HEALTH INDEX PREDICTION

DEVELOPER GUIDE CAN 6.0

DECEMBER 23, 2022

AVANSEUS HOLDINGS PTE. LTD.

REVISION HISTORY

Version	Date	Change description	Created by	Updated by	Reviewed by
V 1.0	March, 2022	Release 6.0	Hemanth	Raksha	Chiranjib

Table of Contents

avanseus

1. Introduction

This document describes the sequence of steps or protocol to be followed for Health Index (HI) based prediction. It also includes a list of master tables, reference tables and transaction tables with description. Health index prediction is also referred to as **Superposed Prediction**.

2. Master tables

Table Name	Description
PerformanceCounterEquipmentComponent	Contains all distinct equipment names and its association with nation, region and zone.
PerformanceCounterEquipmentCause	Contains all distinct KPI names and its association with domain, network type, service impacting etc.,
PerformanceCounter	Contains KPI values of all equipment components captured at different intervals of time.
ThresholdConfiguration	Contains threshold type (MIN, MAX, MIN_MAX and RANGE) and threshold values of all KPIs
EquipmentToEquipmentComponent	Contains the hierarchy of Equipment components.
PerformanceCounterNation	Contains all distinct nation names
PerformanceCounterRegion	Contains all distinct region names
PerformanceCounterZone	Contains all distinct zone names

Sample JSON structure of all the above master tables can be found in this attached document.

3. Reference tables

Table Name	Description
AlarmToKPICorrelation	Contains correlation between each Alarm cause to list of related KPIs.
SelectedMasterSequences	Contains alarm associated KPI bit patterns of most recent 3 Alarm occurrences of every alarm type and equipment component.
SelectedPCECMaster	Contains list of possible master equipment components which are selected based on similarity check algorithm.
DynamicHIReference	Contains ‘scaling factor’ and ‘offset’ for each equipment component and alarm cause combinations. These parameters are required for health index calculation.

Sample JSON structure of all reference tables can be found in this attached document.

4. Transaction tables

Table Name	Description
PerformanceCounterDegradation	Contains the output of every prediction run.
HealthIndex	Contains Health index and shortlisted KPIs of every prediction run for all equipment components and alarm causes combinations.

5. List of indices

KPI based Health index prediction will have a huge volume of data because a snapshot of every performance counter will be captured in smaller regular intervals of time ranging from every 15 seconds or every 15 minutes or every 1-hour or every 24 hours. Hence, it is important to create appropriate indices so that the process of searching and data retrieval happens faster. Below document guides in creating appropriate indices required for Health Index Prediction.

https://avanseuscandev.com/releases/CAN/6.0/DeveloperGuide/CAN_6.0_Indices_for_PerformanceCounterPredictions.pdf

6. Prediction input configuration

Since performance counter data will be huge in volume, it is important to derive the below aspects soon after the data load:

1. What should be the data aggregation granularity?
2. What should be the bit sequence historical length for each prediction run?

The decision of data aggregation is mainly dependent on **historical inter-failure gaps** and the **lowest possible frequency** at which performance counter data is available. Based on the above two factors *'pmCounterSlotDuration'* and *'pmCounterSlotLength'* is configured.

The decision of bit sequence length or *'pmCounterBitSequenceLength*' is taken such that the considered bit sequence length must **include at least 3 recent alarm patterns.**

The parameters such as *'pmCounterSlotDuration'*, *'pmCounterSlotLength' and 'pmCounterBitSequenceLength'* must be updated in the configuration table. These 3 configurations can be modified from the advanced configuration screen of CAN dashboard.

For example, KPI data is available at every 15 minutes and the approximate inter-failure gap for a particular alarm cause is 5 days. It is concluded that if aggregate of performance counter data is done at every 2 hours and considering *'pmCounterBitSequenceLength'* as 200, a minimum of 3 recent failures in prediction input vector can be covered. It is always safer to keep *'pmCounterBitSequenceLength'* as a larger value ~500 bits as CAN covers more number of alarms in the equipment components in which inter-failure gaps tend to be very high.

Most importantly, after configuring the aggregation parameters correctly, the prediction engine will generate the prediction input by picking the most significant point available in the aggregated period. Most significant point is decided based on the configured threshold type for each individual KPIs.

7. Prediction procedure

Follow the steps below in the chronological order:

1. **Data Load and Validation:**
   Data load of the performance counter can be achieved from the ‘Parser’ tab of ‘Input Mapper’ screen in CAN dashboard. Mapping of customer provided data to the CAN field should be done correctly by the developers before calling the data load API. After the data load completes, verify for its correctness by comparing it with the JSON structure present in this [document.]()

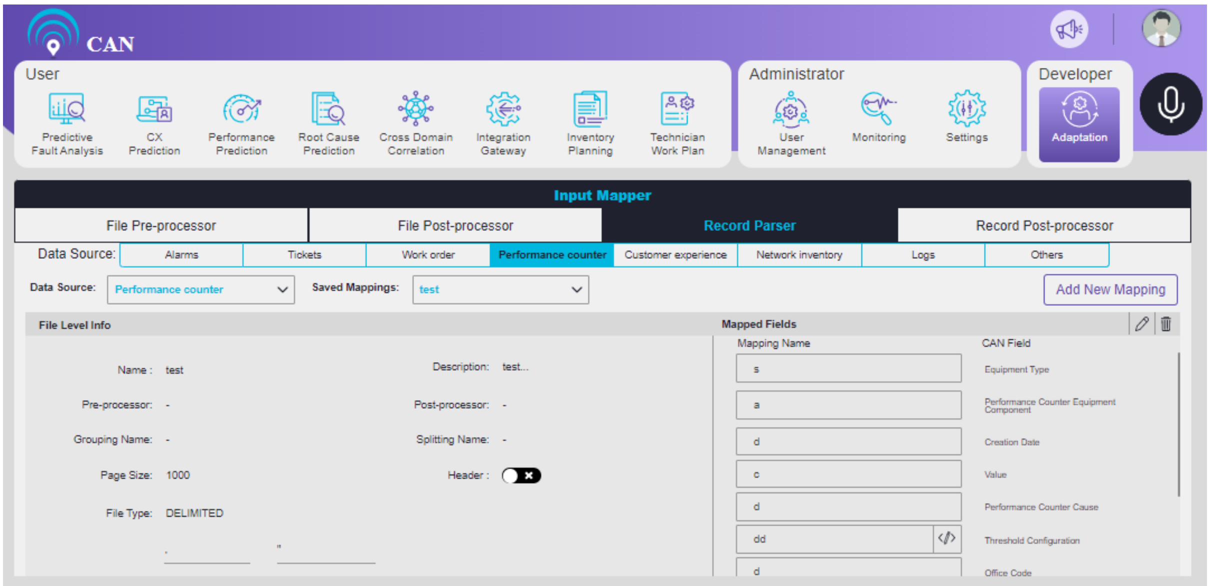

Figure 7-1 - : Performance counter data load configuration screen

2. **KPI Configuration:**
   Classify each KPI by its Domain, Network type, Service impacting or not, Threshold type and its threshold value in ‘KPI Management’ screen under ‘Settings’ tab of web user interface.

avanseus

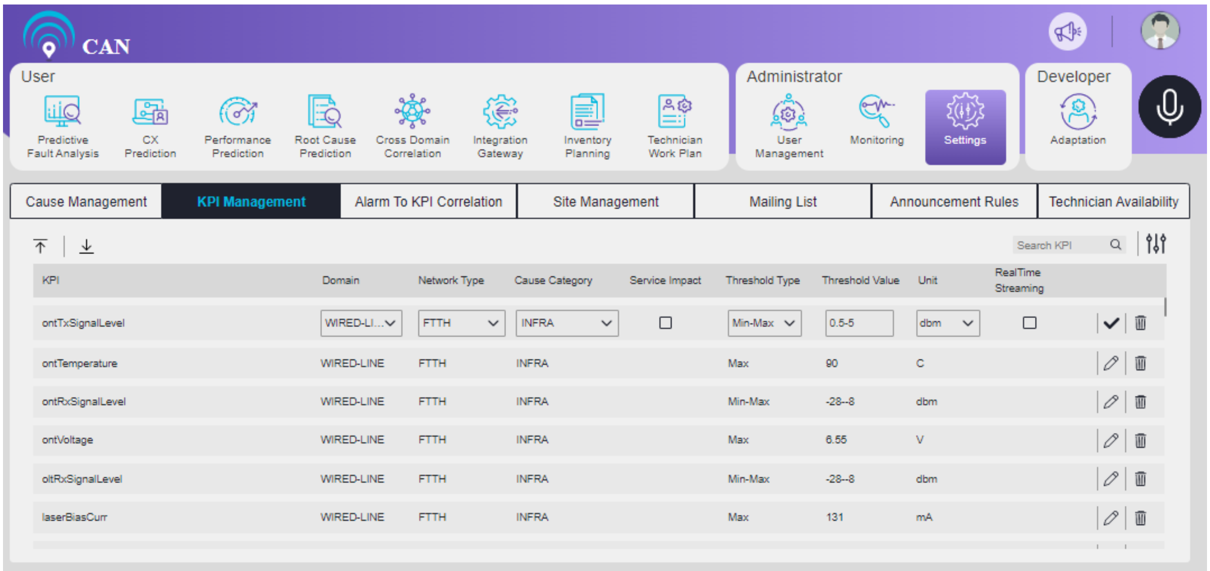

Figure 7-2 - KPI management screen

3. **Alarm to KPI Correlation:**
   KPI related Alarm causes should be identified. Each identified alarm cause must be mapped against the corresponding KPIs. It is important that a single KPI can be mapped against multiple alarm causes. In such cases, multiple input lines will be generated for that KPI w.r.t each alarm cause.

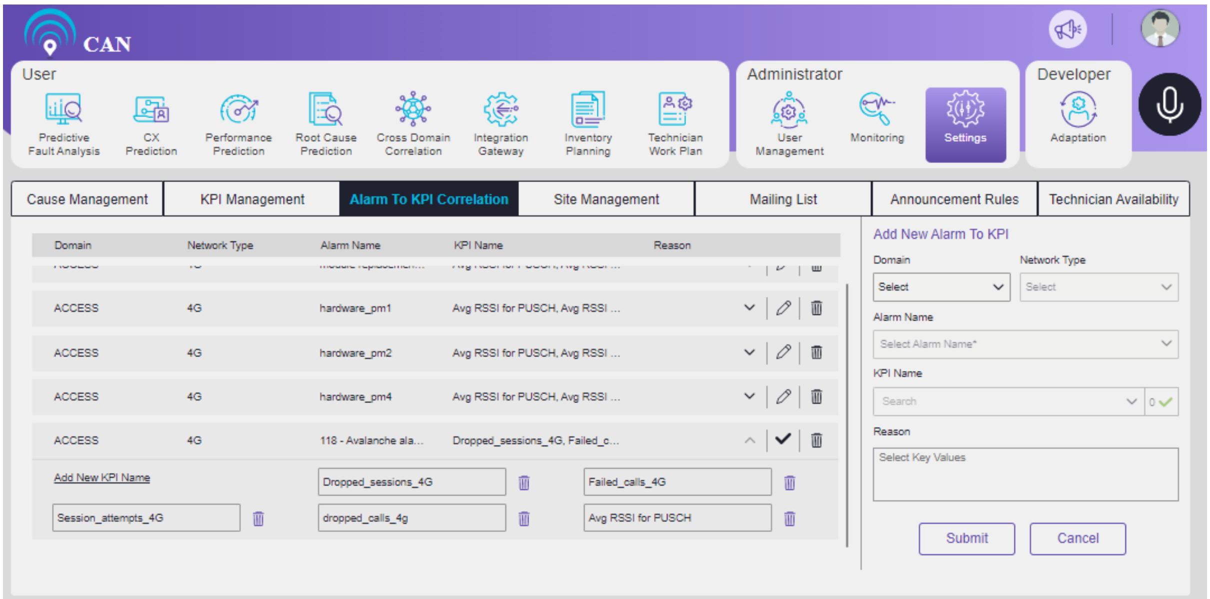

Figure 7-3 - Alarm to KPI correlation configuration screen

4. **Selected Master Sequence Generation (Optional):**
   This step generates input sequences for all the KPIs corresponding to all the alarm causes and for every equipment component. These generated input sequences will be used later by the 'Similarity Check' algorithm. Similarity check algorithm will be invoked for those equipment components which are having insufficient number of alarm occurrences. The role of Similarity Check algorithm is to provide a list of 3 similar equipment components to that of a testing equipment component.

   Master sequence generation can be invoked by calling the below API after logging in to

CAN application:

window.redirect("healthIndexPrerequisites/prepareMasterSequences", {"before": "30-01-2021 00:00:00", "interestedEquipmentComponents":"equipment6,equipment42" },"_blank");

- **"before":** Optional date parameter can be passed in "dd-MM-yyyy HH:mm:ss" format to explicitly specify the end date of the alarm, before which master sequences must be created. By default, if 'before' parameter is not specified, then master sequences will be created by considering the latest available alarm date as the end date.
- **"interestedEquipmentComponents":** Optional string parameter can be passed to specify the interested equipment components for which master sequences must be generated. By default, if this parameter is not specified, then master sequences will be generated for all the equipment components.

  Note:

  1. This step can be skipped if you notice all equipment components are having a minimum of three alarm occurrences for all the interested causes in their respective historical data.
  2. Selected Master sequences must be continuously updated using a cron job to include more recent alarms.
  3. The Cron job responsible for updating master sequences is 'MasterSequenceExtractionScheduler' and by default, this job will be triggered at 2 AM every day. To customize the job triggering time, update the key by name 'MasterSequenceExtractionUpdatingCron' in the configuration table.

5. **Dynamic Health Index Reference Generation:**
   In this Health Index document, to calculate the health indices, we are dependent on offset and scaling factor. Offset and scaling factor will be different for different equipment components and alarm causes combinations. These two parameters must be calculated during the training phase using the average of prediction outputs obtained in the nearby vicinity of historical alarm occurrences.

   Select the duration for which the model has to be trained. Run multiple prediction runs on the selected training duration. Ensure that the prediction completes successfully. Then, generate the dynamic health index reference parameters by invoking the below API:

   window.redirect("healthIndexPrerequisites/prepareHIParameters", {"before": "30-01-2022 00:00:00","_blank");

   - **"before":** Optional date parameter can be passed in "dd-MM-yyyy HH:mm:ss" format to explicitly specify the end date of the alarm, before which alarm will be considered for Health index reference calculations. By default, if 'before' parameter is not specified, then Health index reference calculations will be done by considering the latest available alarm date as the end date.

     Note:

     1. During this step if an equipment is having insufficient number of alarm occurrences in history, then HI related parameters will not be generated. For such equipment, Health index will be calculated using the offset and scaling factor found for their master equipment components which are selected using Similarity Check algorithm.

2. Generation of health index parameters have to be regularly updated using a cron job to include more recent alarms which results in the change of offset and scaling factor.
3. The Cron job responsible for updating health index parameters is 'HealthIndexUpdationScheduler' and by default this job will be triggered at 1 AM every day. To customize the job triggering time, update the key by name 'HealthIndexParametersUpdatingCron' in the configuration table.

6. **Running Prediction:**
   After successful completion of all the above steps, a decision has to be made regarding how frequently the prediction should run. This mainly depends on inter-failure gaps observed in the past.

   Input generation during prediction run is one of the key steps in the entire prediction process. Consider that input generation will happen for all equipment components and alarm cause combinations. During input generation, the latest 3 alarm occurrences will be superimposed on the timeline of historical KPI data. If an equipment component is having no or insufficient number of alarm occurrences, then failure sequences of other equipment components (a.k.a master equipment components) will be combined to the input vector. Selection of master equipment will be done using the 'Similarity Check' algorithm. A note on different scenarios of input generation and similarity check can be found in this Test case document (From row no 10 – 17 in Backend activity requirements sheet)