**X.X Approaches to automatically identify and isolate differently behaving assets in pre-processing steps of model building**

# **X.X.1 Objective**

During the journey of POE defect prediction model, it has been observed that few POE assets behave differently from the rest of the asset list. When visualized, the traces from these assets were found to be different from rest of the assets. This was also evident after observing the Isolation Forest (one class model) results where in for these differently behaving assets many traces were marked as anomalous by the model. This resulted in a drop in the performance of the current solution. So, the objective was to automatically identify and isolate these assets in the pre-processing step (before training the model) so that model performance can be improved. After isolating these differently behaving assets, the final model was be built/trained on the assets which had similar types of traces.

**X.X.2 Why have we considered this approach?**

Previously, the process of identification and isolation of differently behaving assets was done manually by inspecting and comparing the traces of every asset in pre-processing steps. This manual method presented certain challenges for the team:

* Time consuming – The manual trace inspection and identifying which assets are behaving differently was a time-consuming process.
* Human biasness - there was a potential human error and biasness while checking the traces manually
* Not scalable – The current manual process was not scalable for a large number of assets

As such, an automatic process of identifying and isolating differently behaving assets from existing pool of assets was ideated and implemented.

This automated method was considered in pre-processing step. There were no changes to the logic of extracting the traces, segmentation of traces, feature extraction and modelling logic.

**X.X.3 Example of differently behaving traces from different assets**

For example, fig X.X.3.a plot shows few traces from different Clamp Lock Mark 2 assets on trace direction (NR & RN) where green colour traces represent RN trace and red traces represents NR trace. The Y- axis of the plot represents the current in amperes and X- axis represents the swing time. The traces of these assets (in this plot) exhibit different behaviour from rest of the assets. There are many other assets show different behaviour, however, showing few assets for visualisation purpose.

Chart

Description automatically generated

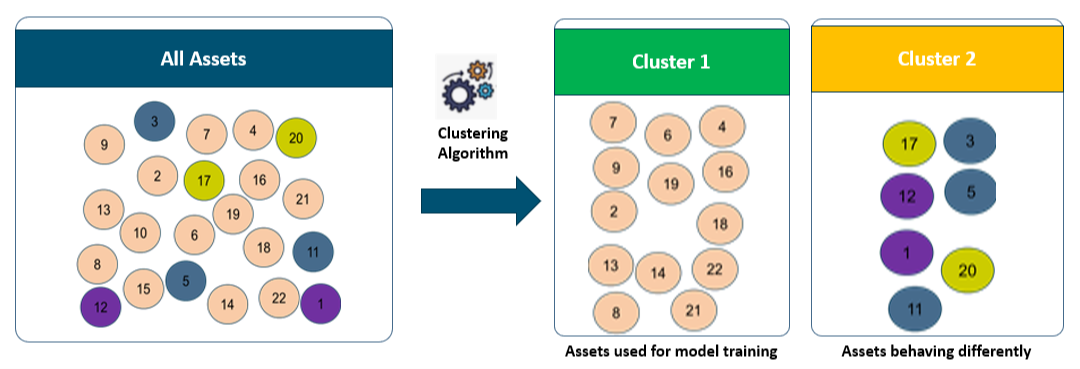
**Fig X.X.3.a: Example of traces with different behaviours**

**X.X.4 Solution to automatically identify and isolate differently behaving assets**

To overcome the challenges of manual inspection, various clustering-based algorithms were considered and finally, Meanshift, one of the clustering-based algorithms was selected to replace the manual process. Meanshift is an unsupervised learning algorithm that assigns the data points to the clusters iteratively by shifting points towards the mode (mode is the highest density of data points in the region, in the context of the Meanshift). The selected clustering-based algorithm was used in pre-processing step and would help in clubbing similarly behaving assets in one cluster and isolating different behaving assets in a different cluster.

Fig. X.X.4.a depicts the approach where ‘Meanshift algorithm’ has been applied on a master asset group and the selected algorithm automatically separate the assets into two groups based on their behaviour. Meanshift algorithm works at asset level. For this, first features were extracted from traces and then features were aggregated (average) at asset level. The algorithm differentiates between ‘differently behaving’ and ‘anomalous’ assets based on the following features:

1. 'SwingTime',
2. ‘First\_phase\_min\_current\_deviation’,
3. 'midcurrent\_deviation',
4. 'MidMax\_Current\_dev\_from\_7amp'

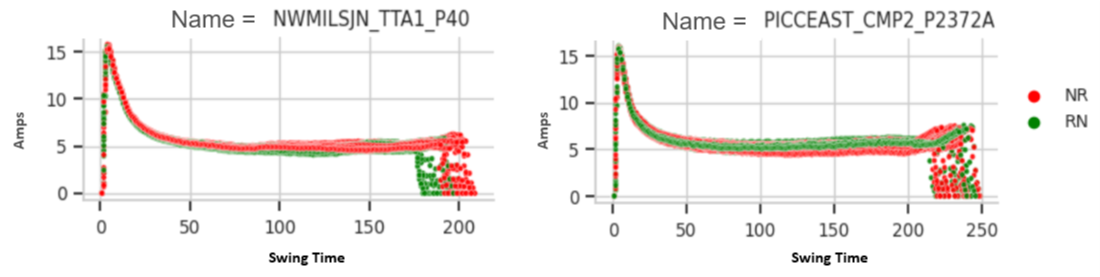


**Fig X.X.4.a: Example of a clustering-based approach to separate the groups based on the behaviour of assets**

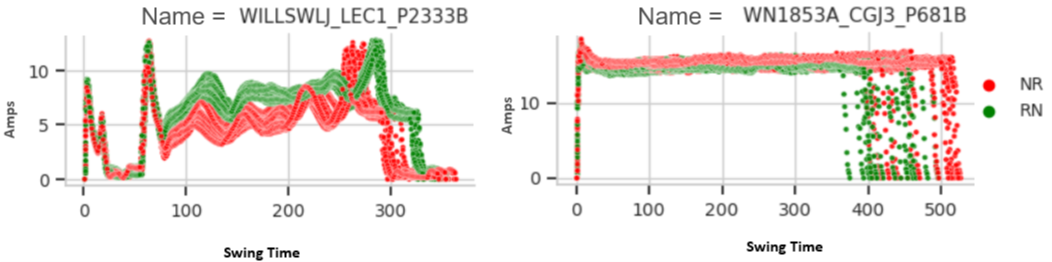
After clustering, differently behaving assets were isolated and the remaining assets were used for training the model.

**X.X.5 Examples of assets in two clusters after applying clustering approach**

After using the clustering-based approach on a group of Clamp Lock Mark 2 assets, two clusters were created and the assets were distributed into those clusters based on the selected features. Fig X.X.5.1.a and Fig X.X.5.1.b plots shows an example of traces of those assets belonging to those two clusters. Fig X.X.5.1.a plot shows the traces of those assets which are in cluster 1 and exhibit similar trace pattern as traces from majority of the assets. Fig X.X.5.1.b plot shows the traces of those assets which were isolated as they were behaving differently from rest of the assets.



**Fig X.X.5.1.a: Example of traces of assets in cluster 1**

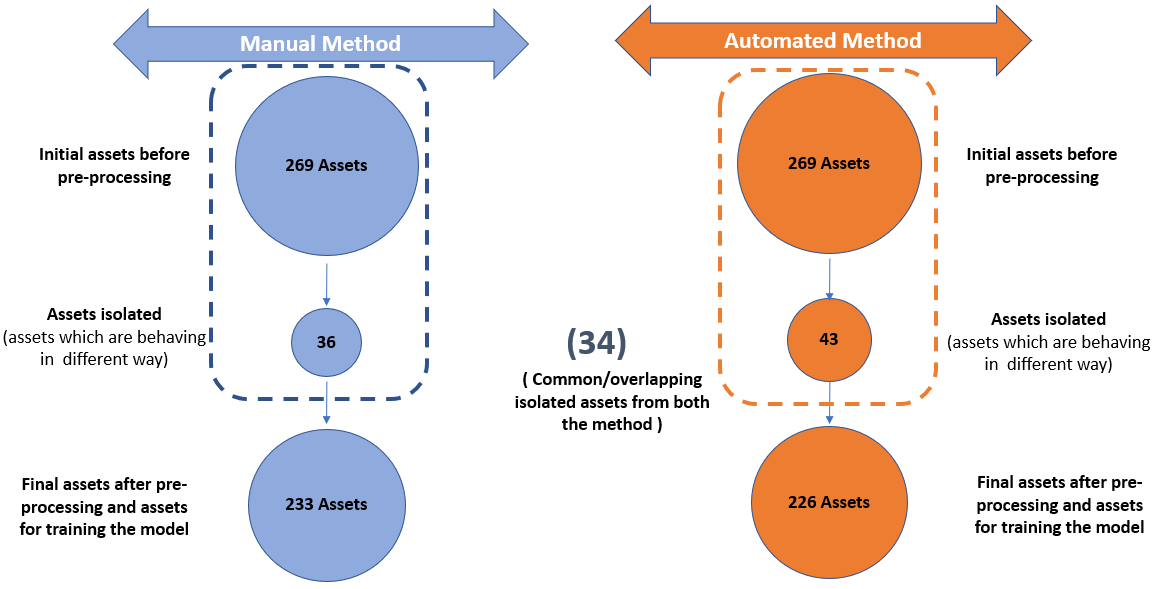


**Fig X.X.5.1.b: Example of traces of assets in cluster 2**

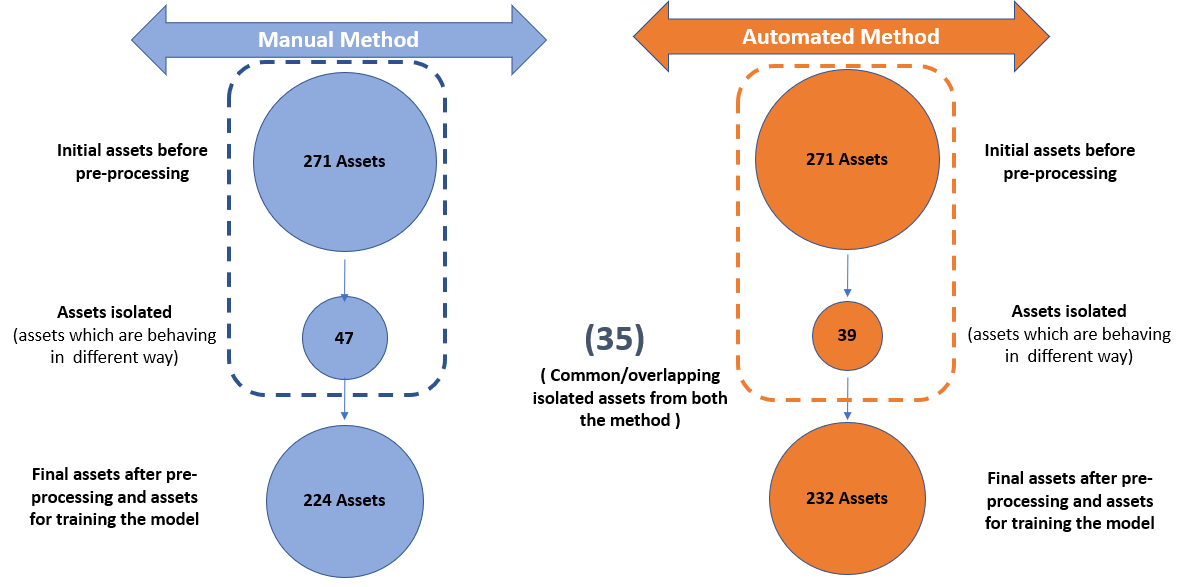
**X.X.5.2 Result comparison between the number of assets isolated by manual and automated (clustering-based approach) method**

For linearity experiment (in section 9.8 in the master solution document), 805 assets from the Clamp Lock Mark 2 assets were divided into three groups. The same three groups and their corresponding assets were selected to proceed with this experiment. Fig X.X.5.2.c. X.X.5.2.d and X.X.5.2.e plots shows the comparison of assets isolated using manual and automated methods and how the selected assets from each group of Clamp Lock Mark-2 assets were used to train the finalised model.

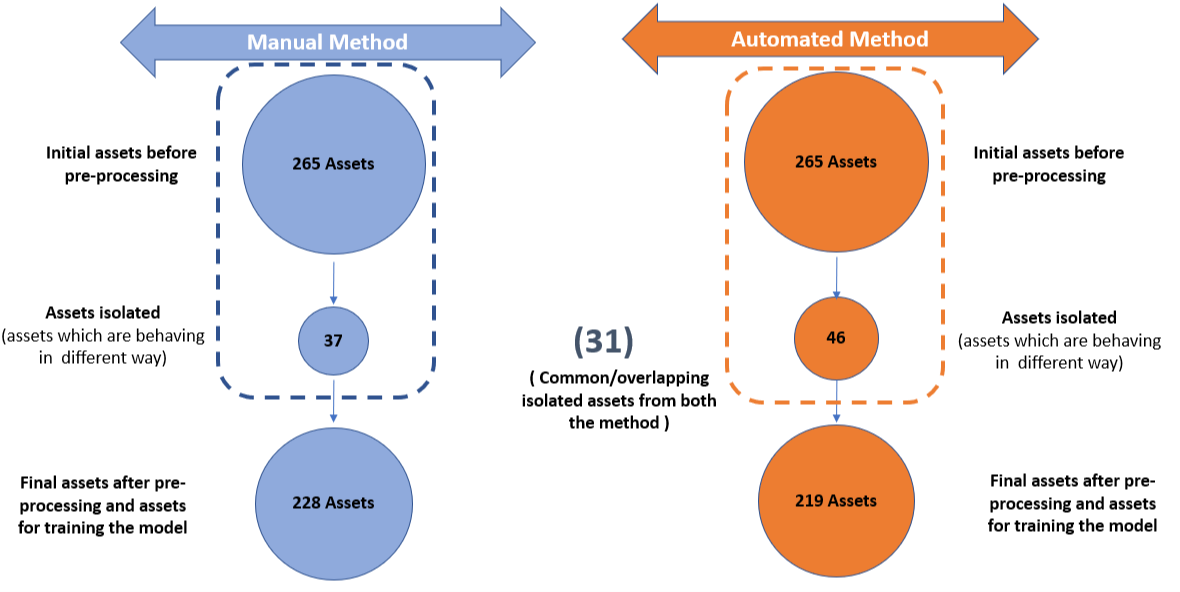
As illustrated in fig X.X.5.2.c, same number of assets (269 assets) were used with both the methods. However, in the manual method a total of 36 assets were isolated and 233 assets were used to train the model whereas in case of automated method, 43 assets (7 more than the manual method) were isolated and a total of 226 assets were used to train the model.

****

**Fig X.X.5.2.c: Comparison of manual and automated method in terms of assets used in model training on Clamp Lock Mark 2 group 1 assets**

****

**Fig X.X.5.2.d: Comparison of manual and automated method in terms of assets used in model training on Clamp Lock Mark 2 group 2 assets**

****

**Fig X.X.5.2.e: Comparison of manual and automated method in terms of assets used in model training on Clamp Lock Mark 2 group 3 assets**

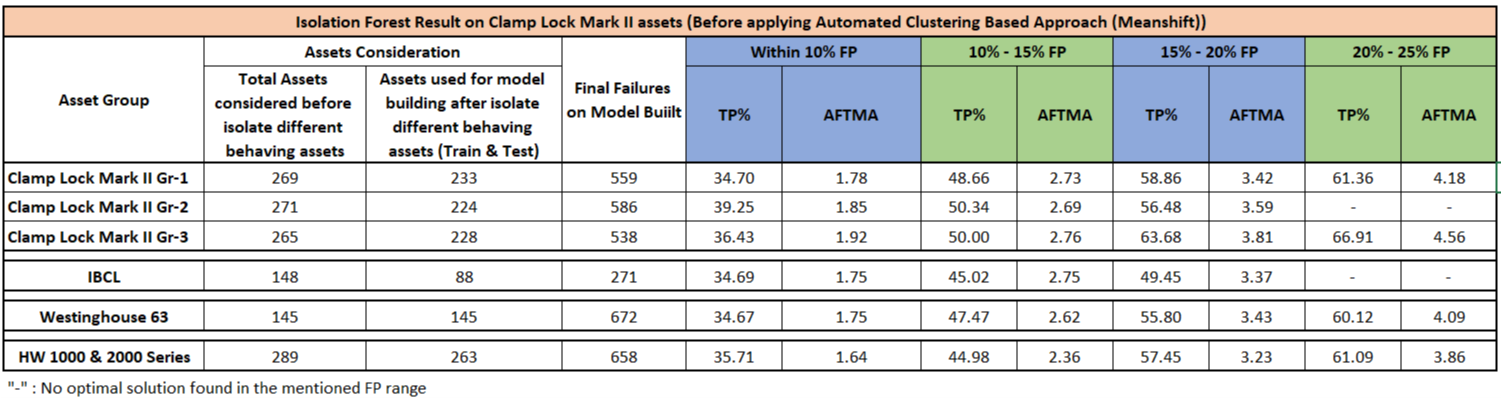
**X.X.6 Comparison of model result before and after using the clustering method**

After isolating the differently behaving assets the machine learning model (Isolation Forest) was trained on selected assets.

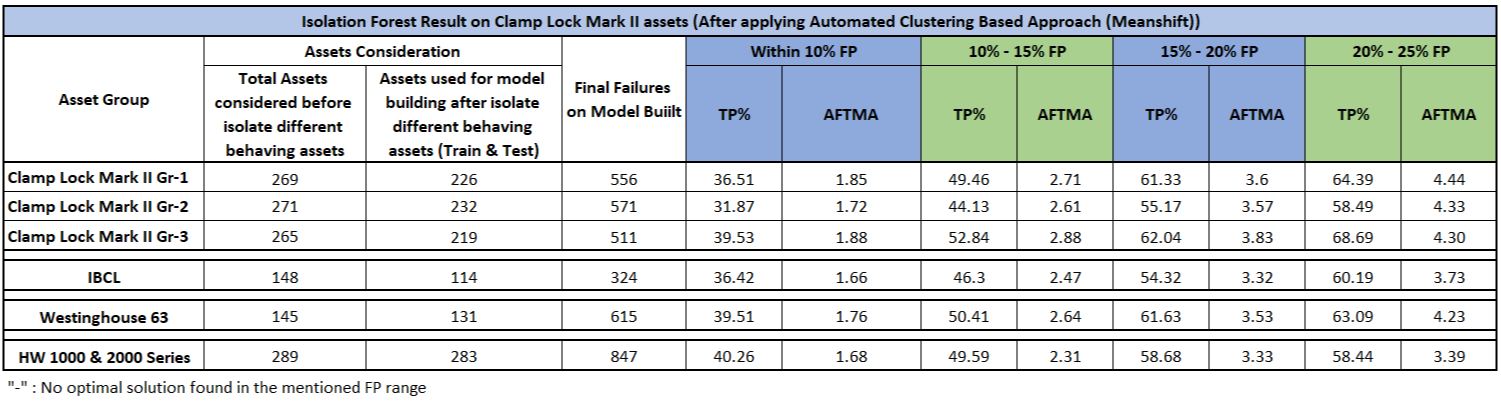
Table X.X.6 and X.X.7 shows the model performance before and after applying clustering algorithms on different POE types - Clamp Lock Mark 2, IBCL, Westinghouse 63 and HW 2000 Series.

The tables below capture the performance metrics - True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA) within specified range of False Positive Percentage (FP%) cut-off.

**Table X.X.6: Model performance on different groups of assets before isolating the differently behaving assets**



**Table X.X.7: Model performance on different groups of assets after isolating the differently behaving assets**

****

**X.X.7 Findings and Next Steps**

The Isolation Forest model performance results (X.X.7) were mostly consistent with the previous results (X.X.6). A dip in performance for Clamp Lock Mark 2-Gp 2 assets was noted and applicable root cause analysis is being done to better understand this difference.