RELEASE DOCUMENT

Cognitive Assistant for Networks (CAN) Release 6.0

MARCH 1, 2023

AVANSEUS TECHNOLOGIES PVT. LTD.

Table of Contents

General Remarks

CAN 6.0 is a yearly general product release for 2023. At the discretion of the delivery team, deployments using earlier CAN releases can be upgraded to CAN 6.0, based on each customer situation, to get the new features & also run CAN in Kubernetes based setup. All new deployments must use CAN 6.0.

Please review this release notes to learn what is new in this version as well as important notes concerning known issues, bug fixes for 6.0 and improvements in the existing features.

New Features

- **Chaos Monkey testing by randomly terminating instances**: This is a testing carried out on many applications by randomly terminating pods to check for availability of the application. This testing ensured no downtime when replicated application instances were run.
- **Python support for IDE**: The existing IDE feature supported only Java. However, this release supports Python by allowing users the flexibility to choose the programming language of their choice. All UI based IDE features remain similar across Java & Python languages.
- **Code snippet version control**: Allows users to have multiple versions of code (error free or with error). This is mostly useful when user wants to come back and update the existing version of the code.
- **Predicted alarm duration**: Earlier release had average alarm duration as one of the parameters in the prediction report that gave the customer some idea of the alarm severity. In this release, we are predicting the alarm duration itself.
- **Cause standardization during alarm ingestion**: This is an activity performed mostly manually in many customer deployments. Existing manual work of finding the logic to extract causes from raw text & writing regular expressions remains as is. However, now a dashboard is available to upload the extraction rules & regular expressions in a standard format. Along with, an API is available to do cause standardization based on the rules uploaded in the dashboard.
- **Rule discovery mechanism for KPI threshold breach prediction**: This feature makes the accuracy of KPI threshold breach prediction better by maintaining a decent coverage.
- **Support for range & min-max type of aggregation for PM KPI**: Enables CAN to deal with KPI values that are normal within certain range (Min-Max) or beyond certain range of values (range). The UI has an option to configure these values along with other edit options from the operationalisation point of view in the KPI management screen.
- **Limiting KPI values up to a multiple of max and min thresholds to make KPI sequence smooth**: CAN uses input data for creation of trends based on which the chances of failures and other related predictions are assessed. Since the input data is sourced from network, the actual breach variations can differ depending on the health status of equipment under monitoring. This feature smoothens KPI sequences to ensure appropriate representation of the data especially in graphical semantics.
- **Anomaly detection in performance KPI**: Anomaly is understood as significant deviation from common pattern followed by the KPI sequence in the past. Anomaly can be created by single abnormal KPI value or by a set of KPI values over a period. In both cases, anomaly is a strong indicator of a breach of KPI or consequential failure of equipment or port. In case of real-time monitoring of streaming data, the key to prediction lies in understanding the performance variations at the earliest and anomaly detection is the best way to understand an underlying or upcoming issue.
- **Visual display of anomaly during stream processing**: Anomaly is understood as significant deviation from common pattern followed by the KPI sequence in the past. Anomaly can be created by single abnormal KPI value or by a set of KPI values over a period. In both cases, anomaly is a strong indicator of a breach of KPI or consequential failure of equipment or port. In case of real time monitoring of streaming data, the key to prediction lies in understanding the performance variations at the earliest and anomaly detection is the best way to understand an underlying or upcoming issue.
- **Selectively picking failures for master sequence generation**: This is one of the major steps in the Health Index Prediction pre-processing phase. The KPI-based health index prediction will have a huge volume of data as the details from performance counters are captured at periodic intervals ranging from 15 seconds to 1 hour. Yet, there could be cases where certain equipment might have an insufficient number of previous alarm occurrences.

Since this affects the overall prediction process, this feature is enabled to overcome the shortcomings of such absence of alarm data availability.

- **Dynamically calculating Health Index, thereby calculating offset and scaling factor**: Health index prediction is dependent on Offset and Scaling factors. The dynamic calculation of health index will require this offset and scaling factor to be calculated dynamically. Hence, there is a need for easy access and method to calculate the offset and scaling factor that are decisive towards calculation of health index dynamically.
- **Calculating health index on cumulative basis**: Health Index is introduced to ensure greater understanding and control of equipment and service abnormalities. Health Index is calculated continuously and any downward trends or sudden dips are directed for immediate investigation. Hence, Health Index need to be made immune to sudden fluctuations in KPI values that can act detrimental to the overall efficiency of Health Index output.
- **Customer experience prediction for broadband customers including PON topology maintenance**: Customer experience prediction focuses on the probability of customer to initiate call with the support centres for network related issue as well as the probability of customer reaching out for booking repair tickets. Call Probability is defined as odds for customer to initiate a call in next 7 days.
- **Changing current function calling based mechanism to micro-services mechanism using gRPC**: Existing REST API based micro-service communication was an HTTP based call that used higher network bandwidth. gRPC is a more efficient and modern Remote Procedure Call that can work seamlessly in any environment to connect application services. It is designed for low latency and high throughput communication. It is developed to speed up data transmission between micro-services and other systems that need to interact with each other. Hence, the transition from REST to more effective gRPC for internal micro-services will enhance the optimization of application performance.
- **Resource Limits for Kubernetes**: Applications run on Kubernetes platform were using the resources like CPU, RAM, etc. on real time basis without any strict bounds and applications were free to use as much resources as possible available on the host system. This sometimes led to a 100% of the resources being consumed and eventually brought down the host machine. Hence, as a standard strategy, resource limits are set for micro-services so that the overall health of the system does not compromise.
- **Viewing historical anomalies in real-time**: UI support is available to display, search and download Historical anomalies. Anomalies are displayed in the line chart with proper indication.
- **Shortlisting worst cells for KPI breach predictions**: This feature identifies worst cells among the threshold breach predictions based on data characteristics observed for individual KPIs with regard to three categories: Accessibility, Retainability and Quality.
- **Topology stitching & topology linking with prediction**: CAN application predicts the upcoming network faults and performance. Each network element that is predicted to have issues are clustered and mapped over user interface to have an easy representation and understanding of overall network that is being monitored. Topology Discovery is an enhanced method by which the end-to-end network is discovered from the inventory data provided by customer and displayed, that will further enhance the operation of CAN application.
- **Integration with Redhat Ansible:** Ansible is an automation tool that provisions configuration management, application deployment, orchestration, and many other processes. It is a plug and play activation system where a set of commands having a workflow is defined. It runs on various interfaces. SSH is the widely used interface. CAN uses Ansible framework to connect with various interfaces for executing high-level commands. These commands are executed remotely through SSH. This feature allows user to define & design workflow schema with multiple tasks. These schemas are then mapped to Predictions based on equipment component & fault. The mapped schemas are converted to Job instances during runtime that internally runs the defined commands per each task.

Enhancements

- **KPI collection from Prometheus**: CAN establishes active link with Prometheus streaming servers for data intake and retrieve the data in accordance with the metrics, configured over the CAN user interface.
- **Grafana & Kiali integration with LDAP**: Enhanced security allows user to access Grafana & Kiali dashboard only after successful user authentication.

- **Alerting on threshold breach for Grafana & Kiali**: Allows alerting via Email to certain contact points based on the threshold levels configured.
- **Fault trace generation configuration**: Allows configuration of fault trace generation technical parameters from the dashboard.
- **IDE - UX review and initial viewport correction**: An enhanced IDE support when compared to last release for users to write code, compile on the fly and save it. IDE now supports Python code writing as well.
- **Post prediction processor with IDE**: Earlier release had the feature to upload Java file to handle post prediction process. However, this has been updated to IDE that gives flexibility to write code not just in Java but in Python too.
- **Configurable location type selection during clustering**: Earlier release had zone as the level on which clustering for cross-domain correlation was performed. The current release has a configuration available to select location types like Nation, Region, Zone, Office code & Topology to do clustering.
- **Session expiry to handle mouse move**: Considers mouse move also as an online activity and keeps the connection/session active with the server. Earlier, the user gets auto-logged out after few minutes if there were no activities performed which involved in reaching the server.
- **Algorithm should return error code and error message in case where gaps in superposed sequences are not same**: CAN application predicts failures based on the historical failures and the due mathematical calculations derived ipso facto. Generally, CAN application requires three historical failures to generate predictions and hence the application needs to have various checks and balances to handle such KPIs where there are no 3 historical failures or very few pre-failure events.
- **Allowing users to configure classifications of PM KPIs**: Provides a comprehensive support in UI to configure the domain, network type, cause category, etc. The network KPIs are identified by the system from the input data and the other relevant attributes. Apart from network KPI names, other attributes are made editable in the UI to ensure that relevant inputs are fed to the system in the most accurate way for processing and decision-making. User will have options to set these values individually through edit option or to bulk upload the same as file for system to process and upload.
- **KPI names should be there (2G, 3G, 4G, Core, Transport, and Device) with separation**: Provides a comprehensive support in UI to filter based on domain, network type, cause category, etc. By default, CAN assigns ‘ACCESS’ Domain and ‘4G’ Network Type. However, this can be modified from advanced configuration tab.
- **Configure linkage between Performance KPI and alarms**: This is KPI to alarm correlation configuration screen. It is a function for relating KPI and alarms as KPI is related to the performance counter configuration. Alarm is triggered for a KPI for a specific reason. It is used for prediction and performance management of KPI.
- **Configure linkage between equipment component name as received in alarms and Performance KPIs**: The equipment component names received in performance counter may differ from the data in alarms. Since both are interconnected, the application need to link these different names together for a comprehensive prediction report.
- **All or major KPI trends to appear in single screen**: In earlier release of CAN, the KPI trends are shown discreetly for the selected equipment component. Since the telecom environment involve monitoring of larger number of KPIs that are critical for providing the adept outlook of network, it is imperative that CAN should show the trends of Multiple KPIs in single screen. This requirement will enable CAN to monitor multiple KPI trends together in a single screen that will improve the ease of operation (Analysis of KPIs).
- **UI changes for changing data granularity while showing KPI trend upon zoom-in**: Provides a zoom tool along with the user interface so that users can zoom over the KPI semantics. As the user zoom in, the granularity of KPI display deepens providing users with the details of KPI changes at seconds/minutes/hours aggregation depending on the input data. On zoom-out, the granularity again changes to provide a peripheral view based on daily aggregation.
- **Format configuration for PM KPI report**: Provides end user a flexibility to customize the daily/matching report template by adding/removing/rearranging the interested columns for both threshold breach as well health index related predictions.

- **Identifying and highlighting parent cause at cluster level**: CAN identifies Root Cause for a given network fault through domain and field learning. Since the application groups network elements as clusters, based on their interconnection, the parent cause of the common fault can also be understood and displayed in UI. Parent and child causes mapping information is configured in the system. Once clustering is performed, parent causes are identified based on the mapping configured.
- **Parent-child correlation - after clustering, we should identify which are local clusters automatically**: Parent child correlation is helpful in root cause analysis. Clusters that have correlated causes and identified by parent-child rules are considered as local clusters. CAN identifies such local clusters automatically and displays it on the UI.
- **Allowing configuration of parent-child groups based on input**: Parent child correlation is an enhanced way of root cause analysis. The parent cause to group of errors are configured or discovered. This will enable seamless understanding of the network irrespective of their domains and enhance ease of root cause analysis.
- **Streaming support for data collection over 3GPP and HTTP REST protocols (Alarm collection and PM counter/KPI collection)**: There are many interfaces supported by CAN to collect data from different data sources. This new streaming interface follows 3GPP specification to collect data over HTTP REST protocol.
- **Realtime prediction - In-memory PM data preparation**: Enables real-time prediction and faster data streaming by storing data related to prediction process in Redis cache. This is applicable in the case of KPI prediction when data is continuously collected and predicted.
- **Changing existing data flow to micro-services**: Existing method of sequential batch file processing for data loading is time consuming. Micro-services architecture allows distributed processing of records that reduces the overall time taken for complete data loading process. Currently, in Kubernetes based setup, the data flow process is implemented using gRPC protocol.
- **Solving approximate sequence comparison problem needed for synthetic failure history generation**: Prediction of faults that happen after long time gap, may not always have minimum required failure or degradation instances (typically 4 failures) for the equipment component CAN is predicting. This is also expected since faults are happening after long time and historical data is limited to typically 6 months to 1 year. For example, if a fault happens after 6 to 9 months, there may be just one or no failure within the historical data. The way to overcome this situation is to create synthetic history by combining available failure and performance data from multiple similar equipment.
- **Support for LDAPS**: LDAPS allows for the encryption of LDAP data (which includes user credentials) in transit during any communication with the LDAP server, thereby protecting against credential theft. This act as an additional layer of security while integrating with directory servers during interoperation. There is also support to integrate with third party LDAP (customer LDAP). This will make the system ready when it works with LDAP.
- **Support for secure Kafka**: CAN application with Kafka broker optimizes customer operations of sending the alarm, ticket, and performance counter data in a streaming channel. This feature accepts data using secured protocols based by SSL certificate.
- **Support for JSON data parsing**: Input mapper has many types of parsers. This feature allows users to configure and parse JSON data.
- **Support for XML data parsing**: Input mapper has many types of parsers. This feature allows users to configure and parse XML data.
- **Support for multiple KPI values in a single record and vice versa**: Enables CAN to parse a single input data record to multiple records under the same head and vice versa. In other words, this allows parser to group a number of input records into a single record in CAN database and combine a set of input records into a single record in CAN database.
- **Map based Display for PM KPI Threshold Breach Prediction**: CAN supports performance KPI monitoring of the network where the KPI threshold breaches are identified by the system and notified. The plotting of such breaches over a map using the GIS coordinates enables quick understanding.
- **Map based Display for Health Index Prediction**: Health Index is introduced to ensure greater understanding and control of equipment and service abnormalities. Health Index need to be made immune to sudden fluctuations in KPI values that can act detrimental to the overall efficiency of health index output. The plotting of such variations over a map using the GIS coordinates enables quick understanding.

- **Filtering KPI Breach Predictions based on Site Priority**: CAN supports performance KPI monitoring of the network where the KPI threshold breaches are identified by the system and notified to relevant stakeholders. Few office codes will have high priority because of its geographical significance. This feature defines the threshold boundaries along with the impact for different technologies with the user-friendly UI support.
- **Filtering Health Index Predictions based on Site Priority**: Health Index is calculated continuously and any downward trends or sudden dips are directed for immediate investigation. Alarm is triggered for a KPI for a specific reason. It is used for prediction and performance management of KPI. Few office codes will have high priority because of its geographical significance.
- **Predictive fault analysis clustering map view to handle large number of markers and correction of inappropriate lat-long**: Enables zoom-in and zoom-out options in map view to accommodate the mapped network elements. Nearby network elements are shown as clusters is resolved as the user zoom-in. Lat-long values provided by the user are to be validated against the national or regional boundaries and accordingly identify the inappropriate lat-long.
- **Lat-Long Validation**: The predicted faults are displayed in the map view using the latitude and longitude values provided at office code level. Lat-long can be updated and validated from the site management screen. Validation happens at nation level and zone level.
- **Lat-Long mapping at office code level**: Latitude and Longitude details are essentially required for CAN application to display the predicted faults in the map view. Earlier versions of CAN application had lat-long mapping at equipment component level. As the CAN application advances, for better resolution and representation, the lat-long need to be mapped at office code level.
- **Enhanced filter for cross-domain correlation (location, office code, equipment, and domain)**: In previous versions of CAN application, cluster display was limited only to zone based filtering. As CAN 6.0 provides an enhanced clustering algorithm through topology discovery, more filters need to be included to filter out the comprehensive results as per the user requirement.
- **Enhanced bit pattern view for topology based clustering in topology discovery screen**: In previous versions of CAN application, cluster display was limited only to zone based filtering. As CAN 6.0 provides an enhanced clustering algorithm through topology discovery, more filters need to be included to filter out the comprehensive results as per the user requirement.
- **New filter algorithm to enhance precision and coverage**: Enhancement on existing algorithm to control precision and coverage. We can also use older filter rules to remove certain predictions. However, there are some new filter rules provided as part of the tool for both alarm & KPI predictions.
- **Data type restriction on record parser IDE**: File upload/data upload is a critical activity in the operation of CAN application during which sanity of the parsed data is essential for the efficiency of prediction reports. Hence, necessary restrictions need to be in place to avoid junk data/unwarranted fields to interfere with the actual data upload. In addition, this mechanism must ensure minimum rejections during data upload.
- **Data source grouping in record parser UI and back end changes**: CAN processes network data from multiple sources and domains to generate prediction reports. Hence, it is imperative to provide UI support towards grouping of such input data that come from similar sources. This will enhance efficiency of operations of the CAN application.
- **Master table caching for data load**: Master table caching improves the data loading time. In comparison to the previous versions, CAN 6.0 will have an edge for faster data loading and hence faster prediction.
- **Identifying parent-child correlated predictions**: Predicted faults are generated on daily basis. Among those predictions, some of them are leads to critical root causes that needs to be prevented. Hence, group of predicted faults pointing to such critical root causes can be identified based on the parent child correlation rules. User can take action based on these root causes.
- **PAT generation configuration**: PAT is a unique predictive ticket identifier that is required to track the predictions. Multiple predictions are clubbed to generate a single PAT. In-order to group multiple predictions there should be a common point among predicted faults. Older

versions of CAN has Equipment based PAT generation and it was not configurable. In CAN 6.0, PAT can be changed/configured at generation level.

Removed Features

N/A

Bug Fixes

- Provision to allow deployment team to configure heap space memory for CAN Tomcat application through ConfigMap. Earlier, the docker image itself had to be configured with heap space details.
- IDE features had cross browser inconsistencies. They are resolved to have a look and feel similar experience across different browsers.
- Spring beans XML configuration was pointing to Spring beans XSD with version 4.2. Due to strict validation in the new update, it causes builds to fail. Necessary corrections have been made to change the Spring beans configuration to point to Spring beans XSD with version 4.1.
- Excel writer's default decimal number representation limited to 2 decimal points for Double & Float type.
- Repeat predictions highlight w.r.t RoE feature in excel report had an issue where the sheets that are not marked configured as RoE sheets were also being coloured with a generic background color if they were repeated from another sheet. Now the code is fixed to consider colouring of only those sheets configured as RoE sheets.
- Maven build issue caused due to Eclipse library dependency is fixed.
- Updated react build files to do strict syntax check. Earlier react build was bypassing syntax check and caused UI to fail during runtime. React-scripts package was upgraded to v5.0.1 for this fix.
- Email manager library code updated to support StartTLS encryption (Especially for MS 365 relay server).
- Passwords are URL encoded while sending over HTTP requests. There was critical bug seen from GSC module during user authentication if anyone put HTML characters in the password.

Platform Support

- **Client OS & Browsers supported**:

OS	Browsers supported
Microsoft Windows 7 & 8	Google chrome 60.0 and above Mozilla Firefox 66.0 and above
Microsoft Windows 10	Google chrome 60.0 and above Mozilla Firefox 66.0 and above Internet Explorer 11 and above
Apple MacOS 10 and above	Google chrome 60.0 and above Mozilla Firefox 66.0 and above
Ubuntu Linux 15.0 and above	Google chrome 60.0 and above Mozilla Firefox 66.0 and above
Fedora Linux 29.0 and above	Google chrome 60.0 and above Mozilla Firefox 66.0 and above

- **Kubernetes server**: v1.21 and above

Known Issues

- Although CAN has been configured to support internationalization, it is currently available only in English and the CAN binary is packed with English text only. When there is a requirement from the Customer to port the application to support some specific language, then it becomes the responsibility of the delivery side developer to map the translations into a file and then pack the binary with that language.