# Autonomous Memory Sizing Formularization for Cloud-based IoT ML Customers

Guang Wang (Presenter), Oracle Labs
Jason Ding (Co-presenter), Oracle Cloud Infrastructure
Kenny Gross, Oracle Labs
Prasad Ballingam, Oracle Cloud Infrastructure
Syed Fahad Allam Shah, Oracle Cloud Infrastructure

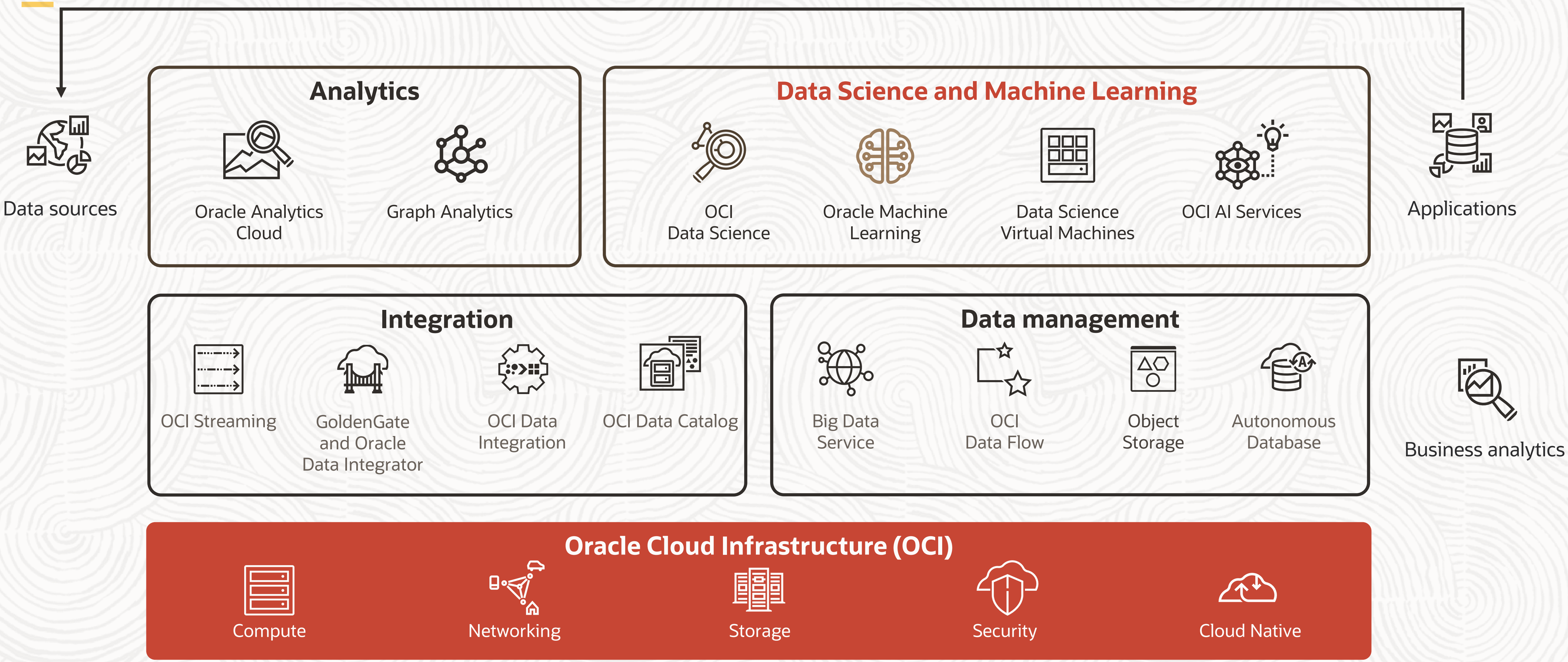
# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# Anomaly Detection at OCI AI Services

# Oracle DS and AI Platform



# OCI AI Services

## Unified AI/ML platform spanning cloud services, apps and data assets

**Applications**

- OCI Digital Assistant
- OCI Language
- OCI Speech
- OCI Vision
- OCI Anomaly Detection
- OCI Forecasting

**AI Services**

- OCI Data Science
- Oracle Database Machine Learning
- OCI Data Labeling

**Machine Learning Services**

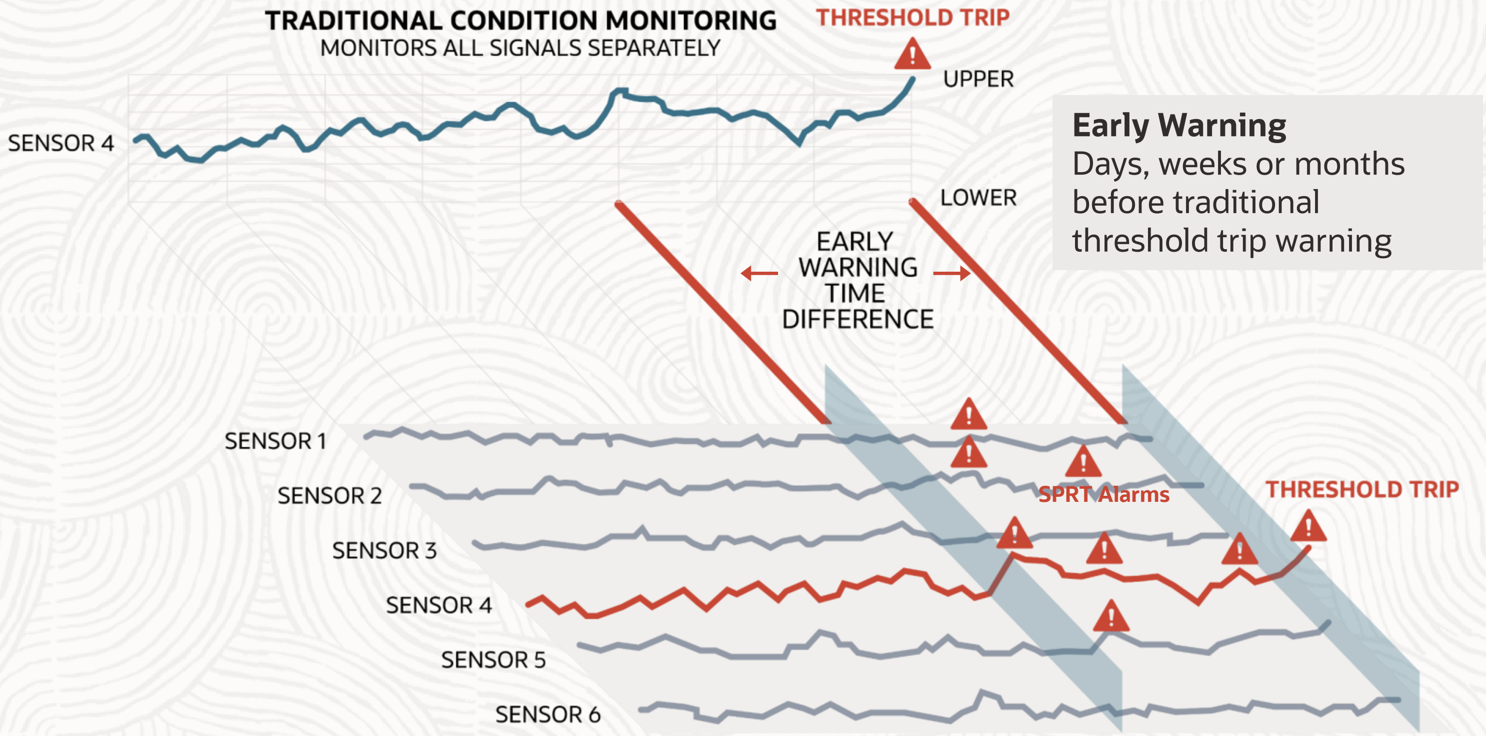
**Data**

# OCI Anomaly Detection (AD) Service

Builds multiple anomaly detection models and automatically selects the most accurate to flag critical incidents earlier

Automatically identifies and fixes data quality issues

Detect anomalies that span across multiple sensors at the earliest time with least number of false alarms using Oracle's heavily patented (150+ patents) MSET2 algorithm



# A Perfect ML Prognostic Solution for IoT Use Case on Oracle Cloud

**Oracle Labs**
MSET2 (Multivariate State Estimation Technique)

+

**OCI AI Platform**
Cloud Infrastructure with ML Kernels

=

**OCI Anomaly Detection**

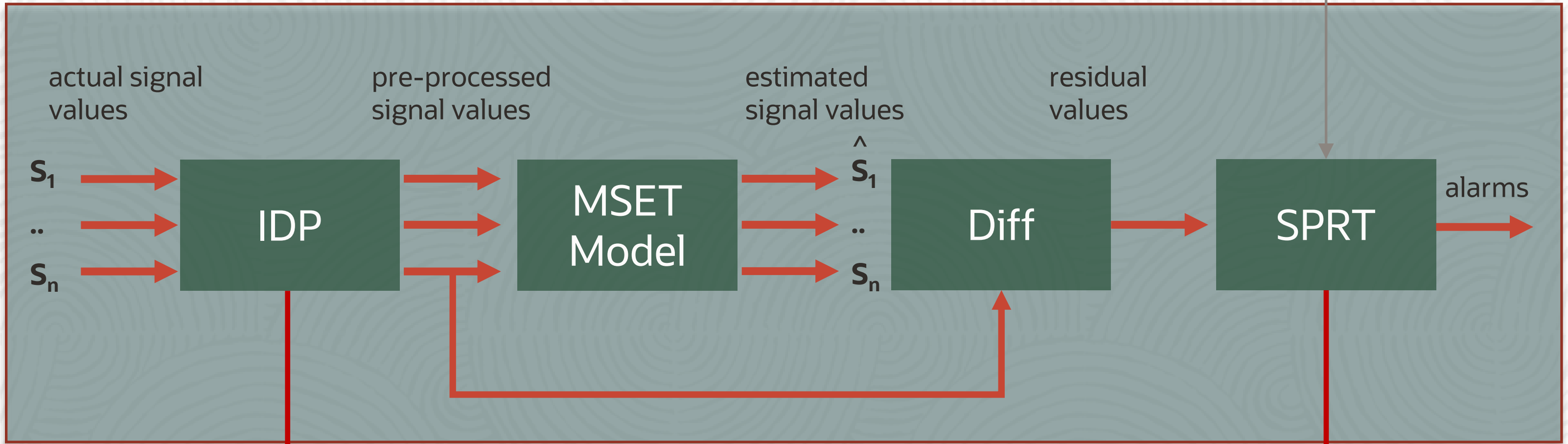
Powered by

# Multivariate State Estimation Technique (MSET2)

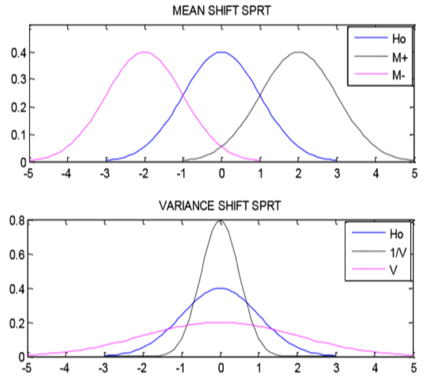
The Core of OCI Anomaly Detection Service

# MSET2 Data-Flow Framework

Real world prognostic applications with suboptimal telemetry data




*Abraham Wald, (June 1943).*
*- Oracle uses modified SPRT*




Sequential test for residual signals is based upon two hypotheses:

$H_0 : \mu = \mu_0 = 0$

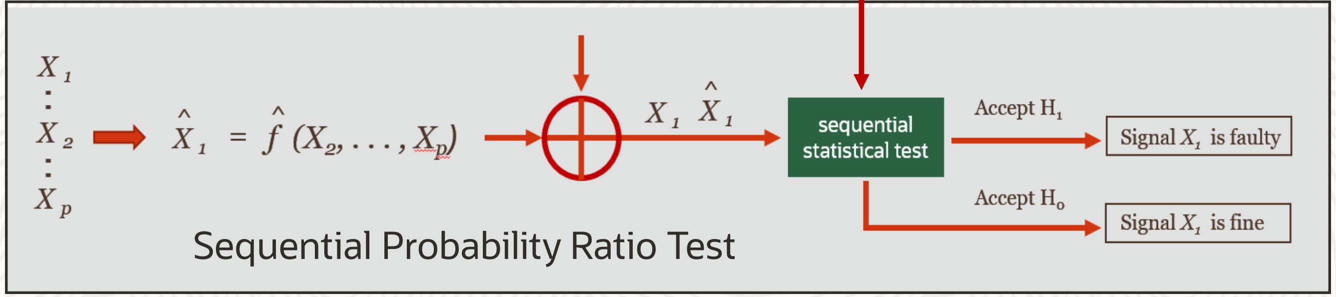
$H_1 : \mu = \mu_1 = M$

**Data Historian Database**

**Telemetry Data Sensor Farm(s)**

Intelligent Data Preprocessing

- UnQuantize
- Analytical Resampling Process
- Missing Value Imputation
- Inferential Sensing
- Provenance Certification & Auditability
- TriPoint Clustering (TPC)
- Telemetry Parameter Synthesis System




- High sensitivity for subtle anomaly detection without increasing false alarm probability.
- Can accommodate any measurement noise, work with non-Gaussian noise signals.
- Sequential-binary hypothesis test compares reference distribution ($H_0$) vs degraded distribution ($H_1$)

# The Idea of MSET2 Algorithm

- Consider a system with N signals and M observations under normal operation
- A data subset of the historical measurement consisting of N signals and m observations

$$D = \begin{pmatrix} X_{1,1} & \cdots & X_{1,\mathrm{N}} \\ \vdots & \ddots & \vdots \\ X_{m,1} & \cdots & X_{m,\mathrm{N}} \end{pmatrix} \in \mathbb{R}^{[m \times \mathrm{N}]}$$

- Given a current observation, $X_{\text{obs}}$, is the system behaving normally or abnormally?

- Compute estimate, $X_{\text{est}}$, given $D$
  - The closest normal behavior
- Compute residual, $X_{\text{est}}$ - $X_{\text{obs}}$
  - Make a decision based on residual

# Ordinary Least Squares

- Estimate is a linear combination of weights
  - $X_{\text{est}} = D\omega_{\text{est}}$
  - $\omega_{\text{est}} = (D^{\mathrm{T}}D)^{-1}D^{\mathrm{T}}\, X_{\text{obs}}$
  - $X_{\text{est}} = \mathrm{D}(D^{\mathrm{T}}D)^{-1}D^{\mathrm{T}}\, X_{\text{obs}}$

- But… systems are typically non-linear
  - Output is not proportional to the change of input
  - Collinearity due to the repeated or highly dependent sensor signals, causing amplification of uncertainties or crashes from singularities

# The Core of MSET2

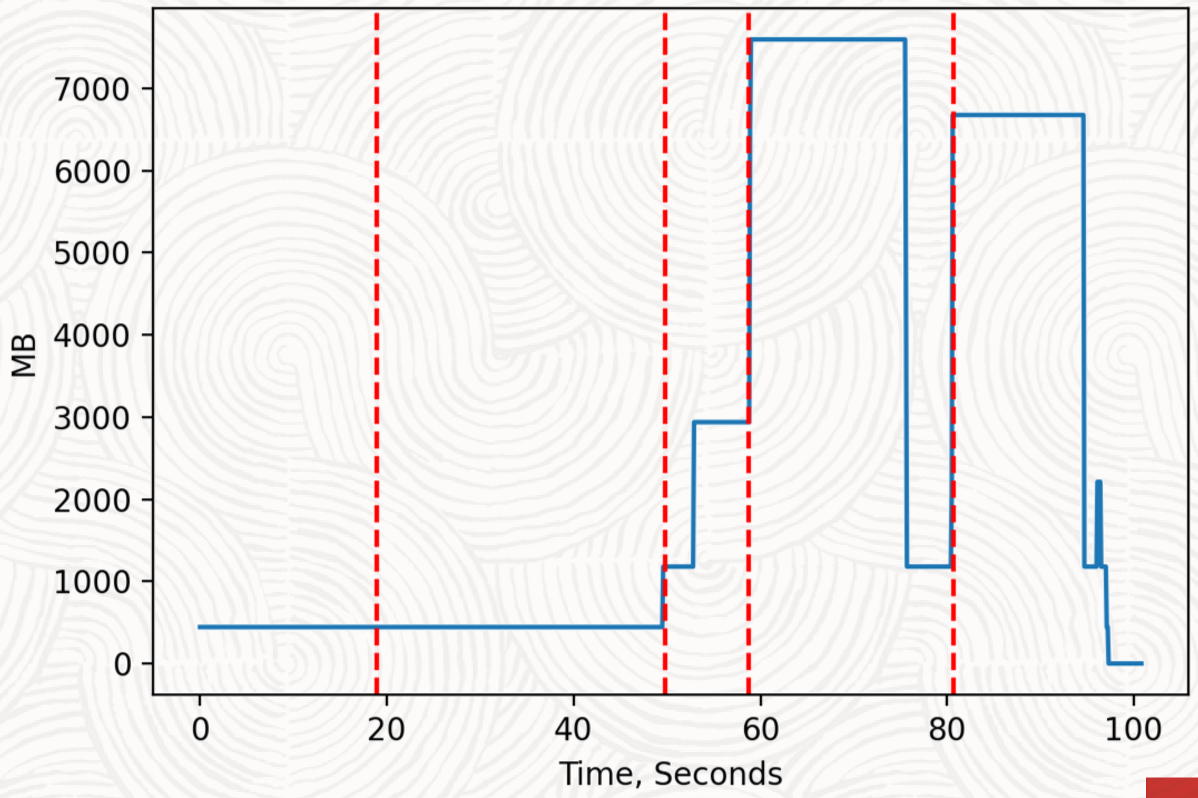
- Use a different binary operator, $\otimes$ to perform a non-linear comparison
  - $\omega_{\text{est}} = (D^{\text{T}} \otimes D)^{-1} (D^{\text{T}} \otimes X_{\text{obs}})$
- Use pseudo-inverse
  - $\omega_{\text{est}} = (D^{\text{T}} \otimes D)^{+} (D^{\text{T}} \otimes X_{\text{obs}})$
  - $X_{\text{est}} = D(D^{\text{T}} \otimes D)^{+} (D^{\text{T}} \otimes X_{\text{obs}})$
  - $D^{\text{T}} \otimes D$ is symmetric and positive definite, characterizing the pairwise correlation between the measurements in D

# Memory Sizing Formularization for AD Service

# Challenge of Right-Sizing the VM Shape (RAM Configurations)

- Only the size of the data is known prior to the ML run
- Peak memory usage determines the memory capacity requirement
- Peak memory usage significantly larger than the size of the data
- Scales with the square of the number of signals

Typical memory utilization profile



# Motivation

- Typical sizing approach for a big use case:
  - Run a small use case, figure out the RAM req., scale the numbers up accordingly
  - Likely require shape-changing later
- What is preferred:
  - Predict the peak memory usage upfront quickly, accurately, and autonomously
  - Avoid exhaustive memory pre-allocation assessments to save operational cost
- As a result, the RAM of the VM Shape can be optimally configured:
  - Accommodate the performance needs while saving cost for customers

Change the shape when the current shape is no longer a good fit for the workloads that run on the instance.

**Current shape:** VM.Standard1.1

Choose the target shape based on the requirements of your workload

| | Shape Name | OCPU | Memory (GB) | Local Disk (TB) | Network Bandwidth |
|---|---|---|---|---|---|
| ☐ | VM.Standard1.1 | 1 | 7 | Block Storage only | Up to 600 Mbps |
| ☐ | VM.Standard1.2 | 2 | 14 | Block Storage only | Up to 1.2 Gbps |
| ☐ | VM.Standard1.4 | 4 | 28 | Block Storage only | 1.2 Gbps |
| ☐ | VM.Standard2.1 | 1 | 15 | Block Storage only | 1 Gbps |
| ☐ | VM.Standard2.2 | 2 | 30 | Block Storage only | 2 Gbps |
| ☑ | VM.Standard2.4 | 4 | 60 | Block Storage only | 4.1 Gbps |
| ☐ | VM.Standard2.8 | 8 | 120 | Block Storage only | 8.2 Gbps |
| ☐ | VM.Standard2.16 | 16 | 240 | Block Storage only | 16.4 Gbps |
| ☐ | VM.Standard2.24 | 24 | 320 | Block Storage only | 24.6 Gbps |

1 Selected

# Mathematical Formulation

## Memory Usage Breakdown for Training

| | |
|---|---|
| Initial Training Data | $4N * \frac{\tau}{\epsilon} + a$ |
| Signals Dynamics Characterization | $(4N + (N + \mathrm{m}) * \mathrm{m}) * \frac{\tau}{\epsilon} + a$ |
| Least Squares Approximations | $(4N + (N + 4.6\mathrm{m} + 141) * \mathrm{m} + 32962) * \frac{\tau}{\epsilon} + a$ |
| Model Validation | $\big((M + 4) * N + (N + \mathrm{m} + M) * \mathrm{m} + M\big) * \frac{\tau}{\epsilon} + a$ |

N: number of signals
M: number of observations for training
m: a subset of M used for training
$\tau$: precision
$a$: deterministic memory usage of the CUDA Toolkit (ver. 10.1.243)
$\epsilon$: B to MB conversion factor = $1024^2$

# Mathematical Formulation – cont.

## Memory Usage Breakdown for Inferencing

Load Model: $$(4N + (N + \mathrm{m}) * \mathrm{m}) * \frac{\tau}{\epsilon} + a$$

Make Inferences: $$((2M' + 4) * N + (2M' + N + \mathrm{m}) * \mathrm{m} + M') * \frac{\tau}{\epsilon} + a$$

$M'$: number of observations for inferencing

The deterministic memory usage can be perfectly characterized as a function of variable size and precision

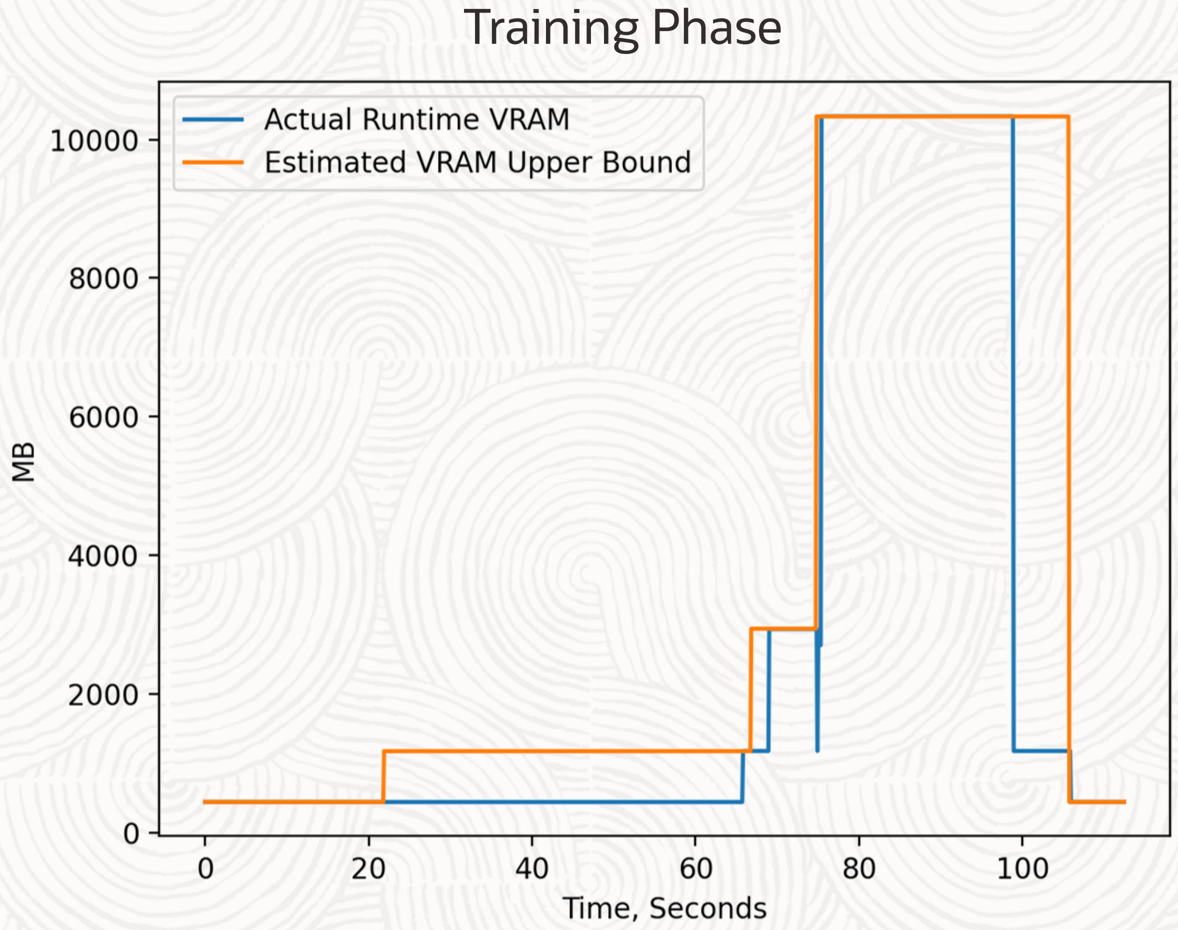
The stochastic memory usage behavior (e.g., proprietary functions in the CUDA library) is characterized leveraging 2D response-surface methodology between the inputs and outputs of the functions

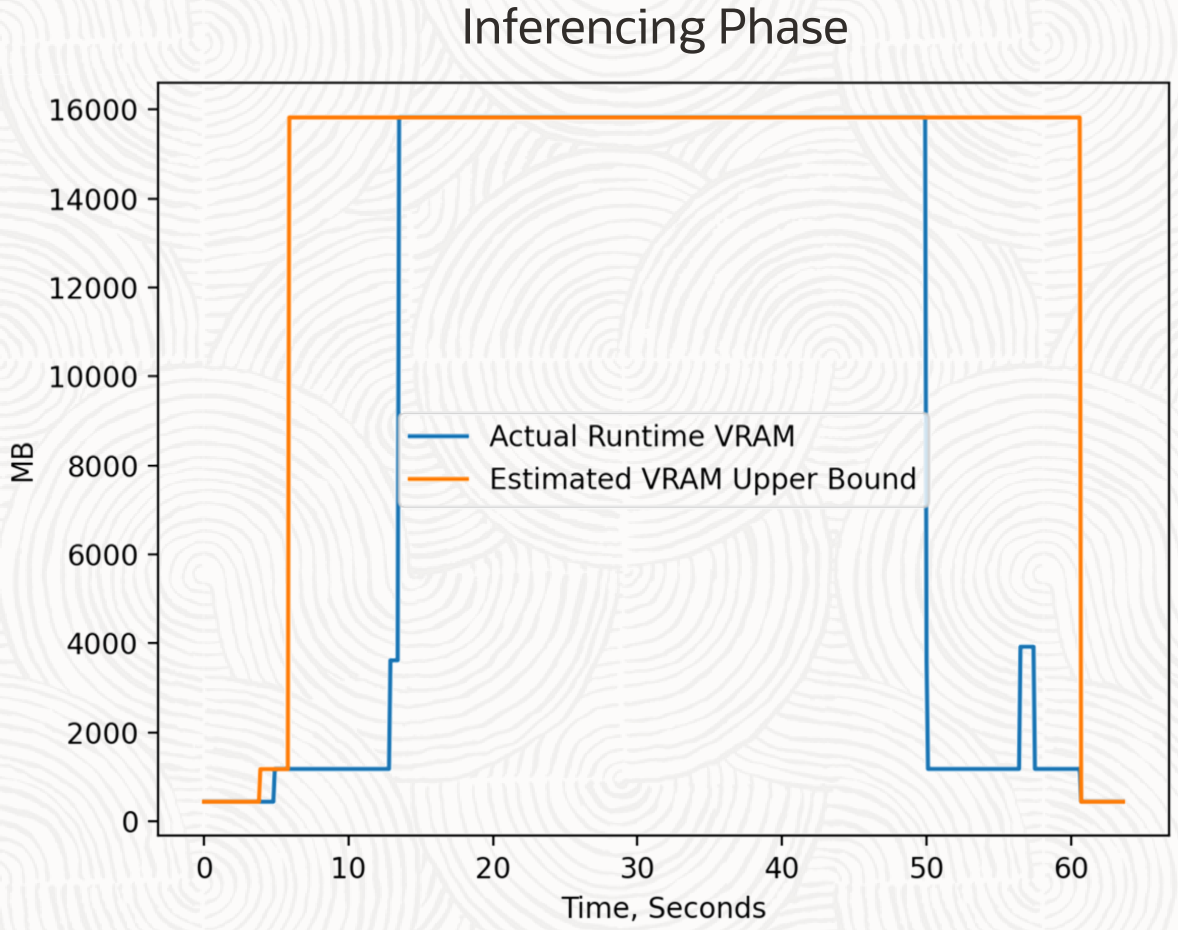
The general methodology employed in the end-to-end framework is adaptable to other nonlinear nonparametric ML prognostic techniques

# Validation on a NVIDIA GPU Instance

- A predictive-maintenance use case on an Oracle AD Service testbed
  - 16 OCPU, 320GB RAM
  - V100 GPU with 16GB VRAM
- Real IoT signals from the O&G industry
  - 4k signals and 100k observations for training
  - 4k signals and 80k observation for inferencing
- A lightweight MSET model with m = 8k observations (i.e., $D \in \mathbb{R}^{[8k \times 4k]}$)
- Computed VRAM usage prior to the run
- Measured VRAM usage during the run
- Each step of both training and inferencing phases is validated

# Validation Results

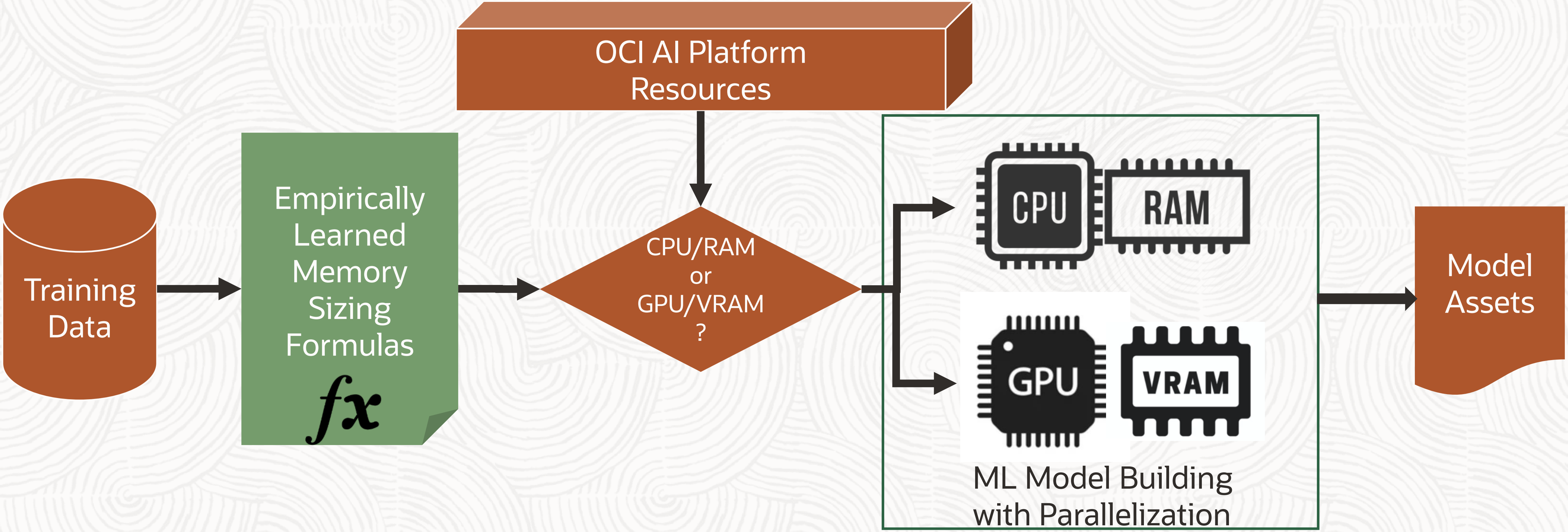


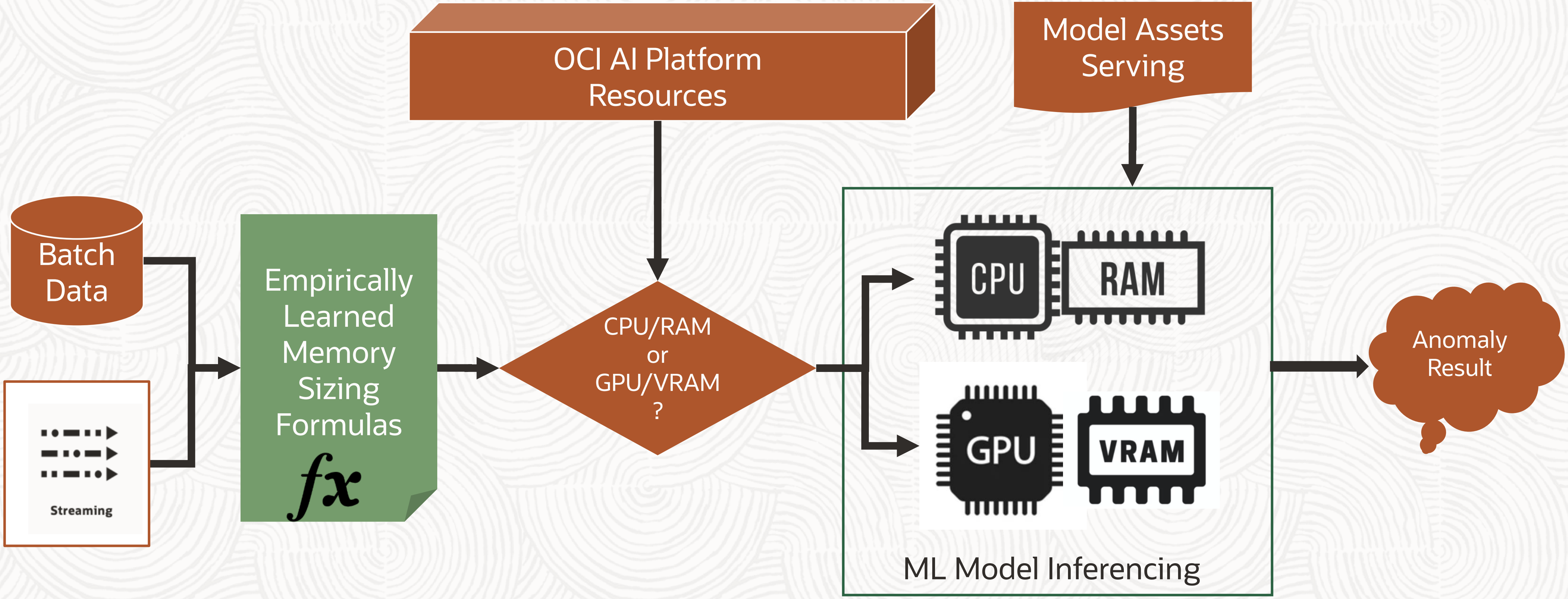

- The memory usage profiles were completely enveloped by our estimates
- The peak memory usages were accurately predicted in both phases with 0.04% residuals

# Leverage Memory-Sizing Formularization in AD Service

# AD Service Model Training Workflow



# AD Service Inferencing Workflow



# OCI AD Service Differentiators

### Automatic data preprocessing

- Imputes missing values based on ML based estimates
- Patented resampling automatically works with differing time interval signals
- Un-quantizes signal values to help build best model for quantized signal monitoring

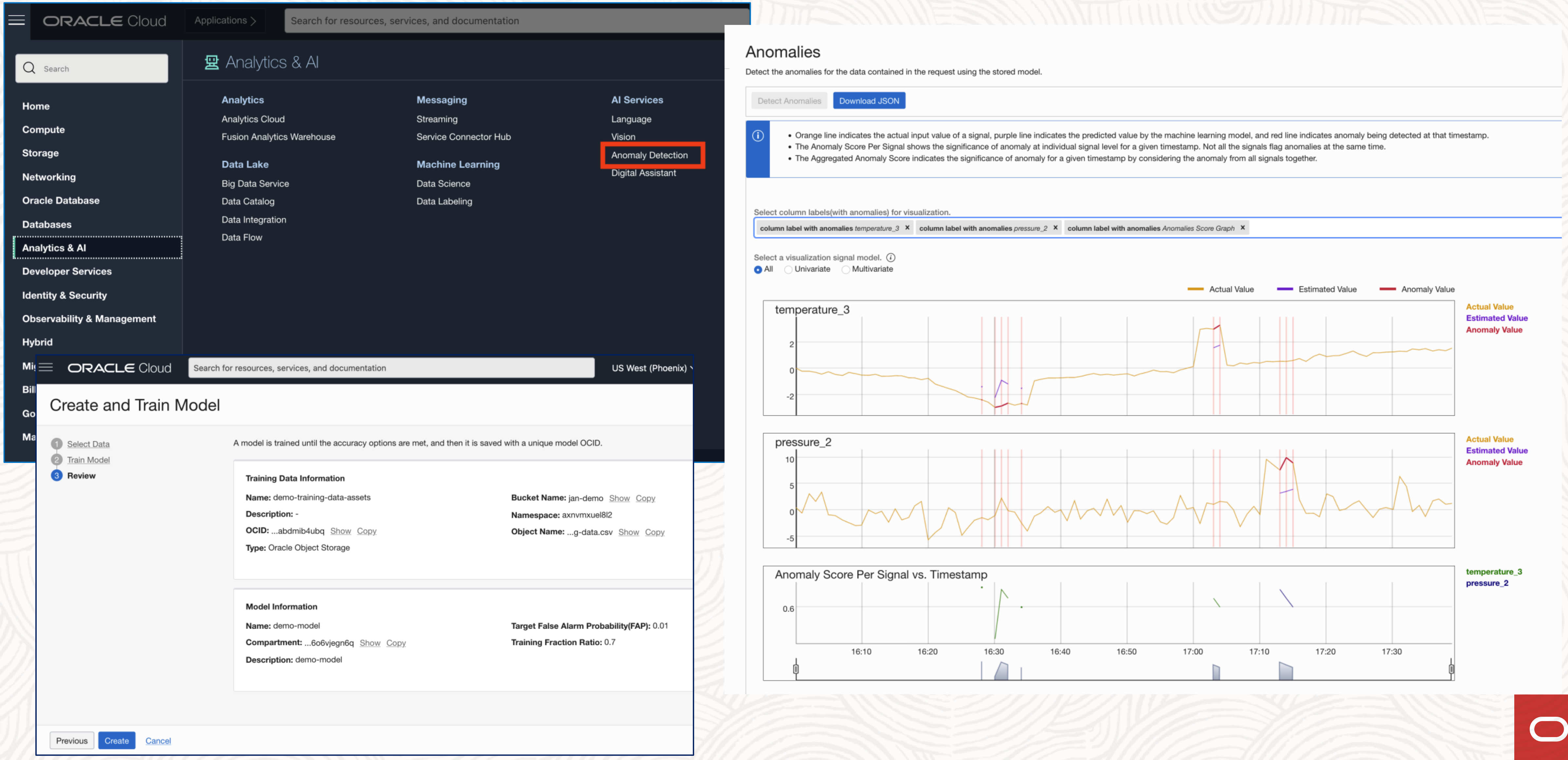
### Developer-focused AI service that automates data science

- Automatic best model creation for the data without needing data scientists
- Model output includes overall model accuracy, specific signal accuracy and signal specific statistics for developers to decide if the model is effective for the business use case

### Automate business workflows for immediate action

- Estimated value for each identified anomaly helps to assess severity of the anomaly occurrence
- Aggregated score of anomaly over time provides whether the anomalies are becoming severe overtime
- Signal specific anomaly score helps to assess relative severity of anomalies across signals

# OCI AD Service in Console



# How to Consume OCI AD Service

- Software Development Kits
  - SDK for Java
  - SDK for JavaScript and TypeScript
  - SDK for Python
  - SDK for .NET
  - SDK for Go
  - SDK for Ruby
- REST APIs
- OCI Command Line Interface

**Docs**
- Release notes
- Docs
- API documentation

**Reference architectures**
- Anomaly detection for managing assets and predictive maintenance
- Detecting anomalies to predict failure

**Blogs**
- Product blog
- Algorithm blog

# Thank You

Interested in Trying Out AD Service?

Viji Krishnamurthy (PM):

viji.krishnamurthy@oracle.com

Technical Questions?

Guang Wang: guang.wang@oracle.com

Jason Ding: jason.ding@oracle.com

Kenny Gross: kenny.gross@oracle.com

Resources:

Anomaly Detection Documentation: https://docs.oracle.com/en-us/iaas/Content/anomaly/using/home.htm

Oracle MSET2 Blog: https://blogs.oracle.com/bigdata/real-time-machine-learning-use-case

Acknowledgement:

The presented methodology is part of the collaborative project with our former colleague Wei Jiang who contributed to the inception and development phases of the work.