Electronics for the Future

Solist-AI

For effective anomaly detection and supervised learning

Solist-AI™ algorithm and learning, AxICORE-ODL features

April 21, 2025
ROHM Co., Ltd.

Contents

- **Neural Network and Algorithm of Solist-AI™**
- Unsupervised Learning by Solist-AI™
- Supervised Learning by Solist-AI™
- Hardware Accelerator for AI Processing, AxlCORE-ODL

- **Solist-AI™ : *<u>S</u>olution with <u>O</u>n-device <u>L</u>earning <u>I</u>c for <u>ST</u>andalone-<u>AI</u>***
  - ➢ On-device learning IC enables standalone AI solution

Two elements that make Solist-AI™ possible

Solist-AI™ MCU

On-device learning technology	AI accelerator
● **No** pre-learning on the cloud is required ➢ **No** network required/**zero** delays ➢ **Low** security risk ● Relearning and additional training can be completed on-site. ➢ Ability to adapt to variability and environmental changes	● **Low-cost** hardware circuit ➢ 1000 times faster than software processing ➢ **Low** power consumption ● Flexible AI configuration via software ● Can be installed on various ICs

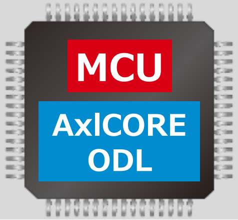
Utilities

Support for considering the introduction/implementation of Solist-AI™ MCU

- **Solist-AI™ Sim**
  - ➢ Check the effectiveness of AI utilization through simulation. Easy, simple, and repeatable.
- **Solist-AI™ Scope**
  - ➢ High-speed real-time debugger
  - ➢ Waveform viewer

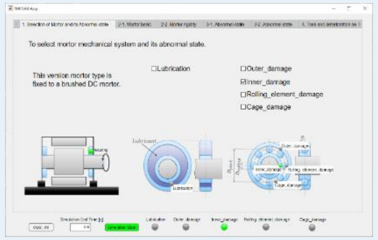
AI processing with only IC and sensor

➢ **Standalone**

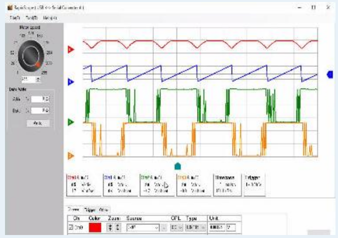


Solist-AI™ MCU

\+



Solist-AI™ Sim

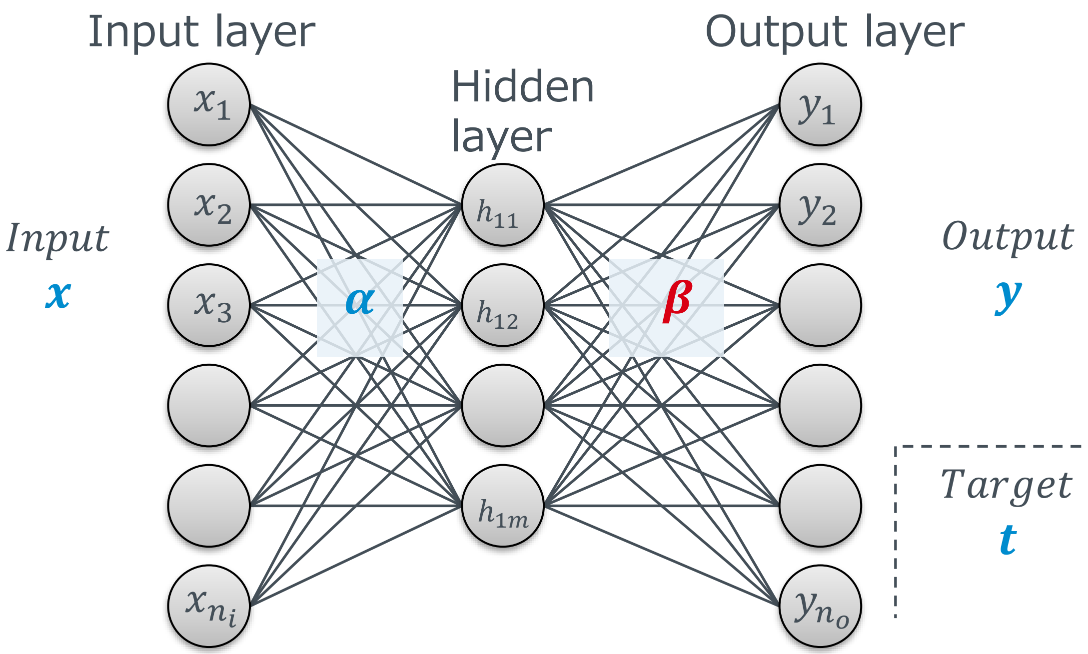


Solist-AI™ Scope

Neural Network and Algorithm of Solist-AI™

- Neural network of Solist-AI™ is an FFNN (feedforward neural network)
  - ➢ The diagram below shows a three-layer neural network with one hidden layer.

- Algorithm of Solist-AI™ is a type of ELM(Extreme Learning Machine), which does not update the weight $\boldsymbol{\alpha}$, but only updates the weight $\boldsymbol{\beta}$.
  - ➢ Our algorithm has been modified to realize On-Device Learning.



Learning and Prediction

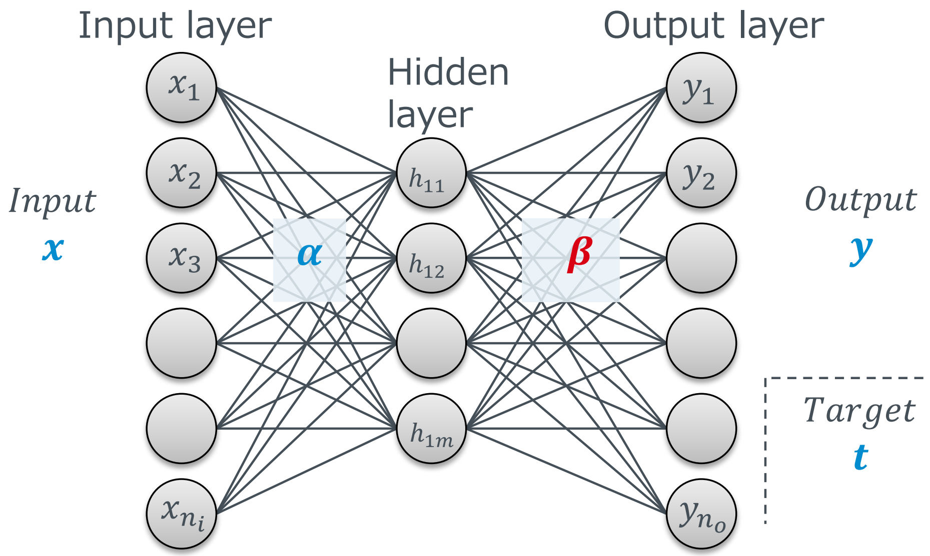
What is Learning?

Updating the weight $\boldsymbol{\beta}$ so that the output layer data $\boldsymbol{y}$ of the AI is closer to the input layer data $\boldsymbol{x}$ ($\boldsymbol{y} = \boldsymbol{x}$) or the target data $\boldsymbol{t}$ ($\boldsymbol{y} = \boldsymbol{t}$) is called **learning**.

In particular, updating weight $\boldsymbol{\beta}$ so that the output layer data $\boldsymbol{y}$ of AI is closer to the input layer data $\boldsymbol{x}$ ($\boldsymbol{y} = \boldsymbol{x}$) is training for anomaly detection, also known as **unsupervised learning**. On the other hand, updating the weight $\boldsymbol{\beta}$ so that the target data $\boldsymbol{t}$ (training data) can be reproduced is known as **supervised learning**.

What is Prediction?

Calculating the output/prediction $\boldsymbol{y}$ when the input $\boldsymbol{x}$ is set to the AI without updating the weight $\boldsymbol{\beta}$ is called inference.

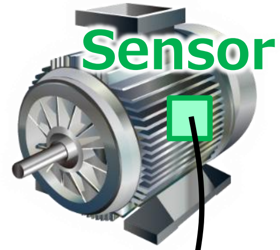


Example of Anomaly Detection using Unsupervised Learning

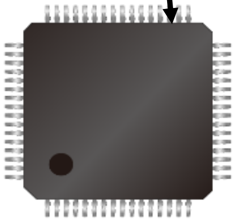
ROHM SEMICONDUCTOR

Solist-AI

Target device of anomaly detection



Sensor signal



AI MCU

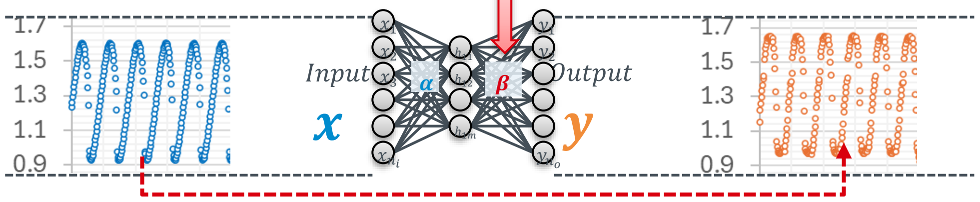
AI MCU starts prediction after unsupervised learning is completed.

Initial learning (Unsupervised Learning)

Normal input signal

Updating weight β

AI output signal (Reproduce normal signal)



Learning to reproduce the sensor input signal under normal condition.

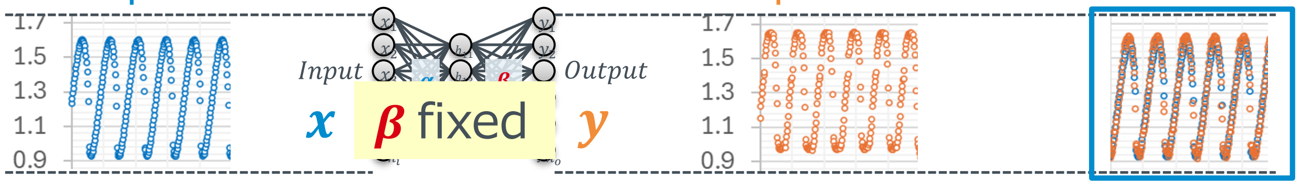
Prediction (Device Monitoring)

Normal device

Input

Output

Difference between x and y is small.



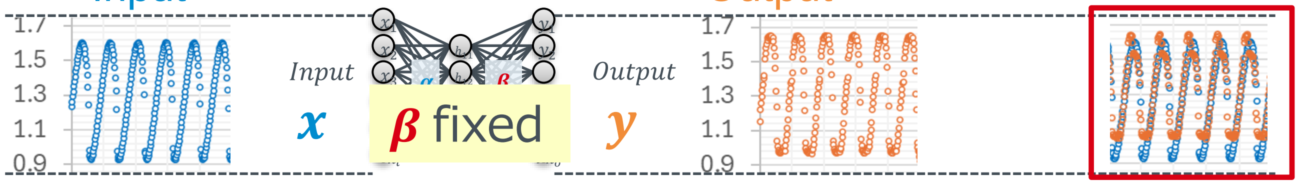
Input signal reproducibility of AI is high ⇒ Input known by AI ⇒ Normal state

Abnormal device

Input

Output

Difference between x and y is large.



Input signal reproducibility of AI is low ⇒ Input unknown by AI ⇒ Abnormal state

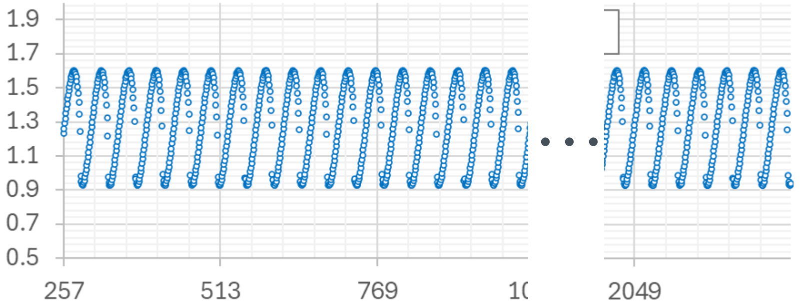
Contents

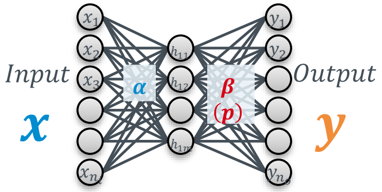
- Neural Network and Algorithm of Solist-AI™
- **Unsupervised Learning by Solist-AI™**
- Supervised Learning by Solist-AI™
- Hardware Accelerator for AI Processing, AxlCORE-ODL

Unsupervised Learning by Solist-AI™

- Let's consider an example where time-series waveform data is input directly into Solist-AI™ without any preprocessing. What we want to do is detect anomalies using unsupervised learning.

Input to Solist-AI™
(Time series waveform data)



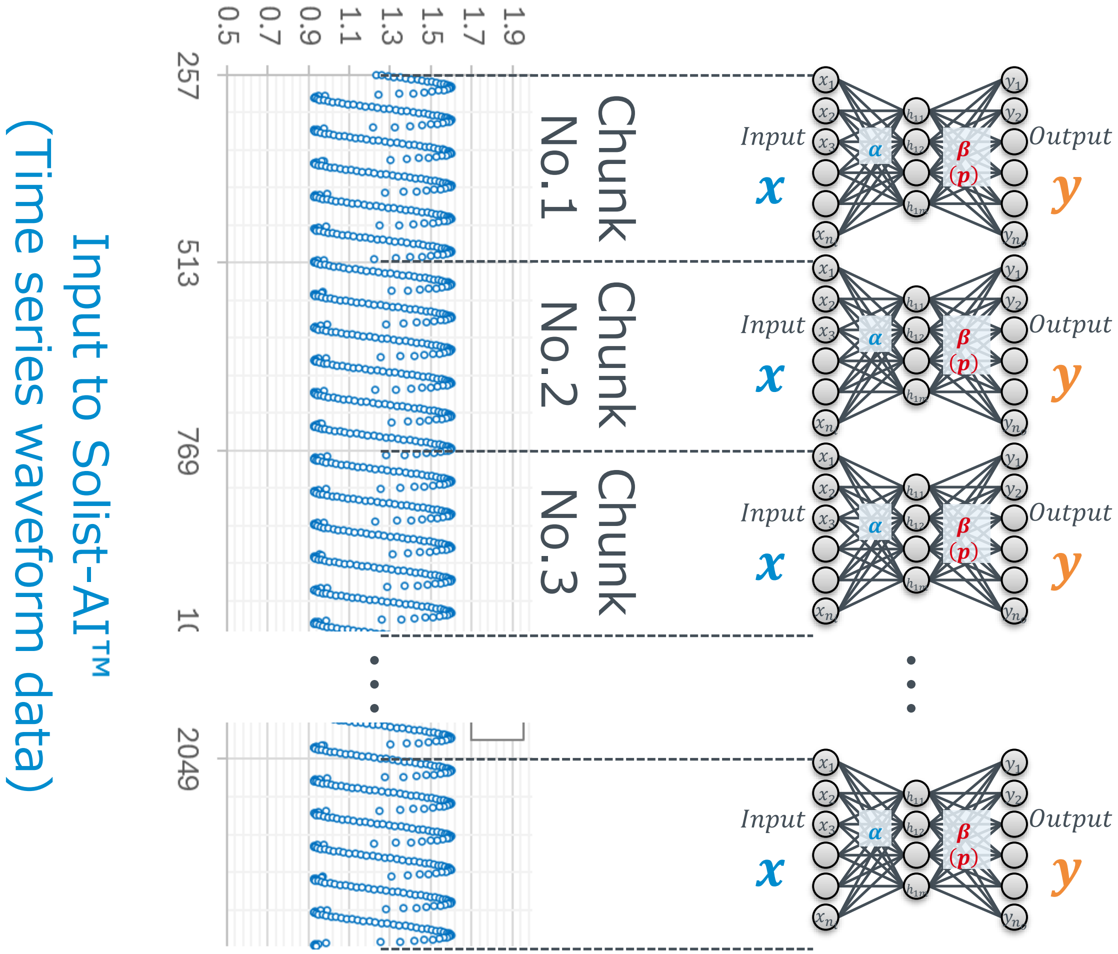


Normal
or
Anomaly

Unsupervised Learning by Solist-AI™

Solist-AI

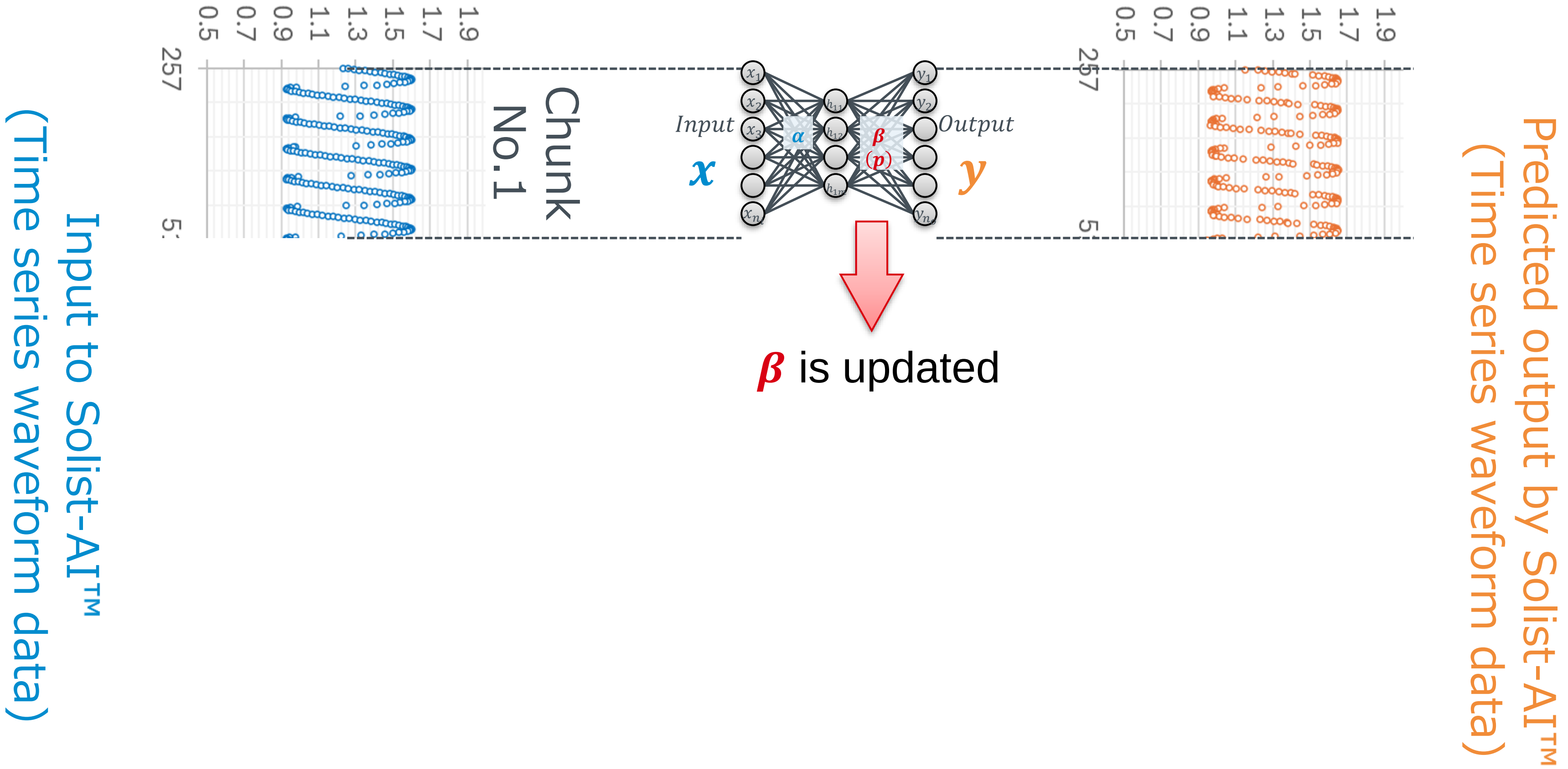
- A chunk is defined as a single set of data that is input into Solist-AI™.



Unsupervised Learning by Solist-AI™

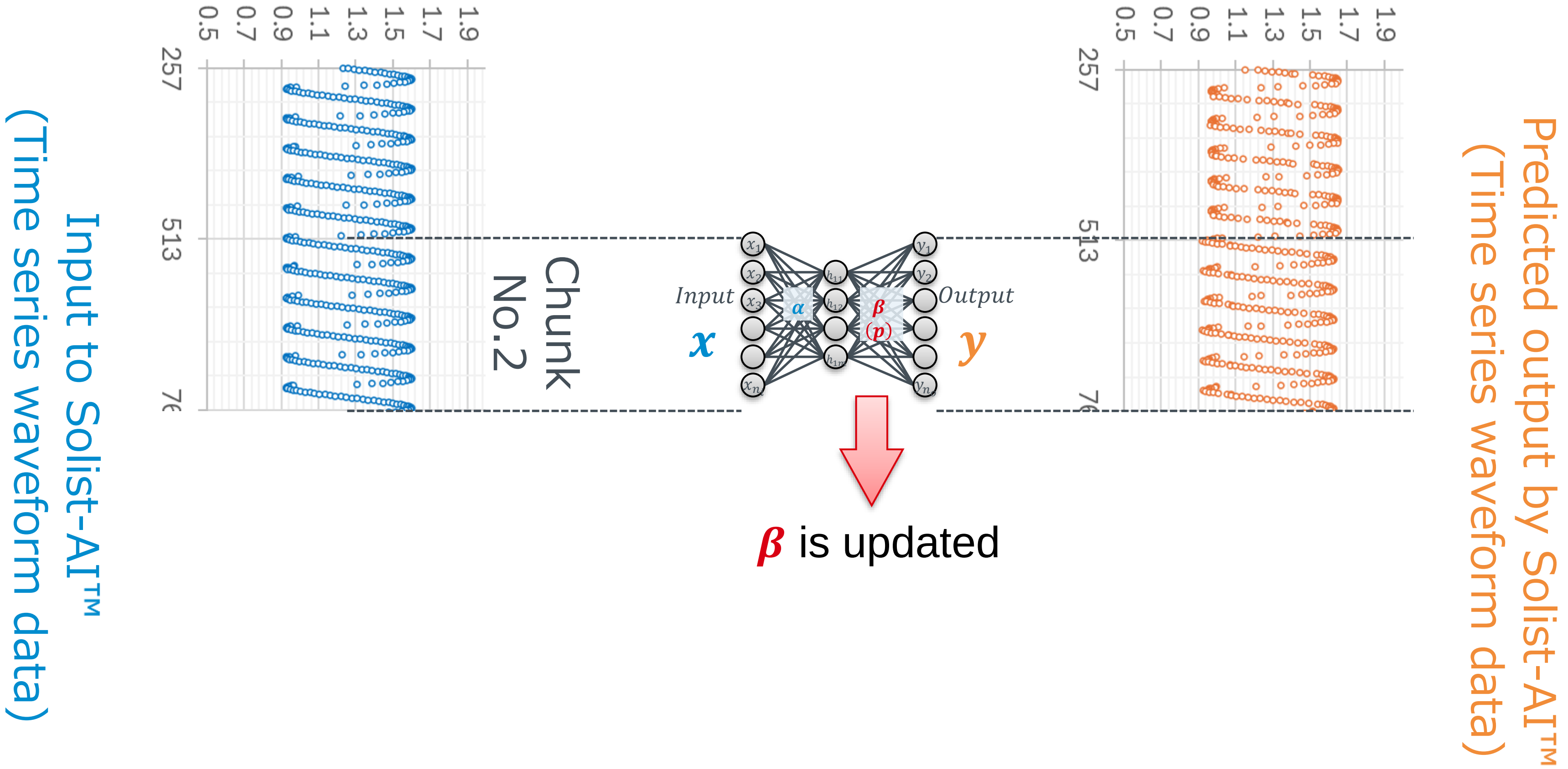
Solist-AI

- In unsupervised learning, the weight $\boldsymbol{\beta}$ is updated so that the output layer data $\boldsymbol{y}$ reproduces the input layer data $\boldsymbol{x}$.



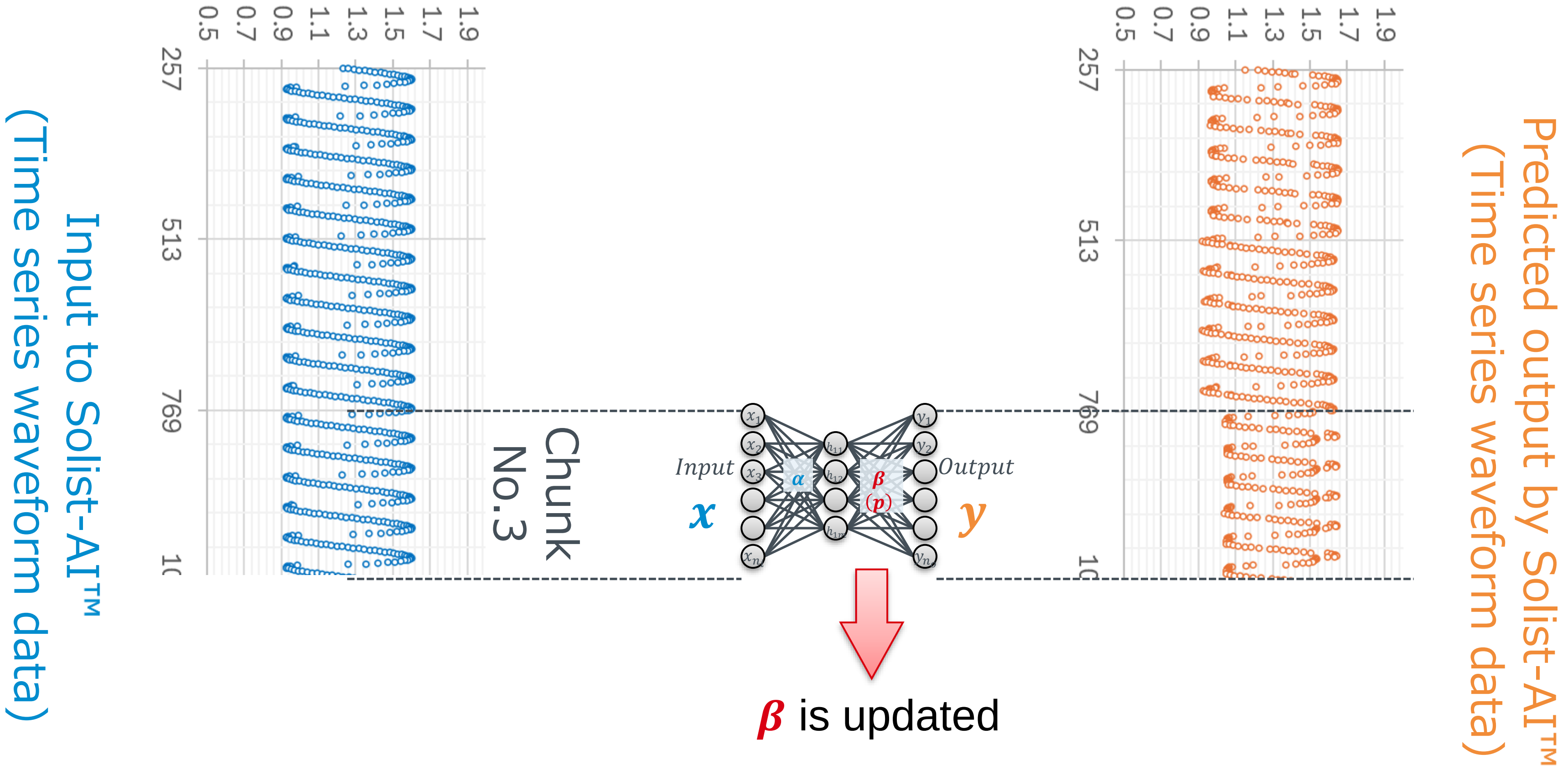
Unsupervised Learning by Solist-AI™

- In unsupervised learning, the weight $\boldsymbol{\beta}$ is updated so that the output layer data $\boldsymbol{y}$ reproduces the input layer data $\boldsymbol{x}$.



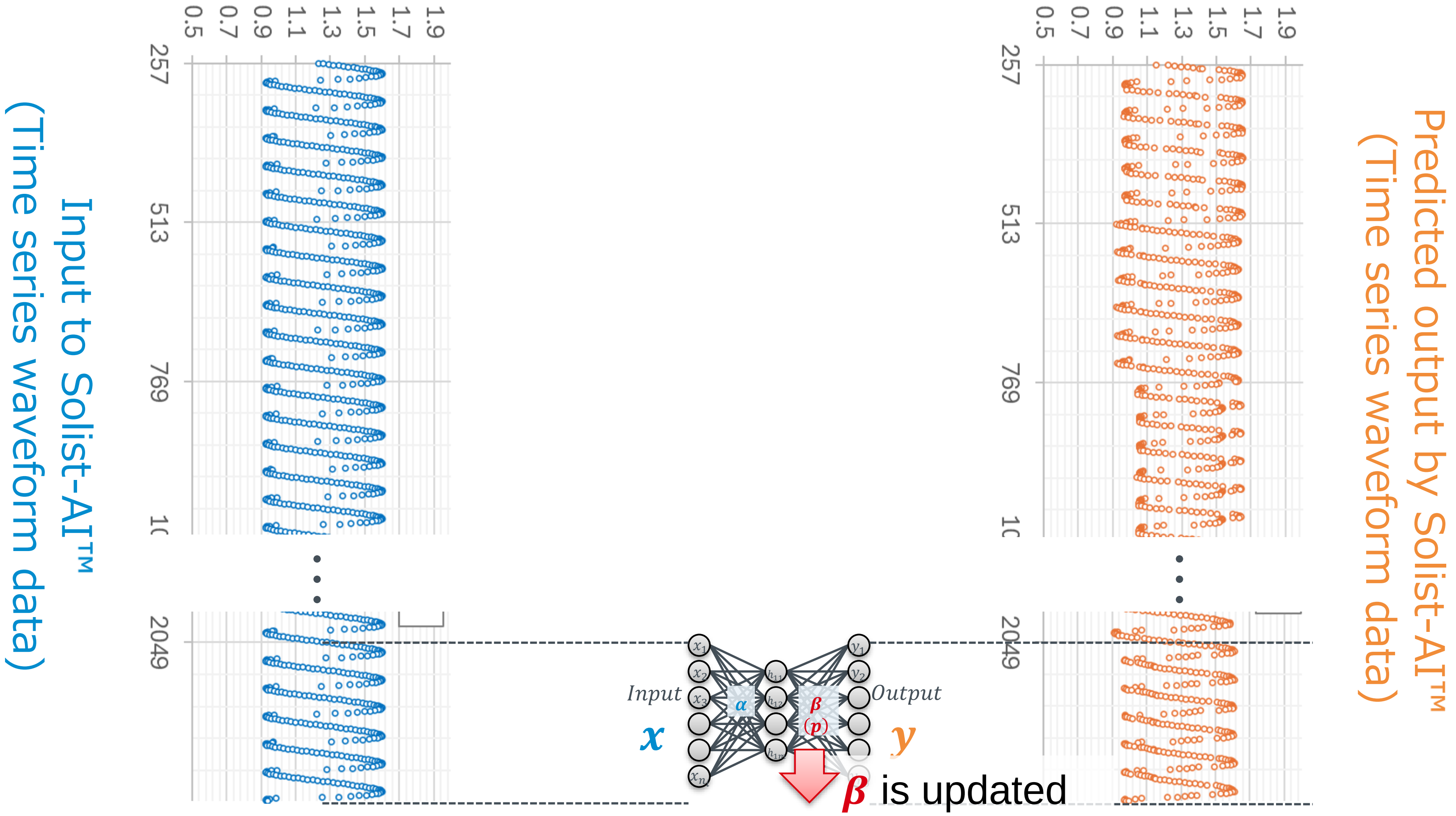
Unsupervised Learning by Solist-AI™

- In unsupervised learning, the weight β is updated so that the output layer data y reproduces the input layer data x.



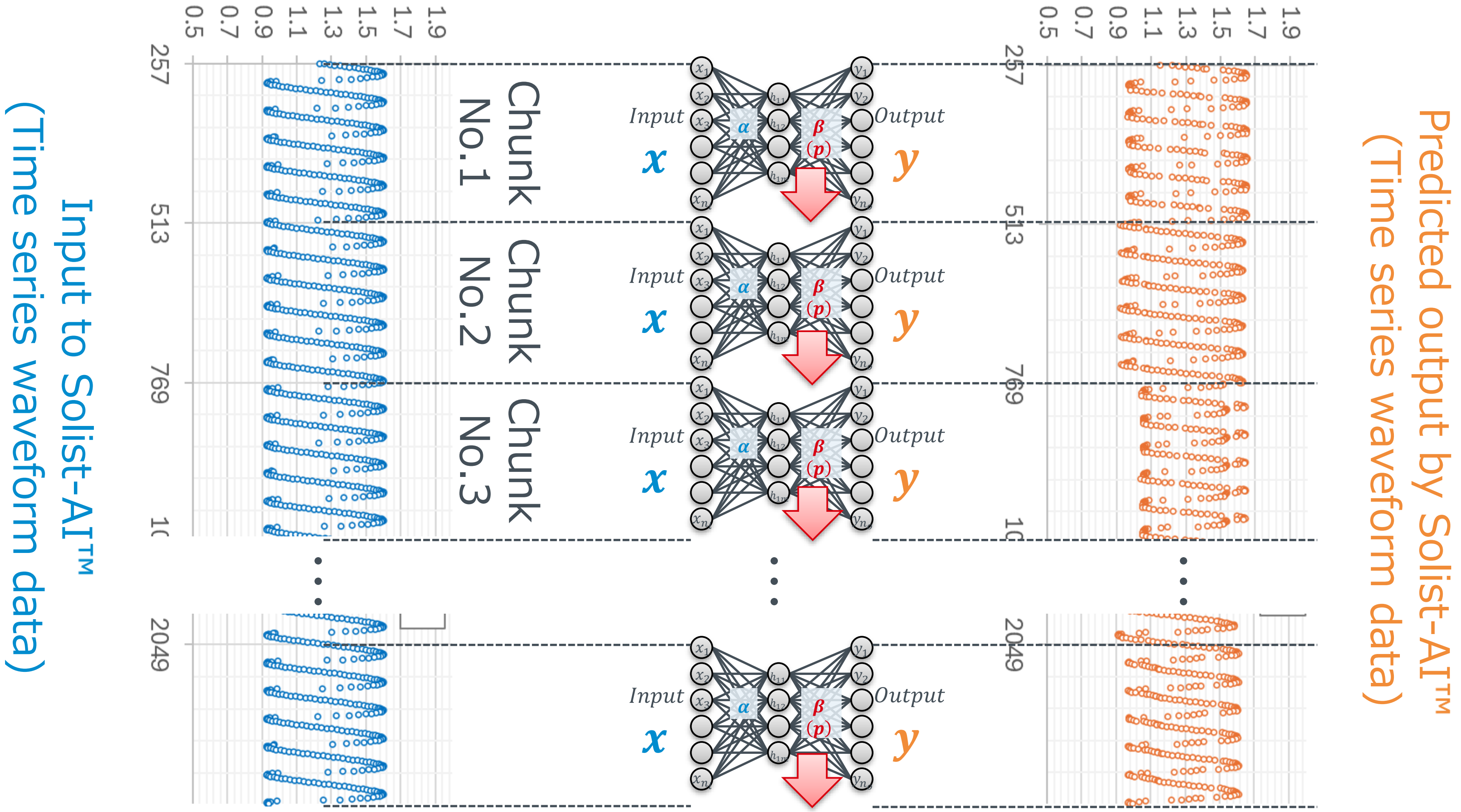
Unsupervised Learning by Solist-AI™

- In unsupervised learning, the weight β is updated so that the output layer data y reproduces the input layer data x.



Unsupervised Learning by Solist-AI™

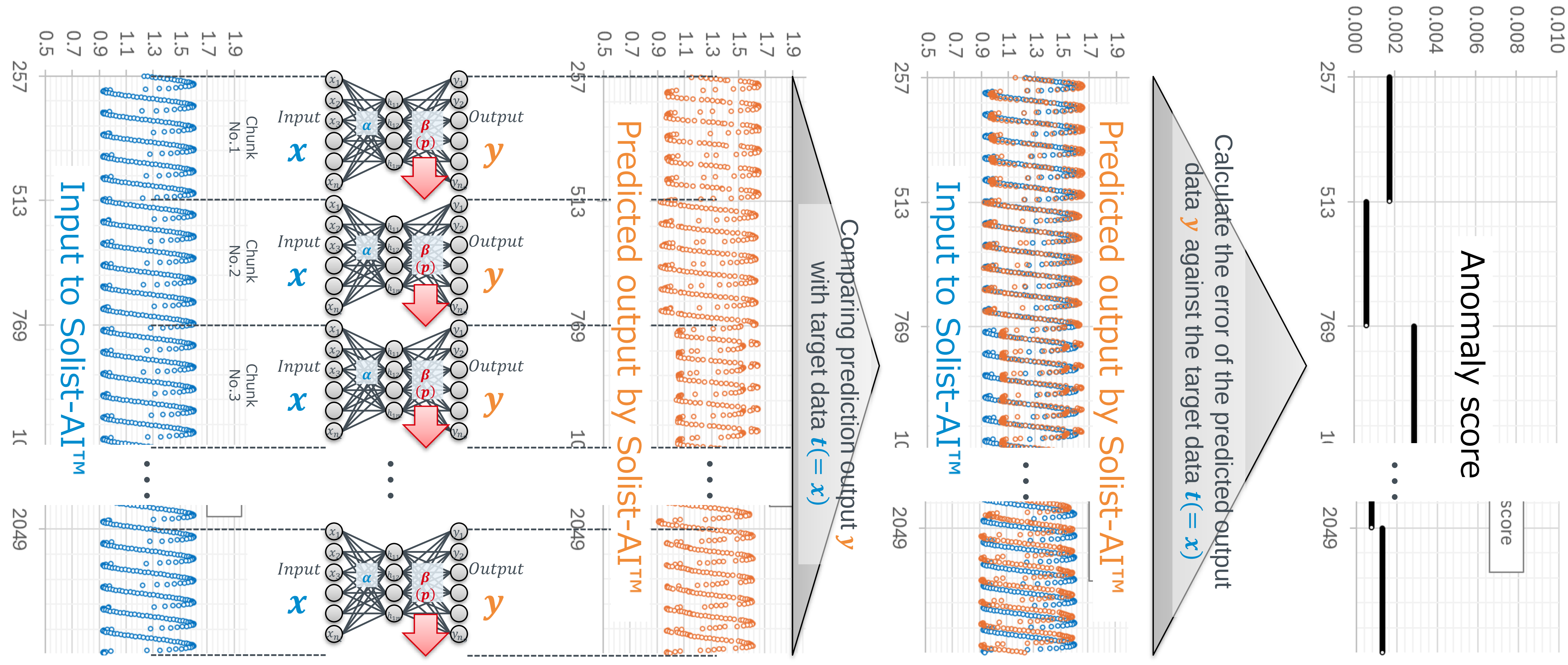
- In unsupervised learning, the weight $\boldsymbol{\beta}$ is updated so that the output layer data $\boldsymbol{y}$ reproduces the input layer data $\boldsymbol{x}$.



Unsupervised Learning by Solist-AI™

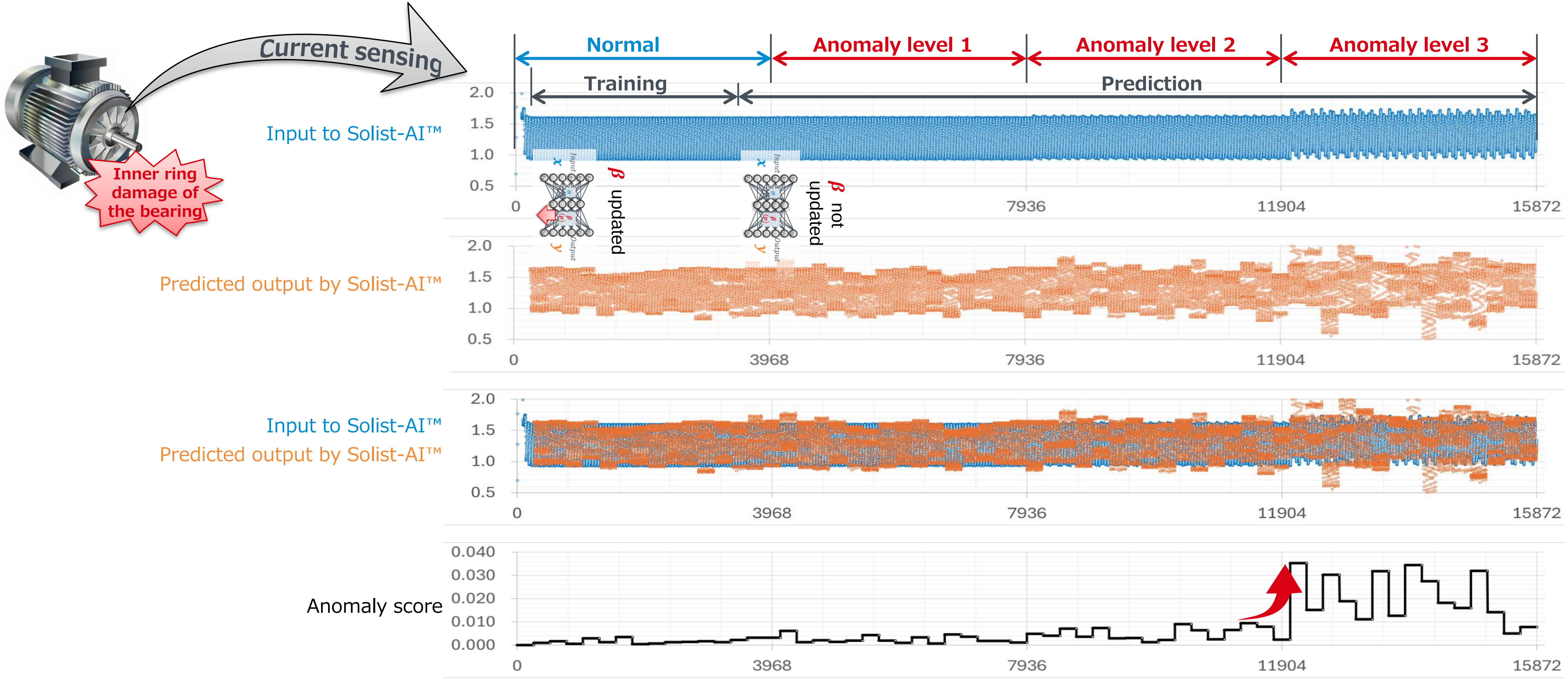
Solist-AI

- Solist-AI™ MCU outputs the anomaly score, which is an index showing how much the input data deviates from what has been learned.



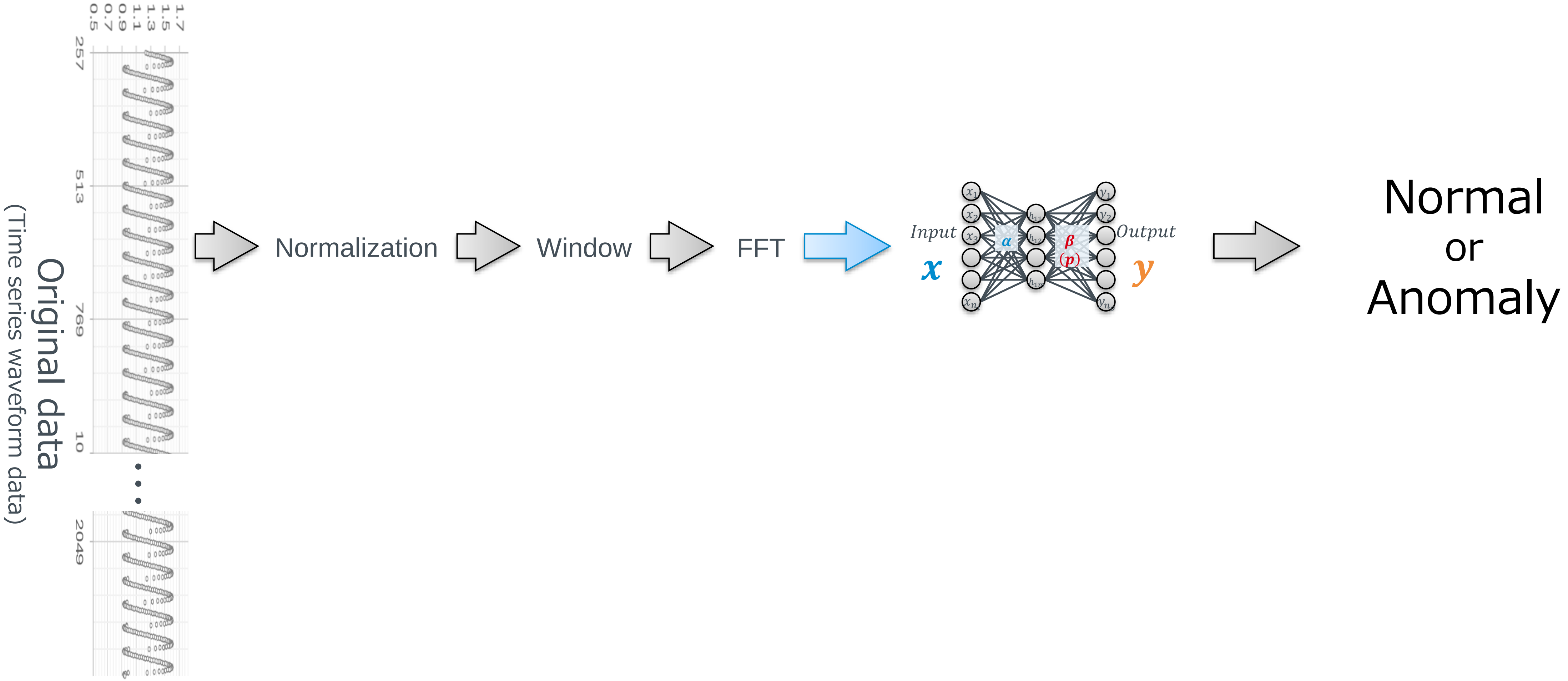
Unsupervised Learning by Solist-AI™

- In fact, the data shown up to this point in the slides is only a portion of the motor current data under normal conditions and when the bearing is damaged.



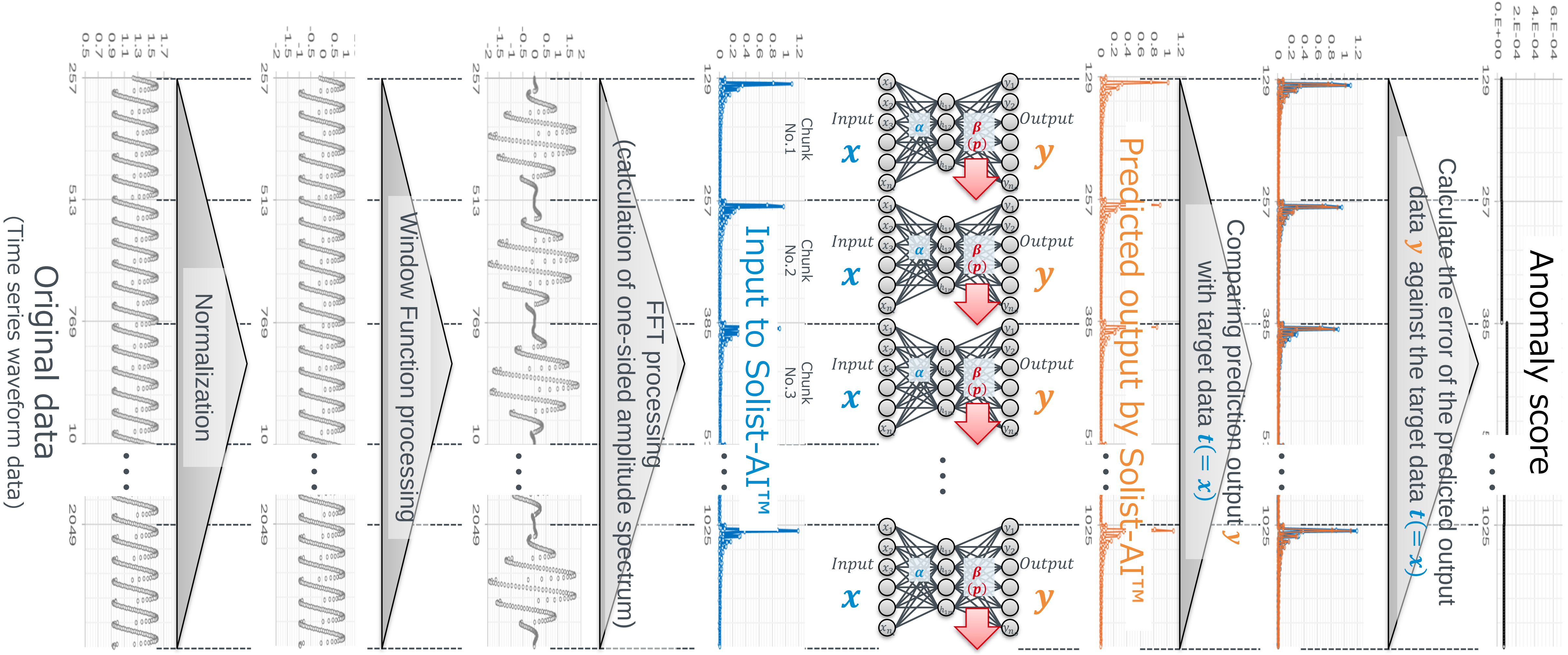
Unsupervised Learning by Solist-AI™, **using the preprocessing**

- Next, let's consider an example of preprocessing time-series waveform data, such as normalization and FFT, before inputting it into Solist-AI™.



Unsupervised Learning by Solist-AI™, using the preprocessing

- Next, let's consider an example of preprocessing time-series waveform data, such as normalization and FFT, before inputting it into Solist-AI™.

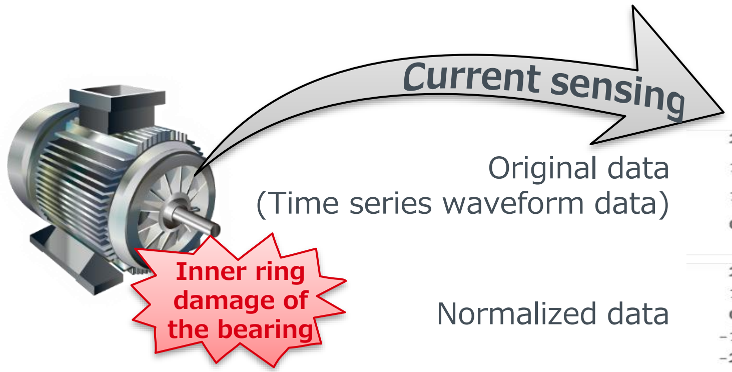


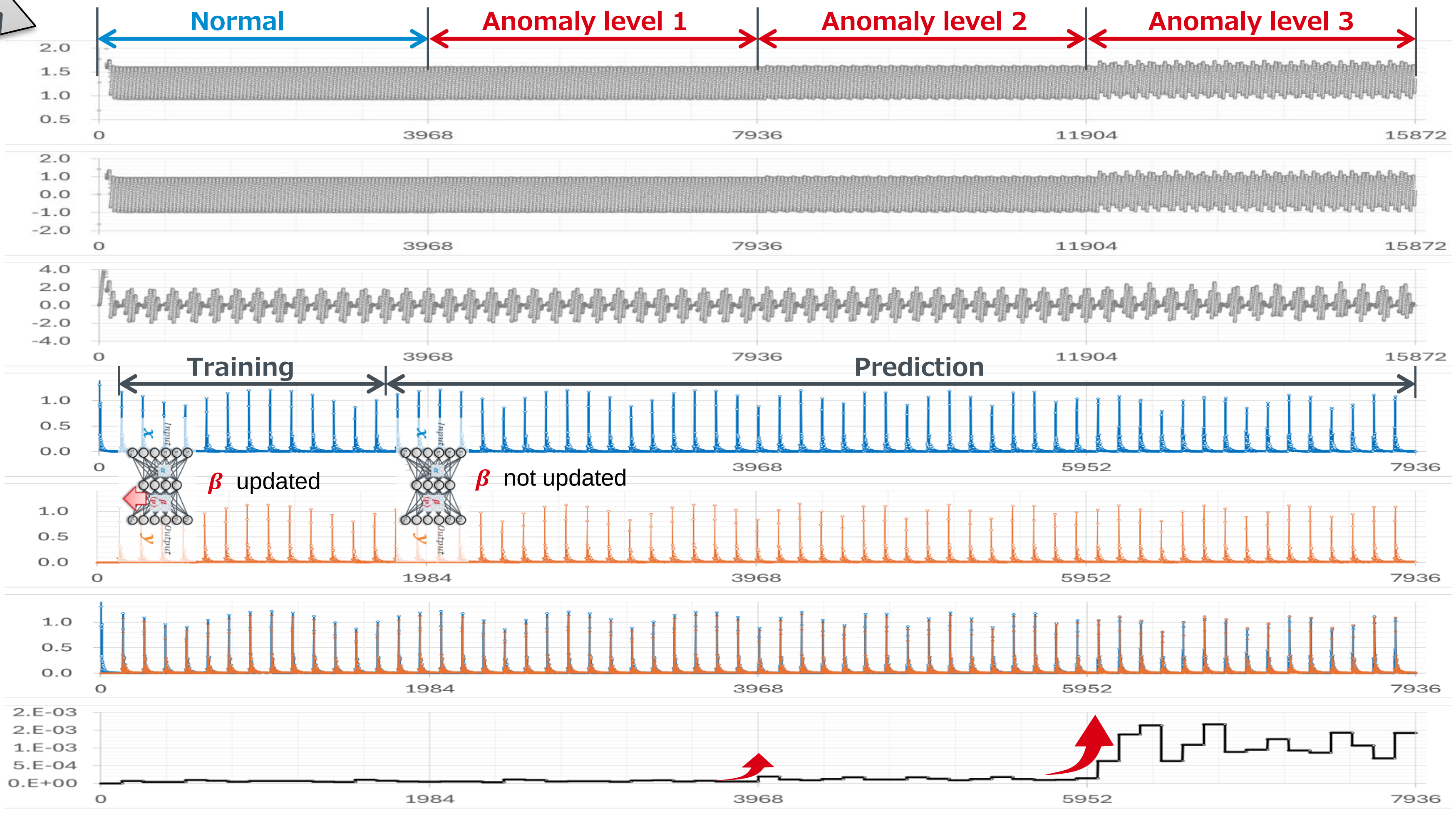
Unsupervised Learning by Solist-AI™, **using the preprocessing**

ROHM SEMICONDUCTOR

Solist-AI

- By preprocessing the motor current data, the anomaly score when bearing damage occurs becomes significantly greater than the anomaly score during normal.



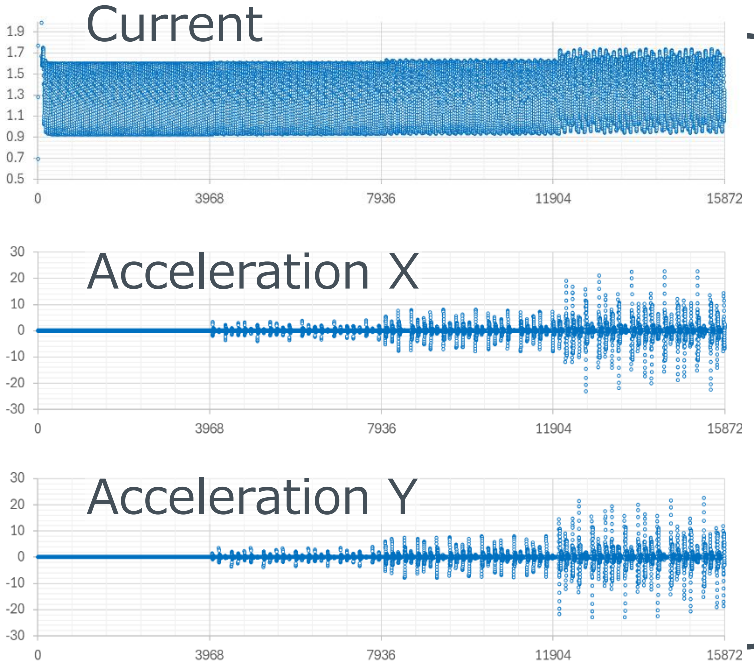


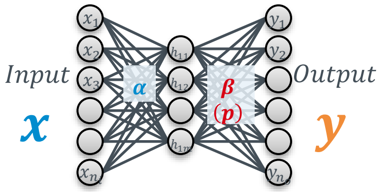
Unsupervised Learning by Solist-AI™, **using multiple types of sensors**

Solist-AI

- Using two-axis acceleration sensor and current sensor data, early signs of motor anomaly can be detected.

Input to Solist-AI™
(Time series waveform data)

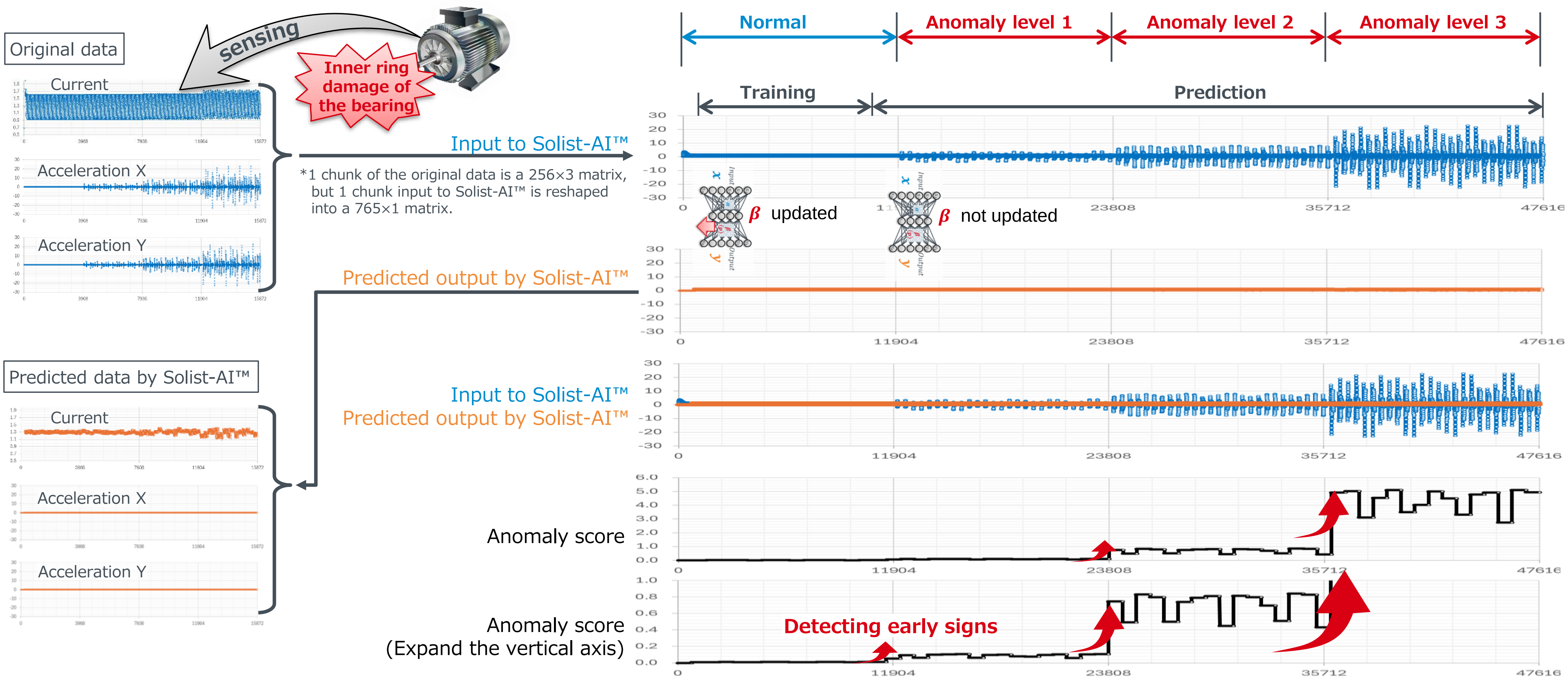




Normal
or
Anomaly

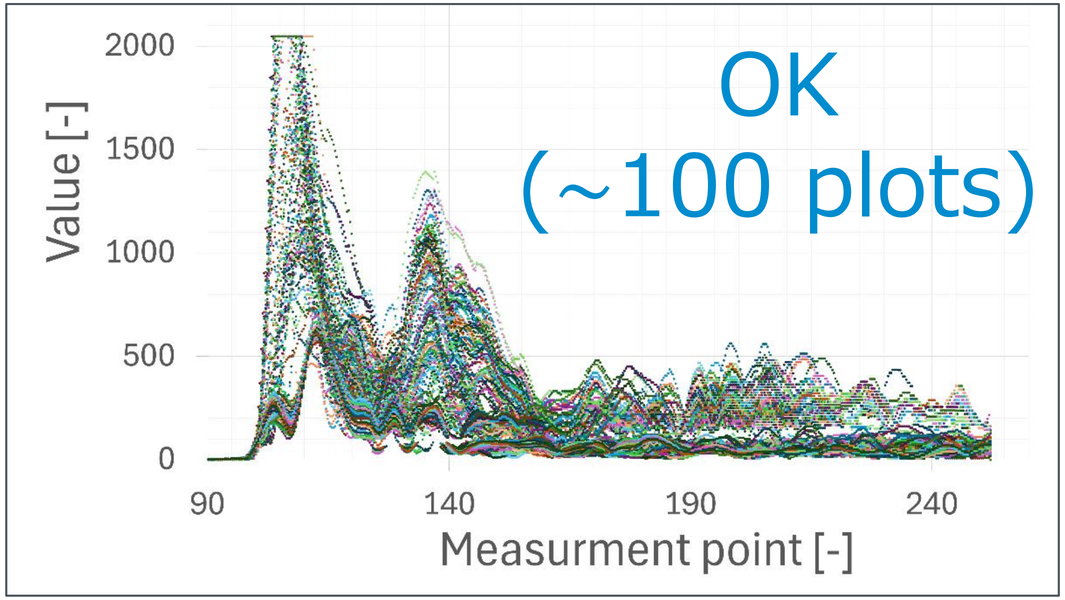
Unsupervised Learning by Solist-AI™, **using multiple types of sensors**

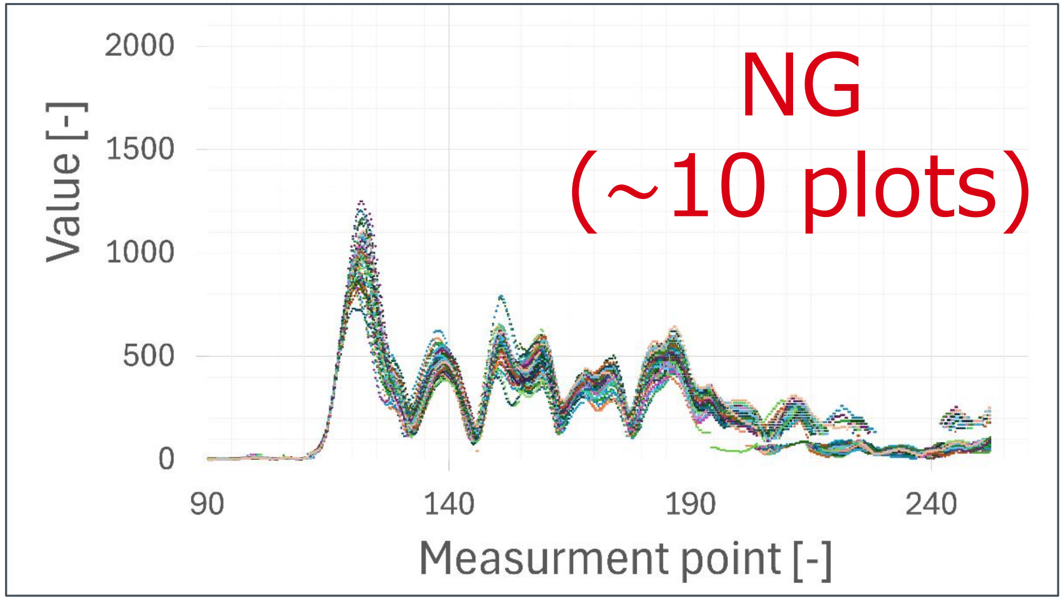
- Using two-axis acceleration sensor and current sensor data, early signs of motor anomaly can be detected.



Unsupervised Learning by Solist-AI™, **using non-time series data**

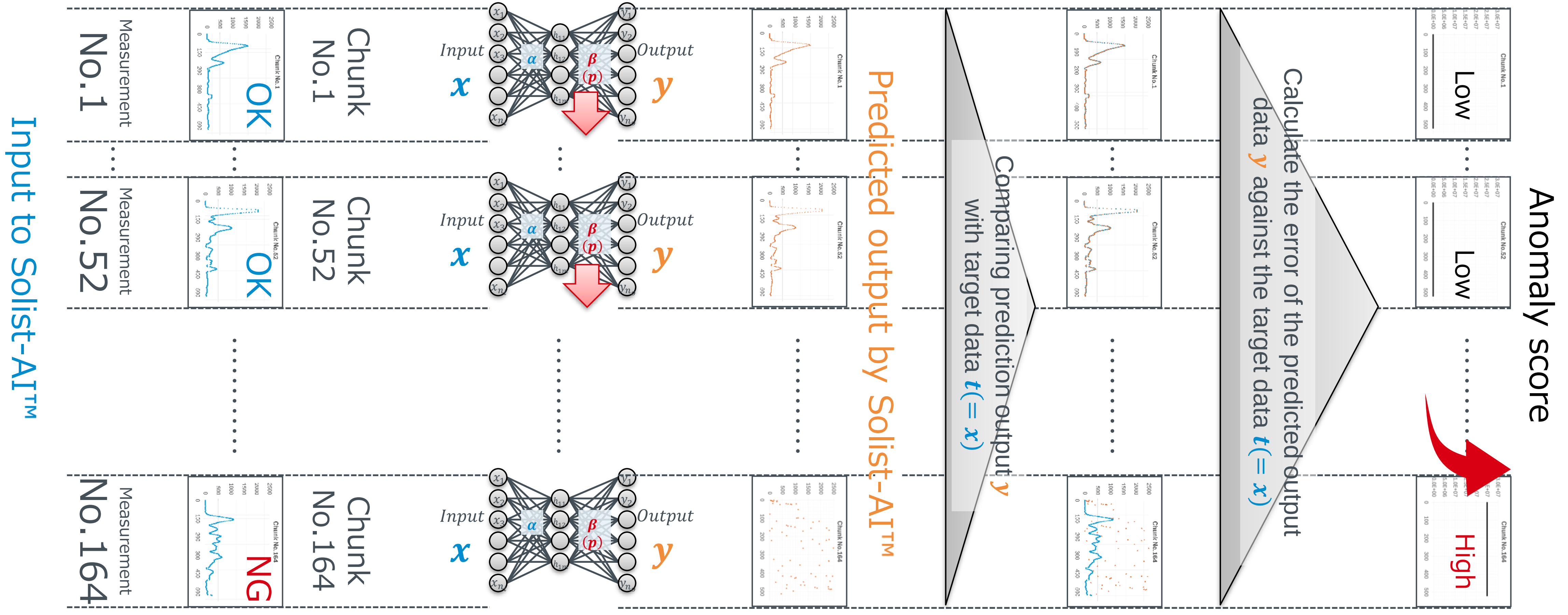
- For each of the OK and NG cases, multiple measurement data were overlaid on a single graph.





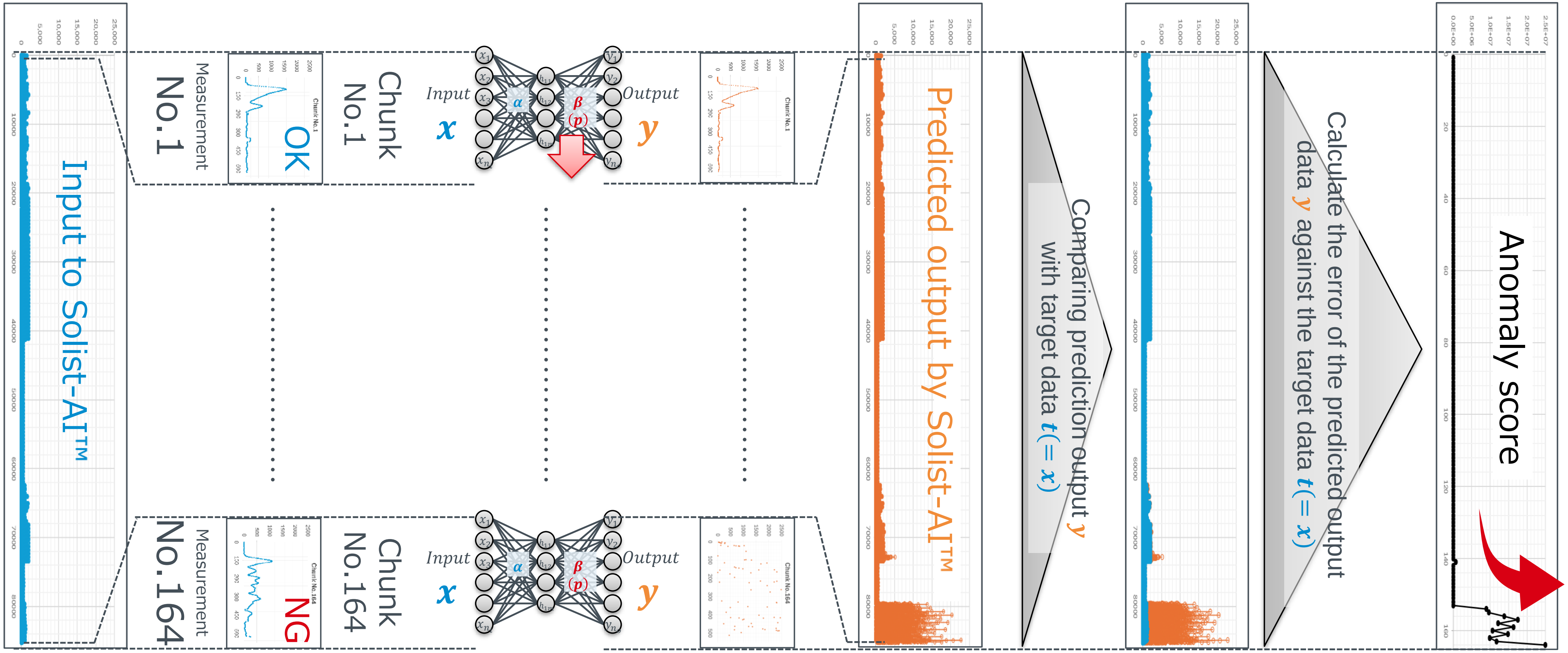
Unsupervised Learning by Solist-AI™, **using non-time series data**

- Each measurement data corresponds to a chunk of data to be input to Solist-AI™. The AI model is trained only on OK data and the weight $\boldsymbol{\beta}$ is updated so that the output layer data $\boldsymbol{y}$ produces the target data $\boldsymbol{x}$.



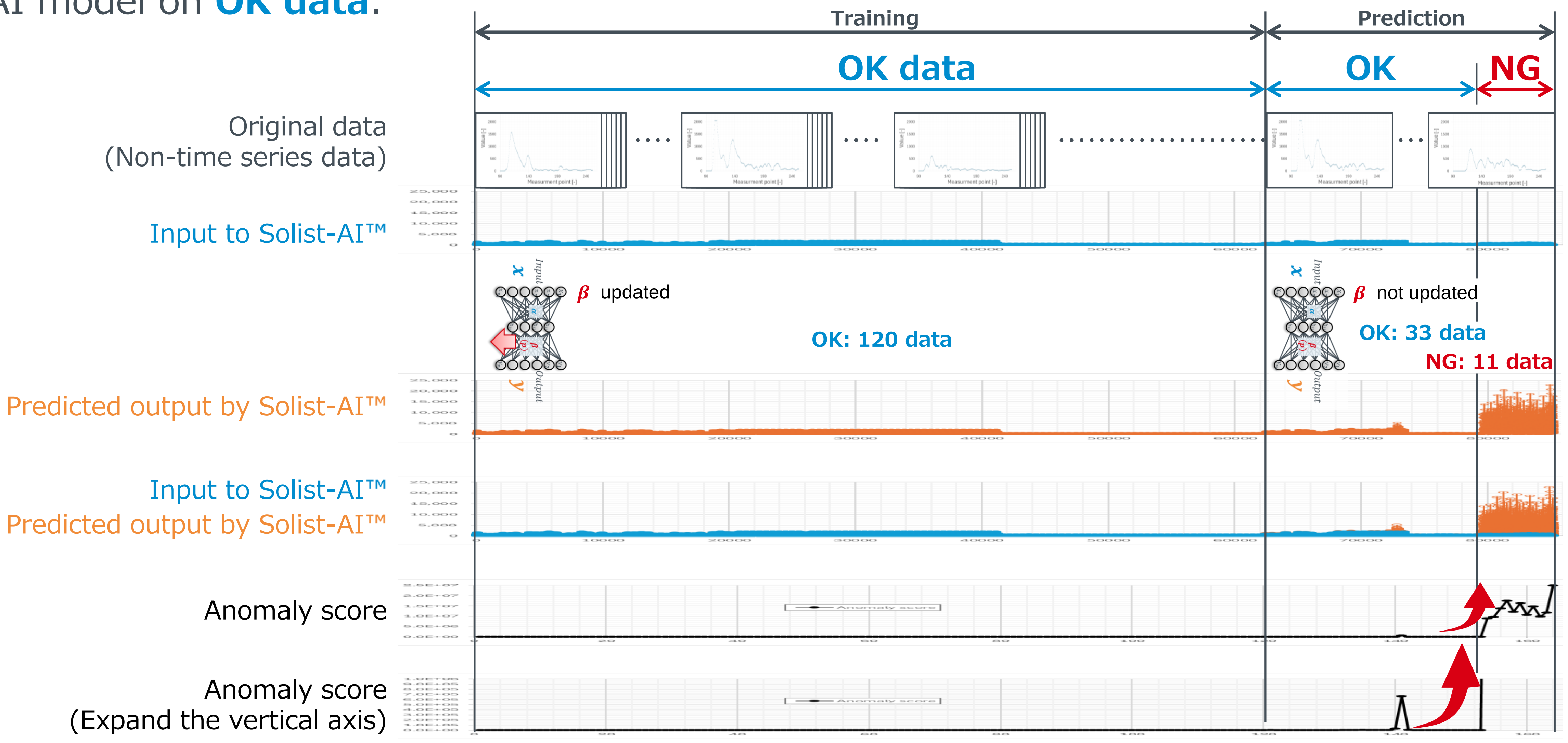
Unsupervised Learning by Solist-AI™, **using non-time series data**

- Each measurement data corresponds to a chunk of data to be input to Solist-AI™. The AI model is trained only on OK data and the weight β is updated so that the output layer data y produces the target data x.



Unsupervised Learning by Solist-AI™, using non-time series data

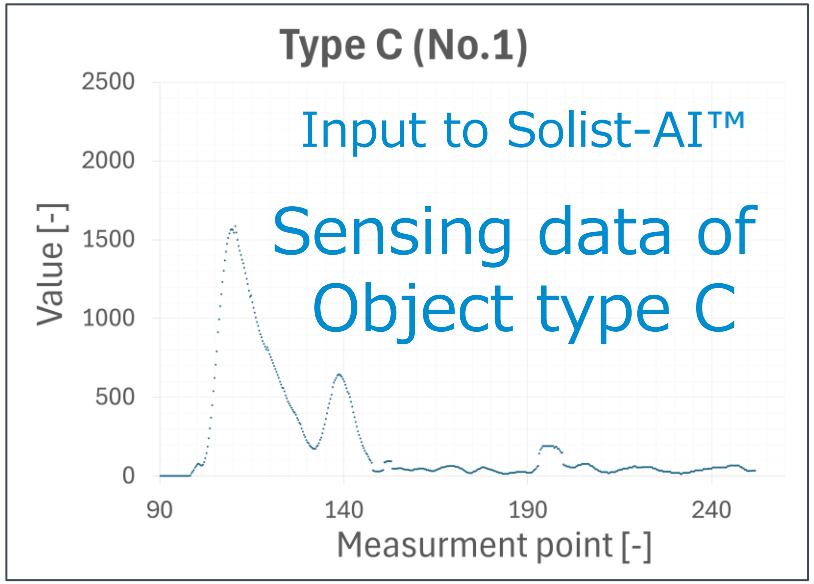
- Even if the data is not time series, **NG data** can be detected by training the AI model on **OK data**.

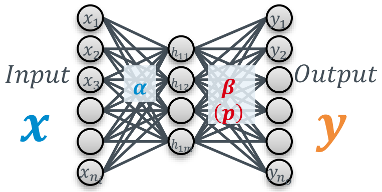


Contents

Supervised Learning by Solist-AI™

- Let's consider an example of object identification using supervised learning.



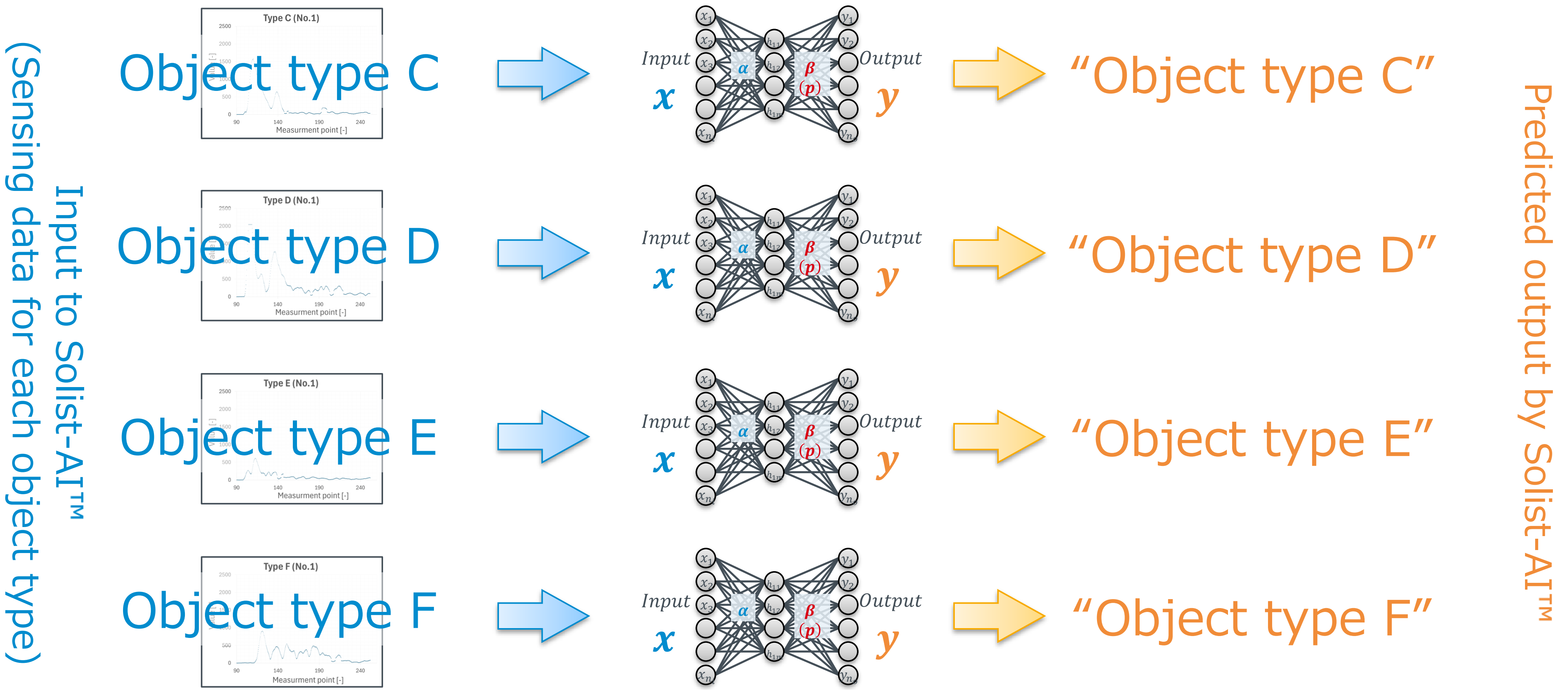


Predicted output by Solist-AI™

"Object type C"

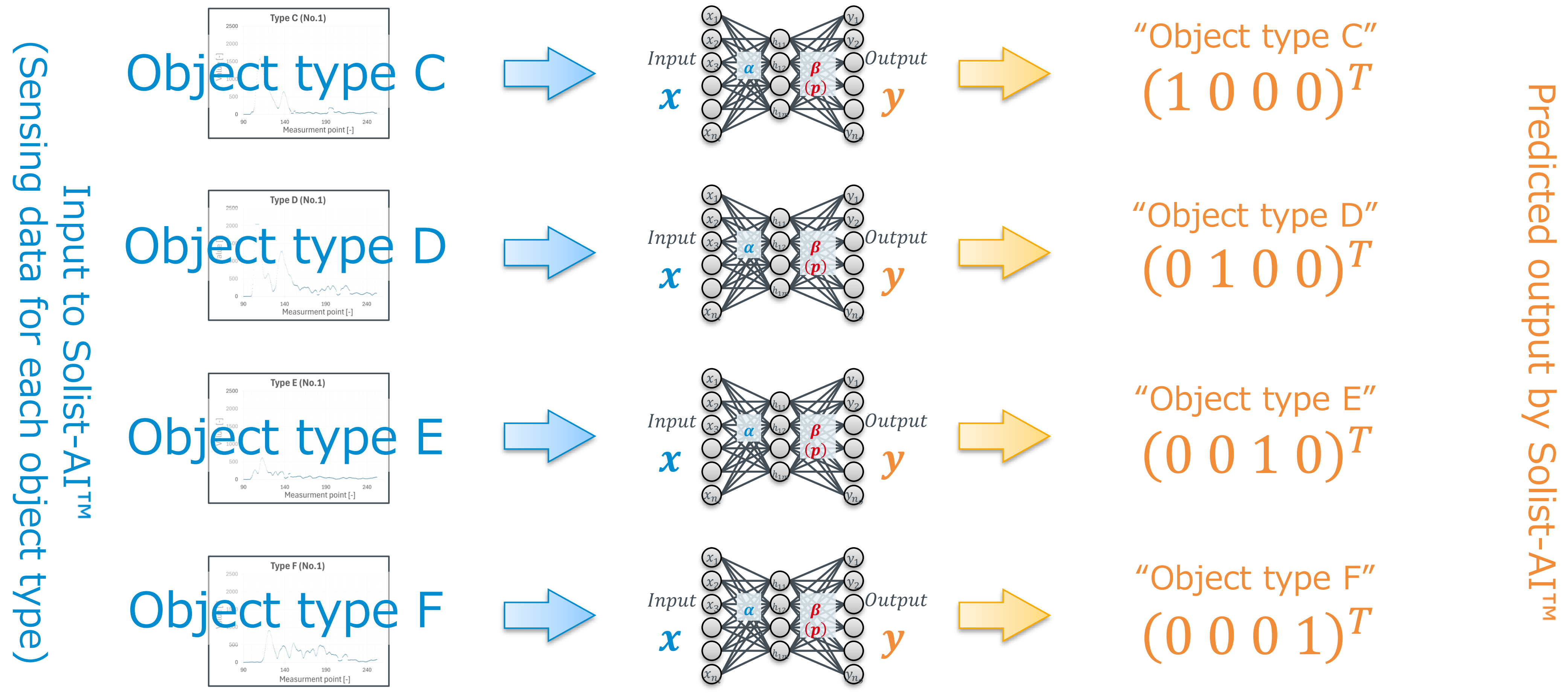
Supervised Learning by Solist-AI™

- Here we assume that there are four object types.



Supervised Learning by Solist-AI™

- It is more convenient to predicted output by Solist-AI™ as numbers, so we will output them as a matrix like $(c\ d\ e\ f)^T$, where c, d, e, and f are type identification indices, and ideally, c is 1 if it is object C and 0 if it is not.



Supervised Learning by Solist-AI™

- There is a lot of sensing data for each object type, and even sensing data for the same object type has variation. We are considering developing an AI model that can identify objects even with such variation.

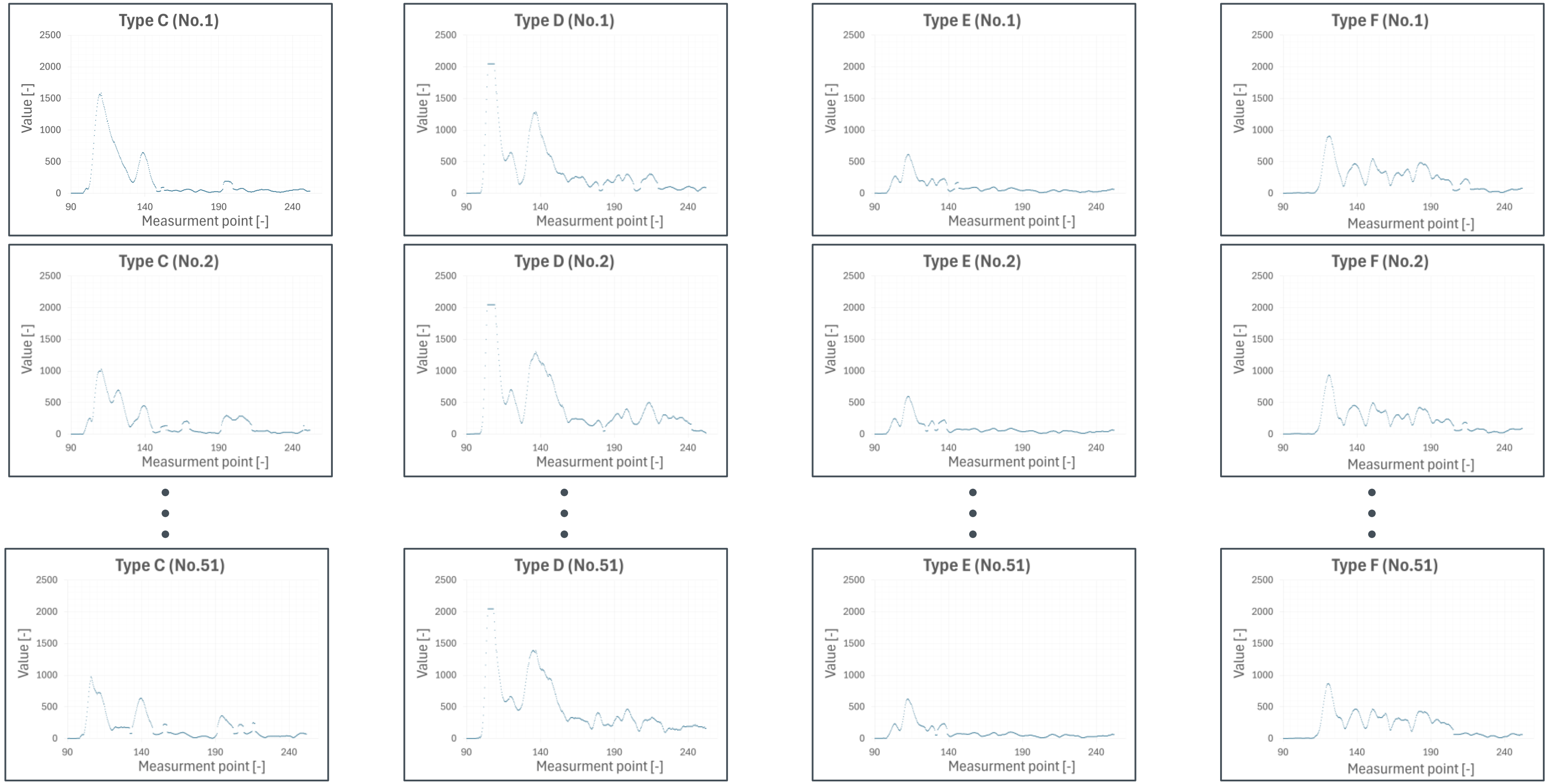
Object type C | Object type D | Object type E | Object type F

Measurement No. 1

No. 2

⋮

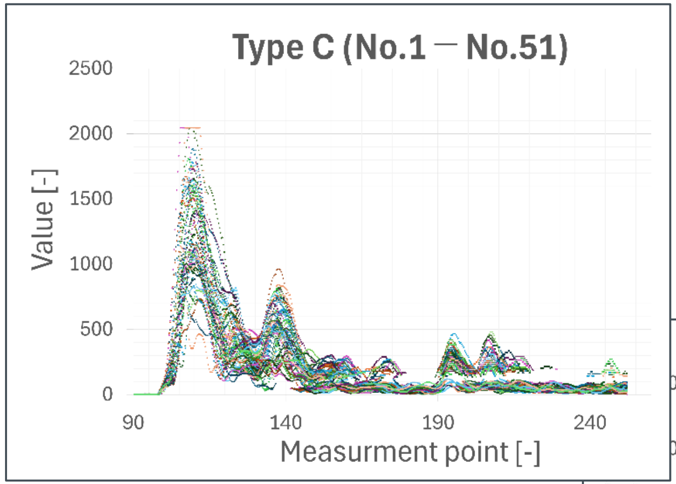
No. 51



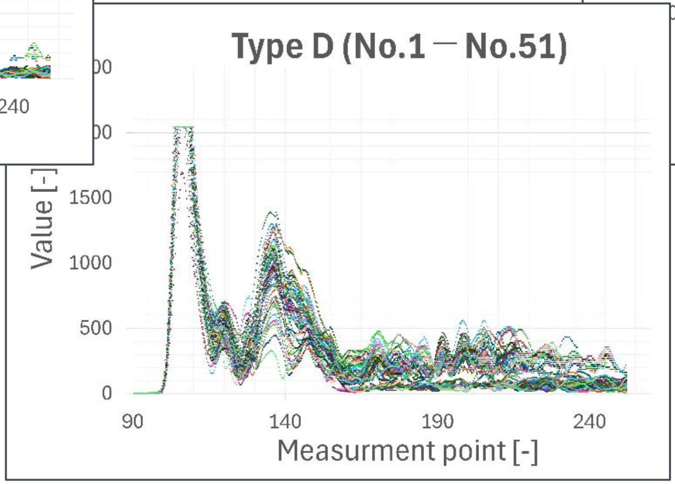
Supervised Learning by Solist-AI™

- Measurement data from No.1 to No.51 for one object type are plotted together on one graph.

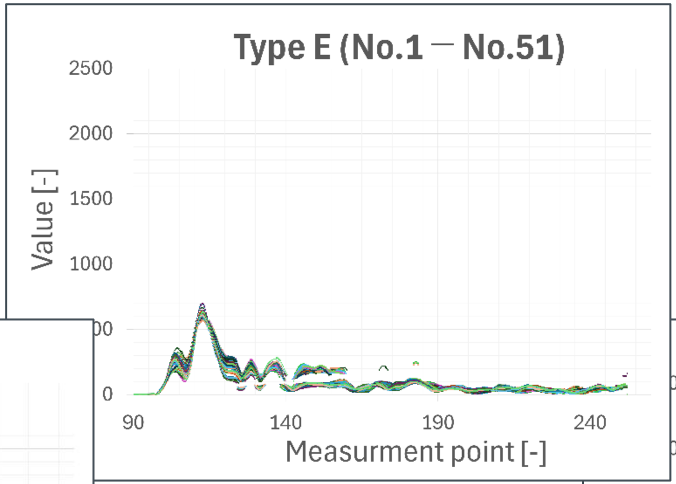
Object type C



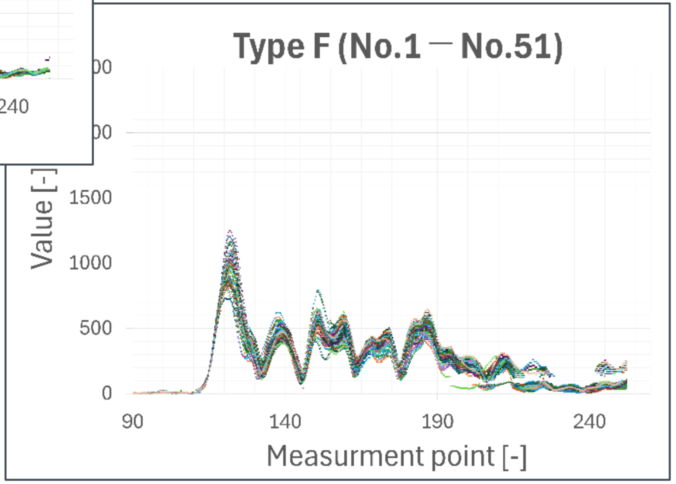
Object type D



Object type E

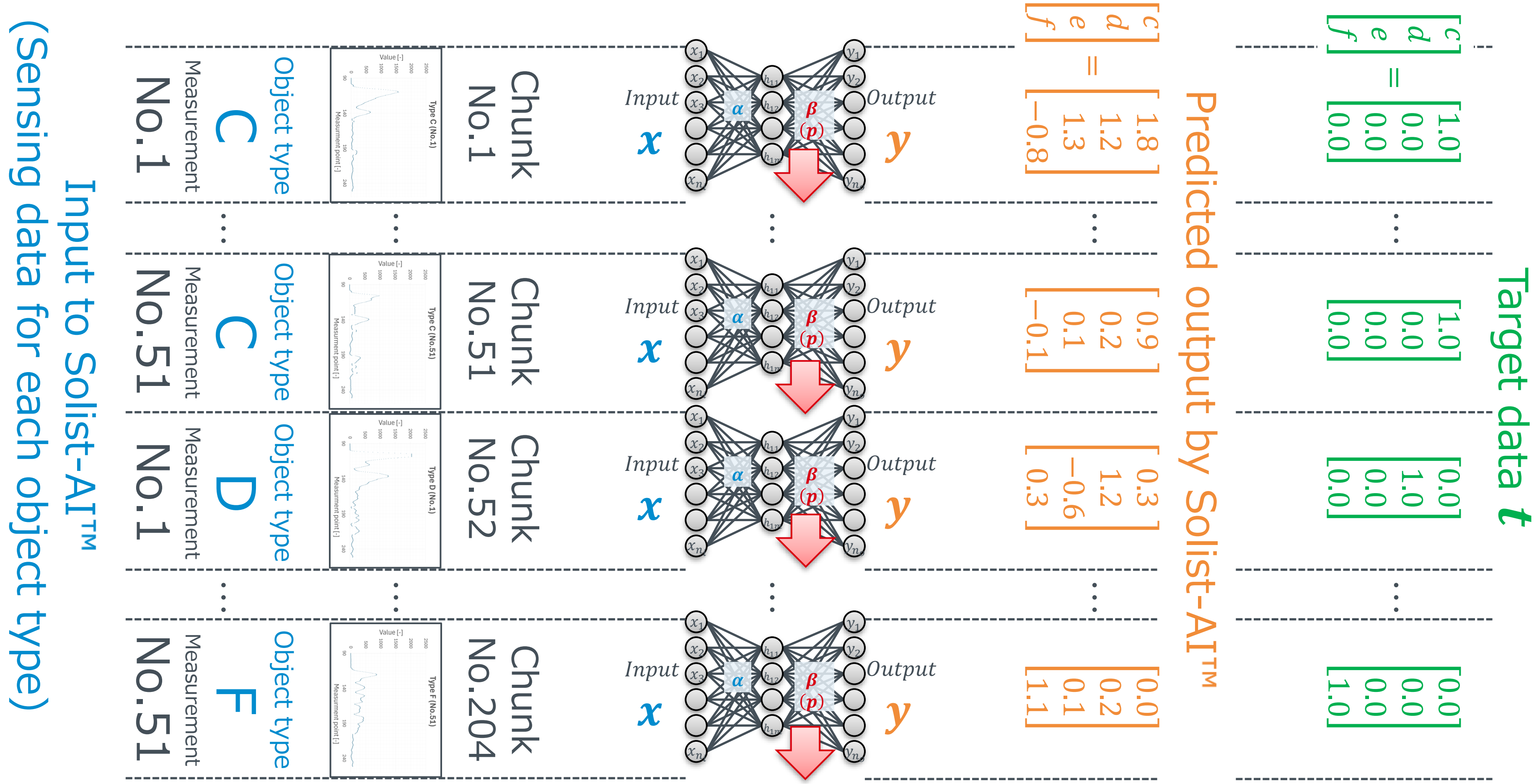


Object type F



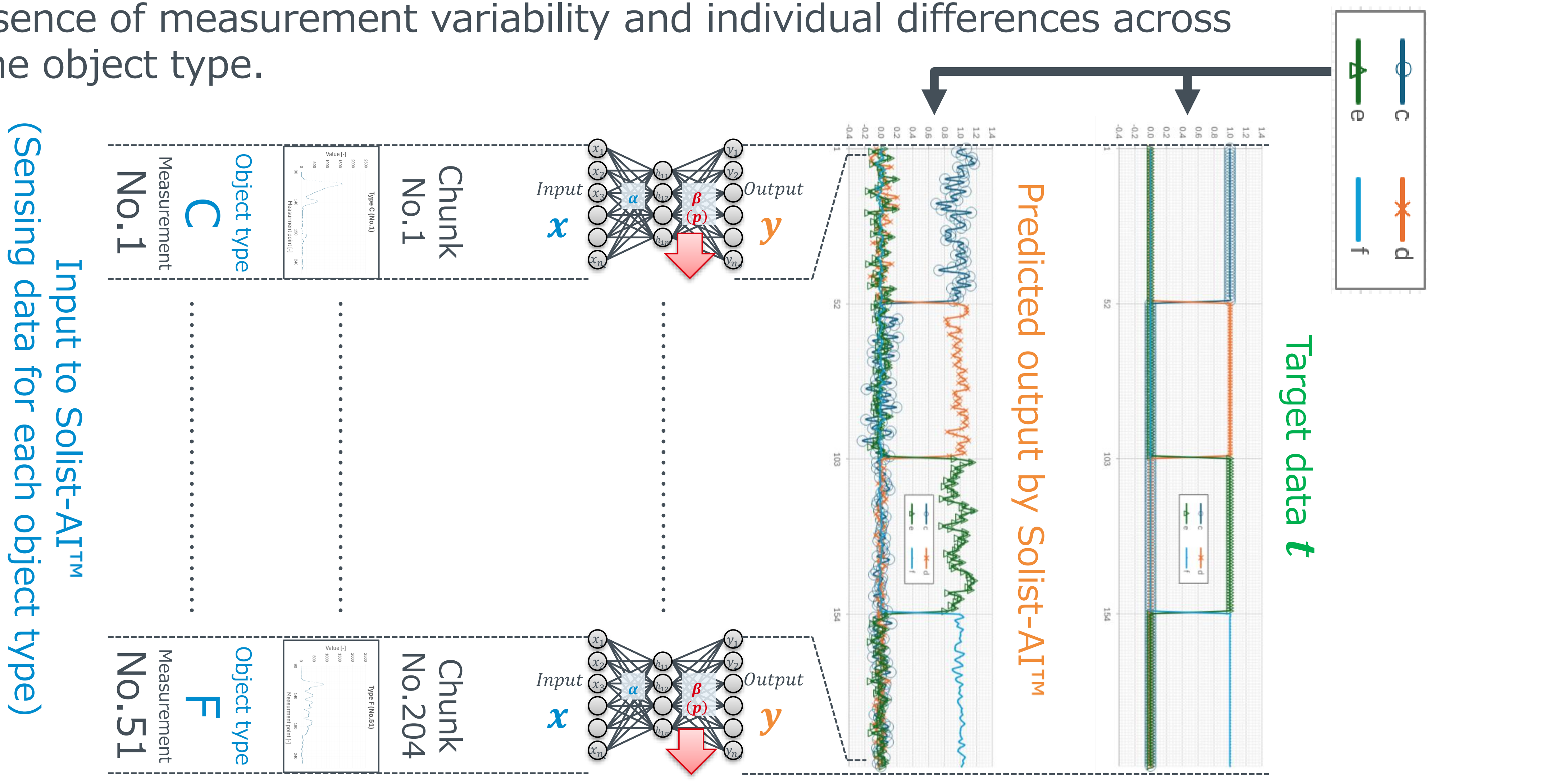
Supervised Learning by Solist-AI™

- Each measurement data corresponds to a chunk of data to be input to Solist-AI™. In unsupervised learning, the weight $\boldsymbol{\beta}$ is updated so that the output layer data $\boldsymbol{y}$ produces the target data $\boldsymbol{t}$.



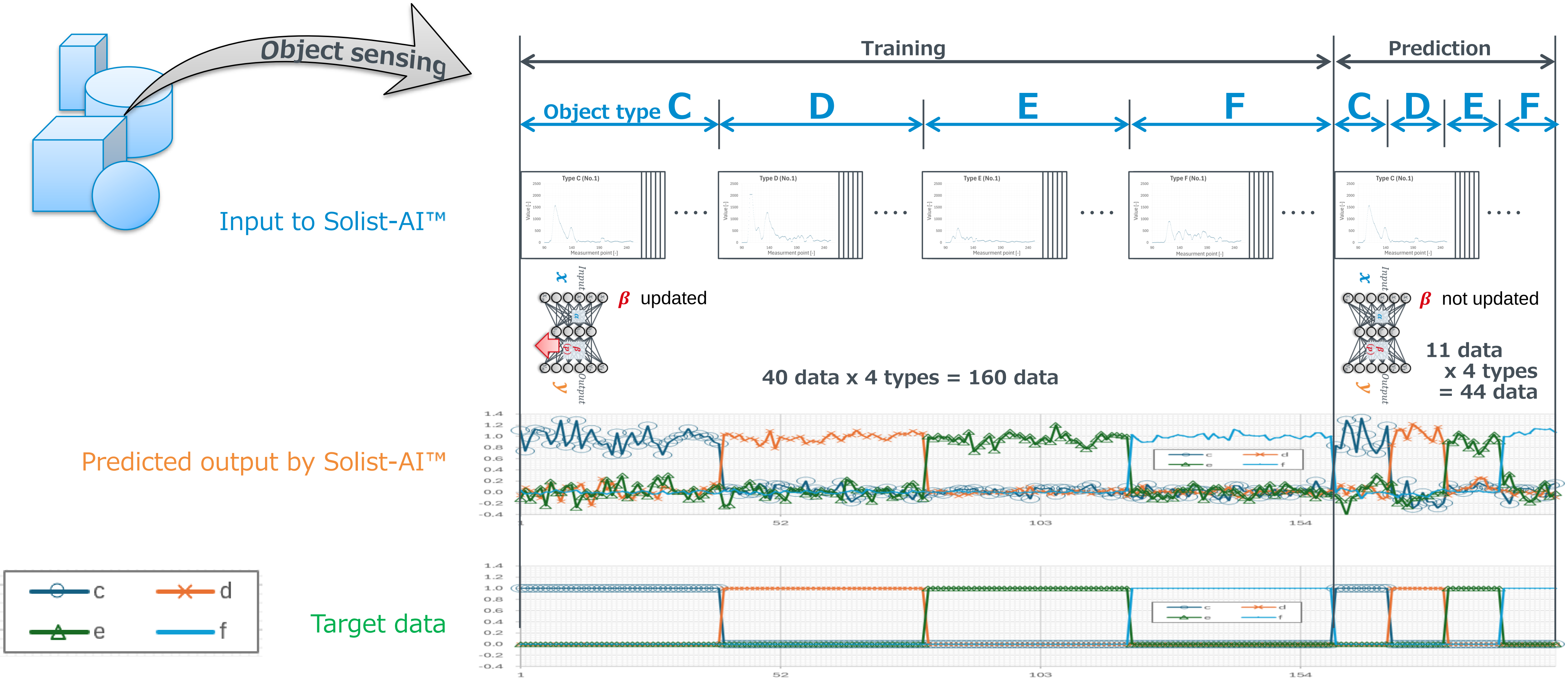
Supervised Learning by Solist-AI™

- By utilizing supervised learning, we are able to predict object type even in the presence of measurement variability and individual differences across the same object type.



Supervised Learning by Solist-AI™

- By utilizing supervised learning, it is possible to realize object identification, condition prediction, and deterioration prediction and so on.

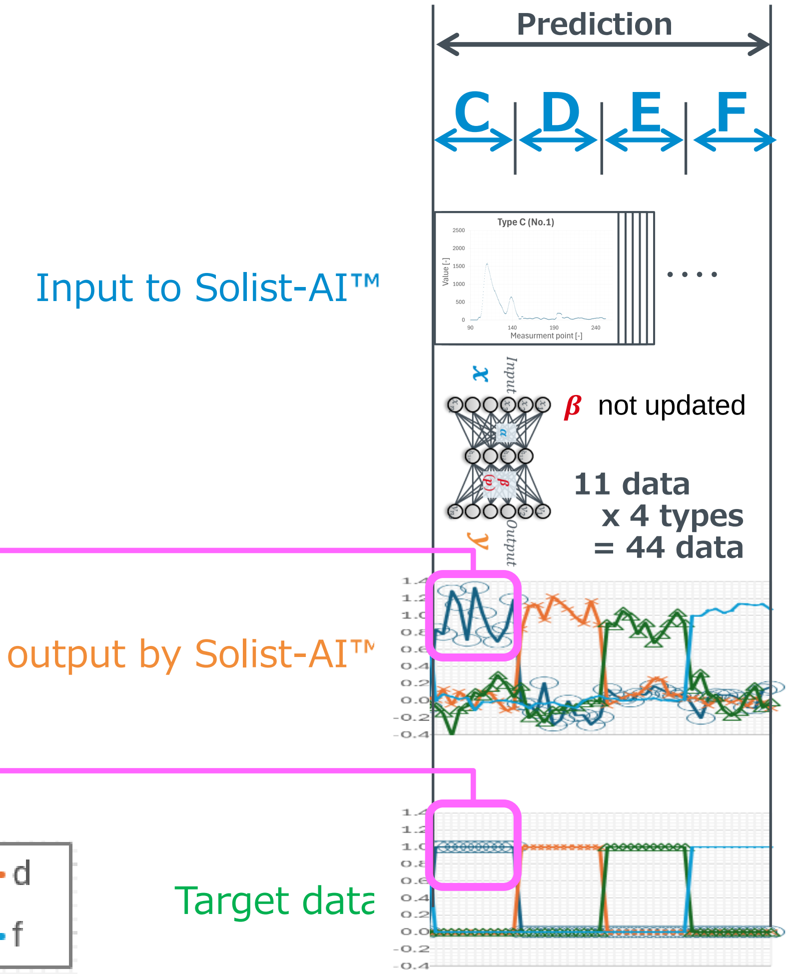


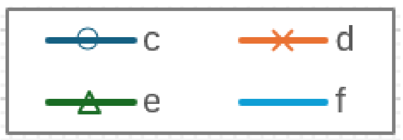
Supervised Learning by Solist-AI™

- By utilizing supervised learning, it is possible to realize object identification, condition prediction, and deterioration prediction and so on.

In the target data within the pink frame, only the object type C identification index is 1, so the actual object type is C.

In the predicted output by Solist-AI™ within the pink frame, only the object type C identification index is close to 1, indicating that the object is likely to be type C.





Contents

- Neural Network and Algorithm of Solist-AI™
- Unsupervised Learning by Solist-AI™
- Supervised Learning by Solist-AI™
- **Hardware Accelerator for AI Processing, AxlCORE-ODL**

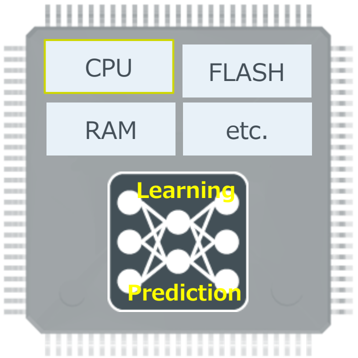
Structure of Solist-AI™

AxICORE-ODL (hardware accelerator for AI processing) is added to a peripheral of MCU.

AxICORE-ODL

- **The original architecture enables autonomous 3-phase neural network processing and high-speed data pre-processing such as FFT with extremely compact circuit.**
- **There is almost no load on the software due to AI processing.**
- **AI processing can be performed 1000 times faster and with ultra-low power consumption compared to software.**

Non-AI processing can be executed at the same time

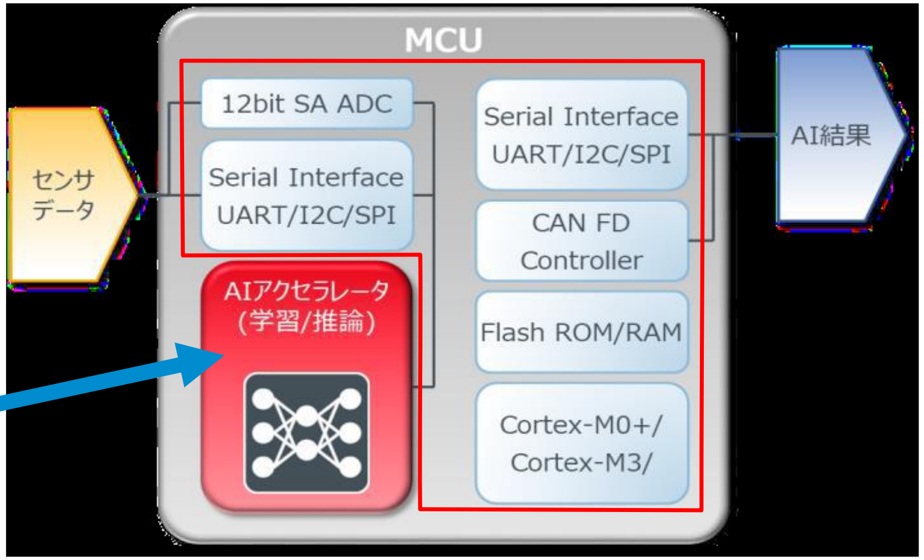


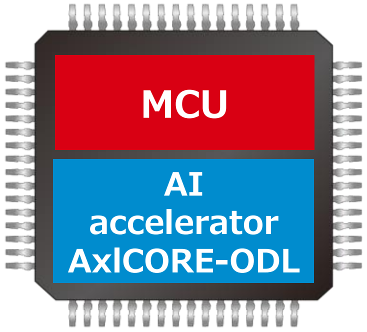
Start of AI prediction with AxICORE-ODL

Non-AI processing with MCU
- Calculation
- Serial I/F control
:

End AI prediction with AxICORE-ODL

Judgment processing of AI prediction results





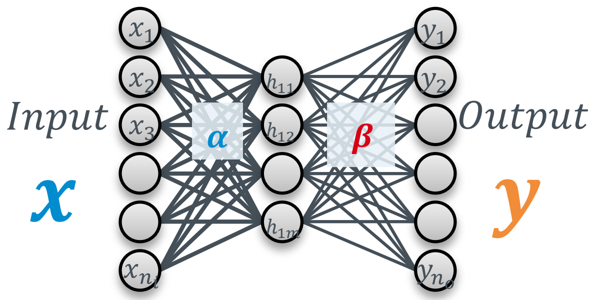
General-purpose MCU
+
AxICORE-ODL

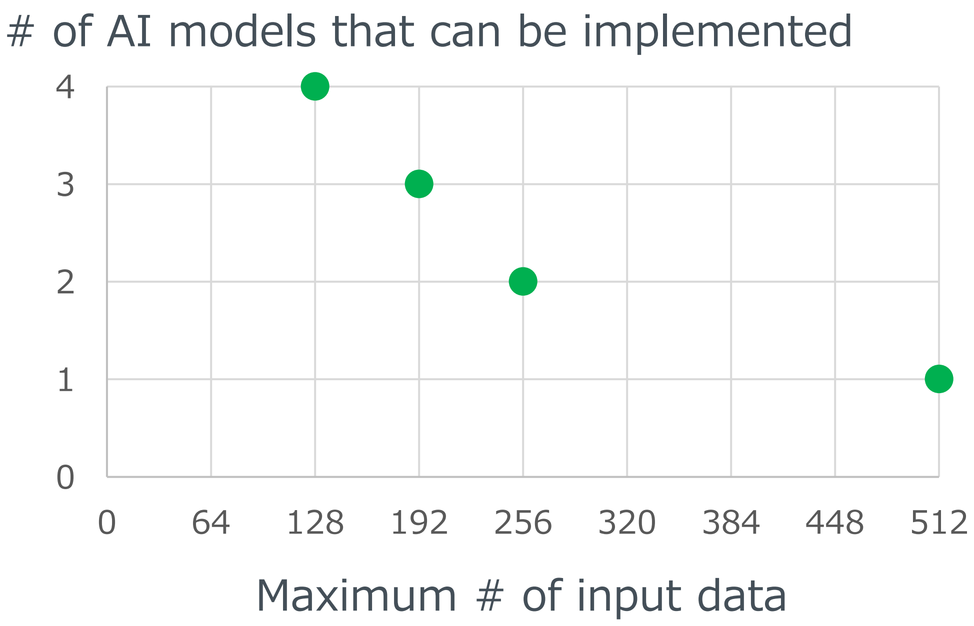
Solist-AI™ MCU

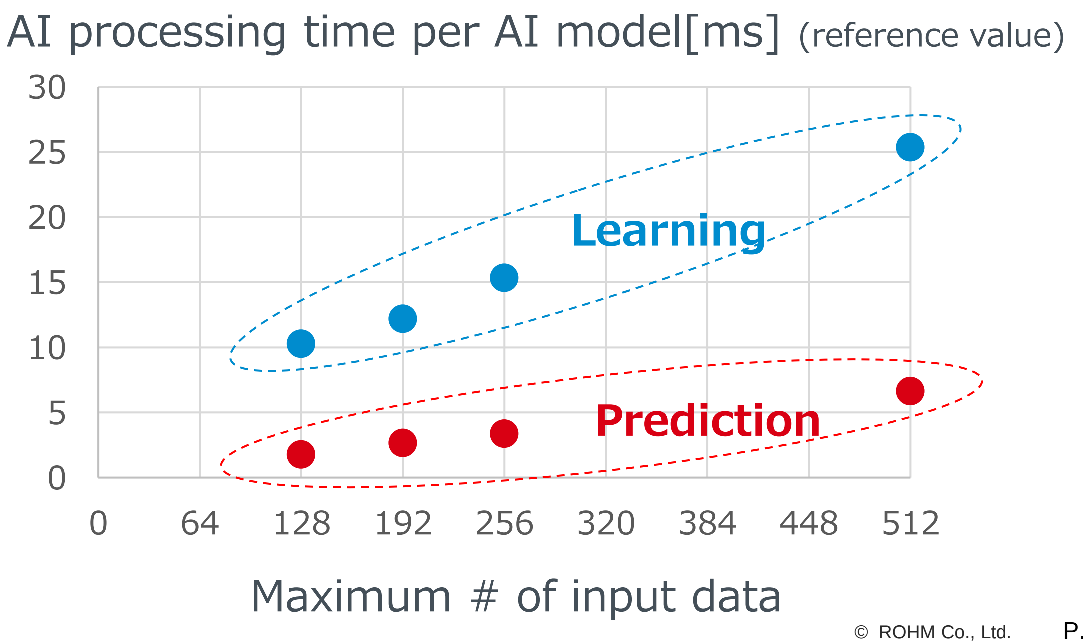
- **Solist-AI™ MCU** with AxICORE-ODL enables completely independent AI processing.
- If user knows how to use **AxICORE-ODL**, it is equivalent to developing ordinary MCU software.

of AI Models and its Processing Time in ML63Q2500 Group

- **Multiple AI models can be implemented on AI microcontrollers.**
  - Each AI model handles different data.
  - # of AI models that can be used at the same time is determined by the maximum # of input data per AI model
  - In the software, it is necessary to use AI library that matches # of AI models implemented in the AI microcontroller.
- **Use of AxICORE-ODL makes it possible to execute AI processing at high speed without a load on MCU**







Notes

- The information contained in this document is intended to introduce ROHM Group (hereafter referred to as ROHM) products. When using ROHM products, please verify the latest specifications or datasheets before use.
- ROHM does not warrant that the information contained herein is error-free. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties resulting from errors contained in this document.
- The information and data described in this document, including typical application circuits, are examples only and are not intended to guarantee to be free from infringement of third parties intellectual property or other rights. ROHM does not grant any license, express or implied, to implement, use, or exploit any intellectual property or other rights owned or controlled by ROHM or any third parties with respect to the information and data contained herein.
- When exporting ROHM products or technologies described in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, such as the Foreign Exchange and Foreign Trade Act and the US Export Administration Regulations, and follow the necessary procedures in accordance with these provisions.
- No part of this document may be reprinted or reproduced in any form by any means without the prior written consent of ROHM.
- The information contained in this document is current as of April 2025 and is subject to change without notice.

ROHM Co.,Ltd.
21 Saiin Mizosaki-cho, Ukyo-ku,
Kyoto 615-8585 Japan

www.rohm.com

Catalog No.：68AN011E Rev.001
Doc. No. : CTD_SOL_68_002