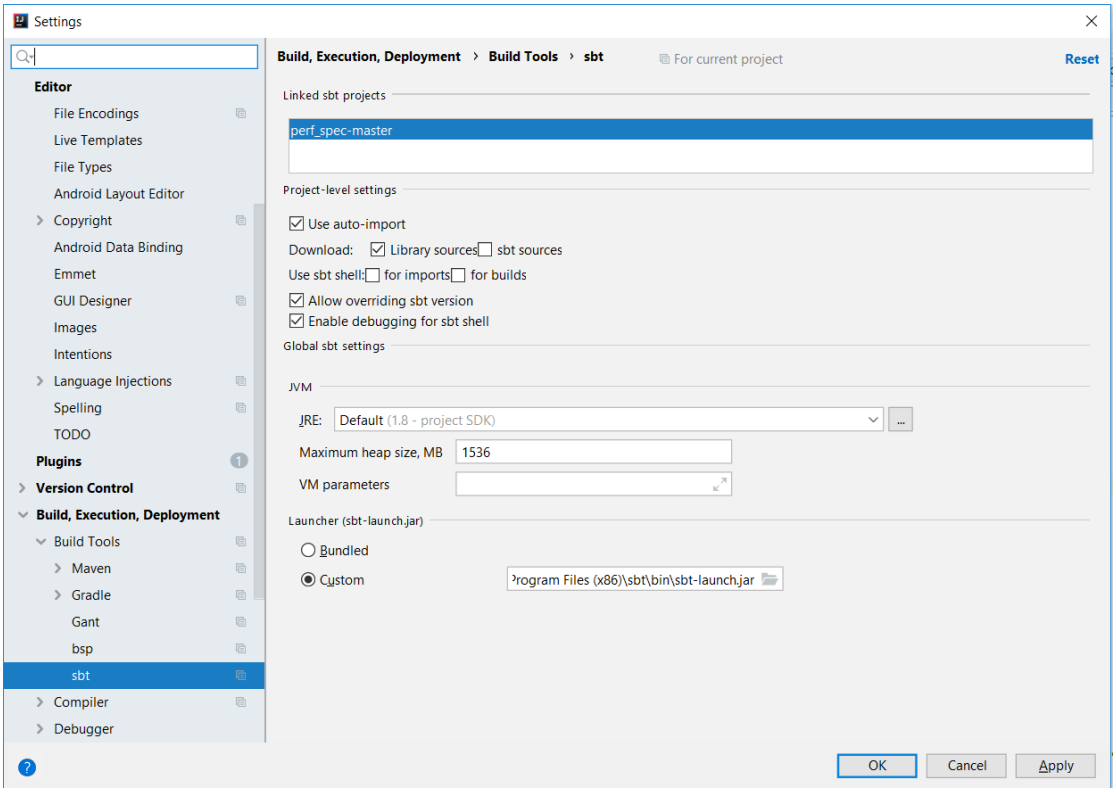


Figure 2: sbt-launch.jar configuration

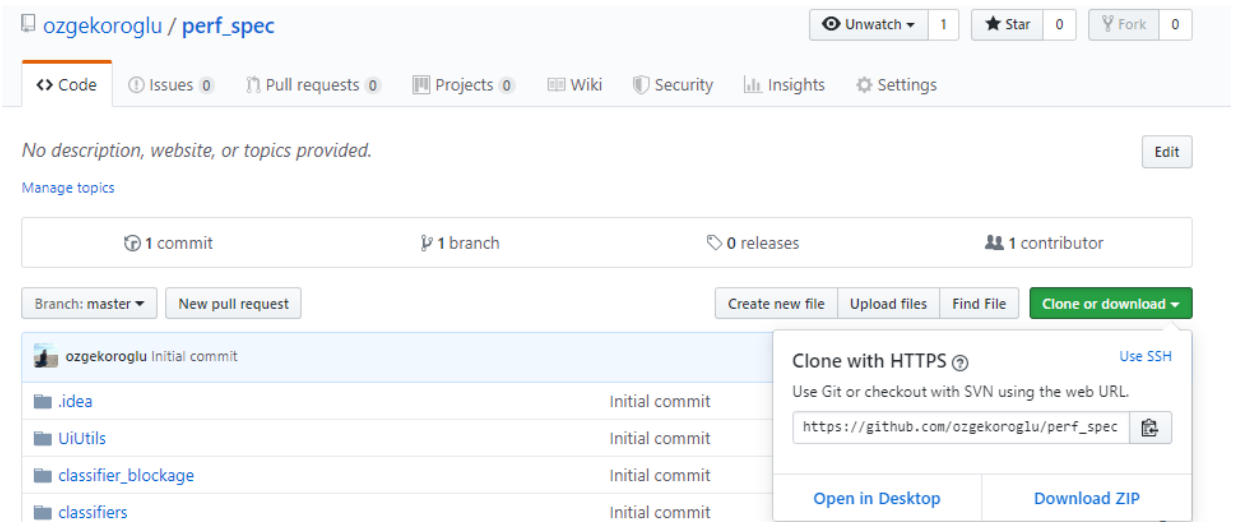


Figure 3: PSM repository

4. INSTALL INTELLIJ

1) From its website, download the community version of Intellij.
2) To create configuration :
    1) Open the Run/Debug Configuration dialog:
        - Select Run | Edit Configurations from the main menu.
        - With the Navigation bar visible (View | Appearance | Navigation Bar), choose Edit Configurations from the run/debug configuration selector.
    2) In the Run/Debug Configuration dialog, click + on the toolbar. The list shows the default run/debug configurations. Select the ***Application*** configuration type.
    3) For a new run/debug configuration:
        - Specify its name in the Name field. This name will be shown in the list of the available run/debug configurations.
        - In the Configuration tab, specify the class that contains the main() method, VM options, program arguments, working directory and other configuration-specific settings.
3) Install scala plugin: Open IntelliJ IDEA, go to File Menu --> Plugins --> [Or directly press Ctrl+Alt+S] Click on "Browse repositories" button and enter "Scala". Select Scala plugin to install it.

5. EXECUTE THE OUTLIER DETECTION ALGORITHM

1) Open the downloaded project (perf_spec-master) in Intellij by selecting **File | Open**.

2) Main event log extraction: The initial dataset (BPI reports for Heathrow T3) is used to extract a log that can be used to extract segments (for a Performance Spectrum (PS)). For example, such a log can contain the bag movement (events from sensors and autoscans). In this step, if the log could not fit the memory then it is separated into smaller files (by days or weeks), for the further pre-processing in memory. We execute ***LogSplitter*** method to obtain the event log from the BPI reports. To do that, we create a configuration with the following arguments:

1) Location of the BPI reports
2) Output of the event log

The arguments should be given respectively with a space in between. The configuration should be saved by clicking Apply. Then we run the configuration by clicking the run icon in the navigation bar.

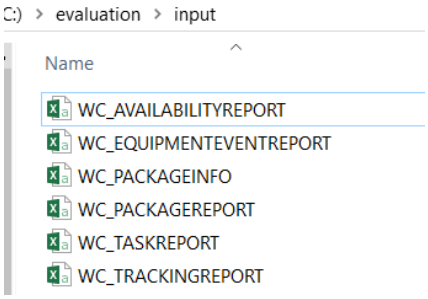


Figure 4: BPI reports

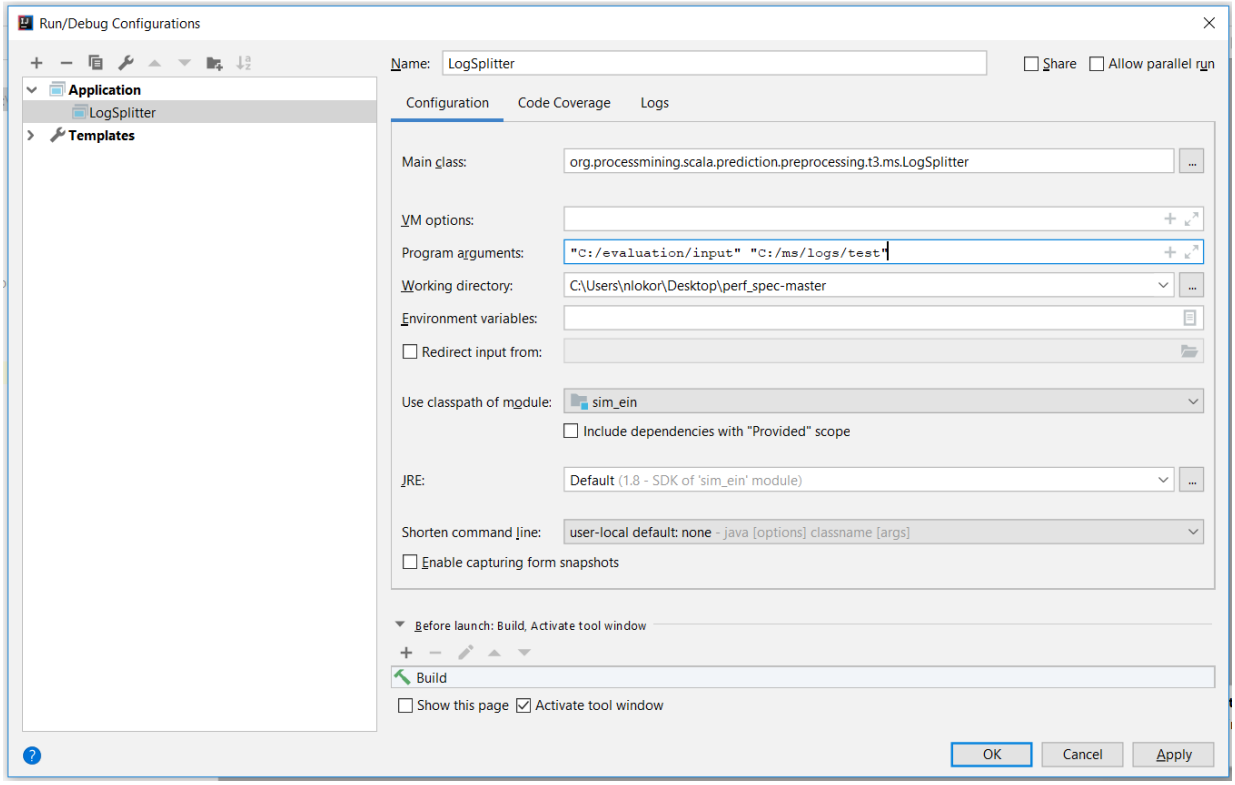
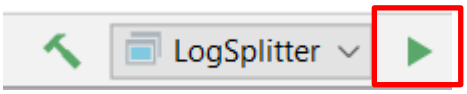


Figure 5: LogSplitter configuration



3) Segment extraction: File by file, segments are extracted, aggregated (if required). Non-relevant segments can be filtered out. We execute the ***T3LogsToSegmentApp*** method with the following parameters : location of the event log that obtained in step 2, aggregator location, location for the output. The arguments should be given respectively with a space in between. The configuration should be saved by clicking Apply. Then we run the configuration by clicking the run icon in the navigation bar.

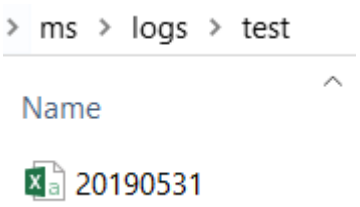


Figure 6: Input of T3LogsToSegmentApp

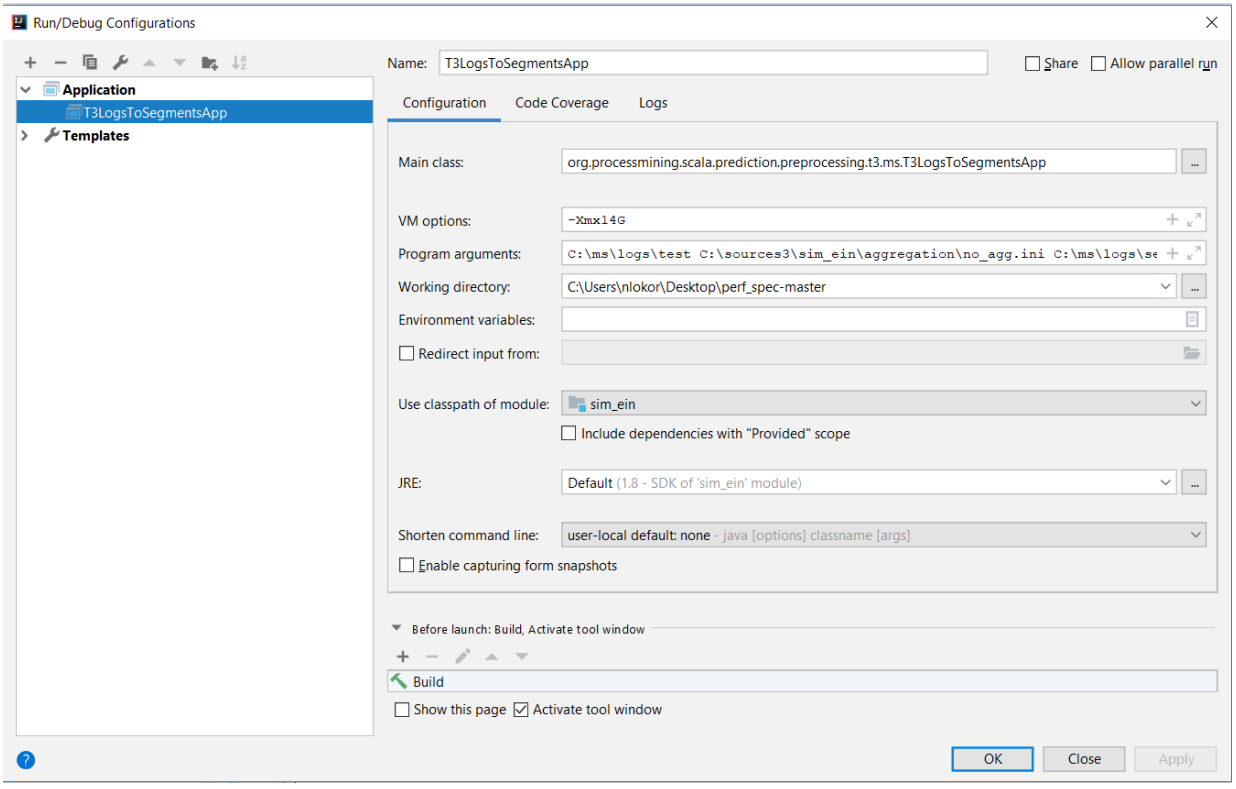
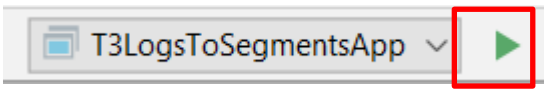


Figure 7: T3LogsToSegmentsApp configuration



4) Download the outlier detection project from https://github.com/ozgekoroglu/Outlier_Detection_BHS as a ZIP file.

5) Unzip the *Outlier_Detection_BHS-master* zip file.

6) Unzip the dist_clusters.rar file

7) (Optional) Execute the to obtain all the clusters that consists of similar days for each segment and weekday. These clusters are representative of the segment behavior. The result for each segment and weekday is stored in the folder called **dist_cluster.** We provide this folder with the results obtained for the period between 29.09.2017 and 30.03.2018 for Heathrow T3. Hence, execution of this step is not necessary unless someone wants to change the baseline for the analysis. If the baseline must be changed than the steps 1, 2, and 3 must be performed again. Execution is performed with following commands in the Anaconda prompt.

```
(outlier_detection) >cd location_of_clustering.py

(outlier_detection) >python clustering.py
```

8) Execute the Outlier_Detection_in_BHS.py with the argument that specifies the location of the segments we obtained in the step 2 with following commands in the Anaconda prompt. As a result of this step we obtain **test_day_weekday.csv** and **weekday.csv** in *analysis_psm_general* and *blockage_psm_general* folders.

```
(outlier_detection) >cd location_of_Outlier_Detection_in_BHS.py

(outlier_detection) >python Outlier_Detection_in_BHS.py location_of_segments
```

9) Open the folder ***psm_outlier***. Put the original event log that had been produced in the step 2 into this folder. Also put the *test_day_weekday.csv* file by changing its name to *outliers.txt.*

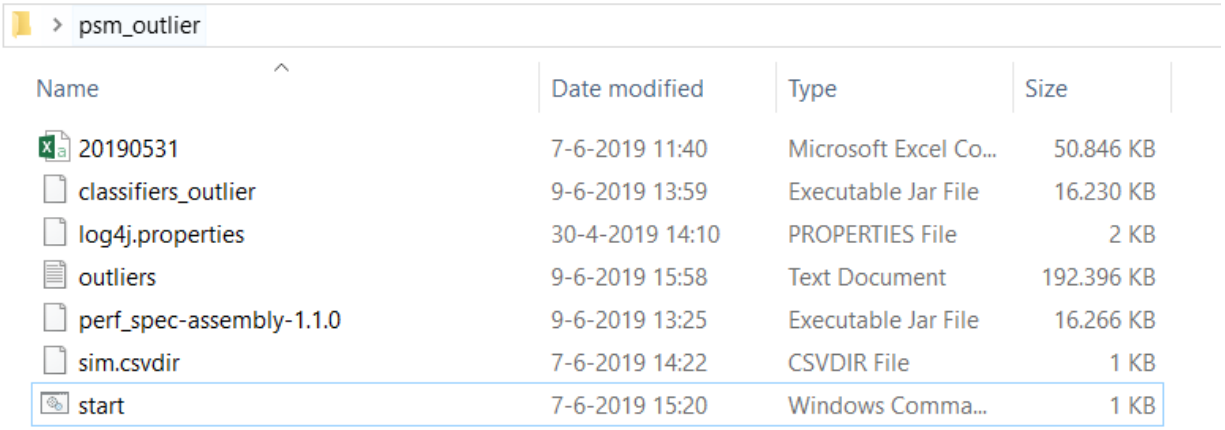


Figure 8: psm_outlier folder

10) To prepare CSV file(s) for import, put the file(s) into a directory and provide a description as a text ini file with extension .csvdir. This file must include the following fields:

Field	Sample value	Comment
dateFormat	dd-MM-yy HH:mm:ss.SSS	Datetime format in Java DateTimeFormatter format
zoneId	Europe/Amsterdam	Time zone ID in Java ZoneId format
startTime	31-05-19 00:00:00.000	Since then the performance spectrum should be computed, in the format described above
endTime	01-06-19 00:00:00.000	Until then the performance spectrum should be computed, in the format described above

caseIdColumn	id	Column name for case ID
activityColumn	activity	Column name for activity
timestampColumn	timestamp	Column name for timestamp

Table 1: Content of .csvdir file

11) Execute ***start.cmd*** to open the Performance Spectrum Miner.
12) Import the .csvdir file via the *Open...* button.

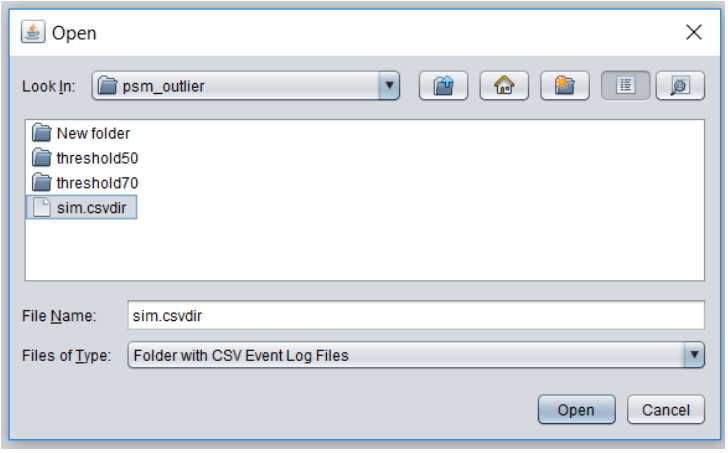


Figure 9: Importing .csvdir file into PSM

13) Choose parameters for generating the performance spectrum data.
    - The transformed data will be stored on disk in the *Intermediate storage directory* together with a meta-data file (session.psm). You can load this transformed data also later via the *Open...* button.
    - Type the following into the Custom Classifier Section. This classifier separates the outlier types in the performance spectrum miner.

      org.processmininginlogistics.classifiers.bp.example.SegmentClassifierExample

    - Choose *Process & open*
    - The transformation may require some time and main memory depending on the *Bin size* chosen. Transformation for larger bin sizes are faster and require less memory.

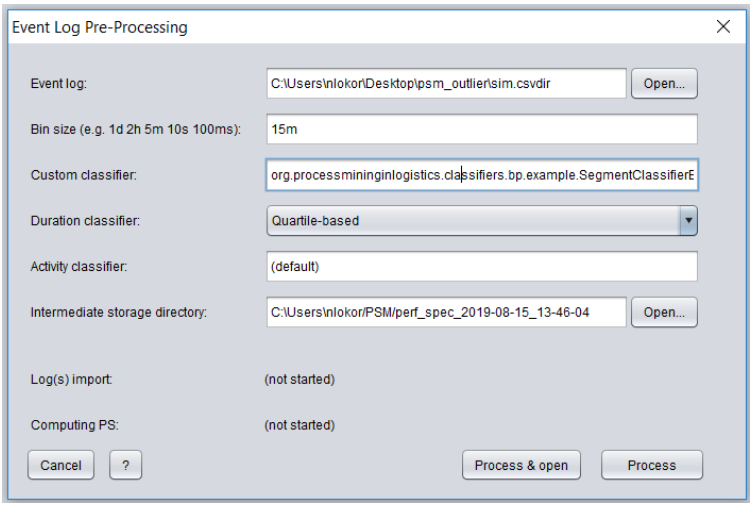


Figure 10: Event log pre-processing parameter selection

14) Click open in the *open pre-process dataset* window.

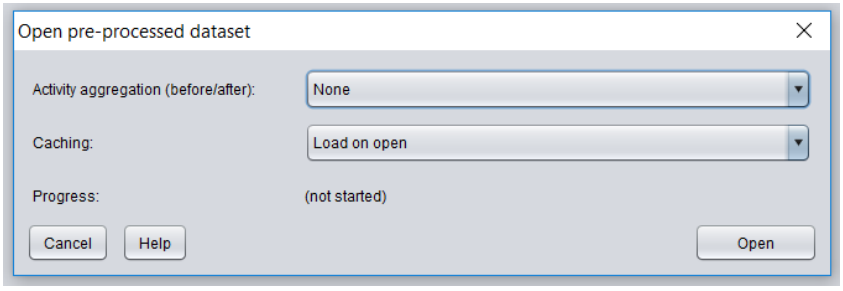


Figure 11: Opening pre-processed dataset

15) Close the PSM.

16) Open the *Intermediate storage directory* for the performance spectrum that was obtained in the step 13. The **intermediate storage directory** is a path to an empty or non-existing folder where the performance spectrum data of the imported event data is stored. (Refer to user manual for the Performance Spectrum Miner for further information)

17) Add a config.ini file with the following content:

    [GENERAL]
    paletteId = 4

18) Copy the sorting_order.txt file from the repository and paste it in the *Intermediate storage directory*.

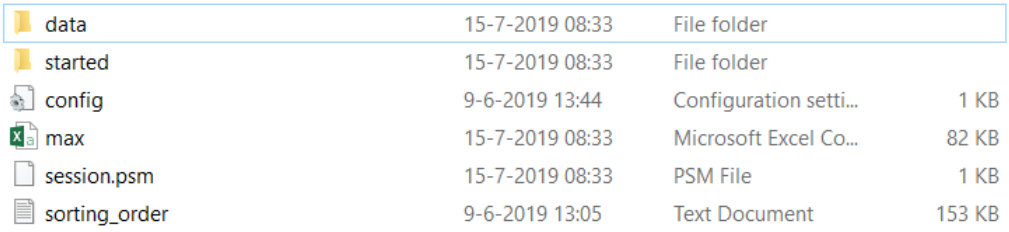


Figure 12: Intermediate storage directory

19) Go to the psm_outlier folder and execute start.cmd again. But, this time select the session.psm in the *Intermediate storage directory*.

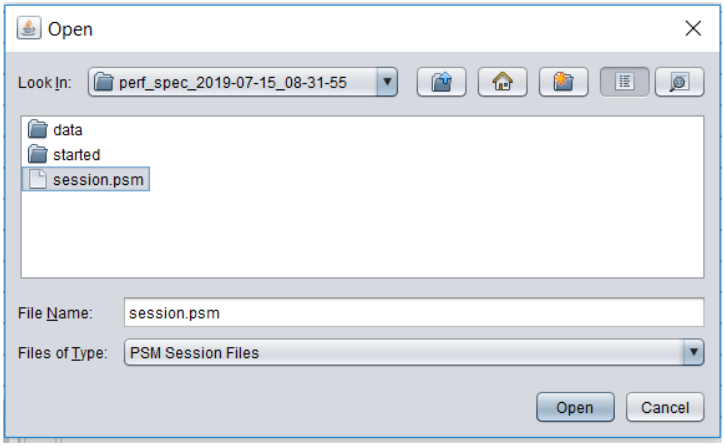


Figure 13: Opening session.psm

20) Explore the performance spectrum miner. For further information refer to the user manual for the PSM . Legend button shows the colors for different outlier types.

21) To see the problematic segments click on the *Legend* button and close it. A separate window is opened after closing the *Legend* window. To see the segments that behaved in their worst behavior select 1 for both comboboxes in the red rectangular and sort the values according to the importance variable.

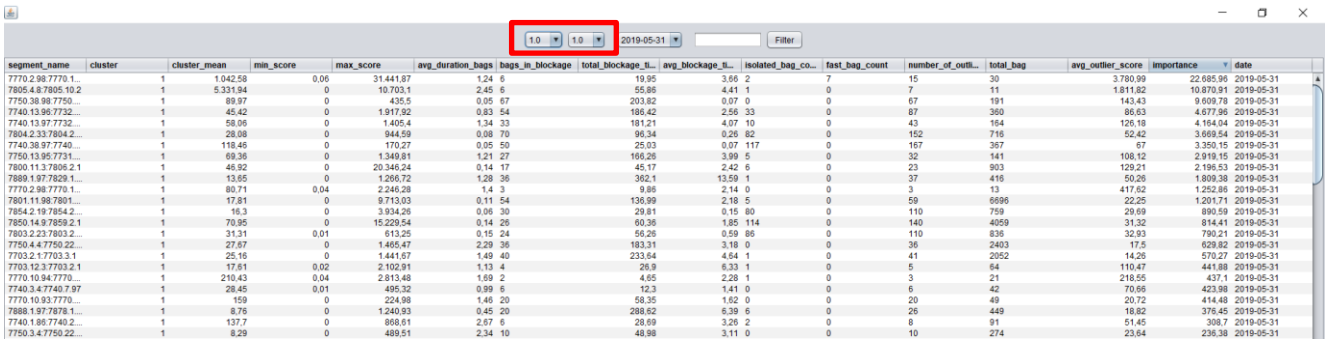
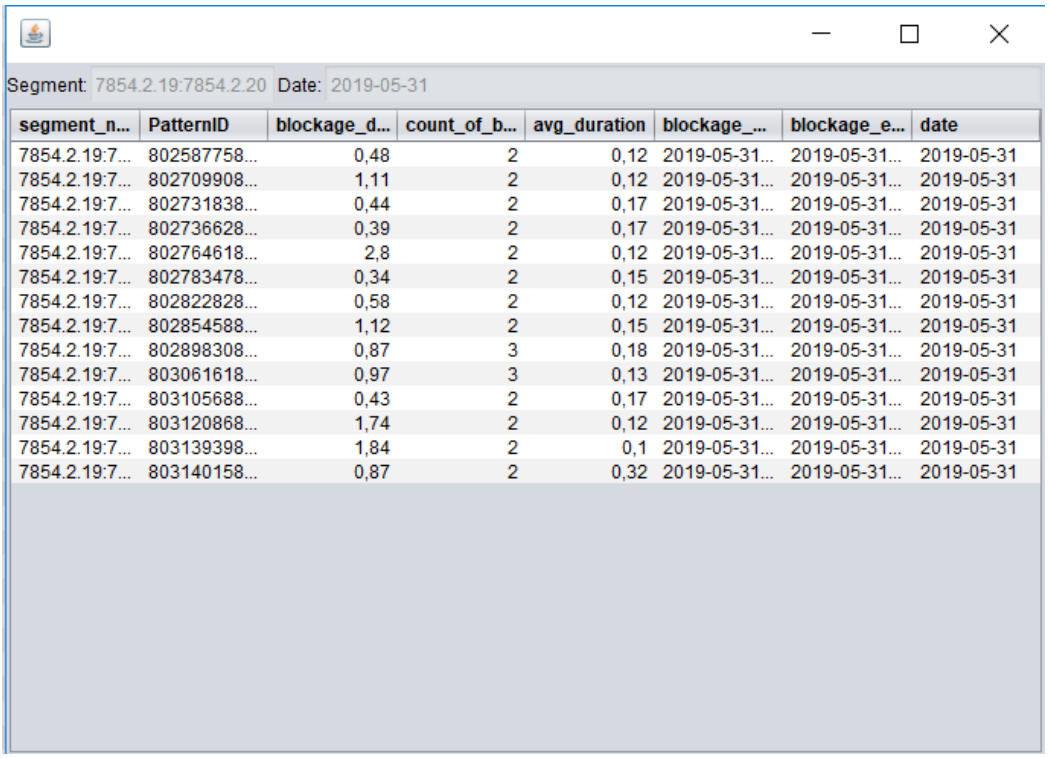


Figure 14: Visual Dashboard for Problematic Segments

22) By clicking on any line, one can see the blockages regarding the segment for the given day in the blockage window.



segment_n...	PatternID	blockage_d...	count_of_b...	avg_duration	blockage_...	blockage_e...	date
7854.2.19:7...	802587758...	0,48	2	0,12	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802709908...	1,11	2	0,12	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802731838...	0,44	2	0,17	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802736628...	0,39	2	0,17	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802764618...	2,8	2	0,12	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802783478...	0,34	2	0,15	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802822828...	0,58	2	0,12	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802854588...	1,12	2	0,15	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	802898308...	0,87	3	0,18	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	803061618...	0,97	3	0,13	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	803105688...	0,43	2	0,17	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	803120868...	1,74	2	0,12	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	803139398...	1,84	2	0,1	2019-05-31...	2019-05-31...	2019-05-31
7854.2.19:7...	803140158...	0,87	2	0,32	2019-05-31...	2019-05-31...	2019-05-31

Figure 15: Blockage window

23) By clicking on any of the blockages, the user can see the selected blockage on the performance spectrum miner.

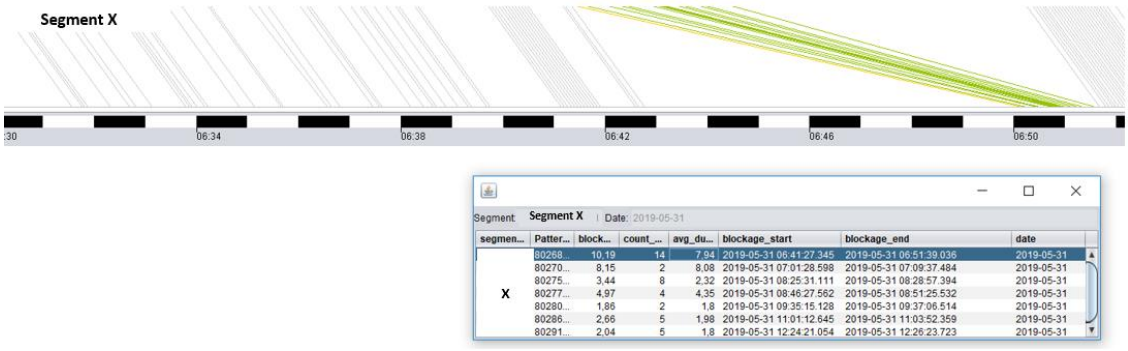


Figure 16: Blockage visualization on the PSM