NEC Advanced Analytics – Invariant Analyzer

- Behavior Learning and Visualization Tool -

March, 2020
NEC Corporation
1. Outline of System Invariant Analysis Technology
What is System Invariant Analysis Technology?

It is a machine learning technology that analyzes time-series numerical data, learns normal system behavior as a performance model from a period when a system works normally, and detects anomalies from numerical data collected in real time.

1. Since anomalies are automatically detected by analyzing whether the correlations extracted as a performance model (Invariant) are valid or not, there is no need to set different thresholds based on the properties of each metric data.

2. It is regarded as normal behavior as long as the values between numeric data change according to the correlations, regardless of the sizes of the values. Thus, the same model can be used even if the workload is different, such as between a busy day and an ordinary day.
Points of System Invariant Analysis Technology

1. Focus on “line”, not “dot”

- Focus on relationships between sensor data, not each of the values.

\[ m_2 = f(m_1) \]

2. A line indicates a limited relationship. However, seeing many lines enables to get a whole picture.

One “relationship” (= Invariant)

"Any relationship" in a system (= Invariants)
Comparison with Conventional Monitoring Technologies

Different from conventional monitoring techniques, the Invariant Analyzer do not need to configure complex monitoring settings and a number of threshold settings. Therefore, it can reduce operator’s burdensome efforts for performance monitoring.

**Conventional Monitoring**

For monitoring a large-scale system, a massive number of thresholds need to be set.

When some data exceeds the threshold, it is necessary to identify the associated events to analyze the causes.

**Invariant Analyzer**

Just import the sensor data. **No need to configure complex settings**

Since it focuses on relationships between performance data, anomalies can be detected **before threshold is exceeded**.
Failure Prediction using Invariant Analyzer

1. Displays the anomaly of the plant on a real-time monitoring.

2. Displays the anomaly of each sensor on the screen. Also displays the ranking of anomalies.

3. Enables further analyses of the anomaly after an error occurred in equipment. (Real-time analysis can be also made.)
2. Software Product:
NEC Advanced Analytics – Invariant Analyzer
System Structure (Real-Time Monitoring)

- NEC Advanced Analytics - Invariant Analyzer
  - WebUI
  - Manager API
  - Manager
  - Engine API
  - Invariant analysis engine

- Provided by SI
- Real-time metric data

- Accumulation server
  - Time-series DB
  - DB

- GW Server

- DB

- Data

- PI System™
  - DCS
  - RDB
  - PLC

- Provided by SI
Monitoring (Summary display)

1. Alarm List
When a model invariant is broken and exceeds its set ratio, an alarm occurs. The list shows the occurrence date and time, monitoring name, model name, confirmation status, sensor ranking, and more.

2. Anomaly graph
Displays the anomaly in the entire model. The graph can be zoomed in/out.

3. Physical map anomaly graph
The invariant status is mapped to the physical map. Users can create the physical map.

4. Ranking of anomaly sensors
Displays the sensors causing broken invariants in the descending order of anomaly (in descending order of potential main causes).
(Detailed) Invariant Status

Displays the detailed information on the broken invariants status.

1. Difference from predicted value
   Shows differences between the sensor value and the value predicted by the model in time series.

2. Shows the value of $x$ in the relational expression in time series.

3. Shows the value of $y$ in the relational expression in time series as blue line and values of $y$ predicted as pale blue broken line.
Select learning data & Create models

For model creation, select the sensors used for learning (creating models) and specify the learning period.

1. Model creation page
   A model is created by specifying the parameters. Specify a model consolidating multiple periods, creation timing, sensors for models and the like. Furthermore, it is possible to select whether to include the range of time-delayed tracking and autocorrelation.

2. Sensor Selection Screen
   Select sensors for the model creation from the list. It is possible to select sensors in a group by grouping them in advance.
Confirming Models (Model Viewer)

The detailed information of a model and filtered model are displayed.

1. Invariant
   Shows whether or not there is any invariant on the physical map. In “Fitness” slide bar shows the state where the model is filtered.

2. Details of invariants
   Shows the details after an invariant is selected on a physical map. The model formula and parameters are displayed.

3. Model summary
   Shows the number of invariants included in the model, the effective number and the like.
Monitoring Settings

The page below shows the real-time monitoring settings for the created model.

1. Set the model to monitor. Specify the model monitored under these monitoring conditions. Multiple models can be monitored by switching them according to conditions. Ex. Automatically select a model for each of the phases in the starting process. Switch the models according to loads.

2. Alarm Detection Settings
   It is possible to prevent a flood of alarms due to flapping by specifying the threshold for issuing alarms, the period when the alarm is suppressed, and the condition for continuous Period.
On-demand Analysis (Past Data)

The page below shows the functions of analyzing past data to review/inspect models and to analyze the causes of errors.

1. Specifying analysis conditions
   Specify the model to use, time-series data for detecting errors, and the like.

2. Anomaly Graph

* In the same way as the monitoring settings, the details of the physical map and broken invariants can be confirmed in analysis results.
# System Requirements

## NEC Advanced Analytics – Invariant Analyzer V2.2

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Red Hat Enterprise Linux  7 (*1)</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Dual-Core Xeon, or compatible / higher processor (8 cores or more recommended)</td>
</tr>
<tr>
<td>System Memory</td>
<td>4GB (32GB or more recommended)</td>
</tr>
<tr>
<td>Storage (free space) (*2)</td>
<td>10GB (200GB or more recommended)</td>
</tr>
<tr>
<td>Network</td>
<td>100Mbps LAN or faster recommended</td>
</tr>
<tr>
<td>Required Software (*3)</td>
<td>Analysis Engine/Manager Package: psmisc (64bit) Package: libicu (64bit)</td>
</tr>
<tr>
<td></td>
<td>WebUI Package: Tomcat7 or 8 Package: Java-1.8.0-openjdk Package: PostgreSQL</td>
</tr>
</tbody>
</table>

*1) Please set SELinux to disabled in advance.
*2) Areas such as data files created after installation are not included.
*3) Required software describes packages that need to be added based on the OS minimal installation.
3. New feature in V2.2
Quality Factor Analysis Engine

V 2.2 enhancements:

- **New analysis engine is available (able to standalone use).** If you specify a time period of poor manufacturing quality, Quality Factor Analysis (QFA) ranks possible cause sensors that have different characteristics in specified time period. By linking the Invariant Analyzer's sensor data and the anomaly detection result, operators can reduce the time and effort to find countermeasure points to prevent behavioral anomalies.

- Usability enhancement has been made in Invariant Analysis. Expanded import CSV format (columns correspond to sensors, rows to time series), support for user role editing, model switching by schedule, support for multiple models of on-demand analysis, support for update / WebUI-additional installation, etc.

- And other error check enhancement and cumulative bug fix.
Quality Factor Analysis Engice

1. Extract feature time series from sensor data

- Facility operation data (time series of sensor data)

2. Select features which contribute to quality change with several feature selection techniques

- Quality data (specify the time period of good quality or not)

3. Ranking the impact of sensors

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>E</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Software Architecture

Behavior learning / monitoring

Web Browser

Root cause analysis of quality degradation

WebUI
Manager
Analysis Engine

Invariant Analyzer

QFA Web

QFA Engine

Quality Factor Analysis

ranking possible cause sensors by its features