A picture containing person, orange, outdoor, man

Description automatically generated

Network Rail II R&D Workstream CF10 - Predict & Prevent

Point Operating Equipment

Failure Prediction Solution Document

A close up of a logo

Description automatically generated

Document control

|  |  |
| --- | --- |
| Document control | |
| Name of the document | Point Operating Equipment Failure Prediction solution |
| Version | V 5.0 |
| Author | R&D team |
| Date created | 13-Oct-2020 |
| Date Last amended | 13-July-2021 |

**Approvals / Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Name | Description |
| 0.1 DE | 13-Oct-2020 | Sankalp | DE input |
| 0.1 DE BA | 13-Oct-2020 | Aniket | + BA Input |
| 0.2 | 14-Oct-2020 | Gerard | Review GC |
| 0.5 | 16-Oct-2020 | Debasis, Bhargav | + DS Input |
| 0.6 | 16-Oct-2020 | Pawel | + DS Input |
| 0.7 | 17-Oct-2020 | Gerard | Review |
| 0.8 | 19-Oct-2020 | Aniket, Pawel, Gerard | + BA & DS Input, Review |
| 0.9 | 20-Oct-2020 | Pawel, Aman | + DS Input |
| 0.10 | 21-Oct-2020 | Gerard | Review |
| 0.11 | 21-Oct-2020 | Sankalp | Reordering |
| 0.12 | 21-Oct-2020 | Gerard | Review |
| 0.13 | 22-Oct-2020 | Gerard | Input from Pawel, Aman |
| 0.14 | 24-Oct-2020 | Viv | Review |
| 0.15 | 30-Oct-2020 | Vinayak | Review |
| 2.1 | 25-Nov-2020 | Gerard | Review |
| 2.2 | 25-Nov-2020 | Debasis | + DS Input |
| 2.3 | 27-Nov -2020 | Debasis | + DS & DE Input |
| 2.4, 2.5 | 27/28-Nov-2020 | Gerard | Review & Formatting |
| 2.8 | 29-Nov-2020 | Gerard | Incl. Steve C comments |
| 2.9 | 3-Dec - 2020 | Sauryya | Updated table 4.3 |
| 3.0 | 4-Dec-2020 | Viv | Review |
| 4.0 | 04-May-2021 | Saket M | Included Phase 3 + with HW series changes |
| 5.0 | 19-Aug-2021 | Saket M | Included Clamp Lock Mark 2 linearity test |
| 6.0 | 16-Mar-2022 | John | POE New User Stories document(link) added in reference |
| 7.0 | 05-May-2022 | John | POE Physical attribute analysis Solution document(link) added in reference |
| 7.0 | 05-Aug-2022 | John | POE New Enhancement Work 1 document(link) added in reference |

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Document name | Location | Version |
| 1 | FMECA\_CF10\_Show & Tell\_V1.2 |  |  |
| 2 | FMECA\_CF10\_Show & Tell\_V1.3.1 |  |  |
| 3 | FMECA\_CF10\_Show & Tell\_V1.4 |  |  |
| 4 | FMECA\_CF10\_Show & Tell\_V2.1.1 |  |  |
| 5 | FMECA\_CF10\_Show & Tell\_V2.2.2 |  |  |
| 6 | FMECA\_CF10\_Show & Tell\_V2.3.1 |  |  |
| 7 | POE New User Stories | [Userstory\_01\_documentation](https://networkrail.sharepoint.com/:f:/r/sites/IIAnalyticsWorkstream-Innovation/Shared%20Documents/Innovation/POE/POE_userstories/Userstory_01_documentation?csf=1&web=1&e=VnSEe9) |  |
| 8 | POE Physical attribute anlaysis | [POE Physical attribute analysis](https://networkrail.sharepoint.com/:f:/r/sites/IntelligentInfrastructure/Shared%20Documents/General/II%20R%26D%20team/Solution%20Document/POE%20Physical%20attribute%20analysis?csf=1&web=1&e=KPjXXx) |  |
| 9 | POE New Enhancement Work 1 | [POE New Enhancement Work 1](https://cognizantonline.sharepoint.com/:f:/r/sites/NRCF10InnovationData-PreventionPredictionCognizant/Shared%20Documents/General/Solution%20Document/POE%20New%20Enhancement%20Work%201?csf=1&web=1&e=twC9pb) |  |

**References**

**Glossary of terms, definition and acronyms**

|  |  |
| --- | --- |
| Term | Definition |
| POE | Point Operating Equipment |
| TC | Track Circuit |
| NLTK | Natural Language Tool Kit |
| EMGTPA | Equivalent Million Gross Tonnes Per Annum |
| ADF | Azure Data Factory |
| R&D | Research and Development |

# **Contents**

[Contents 4](#_Toc71613931)

[1 II R&D background and document summary 6](#_Toc71613932)

[1.1 Innovation stream aim & objectives 6](#_Toc71613933)

[1.2 Point operating equipment problem statement 7](#_Toc71613934)

[1.3 Success criteria 7](#_Toc71613935)

[1.4 Document scope 7](#_Toc71613936)

[1.5 Approach for point operating equipment use case 7](#_Toc71613937)

[1.6 Underlying assumptions 8](#_Toc71613938)

[1.7 Assets in scope 8](#_Toc71613939)

[1.8 Out of scope 9](#_Toc71613940)

[1.9 Phase 1 results 9](#_Toc71613941)

[1.10 Phase 2 update 10](#_Toc71613942)

[1.11 Phase 3 update 10](#_Toc71613943)

[1.12 Summary of business benefits delivered 11](#_Toc71613944)

[2 Business understanding - Point operating equipment 12](#_Toc71613945)

[2.1 Point operating equipment explanation 12](#_Toc71613946)

[2.2 Underlying assumptions 14](#_Toc71613947)

[2.3 Traces 14](#_Toc71613948)

[2.4 Model features 14](#_Toc71613949)

[3 Solution approach 15](#_Toc71613950)

[4 Data requirements and data ingestion 16](#_Toc71613951)

[4.1 Data architecture : dev environment 16](#_Toc71613952)

[4.2 Data sources used 17](#_Toc71613953)

[4.3 Data coverage 18](#_Toc71613954)

[4.4 Data engineering phase 2 18](#_Toc71613955)

[4.5 Data Engineering Phase 3 22](#_Toc71613956)

[4.5.1 Grouping Parameters 22](#_Toc71613957)

[4.5.2 Workorder data 22](#_Toc71613958)

[4.5.3 IBCL & Westinghouse 24](#_Toc71613959)

[4.5.4 Weather data - Ingestion pattern 27](#_Toc71613960)

[4.5.5 Failure Data for HW Assets & In-Order Data– ingestion pattern 29](#_Toc71613961)

[5 Data understanding 31](#_Toc71613962)

[5.1 Quality of data 31](#_Toc71613963)

[5.1.1 Incomplete data 31](#_Toc71613964)

[5.1.2 Shift in range of current 31](#_Toc71613965)

[5.1.3 Identical time stamps 32](#_Toc71613966)

[5.2 Phase 3 – Stage 1 - Data understanding & Quality 34](#_Toc71613967)

[5.2.1 Context of the assets 34](#_Toc71613968)

[5.2.2 Context of failures (on 94 assets) 34](#_Toc71613969)

[5.3 Phase 3- Stage 2 - Data understanding on Grouping Parameters 35](#_Toc71613970)

[5.3.1 S&C Turnout grouping assets (Grouping Parameters) 35](#_Toc71613971)

[6 Data preparation 37](#_Toc71613972)

[7.1 Phase 3 Data Preparation 38](#_Toc71613973)

[7 Exploratory data analysis 40](#_Toc71613974)

[7.1 Phase 1 and Phase 2 Changes 40](#_Toc71613975)

[7.2 Phase 3 41](#_Toc71613976)

[8 Feature engineering 43](#_Toc71613977)

[8.1 Feature creation 43](#_Toc71613978)

[8.2 Phase 1 final features selection 43](#_Toc71613979)

[8.3 Phase 2 update - final features selection 44](#_Toc71613980)

[8.4 Phase 3-Final feature selection 45](#_Toc71613981)

[9 Model building and evaluation of model performance 47](#_Toc71613982)

[9.1 Modelling approach 47](#_Toc71613983)

[9.2 Evaluation of model performance 47](#_Toc71613984)

[9.3 Phase 3 Updates 52](#_Toc71613985)

[9.3.1 Stage 1 Model validation on additional assets (94 assets) 52](#_Toc71613986)

[9.3.2 Stage 1 - Model Comparison (52 Vs 94 Assets) 52](#_Toc71613987)

[9.3.3 Model building approach 53](#_Toc71613988)

[9.3.4 HW 1000 and 2000 series changes 60](#_Toc71613989)

[**9.3.4.1 Data understanding & Quality of data** 60](#_Toc71613990)

[**9.3.4.2 Context of failures (on 289 assets)** 60](#_Toc71613991)

[**9.3.4.3 Model Performance** 61](#_Toc71613992)

[**9.3.4.4 Comparison of results on Clamp Lock Mark II Vs HW 1000 & 2000 series assets** 62](#_Toc71613993)

[10 Appendices 63](#_Toc71613994)

[10.1 Data engineering in detail 63](#_Toc71613995)

[10.1.1 Source & target location 63](#_Toc71613996)

[10.1.2 Wonderware data - ingestion pattern 64](#_Toc71613997)

[10.1.3 Detailed Wonderware data ingestion pipeline 66](#_Toc71613998)

[10.1.4 Wonderware data volume & growth 67](#_Toc71613999)

[10.1.5 Technology landscape/environment - Wonderware 68](#_Toc71614000)

[10.1.6 Azure components for data ingestion 68](#_Toc71614001)

[10.2 Data engineering phase 2 update 68](#_Toc71614002)

[10.2.1 Codes for in order date 68](#_Toc71614003)

[10.2.2 How to find in order date in ADS foundation 69](#_Toc71614004)

[10.2.3 Execution times - in order date ADF pipeline 69](#_Toc71614005)

[10.2.4 ADF resource consumption: in order date 70](#_Toc71614006)

[10.2.5 Technology landscape/environment – in order date: 70](#_Toc71614007)

[10.2.6. Azure components for data ingestion 71](#_Toc71614008)

[10.3 Data Engineering Phase 3 71](#_Toc71614009)

[10.4 Data Science 72](#_Toc71614010)

[10.3.2 Discrepancies (fault resolved and in order dates) 73](#_Toc71614011)

# **1 II R&D background and document summary**

The Research & Development (R&D) Innovation stream is a key component of the NR Intelligent Infrastructure (II) programme. The R&D programme is focused on using Advanced Analytics and Machine Learning (ML) POC's to improve the prediction and prevention of faults. The overall approach is Use Case based with initially a maximum of 6 Use Cases in scope, now extended to 8. Several of the Use Cases have a number of Phases. This document covers Phase 1, Phase 2 and Phase 3 changes (HW series changes, model performance experiment) of the Point Operating Equipment use case. Those business users familiar with POE Phase 1 solution document may wish to jump to section 1.9 Modelling Results. Colleagues who are engaged on converting model output to Prototype and Production may wish to go to section 4, Data Requirements and Data Ingestion.

## **1.1 Innovation stream aim & objectives**

**Aim**

* To refine and build fault prediction analytics products for Track & Signalling workstreams.

**Objectives:**

* Deliver analytics solutions for the Track and Signalling workstreams to enable maintenance managers to determine an intervention based on a prediction within the Decision Support Tool (DST).
* To predict key faults that will support making the correct intervention to reduce delays, while ensuring that the safety standards are met.
* Deliver innovative R&D to both Track and Signalling business workstreams.
* Deliver proof of concept (POC) outcomes to inform Track and Signalling workstreams with the ‘Art of the Possible’.

## **1.2 Point operating equipment problem statement**

To predict POE failures so that proactive intervention can be planned before potential disruption in service occurs. Prediction on failure modes for POE ideally needs to be made at least 2 days ahead of its failure to ensure that the intervention can be planned safely. POE assets for which Wonderware data is available were considered for this exercise.

## **1.3 Success criteria**

The initial success criterion of the Use Case was that POE failures should be predicted ideally at least 2 days ahead of its failure. This was modified following initial exploratory analysis to the following:

* Predictions, particularly those 2 weeks or more in advance, should be better than the existing alerts and alarms that are configured using Wonderware data.

## **1.4 Document scope**

The scope of this document is to present the methodology and results from POE Phase 1 (to 28th September 2020), POE Phase 2 (28th September to 13th November) and Phase 3 (ongoing)

## **1.5 Approach for point operating equipment use case**

As soon as the Innovation workstream began, Programme Management Team organised a series of workshops for key Subject Matter Experts (SMEs) and Business Users. The Business / Data Analysis (BA/DA) team took the output, defined the scope, refined the problem statement and identified all the relevant data sources. Data Engineering (DE) Team constructed a pipeline to ingest the data and Data Scientists (DS) performed Exploratory Data Analysis (EDA), worked with DE team on feature engineering and experimented with a number of different models.

A full description of the BA/DA, DE and DS processes are provided in subsequent sections of this document. Phase 1 considered a well-defined set of assets, faults and ELRs.

Phase 2 of POE (this phase) concentrated on considering additional factors that may affect the health of a POE asset, and on refining the models to reduce both false positive and false negative rates while maintaining precision and accuracy and extending the range of assets included.

During Phase 3, the team focussed on finalising the feature set and building supervised model.

## **1.6 Underlying assumptions**

In the absence of a predefined label of the degradation state of an asset, the pattern of the trace of the electrical current through an asset (both Track Circuit and Point Operating Equipment (POE)) may be analysed using advanced analytical methods and the differences in the patterns used to identify faults from non-faults.

*All the mentioned Data science solutions from the R&D team should only be treated as a POC (Proof of concept). The workstream team should do applicable analysis and changes to these solutions for making it production ready.*

## **1.7 Assets in scope**

Phase 1 and 2

* In both the first and second phases of statistical modelling, the only asset type considered was POE - Hydraulic/Pneumatic (Clamplock Mark 2)
* All types of faults were considered in scope
* Routes used in this phase: LNW South, LNW North
* Scope for Data Coverage for below sources are as follows:

- Ellipse Data : January 2017 - June 2020

- FMS Data : January 2017 - June 2020

- Wonderware Data : January 2017 - June 2020 (No Live Data)

Phase 3

Team worked with HW 1000 and 2000 series assets.

## **1.8 Out of scope**

Phase 1 and 2

* High Cost failures were excluded if their frequency was very low.
* Assets which don’t have Wonderware installed (c50%).

Phase 3

* 11 HW series assets were filtered due to below reasons:
  + 8 assets were not present in the POE PRD environment
  + For 2 assets there was a name mismatch with Ellipse ID
  + For 1 asset there was no raw Wonderware data

## **1.9 Phase 1 results**

Modelling results from Phase 1 at different thresholds are displayed below (Full results in [9.1 Modelling approach](#_9.1_Modelling_approach))

|  |  |
| --- | --- |
| **No\_ Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **65.7%** | **21.4%** |
| **11% Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **47.3%** | **11.4%** |
| **20% Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **39.3%** | **7.9%** |

[Fig1: Modelling results from Phase 1 at different thresholds]

In Phase 2 these results were refined, updated and compared to Wonderware Alerts and Alarms (to the extent possible - please see note 2 below).

Note 1: These results should be considered as interim and are subject to additional verification and updating during phase 3.

## **1.10 Phase 2 update**

As with Phase 1, a number of distinct models were produced, each with its own particular combination of threshold value, ‘true positive’ and ‘false positive’. These enable the business user to choose the most appropriate model based on the trade-off between correctly identifying POE failures in advance against the time wasted in investigating un-necessarily. Section 9.2 displays results from 5 different model variants and illustrates the comparison between these models and wonderware alerts and alarms.

Note 2: Results are also subject to the important caveat that for a number of reasons a direct comparison is not valid - to an extent apples are being compared to pears! Reasons include differences in calculation methodology for TP and FP and differences in the exact performance window used.

From section 9.2 results, one can see that model S20 gives a TP rate of 44.6% and a FP of 11.1% as compared to wonderware alerts & alarms TP of 45.6% and FP 11.8%. Similarly, model S21 gives a TP rate of 40.2%, FP of 9.4% while the existing Wonderware alarms based system gives TP 36.6% and FP of 5.7%

## **1.11 Phase 3 update**

There were two stages in POE Phase 3 (Stage 1 & Stage 2). As per discussions with the NR SMEs certain tasks were considered to improve the model performance. These steps were mainly:

1. Creation of different segmentation logic on traces,
2. Creation of new features based on the new trace segmentation logic,
3. Selection of different assets to validate the existing statistical model etc.

Basically, Stage 1 included the steps that were taken before deep dive session with NR SMEs and Stage 2 included the steps that were taken after the deep dive session.

Below mentioned changes in the modelling approach were enabled in Phase 3:

1. ***Revisited feature creation and engineering***
   * Traces were now being programmatically segmented in a way which is close (but, not exact) to the ‘engineering’ based definition of the event-phases
   * Logic to extract most features were implemented , additional features were explored for further inclusion
2. ***Optimal solutions were being identified through a robust exhaustive search mechanism***
   * True positive % and approximate average number of false triggers per month per asset would allow the work-stream to select solutions for the next stage based on business requirements
3. ***Grouping of assets with similar physical properties*** 
   * Grouping parameters was considered based on the hypothesis that grouping assets with similar physical properties will lead to an improvement in the performance of the solutions
4. ***Standard job analysis***
   * To reduce the number of false positives
   * Analysis of work orders and standard jobs to identify the negative impact on the health of the POE assets

Also, to validate that the final 7 features, the trace segmentation logic and statistical model as used in POE Phase 3 can also be reused for different type of assets, the team worked with HW 1000 & 2000 series assets data. The idea was to check whether the model built for Clamp Lock Mark II can be used for different other type of assets.

Additionally, the team performed model performance experiment during the iteration ending 6th July'21. The details of the experiment are captured in section 9.4.

## **1.12 Linearity experiment on 800+ Clamp Lock Mark 2 assets**

The objective of this experiment is to test if the performance of the solution improves or deteriorates linearly as more assets are added to the existing pool of assets.

To continue with the experiment, the finalized set of features, the trace segmentation logic and the finalized machine learning model from the previous iteration have been used.

## **1.13 Summary of business benefits delivered**

There are a number of benefits to be gained from improving the current Wonderware based system.

* Prediction of key failures.
* Ability to perform proactive and preventive asset maintenance.
* Longer time to plan interventions.
* Minimization of disruption to customers and services.
* Improved safety for staff and infrastructure.
* Improvement in customer/passenger satisfaction.

# **2 Business understanding - Point operating equipment**

## **2.1 Point operating equipment explanation**

Point Operating Equipment (POE) are devices for operating a switch in order to move a train from one track to another.



[Fig 2: POE machine]

There are currently 4 high level types of POE and each of these have different

Model Types

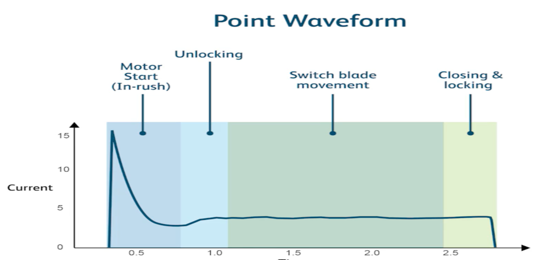
* HPSS 1 Model Type
* Hydraulic/Pneumatic 16 Model Types
* Mechanical 4 Model Types (2 x Signalling and 2 x Track)
* Point Machine 19 Model Types

The major functions of POE are:

* Unlock the POE Facing Point Lock
* Move the switch to either normal or reverse depending on the move being made
* On completion of moving the switch, lock the POE Facing Point Lock
* Indicate that the points have completed the move

Remote Conditioning Monitoring (RCM) can be installed on POE (except Mechanical). At the moment, most of the RCM installed measures current, but new RCM is being installed that also measures pressure (currently around 100 sites installed).

The key areas of the POE waveform are identified below



[Fig 3: POE waveform]

The diagram above shows a typical point waveform. It is important to understand the key phases of point operation and to align these to the waveform. The operation phases may be described as follows:

• **Motor start up inrush current -**When current is first applied to a DC motor the circuit is effectively short circuit for a few milliseconds causing a short spike of current. As the motor starts to turn and speeds up magnetic reluctance occurs and the level of current progressively falls. This phase of operation typically is 100 to 200ms duration.

• **Unlocking Phase -**At the end of the inrush phase current stops decreasing and the motor takes mechanical load; the lock mechanism unlocks, and the switch blades begin to move.

• **Switch Blade Movement** – This phase overlaps the unlocking phase and covers the duration of actual switch blade movement to the point the switch closes to the opposite stock rail.

• **Closing and Locking Phase** – This switch closes up to the stock rail and the lock mechanism engages and locks.

Typically, an increase in the average current and / or the duration or discernible increases in current during one or more of the phases of operation indicate a deterioration of the mechanical point operating condition. Additionally, changes to the pattern of the waveform, e.g. unusual spikes or ripple may indicate an electrical operating issue.

Major Faults Expected in Phase 1 2, & 3: electrical faults, power supply failures, motor start-up issues, and activation related faults.

## **2.2 Underlying assumptions**

In the absence of a predefined label of the degradation state of an asset, the pattern of the trace of the electrical current through an asset (both Track Circuit and Point Operating Equipment) may be analysed using advanced analytical methods and the differences in the patterns used to identify faults from non-faults.

## **2.3 Traces**

Following from Waveform above, Traces were segmented into 4 distinct phases:

1. Start up inrush
2. Unlocking
3. Switch Blade
4. Closing / Locking

Data points that were not consistent with surrounding data points were considered anomalous data-points. It is assumed that a subsequence of such anomalous data-points provides early indication of any problem with POE.

## **2.4 Model features**

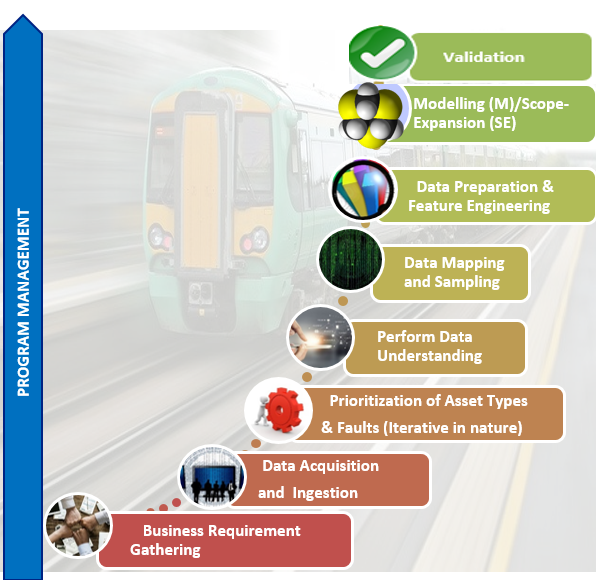
Features used to train the models included:

* Machine characteristics (traces, usage),
* Maintenance (signalling) history,
* Fault history.

The features extracted from traces capture aspects of the trace cycles based on domain knowledge and are largely invariant to individual asset characteristics.

# **3 Solution approach**

Team applied the following structured approach while implementing the Point Operating Equipment use case with the overarching program management methodologies. There are dedicated activities across each of phase of this solution approach.

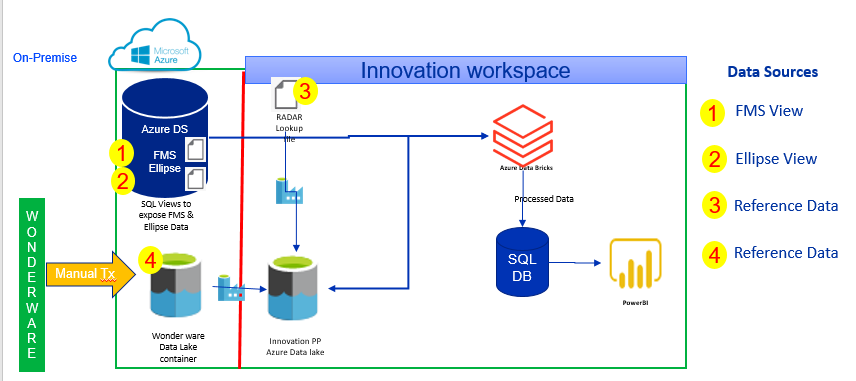


[Fig 4: Solution methodology and steps]

# **4 Data requirements and data ingestion**

The following section describes the different data sources, targets and methods to transform data using Azure Data Factory (ADF) and Azure Databricks.

## **4.1 Data architecture: dev environment**



[Fig 5: Conceptual architecture of the dev environment]

**Note:**

* This architecture was followed for development related work for POC only in DEV DB server, it may change in actual production environment.
* SQL Database was not used to store raw Wonderware data due to huge data volumes.

There are four data sources indicated in above diagram. These data sources are described below.

1. **FMS (Fault Management System) data** – This is asset failure data. This dataset is liberated to sqliidstpochub001.database.windows.net DB – ADS DataMart (Azure Data Store). The data is further enhanced by joining different tables and exposed as SQL view.
2. **Ellipse Data** - Liberated to sqliidstpochub001.database.windows.net DB - ADS\_Data\_mart. The data is further enhanced by joining different tables and exposed as SQL view.
3. **Reference Data** – This file is required to provide a mapping for asset name, used in Wonderware data and asset name used in DIM\_ASSET table. This is a manual data extract exposed as csv file. (For production an updated version of this file may be required)
4. **Wonderware data** – Sensor data from different assets.

## **4.2 Data sources used**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Data Source / Component Name** | **Type of Data** | **Description** |
| 1 | FMS | Structured | FMS is a Fault Management System. It has the Failure data. It is used for Managing / predicting failures. |
| 2 | ELLIPSE | Structured | Ellipse is the system of record for assets, defects, Work Orders, Maintenance Schedules (MST). It has the assets data which can be used for managing assets. |
| 3 | Reference Data | Structured |  |
| 4 | WONDERWARE | Structured | Wonderware has a Real time condition monitoring data and recordings.  It is used for assessing condition / predicting failure. |
| 5 | INORDER DATE | Structured | ADS Foundation DB also contains Fault Management System. It has the Failure data with details of failure, event date, and reference value. |
| 6 | WORKORDER DATA | Structured | Workorder data has the fault related data along with maintenance and standard job related data. |

## **4.3 Data coverage**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Data Source / Component | Database | DB Tables/location | Data Timeline | Data Coverage |
| 1 | FMS | ADS\_Data\_Mart | 1. DIM\_EF\_FAILURE 2. FACT\_EF\_FAILURE 3. DIM\_ASSET 4. FACT\_ASSET 5. DIM\_DATE | 2017 Jan - 2020 Jun | Historical Data Only  No Live Data |
| 2 | ELLIPSE | ADS\_Data\_Mart | 1. DIM\_ASSET 2. DIM\_WORKORDER 3. FACT\_WORKBANK 4. DIM\_STD\_JOB 5. DIM\_DATE 6. DIM\_SNC\_ASSET\_NAMEPLATE | 2017 Jan - 2020 Jun | Historical Data Only  No Live Data |
| 3 | WONDERWARE | NA | ADLS location | 2017 Jan - 2020 Jun | Historical Data Only  No Live Data |
| 4 | INORDER DATE | ADS\_Foundation | 1. ADS\_FND\_FMS\_FAILURE\_EVENTS 2. ADS\_FND\_FMS\_REFERENCE\_VALUES 3. ADS\_FND\_FMS\_FAILURES | NA | Historical Data Only  No Live Data |
| 5 | WORKORDER | ADS\_DATA\_MART | 1. DIM\_ASSET  2. DIM\_WORKORDER  3. FACT\_WORKBANK  4. DIM\_WORKGROUP  5. v\_cf10\_POE\_Assets   1. 6. DIM\_STD\_JOB | 2017 onwards | Historical Data Only.  No Live Data |

## **Data engineering phase 2**

**IN-ORDER DATE Calculation**

The Data Science team found large gaps between Failed Date and Fault Resolved date, so NR SMEs were consulted and ‘InOrder Date’ column was utilised to fix this issue.

**Note:**

* The architecture for “InOrderDate” was implemented for development related work for POC only in DEV SQL DB server, the technical architecture may change in actual production environment.

**Source & Target Location**

For "InOrderDate" use case, a SQL View was created. Data engineering team used supporting table from Source Database to expose information for "InOrderDate" with respect to fault data (FMS data, formed by using DIM\_DATE & other table, mentioned in 4.3). Details of the view are as follows.

|  |  |  |
| --- | --- | --- |
| Server Name | DB Name | SQL View Name |
| sqliidstpochub001.database.windows.net | ADS\_Foundation | 1. [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURE\_EVENTS] 2. [ADS\_FH\_FMS].[ADS\_FND\_FMS\_REFERENCE\_VALUES] 3. [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURES] |
| sqliidstpochub001.database.windows.net | ADS\_Data\_Mart | 1. DST.v\_cf10\_AssetFMS |

**Source (Supporting) View Definition:**

[DST].[v\_cf10\_AssetFMS] :

* Implemented the view so that data is coming from 2017 to 2019 from table ADS\_DM.Dim\_Date
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_ASSET
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_SNC\_ASSET\_NAMEPLATE
* Implemented the view so that data is coming for length of equipment number is 12 from table ADS\_DM.DIM\_ASSET
* Implemented the view so that data is filtered by, Track signalling and POE assets

The logic has been appended in **Appendix** section

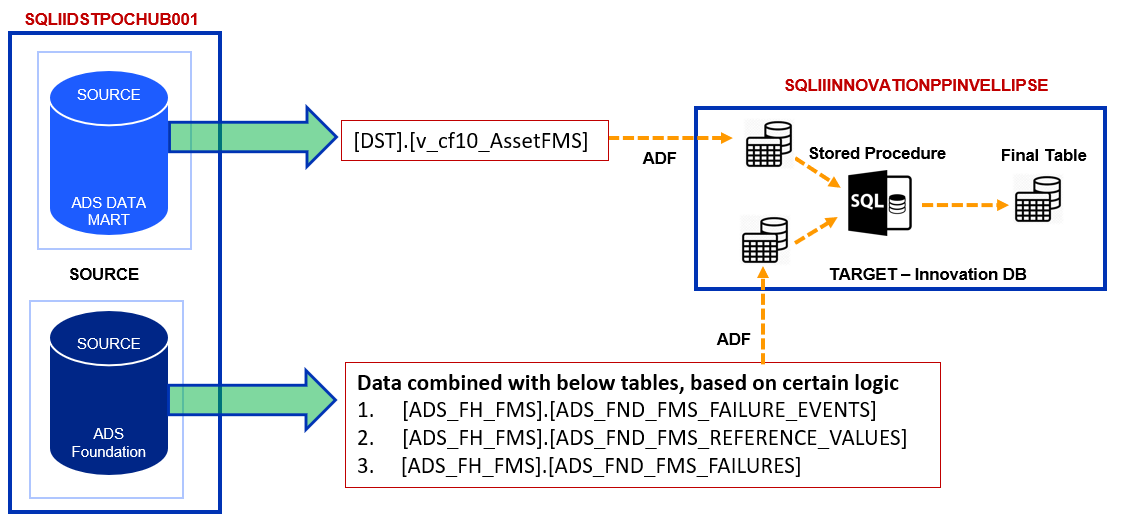
**IN-ORDER DATE Target table Structure:**

For POE "In-OrderDate" use case, a SQL table was created in below target Database to expose information for Failure data of assets. Supporting tables, along with target table are as follows:

|  |  |  |
| --- | --- | --- |
| Server Name | DB Name | SQL table Name |
| sqliiinnovationppinvellipse.database.windows.net | sdbiiinnovationppinvellipse | 1. dbo.tbl\_cf10\_AssetFMS 2. dbo.inorderdate 3. dbo.tbl\_cf10\_AssetFMS\_InDT > Target |

**IN-ORDER DATE Ingestion Pattern: Architecture Diagram**

Below is the high level pictorial representation of Ingestion pattern of InOrder date



[Fig 6: Conceptual architecture diagram of IN-ORDER DATE ingestion]

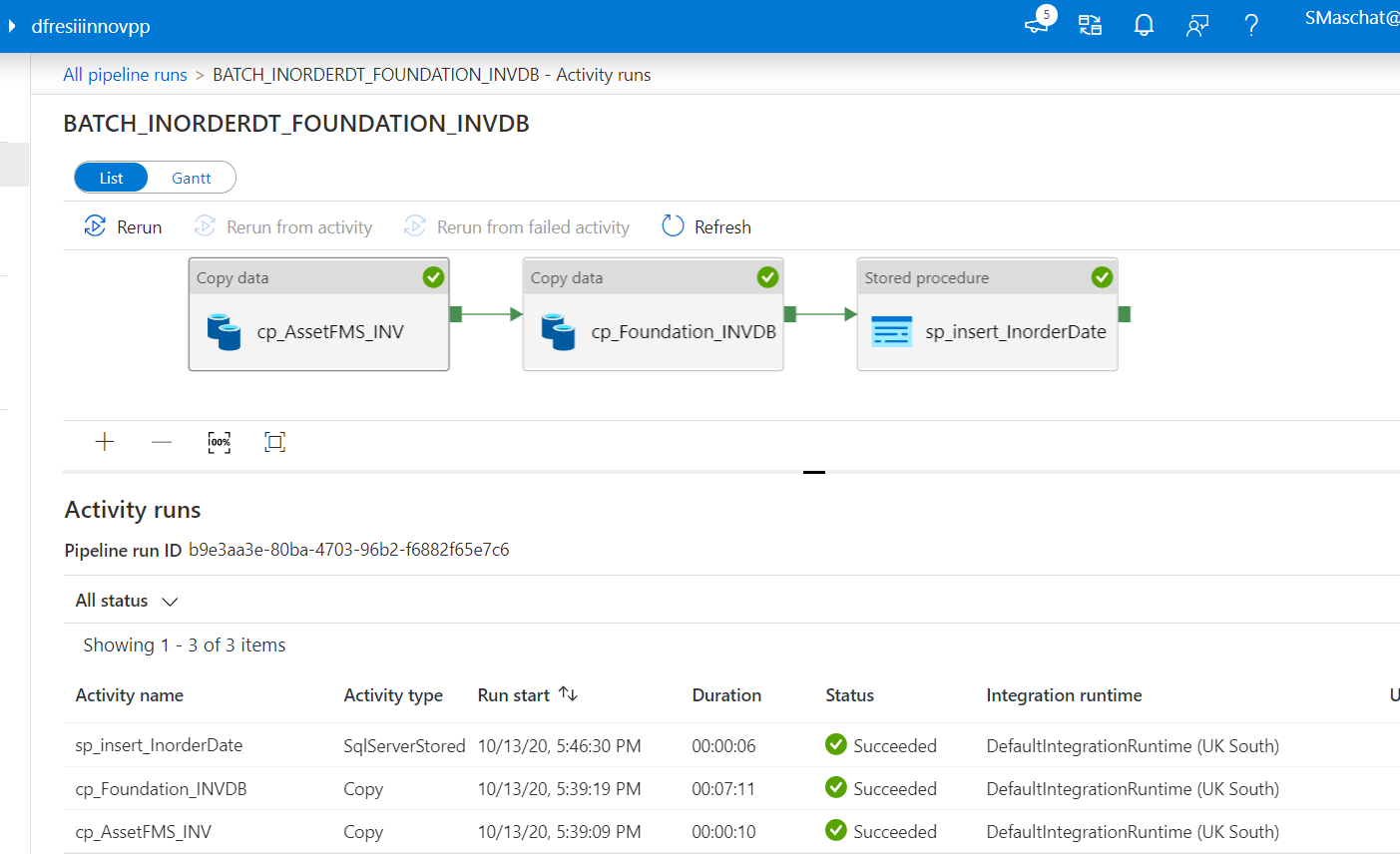
**Note:**

* From Source, AssetFMS data will be pulled directly
* From source ADS\_Foundation, failure related column will be pulled based on certain condition (The logic has been appended in **Appendix (Codes for In-Order Date)** section and details has been mentioned on **Appendix (Brief details to find InOrder date in ADS Foundation DB)**
* In target, Inorder data will be populated through a dedicated store procedure (The logic has been appended in **Appendix (Codes for In-Order Date)** section)

**IN-ORDER DATE Ingestion Pattern: ADF Technical Diagram**

Based on the above pictorial representation, below ADF pipeline was designed to copy the required data from source to target location and populate in order date, with respect to fault data.

Below image shows ADF pipeline & parameter control.



[Fig 7: ADF pipeline and parameter tool]

## **Data Engineering Phase 3**

### **4.5.1 Grouping Parameters**

For Grouping parameters, the analysis has been performed on identifying the below listed parameters in the database:

• Rollers

• Switch type (length)

• Backdrive

• Lubricants

• S&C type

The status of each parameter’s identification was shared with the data science team.

### **4.5.2 Workorder data**

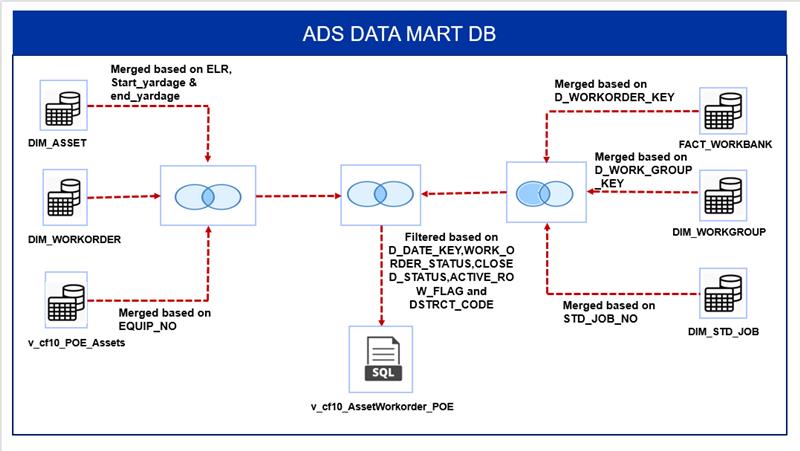
The Data Science team requested for the data in such a format where, the workorder data was generated for the surrounding assets of one particular asset.

**Source and Target location:**

For Workorder data, a SQL view was created. The DE team used supporting tables/views from the source database to expose information for workorder data. Details of the source tables/views and target SQL view are as follows:

**Overview on all the factors**

|  |  |  |
| --- | --- | --- |
| 1 | **Server** | sqliidstpochub001.database.windows.net |
| 2 | **Database** | ADS\_DATA\_MART |
| 3 | **Source Tables/views** | 1. DIM\_ASSET 2. DIM\_WORKORDER 3. FACT\_WORKBANK 4. DIM\_WORKGROUP 5. DIM\_STD\_JOB 6. v\_cf10\_POE\_Assets |
| 4 | **Target View** | v\_cf10\_AssetWorkorder\_POE |
| 5 | **Count** | 157560 |



[Fig 8: Logical Architecture diagram for workorder data]

**Note:**

* Workorder Data is liberated to sqliidstpochub00database.windows.net DB – ADS DataMart (Azure Data Store). The data is further enhanced by joining 5 different tables and 1 view, which is exposed as SQL view.

**Dedicated View definition:**

1. Based upon the Discipline, we have incorporated a new column called ‘Type’ in the workorder data such that:
   1. When Discipline = ‘Signals’/Discipline = ‘Track’, then Type = ‘NA’
   2. For rest of the Disciplines, the Type = ‘WORK AT LOCATION’.
2. Based upon the Yardages, we can extract the workorder related data for the surrounding assets only if any of the below listed conditions are satisfied:
   1. DimAsset.Start\_Yardage between DimWO.Start\_Yardage and DimWO.End\_Yardage
   2. DimAsset.End\_Yardage between DimWO.Start\_Yardage and DimWO.End\_Yardage
   3. DimWO.Start\_Yardage between DimAsset.Start\_Yardage and DimAsset.End\_Yardage
   4. DimWO.End\_Yardage between DimAsset.Start\_Yardage and DimAsset.End\_Yardage
3. In order to obtain the standard job data, we used the following logic such that we can use the latest standard job data.
   1. PARTITION BY STD\_JOB\_NO ORDER BY D\_STD\_JOB\_KEY DESC
4. The view is further filtered based on the following points:
   1. Implemented the view so that data is coming from 2017 based upon the D\_DATE\_KEY > 100120170101000 from FACT\_WORKBANK table.
   2. Implemented the view so that data is coming for WORK\_ORDER\_STATUS not equal to 'Authorised' from DIM\_WORKORDER table
   3. Implemented the view so that data is coming for CLOSED\_STATUS not equal to 'X' from DIM\_WORKORDER table
   4. Implemented the view so that data is coming for DSTRCT\_CODE = 'RTK1' from DIM\_WORKORDER table
   5. Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from DIM\_ASSET table
   6. Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from DIM\_WORKORDER table
   7. Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from DIM\_WORKGROUP table

### **4.5.3 IBCL & Westinghouse**

For Failure and workorder data, dedicated SQL Views were created. Data engineering team used supporting tables/views from Source Database and as well as the list of equipment’s for which the data was generated.

Details of the source tables/views and target SQL view are as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | **Failure** | **Server** | sqliidstpochub001.database.windows.net |
| **Database** | ADS\_DATA\_MART |
| **Source Tables/views** | 1. FACT\_EF\_FAILURE 2. DIM\_EF\_FAILURE 3. Dim\_Date 4. DIM\_ASSET 5. FACT\_ASSET 6. DIM\_SNC\_ASSET\_NAMEPLATE 7. v\_cf10\_IBCL\_Assets 8. v\_cf10\_Westinghouse\_Assets |
| **Target View** | v\_cf10\_AssetFMS\_Westinghouse\_IBCL |
| **Count** | 1293 |
| 2 | **Workorder** | **Server** | sqliidstpochub001.database.windows.net |
| **Database** | ADS\_DATA\_MART |
| **Source Tables/views** | 1. FACT\_WORKBANK 2. Dim\_Date 3. DIM\_ASSET 4. DIM\_WORKORDER 5. DIM\_STD\_JOB 6. v\_cf10\_IBCL\_Assets 7. v\_cf10\_Westinghouse\_Assets |
| **Target View** | v\_cf10\_AssetWorkorder\_Westinghouse\_IBCL |
| **Count** | 31821 |

**In-detailed logic**:

The above listed views consists of the consolidated data for both IBCL and Westinghouse assets.

Below are the conditions used to extract the corresponding Failure data for IBCL and Westinghouse assets from source:

* The Failure data here is obtained from the year 2017 as per the condition: " cal\_Year >=2017"
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from DIM\_ASSET table

Below are the conditions used to extract the corresponding Failure data for IBCL and Westinghouse assets from source:

* The Workorder data here is obtained from the year 2017 as per the condition: " cal\_Year >=2017"
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from DIM\_ASSET and DIM\_WORKORDER tables
* Implemented the view so that data is coming for WORK\_ORDER\_STATUS = 'CLOSED' from DIM\_WORKORDER table
* Implemented the view so that data is coming for DSTRCT\_CODE = 'RTK1' from DIM\_ASSET and DIM\_WORKORDER tables

**Additional feature**:

Apart from the above enlisted conditions, we have incorporated a new column called '**SEGMENT**' in actual target views (v\_cf10\_AssetFMS\_Westinghouse\_IBCL, v\_cf10\_AssetWorkorder\_Westinghouse\_IBCL). This has been incorporated as the dedicated views has the data extracted for both IBCL and Westinghouse assets using the 'UNION ALL' operation. This column specifies whether the asset belongs to IBCL or Westinghouse.

**Record count of each segment**:

Following are the individual record counts of Failure and Workorder data for IBCL and Westinghouse equipment's each:

* Failure data
  + IBCL - 494, out of 150 given priority assets
  + Westinghouse - 799, out of 150 given priority assets
* Workorder data
  + IBCL - 17,301, out of 150 given priority assets
  + Westinghouse - 14,520, out of 150 given priority assets

**Code Location:**

Please find the below location for code, and mail us in case of any queries or access

* Indu.dhulipala@cognizant.com
* Sauryya.Maschatak@cognizant.com

https://networkrail-my.sharepoint.com/personal/smaschat\_networkrail\_co\_uk/\_layouts/15/onedrive.aspx?FolderCTID=0x012000096C36D32FD3D84180F9CA8B1C822602&id=%2Fpersonal%2Fsmaschat%5Fnetworkrail%5Fco%5Fuk%2FDocuments%2FDE%5FCODE%2FDE%2FCode%2FPOE3%2FSQL%2FWestinghouse%20IBCL

### **4.5.4 Weather data - Ingestion pattern**

**Ingestion Architecture:**

A picture containing timeline

Description automatically generated

[Fig 9: Logical Architecture diagram for weather data ingestion]

***Step: 1***

Three types of weather source data are placed in ADLS Gen2 and same has been processed in three different Azure SQL DB table using databricks.

**Source ADLS folder directory:**

The source files are stored In below ADLS directory path

* Storage Account: stgiiinnovationppinv
* Container: CF10
* Directory: WeatherData

**Mapping between source file and target table:**

|  |  |
| --- | --- |
| Source file name | Target table name |
| Weather baseware.parquet.gzip | dbo.tbl\_WeatherData |
| Weather Cells.csv | dbo.tbl\_WeatherCells |
| ELR Yard Info.csv | dbo.tbl\_Weather\_ELRYardINfo |

***Step: 2 (ADF)***

1. Incorporated POE priority asset related information to innovation environment for phase 3
2. For each asset, searched the weather data in source files (tables, mentioned above) by using a dedicated logic (written in the SP - [dbo].[usp\_tbl\_WeatherBaseware\_Output]), and loaded the data into target table - [dbo].[tbl\_WeatherData\_Output]

**Source Code for SQL & ADF:**

ADF:

You will get ADF code in ‘Dev\_Inv’ branch In folder - INV\_ADF, under DevOps project “Innovation Data - Prevention and Prediction”

* BATCH\_WEATHER\_TCPOE\_LOAD
* BATCH\_WEATHER\_TCPOE\_LOAD\_CHILD

SQLDB (Innovation Environment):

Tables:

You will get SQL DB table's code in ‘Dev\_Inv’ branch in below folder under DevOps project “Innovation Data - Prevention and Prediction”

CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/dbo/Tables

* dbo.tbl\_WeatherData
* dbo.tbl\_WeatherCells
* dbo.tbl\_Weather\_ELRYardINfo
* dbo.tbl\_CF10\_DIM\_ASSET\_Weather (asset information is stored here)
* dbo.tbl\_WeatherData\_Output (target table)

SP:

You will get SQL DB SP's code in ‘Dev\_Inv’ branch in below folder under DevOps project “Innovation Data - Prevention and Prediction”

CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/dbo/Stored Procedures

* dbo.usp\_tbl\_WeatherBaseware\_Output

Azure Databricks Notebook:

You will get below notebook code in ‘Dev\_Inv’ branch in below folder under DevOps project “Innovation Data - Prevention and Prediction”

* NoteBook\_STG/smaschat/WeatherData.py

### **4.5.5 Failure Data for HW Assets & In-Order Data– ingestion pattern**

**Ingestion Architecture:**

Diagram, text

Description automatically generated

[Fig 10: Logical Architecture diagram for HW assets and IN ORDER ingestion]

**Note:** The ingestion architecture is similar to point number: 4.4. However, the only difference is here we are including the dedicated POE asset numbers instead of other POE numbers.

**Steps:**

1. ADF pipeline (**BATCH\_INORDERDT\_FOUNDATION\_INVDB\_HW**) is used for ingestion pattern for the HW assets and In Order data. In the first step using copy activity data, which is pulled failure data into [tbl\_cf10\_AssetFMS\_HW] target table in innovation environment for HW Assests. Here source are [DST].[v\_cf10\_AssetFMS], [DST].[v\_cf10\_POE\_HW\_Assets]
2. After 1st step success another copy activity is used for to process the data into dbo.inorderdate as target table in innovation environment. Here source are [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURES] , [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURE\_EVENTS], [ADS\_FH\_FMS].[ADS\_FND\_FMS\_REFERENCE\_VALUES
3. After the 2nd steps store procedure activity is used to Insert the failure record along with InOrder date, into a [dbo].[tbl\_cf10\_AssetFMS\_InDT\_HW] table.

**Source Code for SQL & ADF:**

ADF:

You will get ADF code in ‘Dev\_Inv’ branch in folder - INV\_ADF, under DevOps project “Innovation Data - Prevention and Prediction”

* BATCH\_INORDERDT\_FOUNDATION\_INVDB\_HW

SQLDB (Innovation Environment):

Tables:

You will get SQL DB table's code in ‘Dev\_Inv’ branch in below folder under DevOps project “Innovation Data - Prevention and Prediction”

CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/dbo/Tables

* inorderdate
* ​tbl\_cf10\_AssetFMS\_HW
* ​tbl\_cf10\_AssetFMS\_InDT\_HW

SP:

You will get SQL DB SP's code in ‘Dev\_Inv’ branch in below folder under DevOps project “Innovation Data - Prevention and Prediction”

CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/CF10\_INV\_DBSOL/dbo/Stored Procedures

* dbo. usp\_cf10\_AssetFMS\_InDT\_HW

# **5 Data understanding**

As described above in section 4.2, the following datasets were used:

1. Current values and current waveform direction against timestamps - From Wonderware.
2. Ellipse – Asset information – EGI (Equipment Group Identifier), type of asset.
3. FMS – Fault records (Fault Management System).
4. WO – (Work Order) historical faults and maintenance information.

In total 79 POE assets of type Hydraulic/Pneumatic clamplock lock mark II were considered for analysis in this phase.

The final model was built using 52 assets out of these 79 assets. These 52 assets were selected based on data quality issues and clustering analysis. Asset selection and data quality issues have been discussed in further sections.

## **5.1 Quality of data**

It is very important for data science work to evaluate the quality of data before modelling work commences. Data quality affects model performance adversely if not handled properly. The major quality issues in data are incomplete or gaps in data, shifts in range of current and identical timestamps. These issues have been discussed below.

### **5.1.1 Incomplete data**

The analysis was done using data from Jan 2017 to June 2020, however, for some assets, data in terms of number of months, number of days and number of traces was much less compared to other assets. There is no specific period or month for which gap in data is observed for these assets.

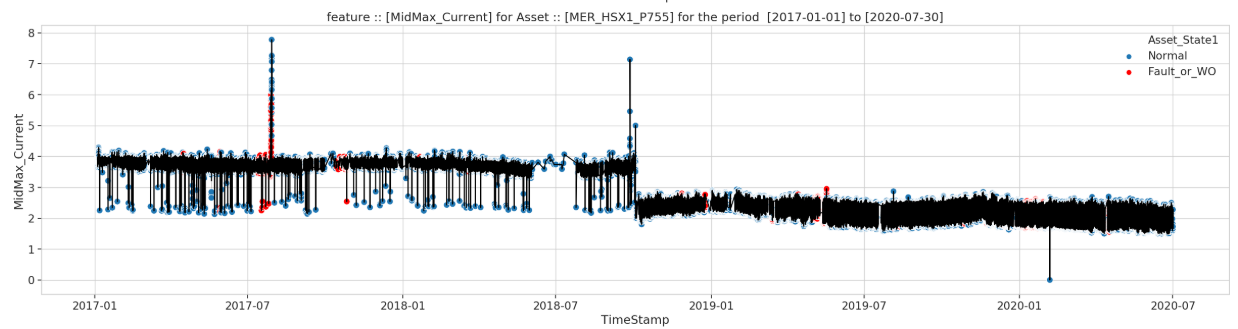
Gaps in data or less data of an asset can be problematic for the predictive model to learn the patterns of traces before a fault especially when the data is missing on the days prior to a fault.

Note: Table 10.4.1 in Appendix contains the data availability information for the assets considered in this phase.

### **5.1.2 Shift in range of current**

For some of the assets a shift in range for maximum value of current in phase 2 was observed. This may constraint the correct learning of model as it is an important feature.

As seen in Fig 5.1, the range of maximum current of phase 2 is around 4 Amp from Jan - 2017 to Oct -2018 and then range shifts between 2 & 3 Amp after this period.



[Fig 11: Range of maximum current value of phase 2 changes after October 2018]

### **5.1.3 Identical time stamps**

It was observed that there were instances of different values of current reading at the same Timestamp for some assets. Due to this, the traces do not get segmented properly. Table 5.1 shows one of the instances for the asset 'GBHYDEJN\_HAJ\_P1006A'.

**Table 5.1: Example of identical timestamp**

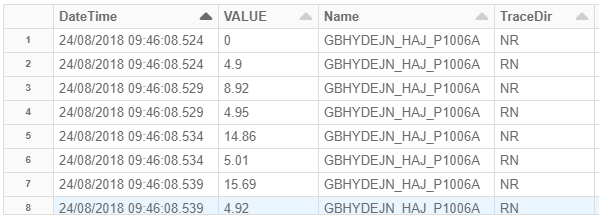


Table 5.2 illustrates asset-wise number of instances of identical timestamp issues and total number of traces where Instances were observed.

**Table 5.2: Identical timestamp issue on number of assets**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EllipseID** | **Name** | **No\_of Traces** | **Avg\_no\_of\_traces** | **timestamp\_issue\_trace** | **%** |
| 000001458741 | MAIRPORT\_MIA\_P2206A | 156129 | 125 | 603 | 0.4% |
| 000002505227 | RYECROFT\_BJW2\_P629B | 22636 | 19 | 558 | 2.5% |
| 000002640105 | SLADELNJ\_CMP2\_P314 | 149417 | 119 | 556 | 0.4% |
| 000001459299 | WNDSRBRS\_MVE1\_P2517 | 77933 | 67 | 425 | 0.5% |
| 000001412867 | CREWCS9\_LEC5\_P776A | 57547 | 47 | 307 | 0.5% |
| 000000839502 | N66#2#3RR\_HNR\_P1417A | 67568 | 69 | 297 | 0.4% |
| 000001412895 | CREWECS3\_LEC5\_P722A | 125790 | 102 | 281 | 0.2% |
| 000001458824 | GBHYDEJN\_HAJ\_P1006A | 56107 | 49 | 275 | 0.5% |
| 000001412869 | CREWCS11\_LEC5\_P810B | 55049 | 46 | 265 | 0.5% |
| 000001412886 | CREWCS1\_SYC\_P719A | 48568 | 57 | 177 | 0.4% |
| 000001459292 | MARWHFJN\_TTA1\_P107 | 65332 | 61 | 166 | 0.3% |
| 000002714646 | MAIRPORT\_MIA\_P2210 | 103042 | 82 | 148 | 0.1% |

**Note:** This issue was observed while evaluating the model’s performance and analysing the traces.

**Phase 2 update**

**Availability of data for predictable faults**

The solutions were provided based on 204 total number of 'predictable' failures out of given 300 total number of failures from Wonderware data (300 - 204 = 96 failures cannot be predicted). However, there were a few issues with 18 failures.

Table 5.3 illustrates the analysis of corresponding Wonderware data (204- 186 = 18 failures) where data is not available for 17 failures and in one case, two faults were reported on consecutive days.

**Table 5.3: 18 Non -availability of data for predictable faults**



**Discrepancies between fault resolved date and in order date**

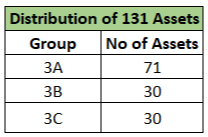
It has been observed that there are 24 Faults out of 204 (11.8%) where the Fault Resolved Date & In Order Date are different. Also, large gaps (more than 20 days) between Fault Date & In Order date have been observed in 33 Faults (16.2%). Refer to [10.3.2 Discrepancies (Fault Resolved and In Order Dates)](#_10.3.2_Discrepancies_(Fault) in Appendices

## **5.2 Phase 3 – Stage 1 - Data understanding & Quality**

### **5.2.1 Context of the assets**

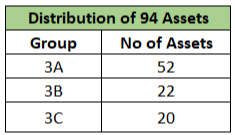
In POE Phase 3 – Stage 1, 131 assets from 3 group of assets (3A, 3B & 3C) were considered to evaluate the existing one class model. The description of 3 grouped assets are - 3A group assets belonging to C type switch, 3B group belonging to all types of switch except C type switch & 3C group assets where there were no failures observed. 3A & 3B groups consisted of 101 assets (71 + 30) where failures were found, whereas 3C group which consisted of 30 assets did not consist of any failures. The group wise distribution of 131 assets was as below:

**Table 5.2a: Group wise distribution of 131 assets**



However, out of 131 assets, 2 assets were removed due to raw circuit current data not available in WonderWare and 35 assets were dropped based on clustering technique. Finally, 94 assets were considered for the final model evaluation. The group wise distribution of 94 assets were as below:

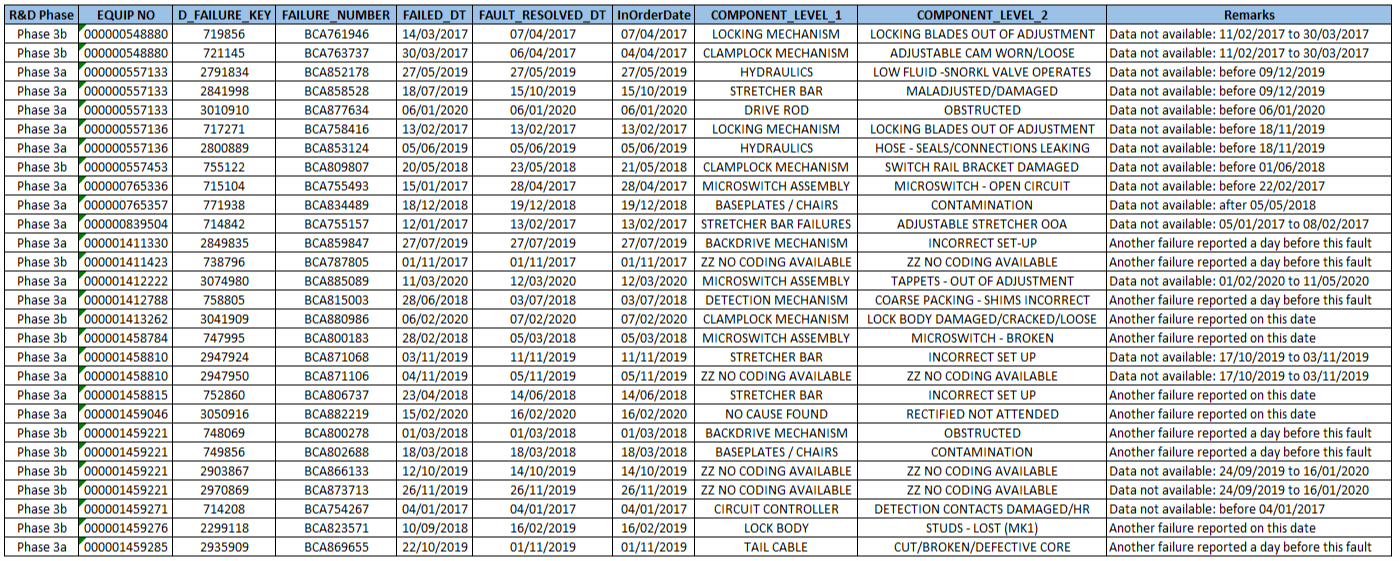
**Table 5.2b: Group wise distribution of 94 assets**



### **Context of failures (on 94 assets)**

In stage 1, a total 401 failures were reported from 3A and 3B group assets. However, there was unavailability of data for predictable faults in 28 failures from 3A and 3B group assets like, Wonderware data missing for 17 failures, 6 faults were reported on consecutive days and in 5 cases, 2 faults were reported on the same day. Table 5.2c illustrates the detail analysis of corresponding Wonderware data (401- 373 = 28 failures) where data was not available for 28 failures.

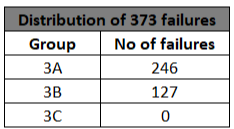
**Table 5.2c: 28 Non -availability of data for predictable faults**



Finally, for the model evaluation, 401 failures were considered from 94 assets out of which 373 failures were from 3A & 3B group of assets (no failures were reported in 3C group of assets) and 28 failures of unavailability of data for predictable faults from 3A & 3B group of assets. However, total 246 number of failures were reported in 3A group assets and total 127 number of failures were reported in 3B group assets.

Table 5.2d, provides the group wise distribution of 373 failures from 3A & 3B group of assets:

**Table 5.2d: Group wise distribution of 373 failures**

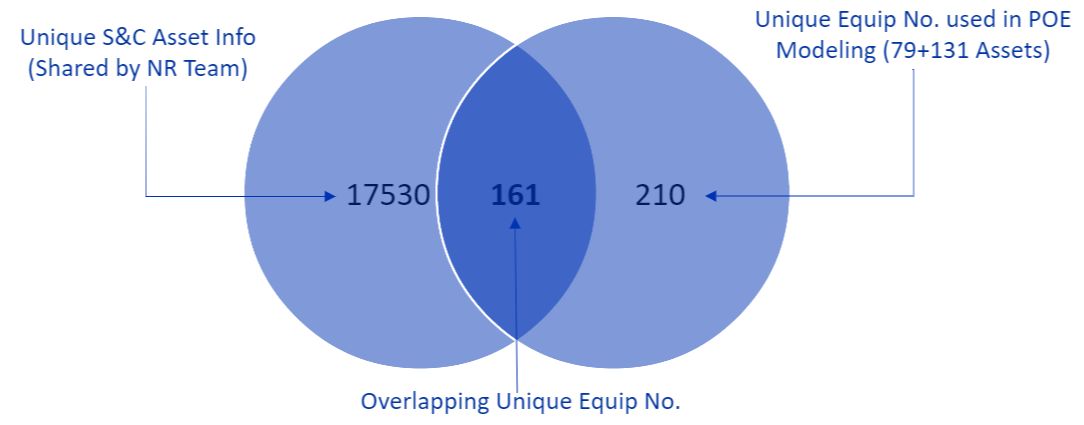


## **Phase 3- Stage 2 - Data understanding on Grouping Parameters**

### **5.3.1 S&C Turnout grouping assets (Grouping Parameters)**

Initially, for the POE modelling in phase 1, 210 Clamplock Mark 2 type assets from LNW route were selected to build the one class model. To validate the performance of the one class model, the same 210 assets out of 17530 S&C Clamplock Mark 2 type assets were provided by the NR team. However, 49 S&C – Turnout Clamplock Mark 2 type assets’ equipment no. were not included. Finally, in POE phase 3 – Stage 2, 161 S&C – Turnout Clamplock Mark 2 type assets were selected to validate the existing one class model.

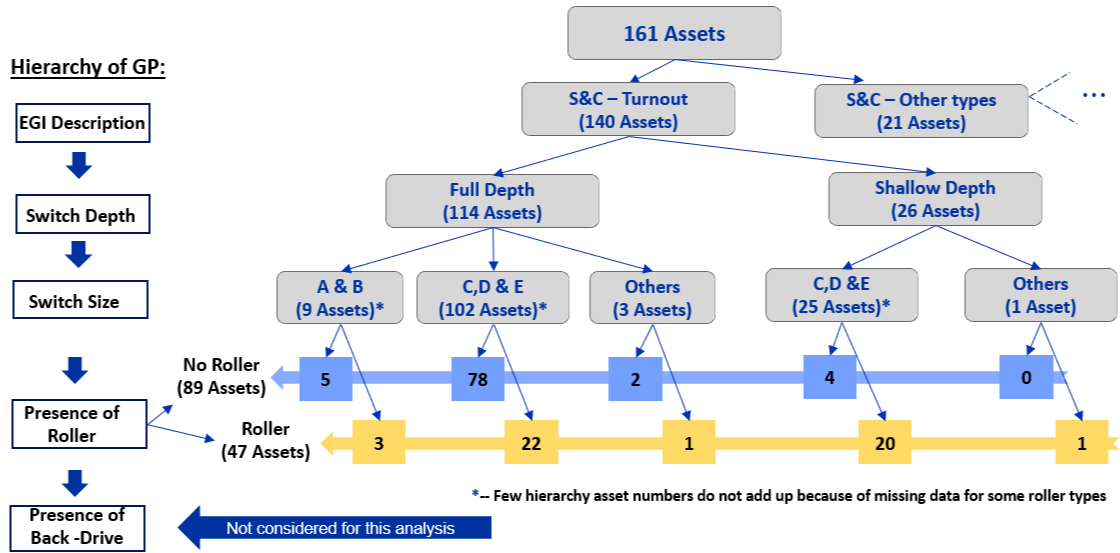
Below figure shows the 161 unique S&C – Turnout Clamplock Mark 2 type assets were used to validate the one class model.



[Fig 12: Selection of 161 unique S&C – Turnout Clamp Lock Mark II type assets]

S&C Turnout grouping assets’ hierarchy and no of assets in each group of hierarchy

Below figure shows the hierarchy of S&C Turnout grouping assets and number of assets in each group of hierarchy.



**[Fig 13: Hierarchy of S&C Turnout grouping assets and no of assets in each group of hierarchy]**

## **Linearity experiment on Clamp Lock Mark 2 assets**

For this experiment, total 848 Clamp Lock Mark 2 assets were received from the NR team and the data period was consider from January 2017 to June 2020. However, out of 848 assets, there were approx. 30% assets that did not have any fault history in the mentioned three and half year’s period.

Out of 848 assets, total 805 assets were used as data and the remaining 43 assets were not considered because of following reasons:

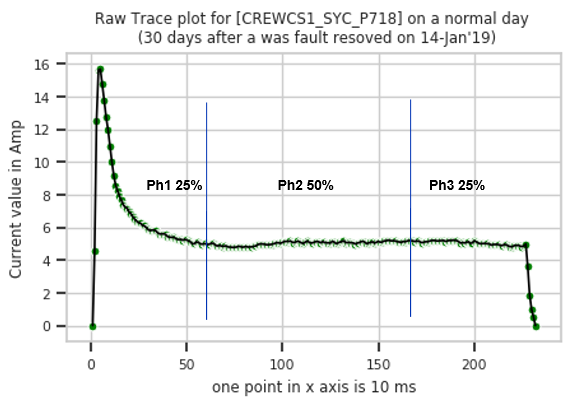
1. Data for 34 assets was not available in local environment.
2. There was some mismatch with Asset Name and EllipseID for 6 assets.
3. Other data issues for 3 assets.

# **6 Data preparation**

Raw data available has timestamps up to millisecond level of precision, current values in Amperes and current waveform direction (Reverse to Normal /Normal to Reverse).

The following steps were taken to extract the traces from raw data

* Data points between 12 am and 5 am were dropped
* The raw data points was segmented into traces based on current readings, trace direction and timestamps on the basis that a trace starts with a 0 Amp reading and ends with 0 Amp reading.
* Three phases were considered to mark unlocking, switch blade movement and closing & locking. Below fig 14 illustrates these marking and division of three phases of each trace
* The traces were segmented into different phases to extract specific features from these phases. For data science work the traces ~~are~~ were sliced into three phases by taking the first 25% of data points as phase 1, next 50% of the data points as phase 2 and last 25% of the data points as phase 3. This represents a rough mapping of 4 phases of a trace as described in Business Understanding (section 2.1) into 3 phases.
* The business understanding definition of segmenting the trace into 4 phases was not used here because start and end of different phases is not well defined and marking them is not straight forward
* After segmenting the traces into three phases, phase 1 represents the motor start & unlocking, phase 2 represents the switch blade movement and phase 3 mark the closing & locking.



**Fig 14: A typical trace and its phases**

The above figure is representation of an exemplary trace after segmentation of raw data. X axis shows the index of data points. One point in X axis equals 10 milliseconds. Y axis shows the current In Amperes of the data points In X axis. Also, traces have been marked into three phases to represent motor start & unlocking phase, switchblade movement phase and locking phase. The motor start & unlocking phase contains first 25% of the data points, switchblade movement phase contains next 50% of the data points and the unlocking phase here contains the last 25% of the data points of a full trace.

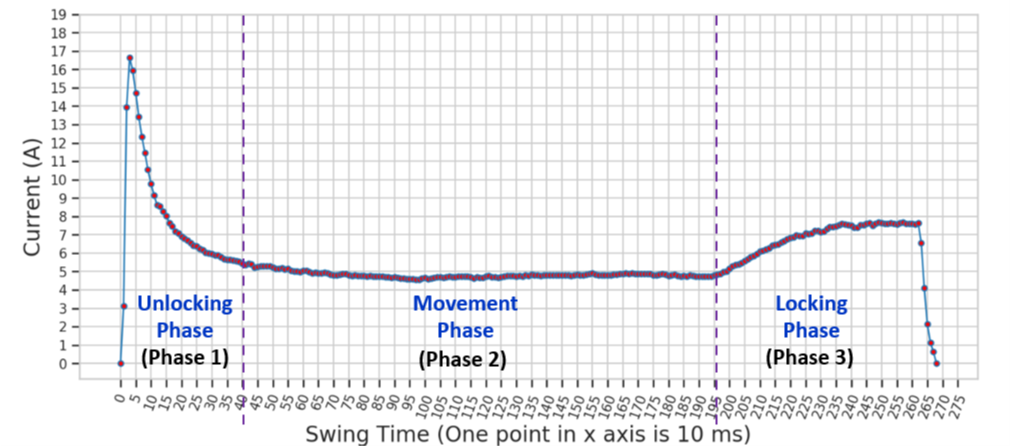
### **6.1 Phase 3 Data Preparation**

**Use of hybrid logic to segment phases automatically in POE phase 3 (after various deep dive session with NR SMEs)**

Previously, the traces were segmented into different phases to extract specific features. The traces were sliced into three phases by taking the first 25% of data points as phase 1, next 50% of the data points as phase 2 and last 25% of the data points as phase 3. This represents a rough mapping of 3 phases.

However, after several deep dive sessions with NR SMEs’, a different approach was considered to create the traces. The traces were segmented programmatically using a hybrid logic and features were extracted using robust mechanism.

The hybrid logic automatically identifies the end of phases of a trace and segment the traces into three phase like unlocking phase, movement phase and locking phase. The identification of end of these phases depended upon the nature of traces.



**Fig 15: A typical trace and its phases using hybrid logic**

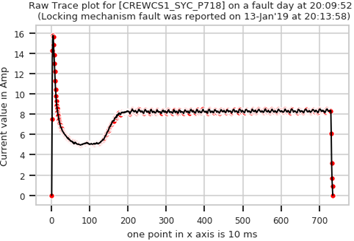
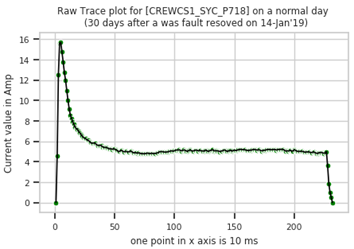
The above figure is a representation of an exemplary trace after segmentation of raw data. X axis shows the index of data points. One point in X axis equals 10 milliseconds. Y axis shows the current in amperes of the data points in X axis. The hybrid logic for segment of trace automatically identify the end of phases.

# **7 Exploratory data analysis**

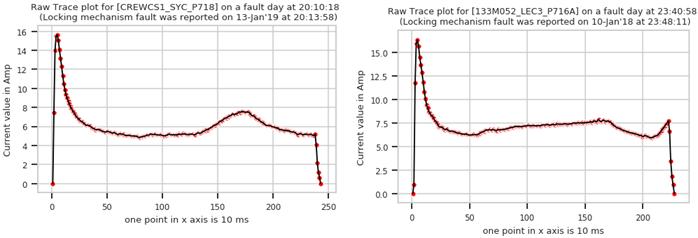
### **7.1 Phase 1 and Phase 2 Changes**

A sample of assets were selected for detailed EDA on traces. Following are the typical plots showing a normal trace (figure 16) and traces (figures 17, 18, 19) exhibiting fault.

These plots were basis for creating features capturing the structural deformities such as the dip in phase one current (17), rise in phase two current observed above 5 Amp (fig 18), dip in phase three current observed (fig 19)

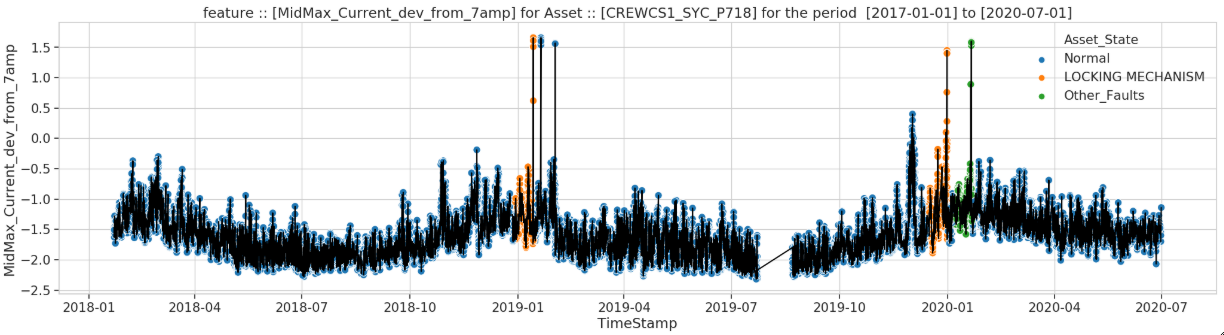


**Fig: 16 shows a normal trace Fig: 17 shows a trace from fault day**



**Fig: 18 shows a trace from fault day Fig: 19 shows a trace from a fault day**

Also, while conducting EDA, seasonality was observed in some features like maximum current in phase 2, hence a feature to capture seasonality was created. Below figure shows the seasonal trend for maximum current feature for asset CREWCS1\_SYC\_P718.

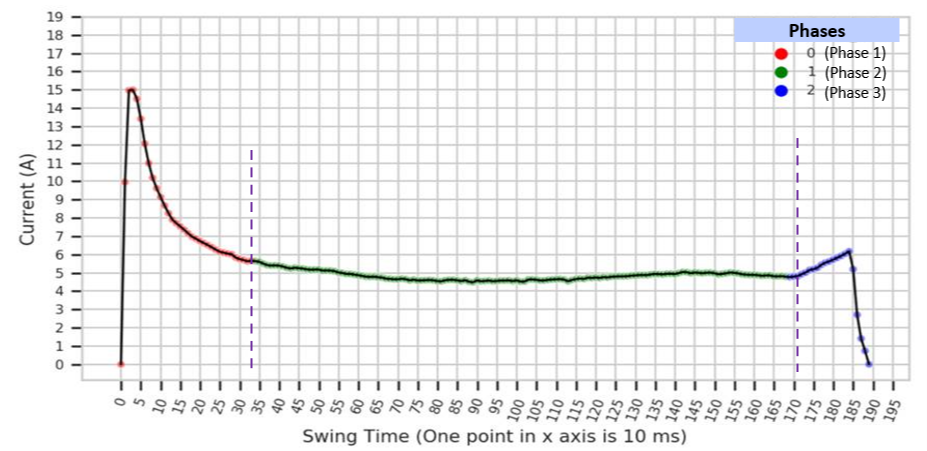


**Fig 20 Seasonal trend for asset CREWCS1\_SYC\_P718**

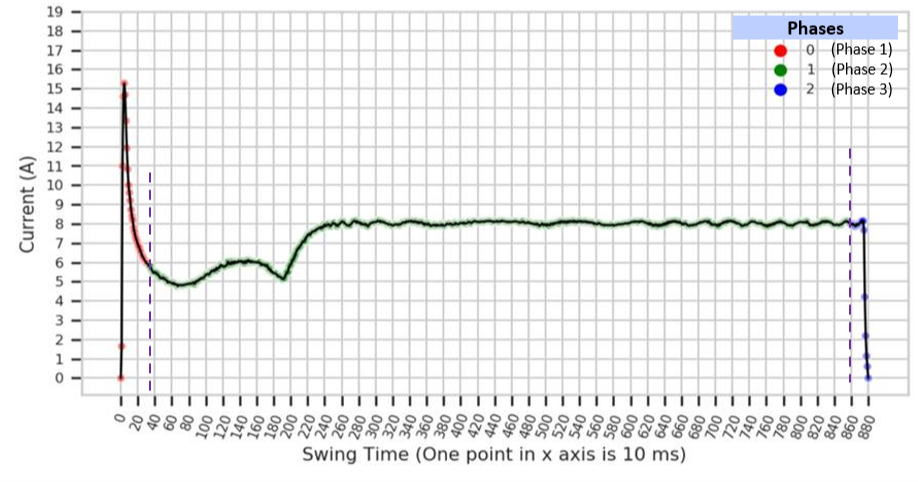
### **Phase 3**

**Representation of an exemplary normal trace and anomalous trace after segmentation by using hybrid logic**

The figures below are representation of an exemplary normal trace and anomalous trace after segmentation of raw data. X axis shows the index of data points. One point in X axis equals 10 milliseconds. Y axis shows the current in amperes of the data points in X axis. For visual inspection, segments of the traces have been marked with different colours into three phases. The unlocking phase marked with red colour, movement phase marked with green colour and the unlocking phase marked with blue colour.



**Fig 21: A typical normal trace and its phases using hybrid logic**



**Fig 22: A typical anomalous trace and its phases using hybrid logic**

# **8 Feature engineering**

## **8.1 Feature creation**

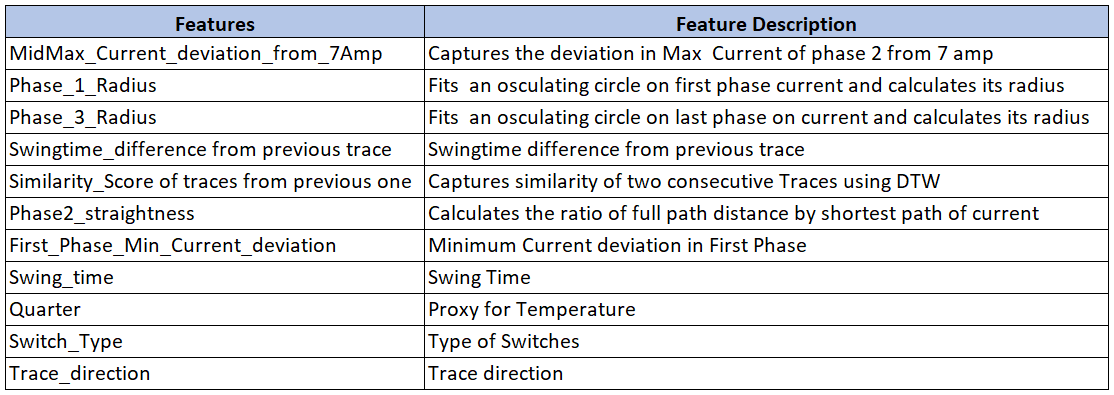
Feature creation was carried out based on outcome of EDA by closely observing the traces.

These features capture structural deformities and other variations in parameters like Swing Time, dip in current of phase one, rise in current of phase 2 above a threshold and other factors.

Trace Direction has also been considered as a feature as it was observed that sometimes the current values such as minimum current of phase 1 and maximum current of phase 2 are different for the two trace directions (RN/NR).

To capture seasonality, months have been grouped into three categories and used as features that would represent the temperature of those categorized months.

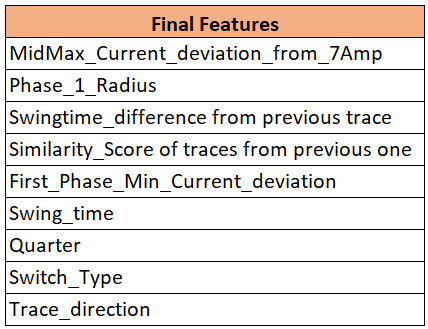
**Table 8.1a: Creation of features and feature description**



## **8.2 Phase 1 final features selection**

By visualising each feature and seeing the pattern before the fault occurred, 9 features were chosen for the final selection. Considering only the most important features from each phase of the trace helped in differentiating between a trace from a normal day and a trace from day of fault or previous days to fault showing some deformations.

**Table 8.1b: Final features used in Phase 1 statistical model**



## **8.3 Phase 2 update - final features selection**

In phase 2, the original 9 features were refined and 7 features were chosen for the final selection used in statistical model. This was done by EDA (Exploratory Data Analysis) and by considering the features from each phase of the trace in turn. This helped in differentiating between a normal or anomalous trace from day of fault or previous days to fault.

**Table 8.2: POE Phase 2 - Final set of features used in Phase 2 model**

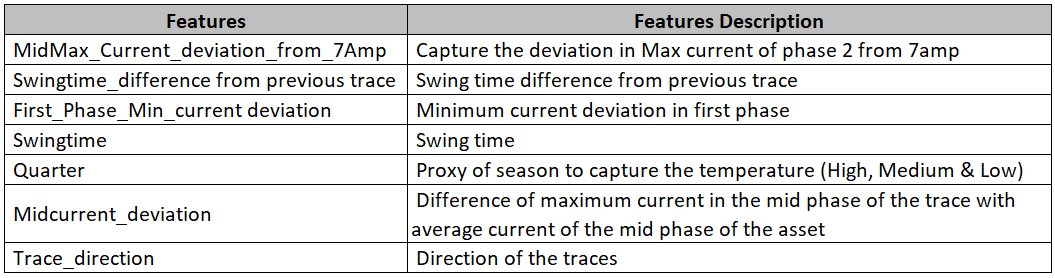


Table 8.3 shows that which are the features were kept as it was in POE phase 1, included or excluded in the final model in POE phase 2 and the explanation of inclusion and exclusion.

**Table 8.3: POE Phase 2 - Inclusion and exclusion of features and reasons**

|  |  |  |
| --- | --- | --- |
| **Features** | **Included/Excluded on Phase 2** | **Reason for Inclusion/Exclusion** |
| MidMax\_Current\_deviation\_from\_7Amp | From Phase 1 | - |
| Phase\_1\_Radius | Excluded | Based on EDA, too much variability for some assets |
| Swingtime\_difference from previous trace | From Phase 1 | - |
| Similarity\_Score of traces from previous one | Excluded | Based on EDA, too much variability for some assets |
| First\_Phase\_Min\_current deviation | From Phase 1 | - |
| Swingtime | From Phase 1 | - |
| Quarter | From Phase 1 | - |
| Switch\_Type | Excluded | Preliminary results indicated that inclusion of this result significantly increased FP% |
| Midcurrent\_deviation | Included | In Phase 2 'It was observed that different assets have different current ranges in mid phase |
| Trace\_direction | From Phase 1 | - |

## **8.4 Phase 3-Final feature selection**

Feature creation was carried out based on deep dive discussions with the NR team and the outcome of EDA by closely observing the traces. By visualising each features and seeing the pattern before the fault occurrence, 13 features were built. Out of these 13 features, 10 features were developed on new logic and 3 existing features from POE phases 2.

Considering only the most important features from each phase of the trace helped in differentiating between a trace from a normal day and a trace from day of fault or previous days to fault showing some deformations. In the current phase out of 13 features, 10 new features were developed from 3 phases of a trace based on the new segmentation logic. The feature creation was carried out like, 2 features from unlocking phase, 7 features from movement phase, 1 feature from locking phase and 3 other features from other data information.

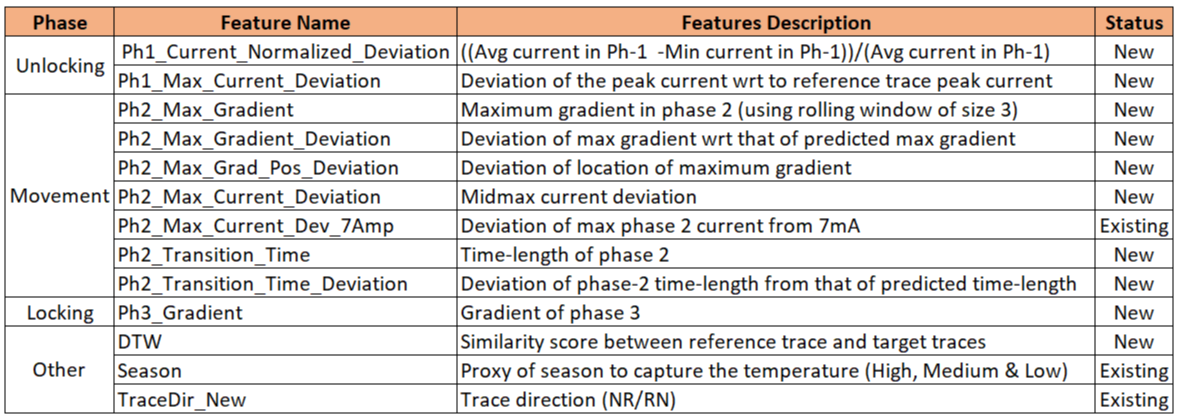
Below figure shows that how many features were developed from which phases of a trace and total number of features built in POE Phase 3 – Stage 2.



Fig 23: Feature creation and extraction in POE phase 3

Table 8.4b shows that which are the features were developed from which phases of trace, feature description and status of the features (‘New’ – Features developed in POE Phase 3 – Stage 2 and ‘Existing’ - kept as it was in POE phase 2).

**Table 8.4b: Phases wise creation of features, feature description and status**



# **9 Model building and evaluation of model performance**

## **9.1 Modelling approach**

52 assets were selected out of 79 Clamp Lock Mark II type assets.

Some assets were not considered due to data quality issues such as incomplete data, inconsistency in current range and clustering analysis.

The data has been segmented into traces and features have been created that attempt to differentiate between a normal trace and a trace relating to a fault.

A typical model needs labels to learn the pattern and differentiate between classes. In this scenario the data can labelled only at a day level and it is challenging to label the traces based on this data. Hence, a one class model was used to classify the traces into anomalous or normal. This type of model does not require labelled data for training.

## **9.2 Evaluation of model performance**

The model was trained on trace level data and then the trace level predictions were aggregated at day level for evaluation. All the days were marked either as -1(anomalous) or 1(normal) based on the number of traces observed as anomalous on that day.

Model evaluation here is challenging as traces are not labelled as anomalous and normal to begin. So, standard definition of True Positives and False positives does not apply here.

Here True Positives (TP) represents the number of successful predictions of unique failures within a 7 day window from a fault date and False Positives (FP) depicts the number of days predicted as anomalous by the model out of all the days outside the 7 day window.

Models performance was evaluated using different thresholds while converting the results from trace level to day level. In table 9.1, Threshold of 11% means a day will be marked as anomalous if more than 11% of the traces pertaining to that day are predicted as anomalous by the model. Model performance has been evaluated for different thresholds.

**Table 9.1: Phase 1 Model Results with different thresholds**

|  |  |
| --- | --- |
| **No\_ Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **65.7%** | **21.4%** |
| **11% Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **47.3%** | **11.4%** |
| **20% Threshold** | |
| 7\_days\_Window (52\_Assets) Day\_level | |
| TP | FP |
| **39.3%** | **7.9%** |

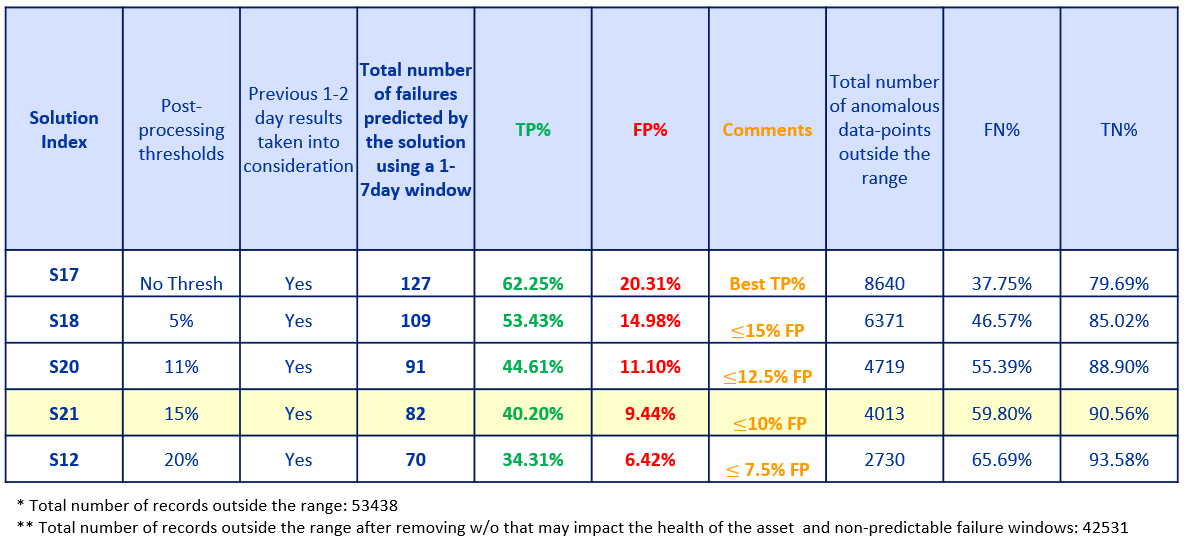
**Phase 2 update**

**Result of POE solutions**

Models performance was evaluated using different thresholds while converting the results from trace level to day level. For this analysis 52 assets were selected out of 79 Clamp Lock Mark II type assets and 204 total number of 'predictable' failures out of 300 total number of failures provided by the NR team. The analysis was done using one class model to classify the traces into anomalous or normal.

Table 9.2 shows the evaluation of model performance of different solutions by using different thresholds and their performance metrics like True Positive Percentage (TP%) and False Positive Percentage (FP%). The best solution from a business perspective may be chosen by selecting the appropriate combination of TP% and FP%.

**Table 9.2: Result of POE solutions**



Note: Please note these results should be considered as interim and are subject to additional verification and updating during phase 3.

Please find the definition of the columns of the above table:

**A. Solution Index:** Solutions were created using one class model based on various thresholds and conditions.

**B. Post processing threshold:** Various thresholds were applied in aggregated results to grade a day level data points as anomalous or normal.

For example, consider the case where 100 traces on a particular day and 5 out of those 100 traces were classified as 'anomalous' by the model (95 traces were classified as 'normal' by the model). If a post-processing threshold of 5% was applied to the aggregated result, then (in this context) the day level data point will be marked as anomalous as 5% or more of the traces were classified as anomalous by the model. Similarly, 15% post processing threshold means that the day has been marked as anomalous if more than 15% of traces on that day are found to be anomalous.

**C. Previous 1-2 day results taken into consideration:** It is believed that in this type of analysis, isolated predictions generally does not lead to a fault but a group of predictions or anomalies leads to a fault. To capture these type of events, an algorithm has been implemented that takes into consideration the previous prediction.

**D. Total number of failures predicted by the solution using a 1-7 day window:** If there were at least one anomalous day within 1-7 days before a failure, it was marked as predicted

**E. TP% (True Positive %):** (Number of at least one prediction within the marked period / Total number of predictable failures)\* 100

**F. FP% (False Positive %):** (Total number of anomalous data points outside the range / Total number of day level data points outside the marked period (adjusted)) \* 100

**G. Comments:** Selection of best solution based on the cut off criteria of FP%

**H. Total number of anomalous data-points outside the range:** How many day level data points were graded as anomalous (number of predictions) by the solution outside the marked period across all assets/failures within the 3.5 year period.

For calculation of False Positive percentage, following data is being removed:

1. 7 days prior to work order days impacting POE
2. Fault resolving days, for calculating fault-resolving days, InOrder\_Date has been considered.

**I. FN% (False Negative):** 100 - TP%

**J. TN% (True Negative)**: 100 - FP%

**Comparison of the performance of the solutions with that of Wonderware alerts & alarms**

It is not like to like comparison. A methodology has been implemented to compare the Wonderware result with POE solution. The methodology is, a single alert or an alarm within the marked period (1-7day) was considered as a True Positive for Wonderware; any alert or alarm outside the marked period was considered as false positive.

Below figure shows the comparison of the performance of the solutions with that of Wonderware alerts & alarms. In this plot, the Y-axis represents the False Positive percentages (FP%) and X-axis represents the True Positive percentages (TP%).Also, the square shape represents the respective solutions using

different thresholds and the diamond shape represents the Wonderware alerts and alarms.

**Fig 23: Performance comparison between Wonderware alerts & alarm and solution index**

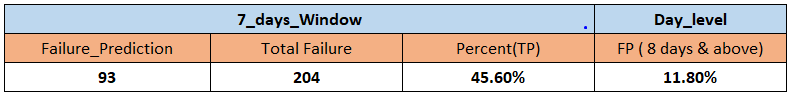
Note: Please note these results should be considered as interim and are subject to additional verification and updating during phase 3

**Wonderware alert and alarms analysis**

**Wonderware Statistics:**

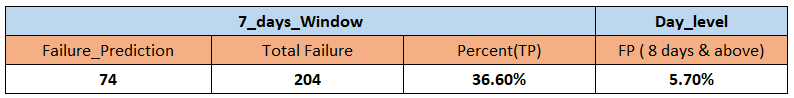
Table 9.3 represents the result of Wonderware both alerts and alarms prediction within 1-7 day window. In this analysis, a day has been marked as anomalous here if there is at least a single Alert or Alarm on that day

**Table 9.3: Result of Wonderware Alerts & Alarms**



Similarly, table 9.4 represents the result of only Wonderware alarms prediction within 1-7 day period (in this table alerts were ignored). However, in this analysis, a day has been marked as anomalous here if there is at least a single Alarm on that day.

**Table 9.4: Result of Wonderware Alarm only**



Note: For Wonderware system only acknowledged and returned alerts and alarms are considered.

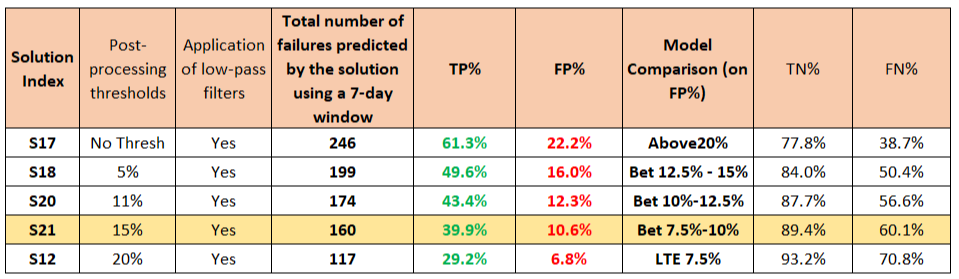
## **Phase 3 Updates**

### **9.3.1 Stage 1 Model validation on additional assets (94 assets)**

As mentioned, 52 assets were used to train the one class model to classify the traces into anomalous or normal and 94 assets were selected out of 131 Clamp Lock Mark II type assets and 401 total number of failures to validate the one class model build on 52 assets. Model performance was evaluated using different thresholds while converting the results from trace level to day level.

Table 9.3 shows the evaluation of model performance of different solutions by using different thresholds on 94 assets and their performance metrics like True Positive Percentage (TP%) and False Positive Percentage (FP%). The best solution from a business perspective may be chosen by selecting the appropriate combination of TP% and FP%. However, the figures in the results are conservation as we need to check the result on “Predictable Failures” for 94 assets.

**Table 9.3: Model Validation result of 94 assets**

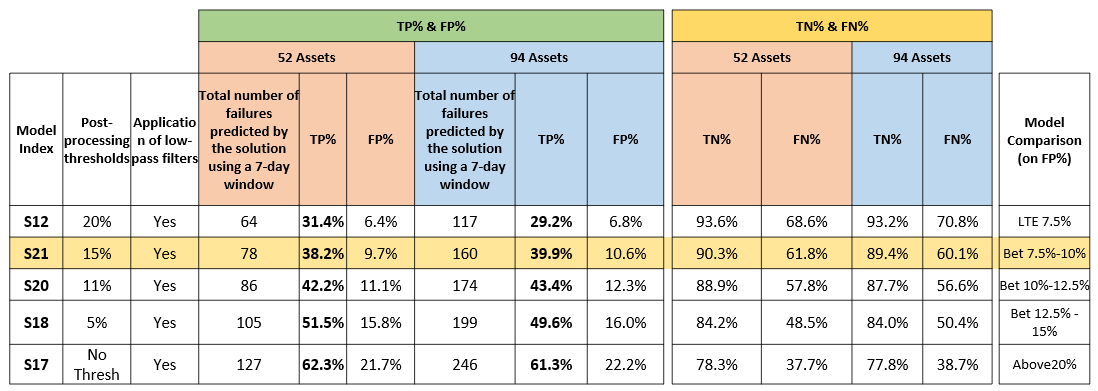


### **9.3.2 Stage 1 - Model Comparison (52 Vs 94 Assets)**

**Comparison of Results on 52 Assets Vs 94 Assets of POE solutions**

Table 9.4 shows the comparison of model performance of different solutions by using different thresholds on 52 and 94 assets and their performance metrics like true positive percentage (TP%) and false positive percentage (FP%). The best solution from a business perspective may be chosen by selecting the appropriate combination of TP% and FP%. The result of 94 assets are consistent when compared with 52 assets result.

**Table 9.4: Comparison of POE solution result - 52 vs 94 Assets**



### **9.3.3 Model building approach**

During phase 3 – Stage 2, model building was same as considered in phase 2 and one class model was used to classify the traces into anomalous or normal.

**Evaluation of model performance**

The model performance was evaluated by using two different approaches. In the first approach, all 13 features were used to train the one class model. The model was trained on trace level data and then the trace level predictions were aggregated at day level for evaluation. All the days were marked either as -1(anomalous) or 1(normal) based on the number of traces observed as anomalous on that day. Table 9.5a shows the list of 13 features used in first approach to train the one class model.

**Table 9.5a: All 13 features used in first approach to train the one class model**

|  |
| --- |
| Features |
| Ph1\_Max\_Current\_Deviation |
| Ph1\_normalized\_current\_deviation |
| Ph2\_Max\_Gradient\_Deviation |
| Ph2\_Max\_Gradient |
| Ph2\_Max\_Grad\_Pos\_Deviation |
| Ph2\_Transition\_Time\_Deviation |
| Ph2\_Transition\_Time |
| Ph2\_Max\_Current\_Deviation |
| Ph2\_Max\_Current\_Dev\_7Amp |
| Ph3\_Gradient |
| DTW |
| Season |
| TraceDir\_New |

However, in second approach, few features were removed statistically (one by one) and final 9 features were selected to train the one class model. Table 9.5b shows the list of 9 features which were used to train one class model out of 13 features.

**Table 9.5b: 9 features used in second approach to train the one class model**

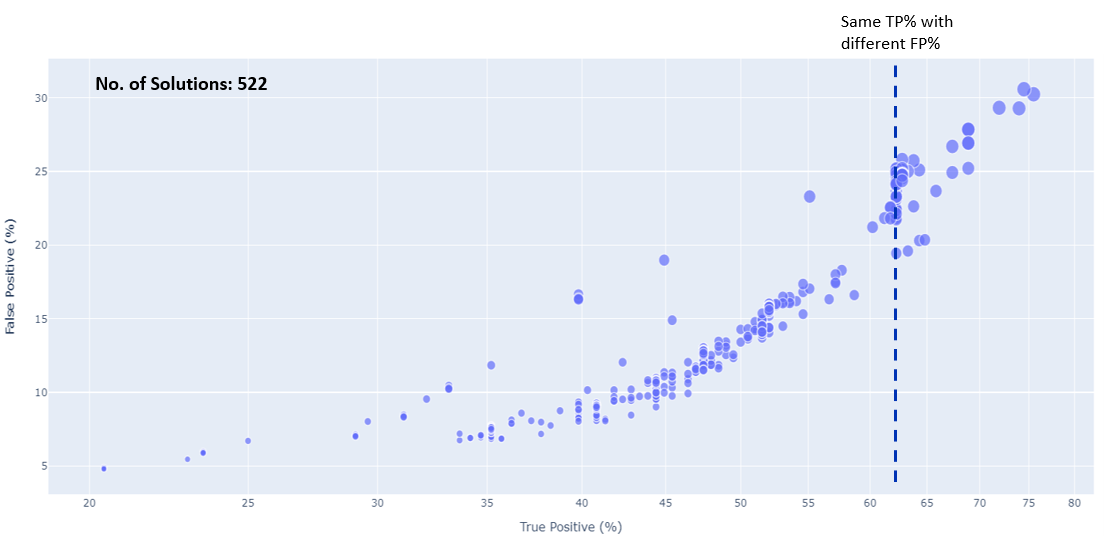
|  |
| --- |
| Features |
| Ph1\_Max\_Current\_Deviation |
| Ph2\_Max\_Gradient\_Deviation |
| Ph2\_Transition\_Time\_Deviation |
| Ph2\_Max\_Current\_Deviation |
| Ph2\_Max\_Current\_Dev\_7Amp |
| Ph3\_Gradient |
| DTW |
| Season |
| TraceDir\_New |

**Optimal – solutions selection approach**

Previously, in phase 2, model performance was evaluated ‘manually’ using different post processing & low pass filter thresholds and a total of 32 solutions were created while converting the results from trace level to day level. Out of 32 solutions, 5 optimal solutions were provided with highest TP% within specified FP% cut-off.

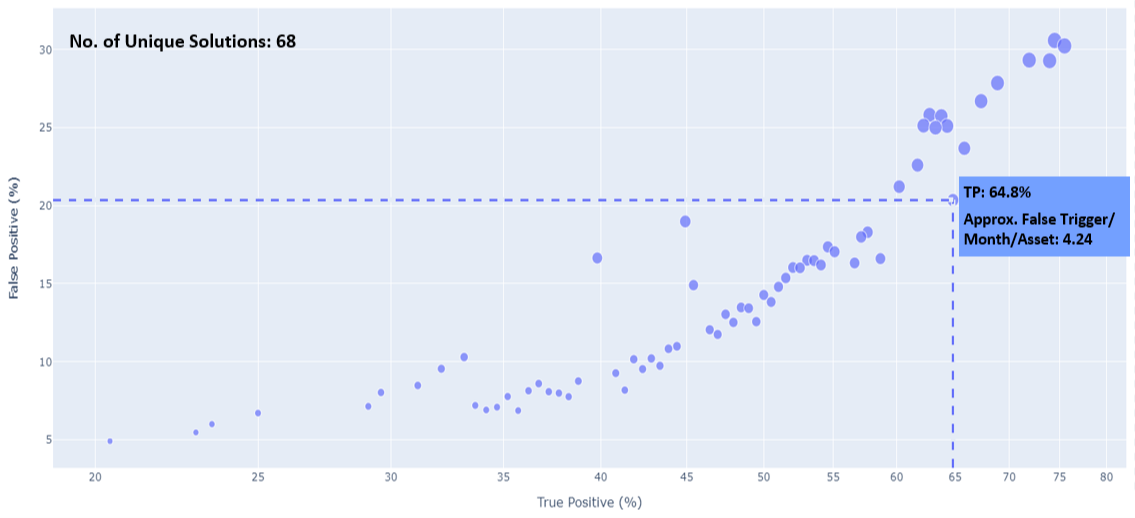
However, in phase 3 – Stage 2, optimal solutions ware calculated using a different logic and “programmatically” creating multiple solutions using different post processing thresholds & low pass filters. The final result from this approach provided multiple unique solutions with highest TP% along with lowest FP% within specified FP% cut-off. This statistical approach provided with optimal solutions and would allow users to consider solutions based on business requirements. Also, ‘**Approx False Triggers/Month/Assets**’ as an indicator was created to get an idea of approximate average number of false triggers/month/assets for every solution. The formula to calculate the Approx. False Triggers/Month/Assets is:

Below figure shows programmatically” creating 522 solutions using different post processing thresholds & low pass filters and their performance metrics like true positive percentage (TP%) and false positive percentage (FP%). X axis represents the True Positive % and Y axis represents the False Positive %. Every circle in the plot represents a solution with its TP% and FP% and the radius of the circle is proportional to approx. average number of False Trigger/ Month/Asset. However, the result of 522 solution returns different FP% for same TP%.



**Fig 24: Programmatically creating multiple solutions (total 522 solutions)**

Figure 25 is the refined version of Fig 9.5c where multiple unique solutions were provided with highest TP% along with corresponding lowest FP%. In the final result, different FP% for the same TP% were filtered out and provided the highest TP% along with the lowest FP%. It shows, after removing the different FP% for same TP% total 68 unique solutions obtained from 522 solutions. The best solution from a business perspective may be chosen by selecting the appropriate combination of TP%, FP% and Approx. False Triggers/Month/Assets.



**Fig 25: Selecting the final optimal solutions**

**Model Performance of POE solutions on various assets**

**Checking the performance of model on 51 and 91 assets**

In Phase 3 – Stage 2, models performance was evaluated on 51 assets out of 79 Clamplock Mark 2 type assets and 91 assets out of 131 Clamplock Mark 2 type assets using different thresholds while converting the results from trace level to day level. For, 51 assets, there were total 196 number of 'predictable' failures out of 300 total number of failures and for 91 assets, there were 401 total number of 'predictable' failures out of 584 total number of failures provided by the NR team. The analysis was done using one class model to classify the traces into anomalous or normal.

The analysis was carried out with 13 features and 9 features separately by using one class model on 51 and 91 assets.

Table 9.5e and 9.5f shows the model evaluation performance on 51 and 91 assets by using 13 and 9 features within specified FP% cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA). The best solution from a business perspective may be chosen by selecting the appropriate combination of TP% and FP%.

**Table 9.5e: Result on 13 features using one class model**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest result using 13 Features** | | | | | | | | |
| **Asset Description** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 51 Assets | 36.22 | 1.80 | 47.45 | 2.71 | 53.06 | 3.23 | 60.71 | 4.53 |
| 91 Assets | 17.71 | 1.98 | 26.43 | 3.03 | 32.92 | 4.00 | 36.41 | 4.96 |

**Table 9.5f: Result on 9 features using one class model**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest result using 9 Features** | | | | | | | | |
| **Asset Description** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 51 Assets | 37.76 | 1.77 | 46.94 | 2.64 | 51.53 | 3.47 | 58.16 | 4.43 |
| 91 Assets | 18.45 | 1.91 | 26.93 | 2.82 | 33.17 | 3.84 | 35.41 | 4.80 |

The summary of the one class model output in Table 9.5e & 9.5f shows that there were insignificant difference between two results while using 13 and 9 features on 51 and 91 assets.

**Checking the performance of model on S&C Turnout grouping assets**

The performance of the one class model was also checked on different hierarchy of S&C Turnout grouping assets and produced the results within specified FP% cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA). Since, there were insignificant changes in model performance on 51 and 91 assets by using 13 & 9 features, keeping that in mind, 9 features were used to build one class model on different hierarchy of S&C Turnout grouping assets.

However, initial hypothesis was that considering different hierarchy of S&C Turnout grouping assets and building model on different hierarchy lead to improvement in model performance. Table 9.5g shows the model performance on different hierarchy of S&C Turnout grouping assets by using 9 features.

**Table 9.5g: Result on 9 features using one class model on different hierarchy of S&C Turnout grouping assets**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest result using 9 Features** | | | | | | | | |
| **Assets Description** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| S&C Turnout | 26.06 | 1.64 | 48.07 | 2.55 | 56.76 | 3.13 | 64.86 | 3.94 |
| S&C Turnout + Full Depth | 23.78 | 1.65 | 47.79 | 2.55 | 57.34 | 3.21 | 66.67 | 3.93 |
| S&C Turnout + Full Depth + C,D&E | 22.37 | 1.73 | 45.55 | 2.54 | 55.80 | 3.41 | 67.65 | 4.00 |
| \* Do not have the "Predictable Failure" for Grouping Assets | | |  |  |  |  |  |  |

However, there were few drawbacks of using S&C Turnout grouping assets:

* Too many groups lead to smaller number of assets in a single group which is statistically incorrect. For example, S&C Turnout + Full Depth + A&B group contain 9 assets only and S&C Turnout + Shallow Depth + C, D&E group contains 25 assets only.
* For every groups, require different model to build which will be very cumbersome.

After careful consideration and analysis of S&C Turnout grouping assets result, there we no evidence that different hierarchy of S&C Turnout grouping assets lead to improvement in model performance.

**Phase 3 update – Stage 2 – Selection of final features by comparing the result on POE phase 2 & 3 features**

For, final selection of features, POE Phase 2 & POE Phase 3 – Stage 2 features were used on 51 and 91 Clamp Lock Mark II type assets using different thresholds while converting the results from trace level to day level. The analysis was done using one class model to classify the traces into anomalous or normal. The final 7 features were used from POE phase 2 and 13 & 9 features respectively were used from POE phase 3 – Stage 2 to build the one class model.

Table 9.6a and 9.6b shows the model evaluation performance on 51 and 91 assets by using phase 2 and phase 3 – Stage 2 features within specified FP% cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA).

**Table 9.6a: Comparison of result on 51 assets with POE phase 2 & 3 – Stage 2 features using one class model**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest results on 51 Assets** | | | | | | | | | |
| **Features used to build model (POE Phase)** | **Asset Description** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 7 Features (Phase 2) | 51 Assets | 44.90 | 2.08 | 55.61 | 3.11 | 66.84 | 4.01 | 72.45 | 4.54 |
| 13 Features (Phase 3 – Stage 2) | 51 Assets | 36.22 | 1.80 | 47.45 | 2.71 | 53.06 | 3.23 | 60.71 | 4.53 |
| 9 Features (Phase 3 – Stage 2) | 51 Assets | 37.76 | 1.77 | 46.94 | 2.64 | 51.53 | 3.47 | 58.16 | 4.43 |

**Table 9.6b: Comparison of result on 51 assets with phase 2 & 3 – Stage 2 features using one class model**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest results on 91 Assets** | | | | | | | | | |
| **Features used to build model (POE Phase)** | **Asset Description** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 7 Features (Phase 2) | 91 Assets | 41.15 | 2.1 | 50.37 | 3.19 | 62.34 | 4.32 | 66.83 | 4.95 |
| 13 Features (Phase 3 – Stage 2) | 91 Assets | 17.71 | 1.98 | 26.43 | 3.03 | 32.92 | 4.00 | 36.41 | 4.96 |
| 9 Features (Phase 3 – Stage 2) | 91 Assets | 18.45 | 1.91 | 26.93 | 2.82 | 33.17 | 3.84 | 35.41 | 4.80 |

The summary of Table 9.6a and Table 9.6b shows that model evaluation performance from POE phase 2 features showing better result where TP% are approx. 6%-8% higher (although AFTMA are slightly higher) within specified FP% cut-off compared to POE phase 3 – Stage 2 results.

**Model Deployment**

After careful consideration on model performance, POE phase 2 features have been provided the best result and considered as final model for deployment.

**Phase 3 update – Stage 2 – Training Conventional Model**

**Training the conventional model-Why are we training conventional model?**

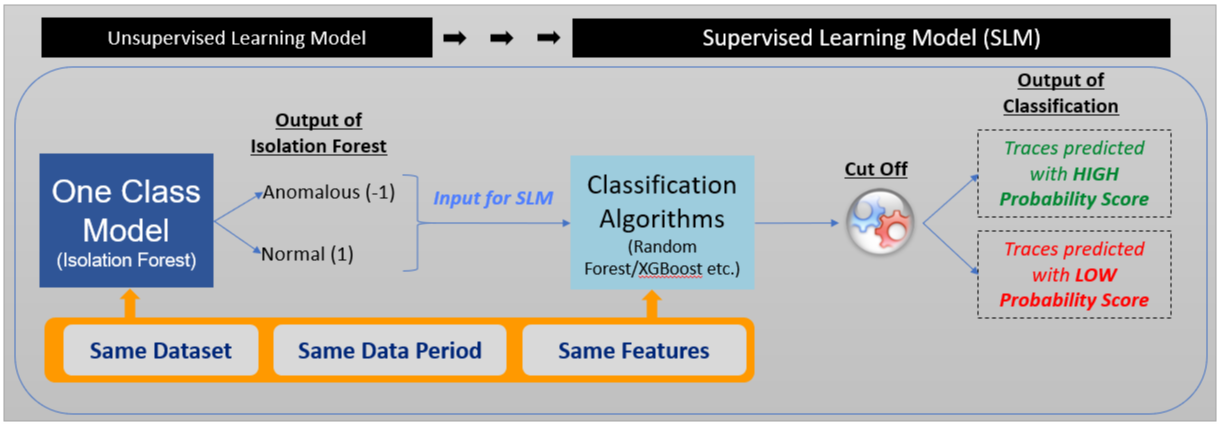
A typical statistical model needs labels to learn the pattern and differentiate between classes. However, in POE, the data was not labelled so that any conventional classification algorithms cannot be used to classify the traces as anomalous or normal. One class model which is one of the best unsupervised learning algorithms has been used which does not require the data to be labelled for training. One class model tried to learn the pattern of the traces during training of a model and classify the traces into anomalous and normal.

In POE, one class model has been used to classify the traces into anomalous (-1) and normal (1) using various features created in different POE phases on different group of assets. The model was trained on trace level data and then the trace level predictions were aggregated at day level for evaluation.

However, there are some drawbacks in one class model related to confidence score and deployment of the final model. In this analysis, classification algorithms like Random Forest, XGBoost etc. help to overcome these drawbacks.

In this analysis, classification algorithms which are also call supervised learning algorithms like Random Forest, XGBoost etc. has been used and a typical classification algorithm, in general, is a function that weighs the input features so that the output separates one class into positive values and the other into negative values. Unlike, one class model, classification algorithms require data to be labelled to classify the traces into anomalous and normal. However, the output of one class model has been used as input to train the classification models. The same assets, datasets, same data periods and same features which were used to train one class model also used the same to train the classification models. However, the output of classification algorithms produce two classes and confidence score which is nothing but probability score belonging to each output class. Based on the selected probability cut-off criteria a trace predicted with high probability score will be mark as anomalous and low probability score will be mark as normal.

Below figure shows the journey of analysis from one class model to classification model and use of conventional algorithms.



**Fig 26: Use of conventional classification algorithms**

### **9.3.4 HW 1000 and 2000 series changes**

The objective of this analysis is to determine if the final existing 7 features, the trace segmentation logic and statistical model which have been used in POE Phase 3 can be reused for different type of assets like HW 1k &2K series assets data. The idea is to check whether the model built for Clamp Lock Mark II can be used for different other type of assets.

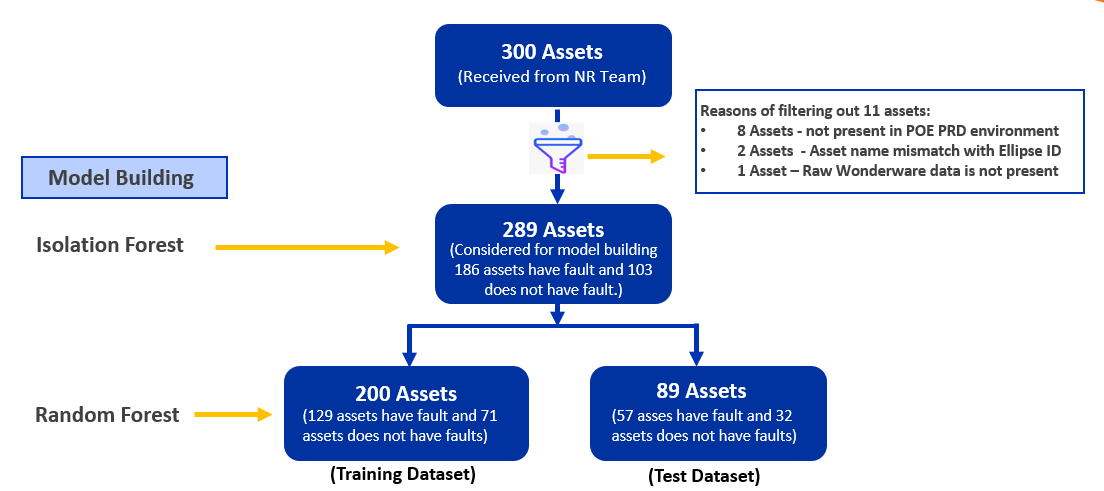
However, to validate the objective mentioned above, the logic to extract the traces, segmentation of traces into three phases, feature creation and modelling logic have been kept same as final phase of POE.

#### **9.3.4.1 Data understanding & Quality of data**

A list of 300 HW 1000 & 2000 series assets were received from the NR team. 289 assets from the list of 300 assets were used for feature extraction and modelling purpose. For 11 assets, data were not available because of following reasons:

1. Data for 8 assets was not available in local environment.
2. There was asset Name and EllipseID mismatch for 2 assets
3. There was one asset for which raw wonderware data was not available

Below figure shows the context of the HW 1K & 2k series assets and model building using how many assets.

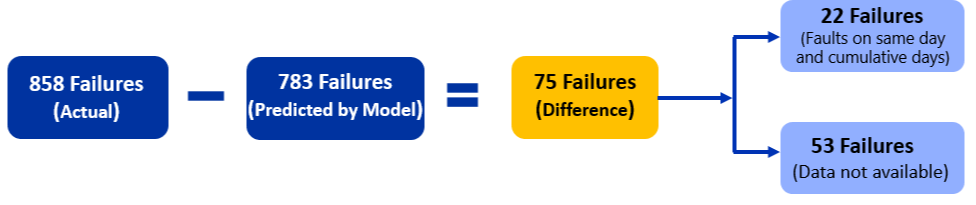


**Fig 27: Context of the assets**

#### **9.3.4.2 Context of failures (on 289 assets)**

Out of 289 assets, 186 assets had historical fault data and 103 assets did not have any historical fault. For 289 Assets, there were total 858 failures were reported against them. However, out of these 858 failures, 22 of them were reported on the same day and raw wonderware data was not available for 1 – 7 days window before the fault for 53 failures.

Below figure shows the context of failures on 289 assets and data quality



**Fig 28: Context of failure reported on 289 assets**

#### **9.3.4.3 Model Performance**

As mentioned above in section 9.7.2, to validate the model performance on HW 1K & 2K series assets, the same algorithms as well as same finalised features were used (which was used in POE Phase 3). Preliminary, 289 assets were used to build the Isolation Forest to classify the traces into anomalous or normal.

Table 9.7.4a shows the model evaluation performance on 289 assets by using 7 features within specified FP% cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA).

**Table 9.7.4a: Result on Isolation Forest**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest result on 289 Assets** | | | | | | | | | |
| **Model Built on Assets** | **Failures** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 289 Assets | 858 | 38.93 | 1.64 | 49.88 | 2.34 | 57.34 | 3.25 | 60.72 | 3.44 |

However, Random Forest was also used to check the confidence of the model on test data. Initially, random forest was trained on 200 assets (training data) and after that same trained random forest model was used to predict on test data (89 Assets).

Table 9.7.4b shows the model evaluation performance on test data (89 assets) by using finalised features within specified FP% cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers / Month / Assets (AFTMA).

**Table 9.7.4b: Result on Random Forest**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Random Forest Result on Test Dataset (89 Assets)** | | | | | | | | | |
| **Asset Description** | **Failures** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 89 Assets | 240 | 32.50 | 1.69 | 46.67 | 2.54 | 53.33 | 3.41 | 57.08 | 4.38 |

#### **9.3.4.4 Comparison of results on Clamp Lock Mark II Vs HW 1000 & 2000 series assets**

Table 9.7.5 shows the comparison of model performance of different solutions by using different thresholds on clamp lock mark II and HW 1K &2K series assets and their performance metrics like true positive percentage (TP%) and false positive percentage (FP%). The best solution from a business perspective may be chosen by selecting the appropriate combination of TP% and FP%. The summary of the model performance is consistent results observed across asset subtypes.

**Table 9.7.5: Comparison of POE solution result – Clamp Lock Mark II and HW 1K & 2K series assets**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Isolation Forest Result on Clamp Lock Mark II and HW 1K & 2K Series** | | | | | | | | | | |
| **Asset Group** | **Asset Description** | **Predictable Failures** | **Within 10% FP** | | **10%-15% FP** | | **15%- 20% FP** | | **20%-25% FP** | |
| **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| Clamp Lock Mark II | 51 Assets | 196 | 44.90 | 2.08 | 55.61 | 3.11 | 66.84 | 4.01 | 72.45 | 4.54 |
| 91 Assets | 401 | 41.15 | 2.1 | 50.37 | 3.19 | 62.34 | 4.32 | 66.83 | 4.95 |
| HW 1K & 2K Series | 289 Assets | 858 | 38.93 | 1.64 | 49.88 | 2.34 | 57.34 | 3.25 | 60.72 | 3.44 |

## **POE Model Performance Experiment**

### **9.4.1 Objective**

The objective of this experiment is to test if the performance of the solution improves or deteriorates linearly as more assets (Clamp Lock Mark 2) are added to the existing pool of assets.

To continue with the experiment, the finalised set of features, the trace segmentation logic and the finalised machine learning model from the previous iteration have been used.

Note: The experiment was carried on Isolation forest model only to classify the traces into anomalous and normal as per the agreement with NR SMEs.

### **9.4.2 Data understanding & selection of final assets**

For this experiment, total 848 Clamp Lock Mark 2 assets were received from the NR team and the data period was consider from January 2017 to June 2020. However, out of 848 assets, there were approx. 30% assets that did not have any fault history in the mentioned three and half year’s period.

Out of 848 assets, total 805 assets were used as data and the remaining 43 assets were not considered because of following reasons:

1. Data for 34 assets was not available in local environment.
2. There was some mismatch with Asset Name and EllipseID for 6 assets.
3. Other data issues for 3 assets.

### **9.4.2 Experimentation approach**

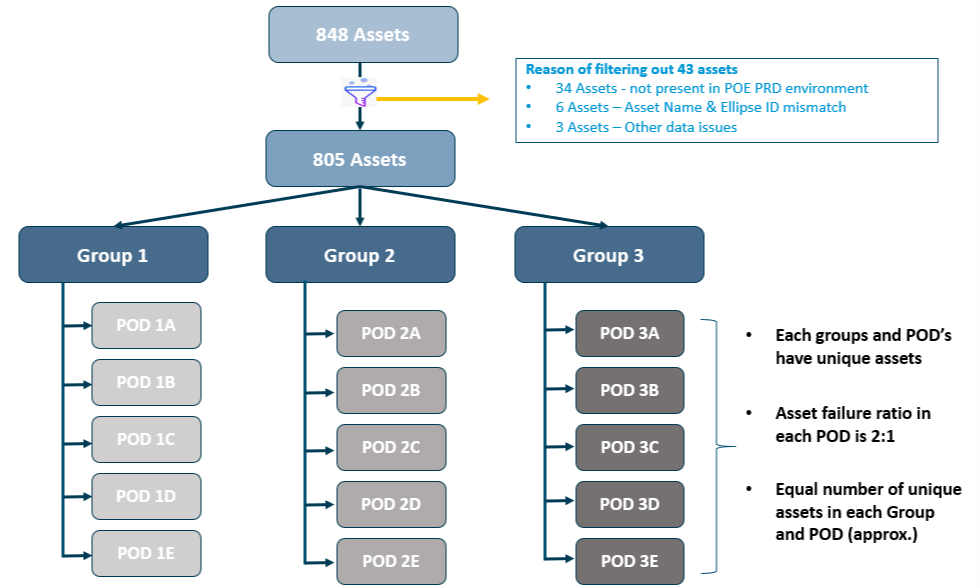
After discussion with the NR SMEs some approaches and rules have been considered to divide the data into groups & PODs and on model iterations. The detail approaches and rules are as follows:

#### **9.4.2.a Asset distribution approach**

Total 805 Clamp Lock Mark 2 assets were divided into three groups and each group consist of total five sub-groups or PODs. The rules that have been followed while dividing the number of assets into groups or sub-groups/PODs are:

1. Each group and POD must have unique number of assets.
2. Asset failure ratio (Assets which have failures in past: Assets which do not have any failures in past within the specified period) in each POD should be 2:1.
3. Each group and PODs within it must have approximately equal number of assets in them.

Fig. 9.4.2.a, shows the asset distribution approach and rules that have been considered to divide the 805 Clamp Lock Mark 2 assets into groups and PODs.

****

**Fig 9.4.2.a: Asset distribution approach and rules**

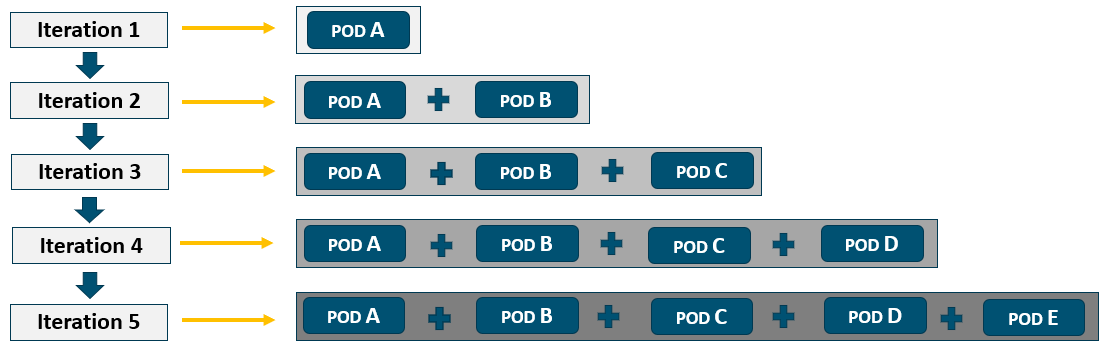
#### **9.4.2.b Model iteration approach**

In this experiment, statistical models (finalised ML Model) were built cumulatively on different PODs of assets to classify the traces into anomalous or normal..

For example, an asset group consist of five PODs/sub-group (POD-A to POD-E) and each POD consist of 50 assets. Also, each POD has equal number of unique assets and with a 2:1 asset failure ratio.

In the first iteration, the model was built on POD-A assets which consist of 50 assets and then the model performance is to be checked. In the second iteration, POD-B assets were added to the existing POD-A assets (with which the model had already been built), i.e., POD-A & POD-B which now consist of 100 assets was used to build model and the model performance was checked. In this way, gradually other PODs were cumulatively added to the existing set of PODs (with whichh the model had already been built) and model performance was observed.

Fig. 9.4.2.b, shows the approach that have been considered for model iterations.

****

**Fig 9.4.2.b: Model iteration approach**

### **9.4.3 Selection of final assets for model building**

Before building the model, few checks were made on the attributes of the traces to select final set of assets. Since our data was not labelled, the statistical model using Isolation Forest algorithm helped to classify the traces of the assets into anomalous or normal. The checks were made through few stages and the stages are as follows.

#### **9.4.3.a Initial stage**

As mentioned above, initially, 805 Clamp Lock Mark 2 assets were used to create three groups and five PODs for each group by maintaining approx. equal number of assets into groups & PODs. Each group consist of approx. 268 assets and each POD consist of approx. 54 assets. However, after building statistical model cumulatively on few PODs, the model evaluation performance were unusual while comparing the result with other sets of previous results on same type of assets.

#### **9.4.3.b Intermediate stage**

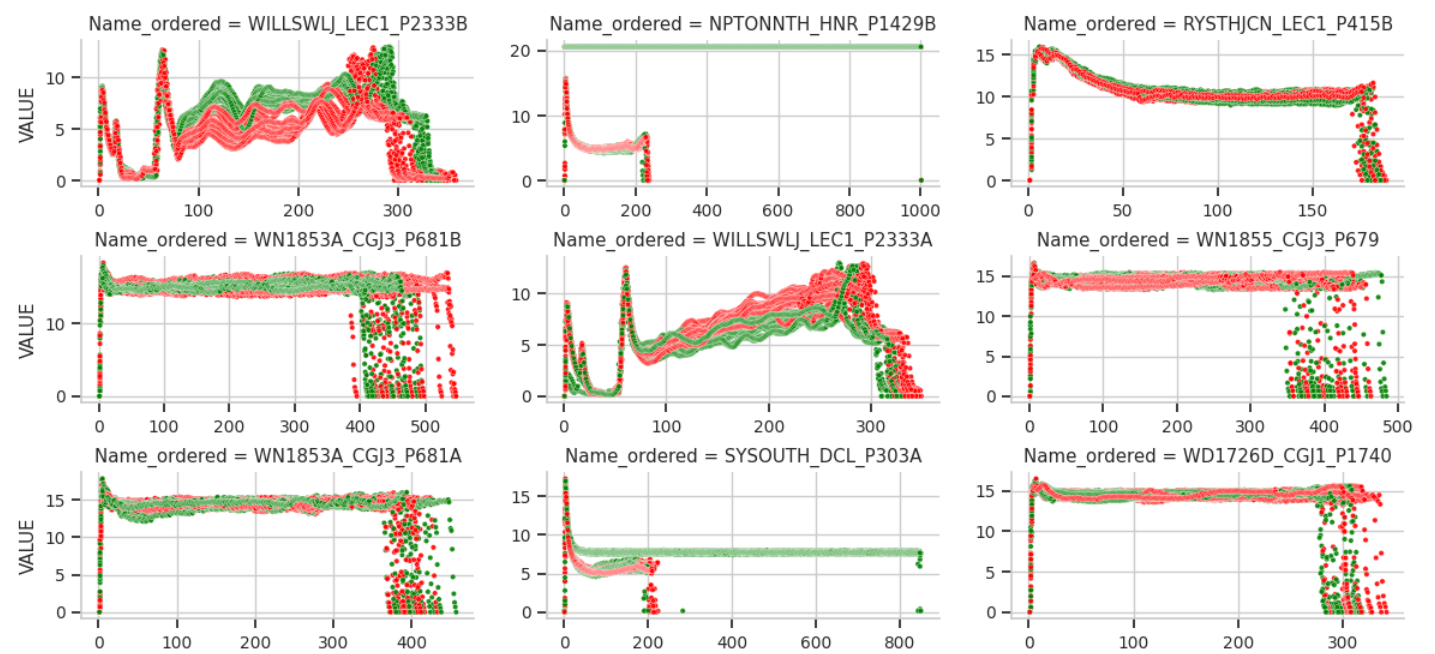
After observing the unusual model evaluation performance on few PODs, a detail analysis was carried out and visualised the traces of the assets. After the detailed analysis and visualisation of traces, It had been observed that a certain portion of assets in each group were behaving in a different way from therest of the assets.

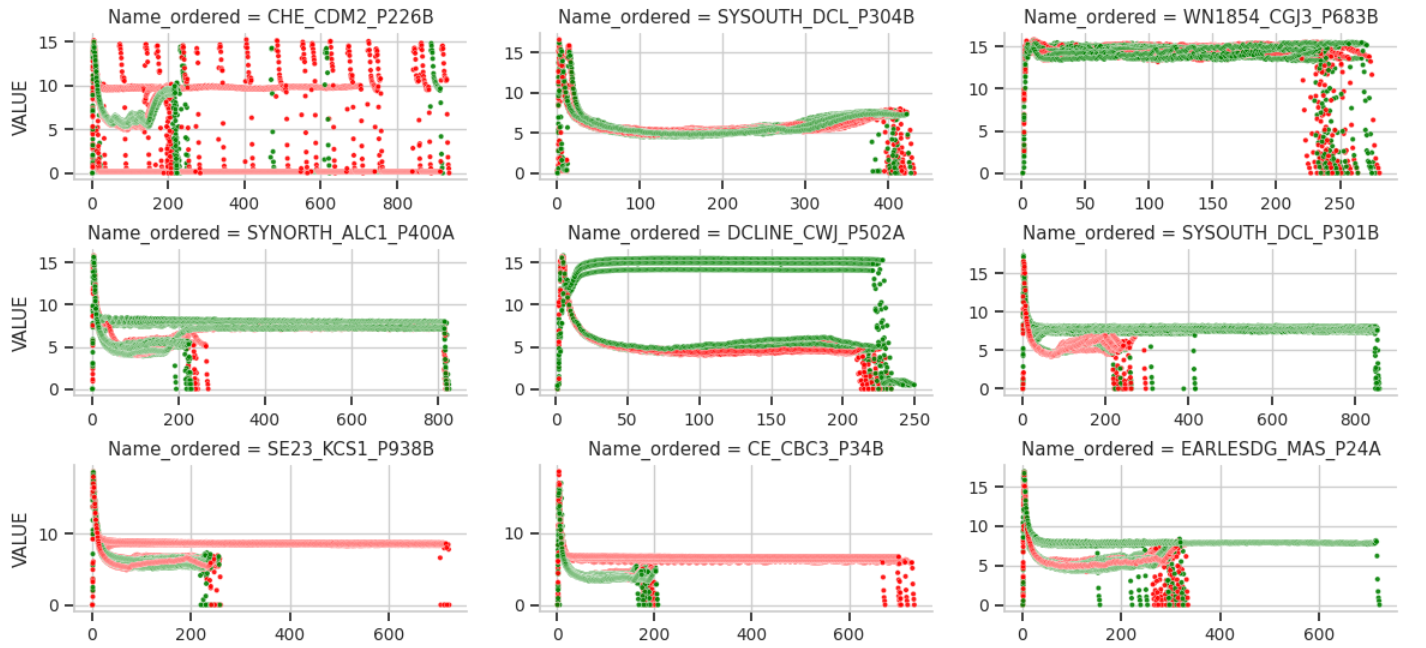
#### **9.4.3.c Final stage**

Finally, the assets which were behaving in a different way from the rest of the assets were removed from the experiment and the groups and PODs were recreated (following the same grouping rules mentioned earlier). After regrouping, each group was having approx. 233 assets and approx. 46/47 assets in each POD. A total of 120 assets were removed from the original 805 list of Clamp Lock Mark 2 assets for building statistical model cumulatively on groups/PODs of assets.

### **9.4.4 Data quality issues**

Fig 9.4.4 plots show the traces of few assets which were behaving in a different way from rest of the Clamp Lock Mark 2 assets. The unusual behaving assets were removed from final set of assets and the team proceed to build the model with the rest of the assets.

****



**Fig 9.4.4: Traces of few assets which are behaving different way from rest of Clamp Lock Mark 2 assets**

### **9.4.5 Model building & results**

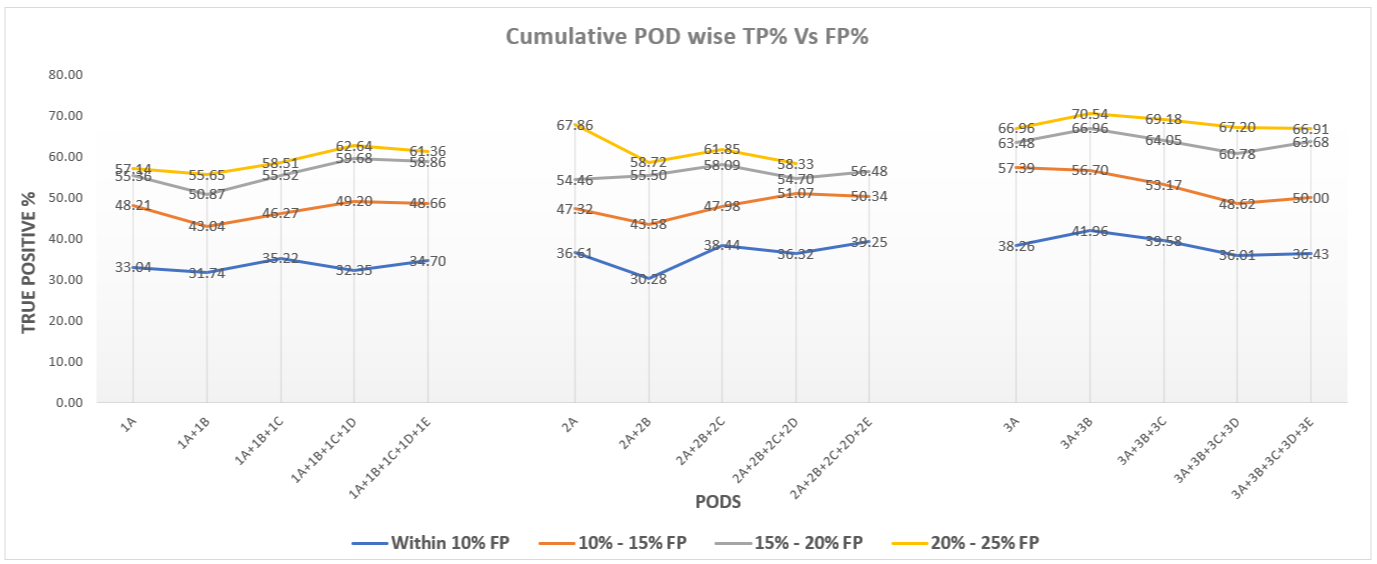
As mentioned above, for this experiment, finalised set of features, the trace segmentation logic and the finalised machine learning model (Isolation Forest) was used to classify the traces into anomalous or normal.

Table 9.4.5 shows the model evaluation performance on cumulative PODs for group 1, 2 & 3 assets by using finalised machine learning model and set of features within specified range of False Positive Percentage (FP%) cut-off and their performance metrics like True Positive Percentage (TP%) and Approx. False Triggers/Month/Assets (AFTMA).

**Table 9.4.5: Model performance on cumulative PODs for group 1, 2 & 3 assets**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model performance on Group 1 Clamp Lock Mark II assets** | | | | | | | | | | | | |
| **PODs** | **Assets Description** | | **Unique Failures** | | **Within 10% FP** | | **10% - 15% FP** | | **15% - 20% FP** | | **20% - 25% FP** | |
| **# of Assets** | **Cum. Assets** | **# on failures** | **Cum. Failures** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 1A | 47 | 47 | 112 | 112 | 33.04 | 1.70 | 48.21 | 2.42 | 55.36 | 3.03 | 57.14 | 3.58 |
| 1A+1B | 46 | 93 | 118 | 230 | 31.74 | 1.18 | 43.04 | 2.60 | 50.87 | 3.18 | 55.65 | 3.80 |
| 1A+1B+1C | 47 | 140 | 105 | 335 | 35.22 | 1.86 | 46.27 | 2.71 | 55.52 | 3.36 | 58.51 | 3.93 |
| 1A+1B+1C+1D | 46 | 186 | 104 | 439 | 32.35 | 1.75 | 49.20 | 2.73 | 59.68 | 3.55 | 62.64 | 4.32 |
| 1A+1B+1C+1D+1E | 47 | 233 | 120 | 559 | 34.70 | 1.78 | 48.66 | 2.73 | 58.86 | 3.42 | 61.36 | 4.18 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Model performance on Group 2 Clamp Lock Mark II assets** | | | | | | | | | | | | |
| **PODs** | **Assets Description** | | **Unique Failures** | | **Within 10% FP** | | **10% - 15% FP** | | **15% - 20% FP** | | **20% - 25% FP** | |
| **# of Assets** | **Cum. Assets** | **# on failures** | **Cum. Failures** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 2A | 45 | 45 | 112 | 112 | 36.61 | 1.89 | 47.32 | 2.74 | 54.46 | 3.78 | 67.86 | 4.6 |
| 2A+2B | 45 | 90 | 106 | 218 | 30.28 | 1.77 | 43.58 | 2.68 | 55.50 | 3.48 | 58.72 | 3.94 |
| 2A+2B+2C | 45 | 135 | 128 | 346 | 38.44 | 1.79 | 47.98 | 2.64 | 58.09 | 3.60 | 61.85 | 4.47 |
| 2A+2B+2C+2D | 45 | 180 | 122 | 468 | 36.32 | 1.76 | 51.07 | 2.74 | 54.70 | 3.13 | 58.33 | 3.92 |
| 2A+2B+2C+2D+2E | 44 | 224 | 118 | 586 | 39.25 | 1.85 | 50.34 | 2.69 | 56.48 | 3.59 | - | - |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Model performance on Group 3 Clamp Lock Mark II assets** | | | | | | | | | | | | |
| **PODs** | **Assets Description** | | **Unique Failures** | | **Within 10% FP** | | **10% - 15% FP** | | **15% - 20% FP** | | **20% - 25% FP** | |
| **# of Assets** | **Cum. Assets** | **# on failures** | **Cum. Failures** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** | **TP%** | **AFTMA** |
| 3A | 46 | 46 | 115 | 115 | 38.26 | 1.77 | 57.39 | 2.53 | 63.48 | 3.55 | 66.96 | 4.15 |
| 3A+3B | 46 | 92 | 109 | 224 | 41.96 | 1.91 | 56.7 | 2.90 | 66.96 | 3.85 | 70.54 | 4.25 |
| 3A+3B+3C | 46 | 138 | 107 | 331 | 39.58 | 1.89 | 53.17 | 2.89 | 64.05 | 3.72 | 69.18 | 4.16 |
| 3A+3B+3C+3D | 46 | 184 | 105 | 436 | 36.01 | 1.92 | 48.62 | 2.81 | 60.78 | 3.69 | 67.20 | 4.72 |
| 3A+3B+3C+3D+3E | 44 | 228 | 102 | 538 | 36.43 | 1.92 | 50.00 | 2.76 | 63.68 | 3.81 | 66.91 | 4.56 |

Fig. 9.4.5.a represents the above table (Table 9.4.5) by plotted the result of group 1, 2 & 3 assets. The individual line chart represents the specified range of False Positive Percentage (FP%) cut-off and their performance metrics like True Positive Percentage (TP%) on cumulative PODs for group 1, 2 & 3.



**Fig 9.4.5.a: Model performance on cumulative PODs and their corresponding True Positive Percentage (TP%) within specified False Positive Percentage (FP%)**

### **9.4.6 Conclusion**

It has been observed from model evaluation performance that the True Positive Percentage (TP%) within the specified range of False Positive Percentage (FP%) were not continuously increasing or decreasing as more assets were added.

# **10 Appendices**

|  |  |
| --- | --- |
| **Navigation sequence** | **Section** |
| 10.1 | Data Engineering |
| 10.2 | Technology Landscape |
| 10.3 | Data Science |

## **10.1 Data engineering in detail**

### **10.1.1 Source & target location**

**View Structure: FMS & Ellipse:**

For POE use case, two SQL Views were created in Source Database to expose information for asset, FMS, work order. Details of the two views are as below.

|  |  |  |
| --- | --- | --- |
| Server Name | DB Name | SQL View Name |
| sqliidstpochub001.database.windows.net | ADS\_Data\_Mart | 1. V\_CF10\_ASSETFMS 2. V\_CF10\_ASSETWORKORDER |

1. **SQL View Structure: FMS & Ellipse**

1. **View Definition**

[DST].[v\_cf10\_AssetFMS] :

* Implemented the view to restrict data from 2017 to 2019 from table ADS\_DM.Dim\_Date
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_ASSET
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_SNC\_ASSET\_NAMEPLATE
* Implemented the view so that data is coming for length of equipment number is 12 from table ADS\_DM.DIM\_ASSET
* Implemented the view so that data is filtered by, Track signalling and POE assets

[DST].[v\_cf10\_AssetWorkorder] :

* Implemented the view to restrict data from 2017 to 2019 from table ADS\_DM.Dim\_Date
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_ASSET
* Implemented the view so that data is coming for ACTIVE\_ROW\_FLAG = 'Y' from table ADS\_DM.DIM\_WORKORDER
* Implemented the view so that data is coming for WORK\_ORDER\_STATUS = 'Closed' from table ADS\_DM.DIM\_WORKORDER
* Implemented the view so that data is coming for DSTRCT\_CODE = 'RTK1' from table ADS\_DM.DIM\_WORKORDER
* Implemented the view so that data is coming for DSTRCT\_CODE = 'RTK1' from table ADS\_DM.DIM\_STD\_JOB
* Implemented the view so that data is filtered by, Track signalling and POE assets

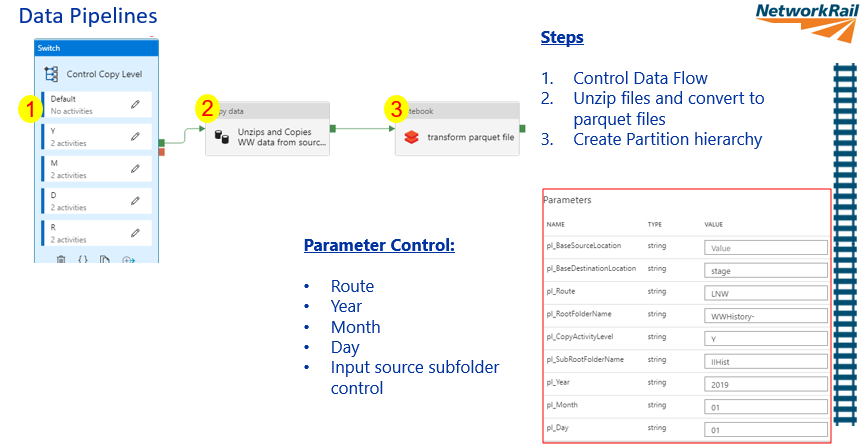
**Wonderware Data:**

|  |  |
| --- | --- |
| Source Storage | stgiidstpochub001 |
| Target Storage | Stgiiinnovationppinv |
| Source Container | nrii-sig-wonderware |
| Target Container | cf10/stage & cf10/prd\_poe |

### **10.1.2 Wonderware data - ingestion pattern**

An ADF pipeline will connect source location & bring the source files into NR innovation environment (ADLS location - Stage) as parquet files, according to user input. The STAGE file will be taken from Azure Databricks and the files will be segregated in an efficient partition structure.

Below image shows ADF pipeline & parameter control.

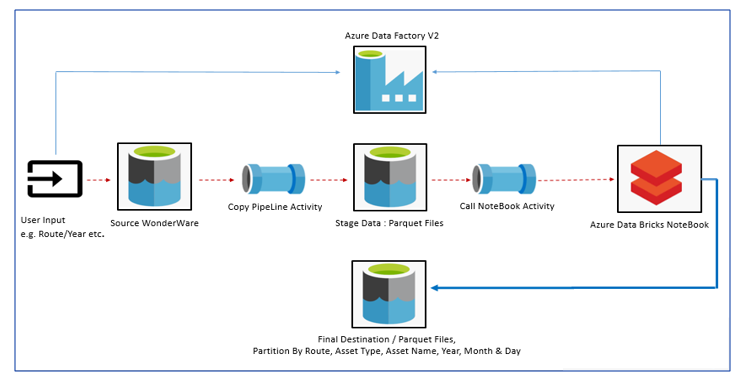


**STEPS:**

1. ‘Parameter control’ functionality allows control of the dataflow from Source location based on the input parameter, e.g. Route, Year, Month, Date etc.
2. Data is ingested from the source ADLS store by Azure Data factory which performs the following operations
   1. Unzip files
   2. Converts the .csv files to parquet files
3. Data from stage layer is converted to partitioned parquet files using Azure Databricks. This step limits only in-scope attributes (e.g. - current waveform) for faster processing and reduced read IO. The partition hierarchy is Route / Asset / Attribute / Year.

### **10.1.3 Detailed Wonderware data ingestion pipeline**

Below is the detailed data flow diagram for the ADF pipeline implementation.



**Source ADLS Folder directory Structure:**

Wonderware data at source location is in .zip format. It is assumed that data from other routes will maintain a similar folder structure in order to support data flow from the pipeline.

Sample folder structure at source location-

WWHistory-LNW/LNW\_201401/IIHistLNW-20140101/\*.zip

**Stage ADLS Folder Structure:**

The folder structure is retained at stage location. From source to stage while copying data files, unzip and csv-parquet conversion is performed.

Sample folder structure at stage location-

stage/WWHistory-LNW/LNW\_201701/IIHistLNW-20170104/IIHist-LNW-20170104-1-1-1.zip/IIHist-LNW-20170104-1-1-1.parquet

**Target ADLS Folder Structure:**

While data movement from stage to ‘prd\_poe’ folder is performed, selective attributes are read and stored in partitioned parquet files for efficient read.

The partition hierarchy is Route / Asset / Attribute / Year.

**Sample folder structure at 'prd\_poe' location:**

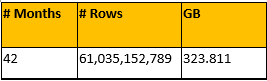
prd\_poe/Route=LNW/Asset=0M032\_SYC\_P1111A/Attribute=Current\_Waveform\_NR/Year=2019/part-00064-tid-8499870759290176134-3f4051da-3af3-4fe2-8db8-04a4ddafb7f9-15791-1.c000.snappy.parquet

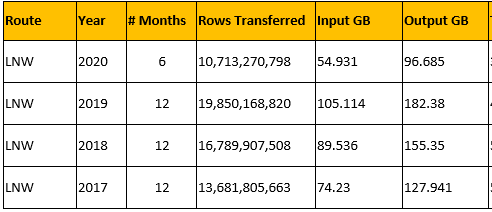
### **10.1.4 Wonderware data volume & growth**

Machine learning models will train and test on historical data.

Data duration ingested is from Jan 2017 – Jan 2020

42 months data volume in Zipped format is observed to be around 323 GB having approximately 61 Billion records.

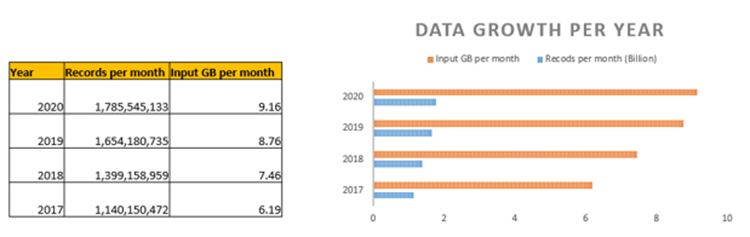


Wonderware data ingestion Pipeline run Stats.

**Wonderware data growth**

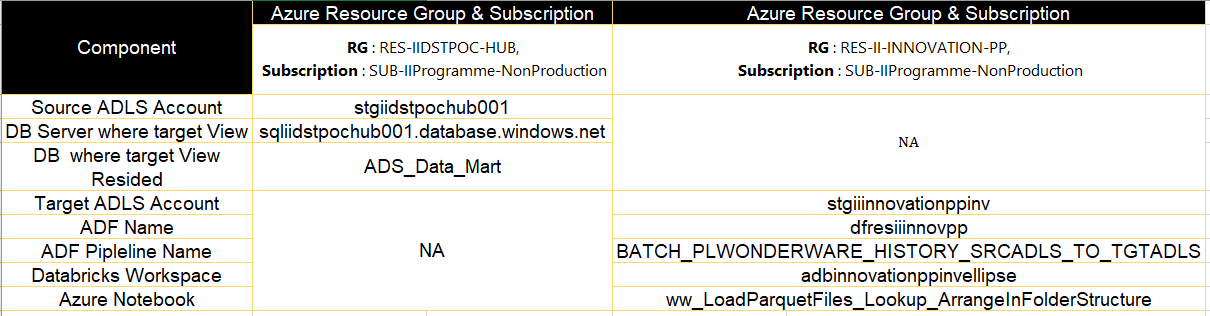
Data is observed to grow at growth rate 10-15% YOY.

Data growth from 2017-2018 is observed high as compared to 2019-2020.



### **10.1.5 Technology landscape/environment - Wonderware**

**Components and resource groups**



### **10.1.6 Azure components for data ingestion**

|  |  |  |  |
| --- | --- | --- | --- |
| Application Component Name | Application Component Role | Application Platform Classification | Version details |
| Azure Data Factory | ETL | Azure PaaS | ADF V2 |
| Azure Data Lake Store | Data Lake | Azure PaaS | Gen 2 |
| Azure Databricks | Data Analytics Platform | Azure PaaS | Databricks Runtime 6.4 |
| Azure SQL DB | Database | Azure PaaS | Hyper scale service tier |

## **10.2 Data engineering phase 2 update**

### **10.2.1 Codes for in order date**





### **10.2.2 How to find in order date in ADS foundation**

* Step : 1

Table [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURE\_EVENTS] & Table [ADS\_FH\_FMS].[ADS\_FND\_FMS\_REFERENCE\_VALUES] merged by using **INNER JOIN technique**, on columns frv\_id & event type, and produced a set (which contains failure no, failure event date, description, ifc no) based on required condition.

* Step:2

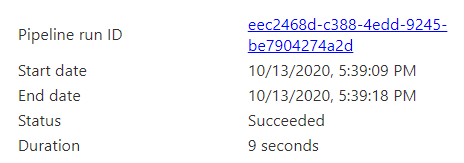
Once we get the above set, then joined the same set with table [ADS\_FH\_FMS].[ADS\_FND\_FMS\_FAILURES] by using **LEFT JOIN method** on columns Ifc no & failure no.

* Step:3

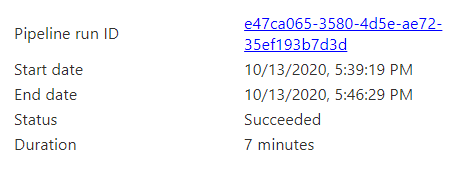
Remember to mark Active row flag as Yes for all the columns, and verify if you get Inorder type for frv Id 5687 In step-1

### **10.2.3 Execution times - in order date ADF pipeline**

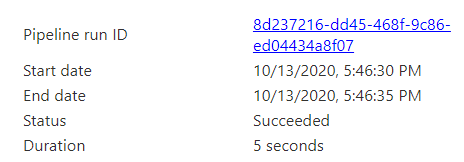
**Execution time of First Copy Activity**



**Execution time of Second Copy Activity**



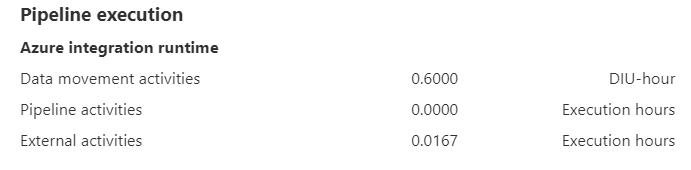
***Execution time of Last SP Activity***



### **10.2.4 ADF resource consumption: in order date**

A Data Integration Unit (DIU) is a measure that represents the power of a single unit in Azure Data Factory. Power is a combination of CPU, memory, and network resource allocation. Below is the details of DIU to execute "InOrder date" pipeline.

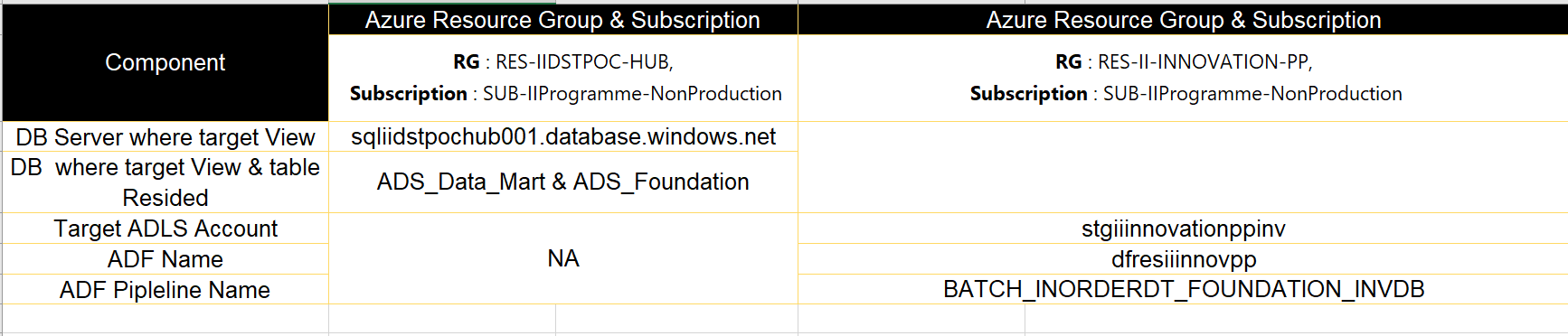
For example, 0.6000 DIU hour denotes cost to execute an Azure Data Factory activity on the Azure integration runtime per hour.



**Note:** The source JSON code for this ADF has been appended in **Appendix (Codes for In-Order Date)** section

### **10.2.5 Technology landscape/environment – in order date:**

Components and Resource Groups



### **10.2.6. Azure components for data ingestion**

*Azure data factory & Azure SQL DB were used only for this POC. The details of the components have been mentioned in Appendix - 10.1.6*

## **10.3 Data Engineering Phase 3**

SQL view script for Workorder data



## **10.4 Data Science**

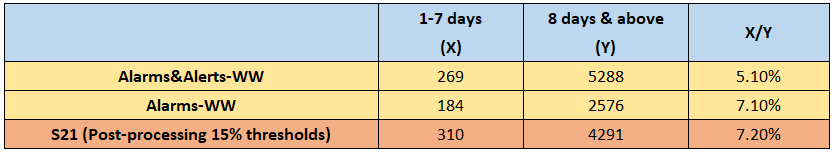
The attached workbook represents the availability of data in number of months, number of days for which data is available, the total number of traces and average number of traces per day for the Assets.



**POE Solution Comparison with Wonderware alert and alarms**

Table 10.3.2 represents the ratio of anomalous data points within 1-7 day window and anomalous data points outside the 1-7 day window. The table shows how many times the operation was alerted during 1-7 day window and outside 7 days window. However, the ratio is higher for S21 (one of the solutions built).

**Table 10.3.2: Result of POE Solution Vs Wonderware anomalous data points within and outside 1-7 day window**



Please find the definition of the columns of the above table:

X = Total number of anomalous predictions within 1-7day window

Y = Total number of anomalous predictions beyond 7days

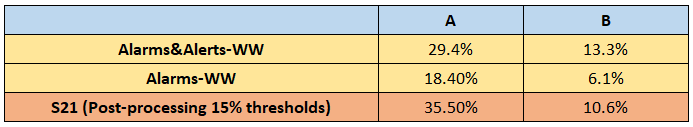
For Wonderware, the number 269 in Table 10.3.2 means that for these 204 failures, Wonderware raised alerts & alarms on 269 days out of (1428 days approx.) And with these 269 Alerts & Alarms it was able to correctly predict 40.2% (82/204) within 7 days window (1-7 days).

Table 10.3.3 represent the result in two folds,

- the first fold result (Column A) represent the total number of true predictions over total regular data points within 1- 7 day window where S21 (one of the solutions built) is higher.

- the second fold result (Column B) represent the total number of true predictions over total regular data points outside 1- 7 day window where S21 (one of the solutions built) is lower compared to Wonderware alerts and alarms.

**Table 10.3.3: Result of POE Solution Vs Wonderware predictions within and outside 1-7 day window**



This shows that S21 raised more predictions than Wonderware within 1-7 day window whereas Wonderware alerts and alarms predicted more than S21 outside 1-7 day window.

Please find the definition of the columns of the above table:

A = Total number of true prediction within 1-7 day window / Total number of regular data points within 1-7 day window.

In above table, 35.5% was calculated as mentioned below:

=310/874

B = Total number of predictions outside the 1-7 day window / Total number of regular data points outside the 1-7 day window

In above table, 10.6% was calculated as mentioned below:

=4291/40621

**Note**: For Wonderware system only acknowledged and returned alerts and alarms are considered.

### **10.3.2 Discrepancies (fault resolved and in order dates)**



**Table 10.3.4: Discrepancies between Fault Resolved and In Order Date**