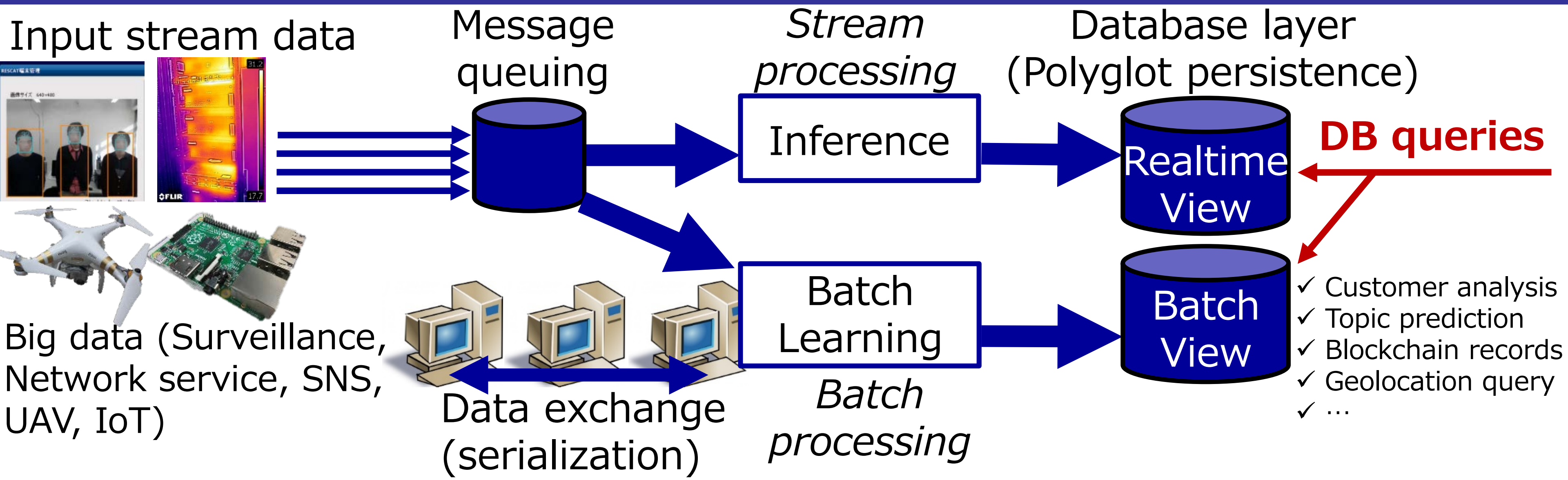
Accelerating Anomaly Detection Algorithms on FPGA-Based High-Speed NICs

Hiroki Matsutani

Dept. of ICS, Keio University

http://www.arc.ics.keio.ac.jp/~matutani

Accelerator design for big data



Input stream data

Message queuing

Stream processing

Inference

Database layer (Polyglot persistence)

Realtime View

DB queries

Batch Learning

Batch View

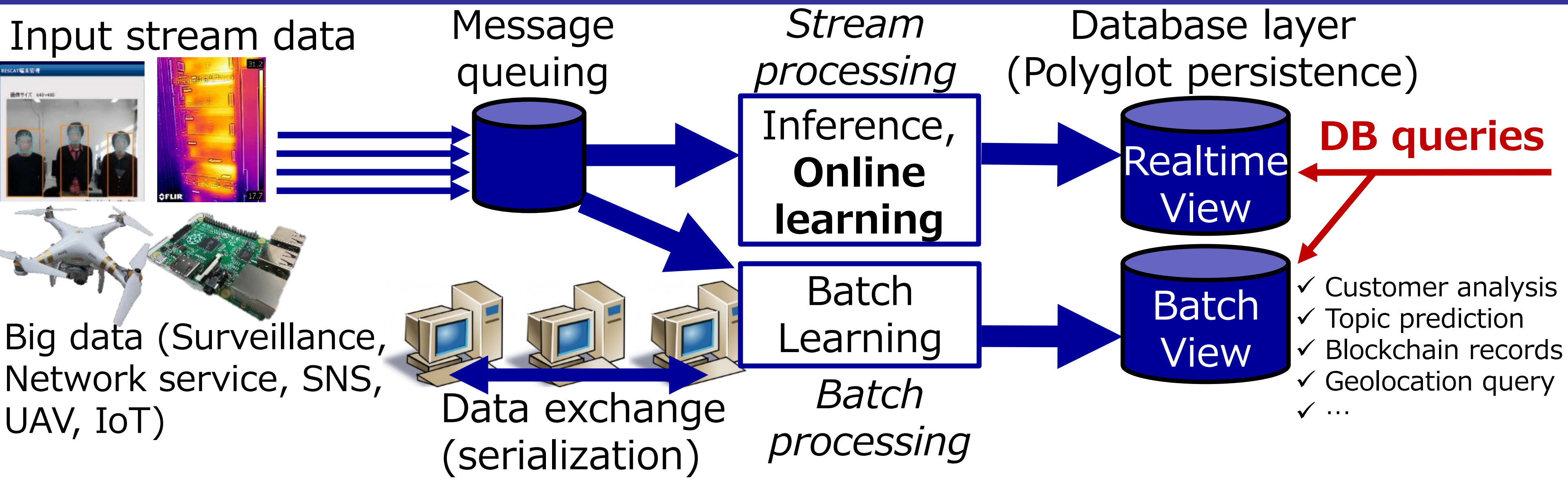
- ✓ Customer analysis
- ✓ Topic prediction
- ✓ Blockchain records
- ✓ Geolocation query
- ✓ …

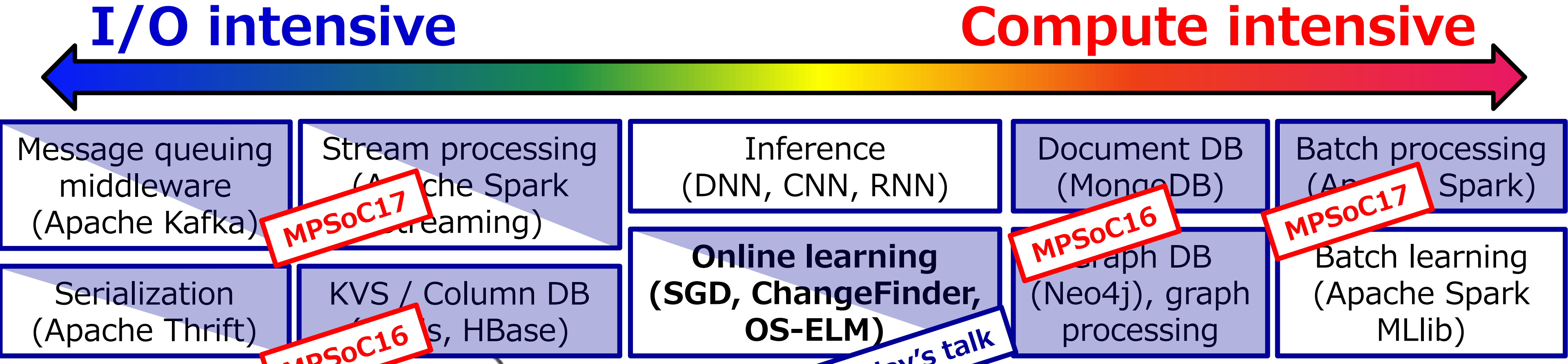
Big data (Surveillance, Network service, SNS, UAV, IoT)

Data exchange (serialization)

Batch processing

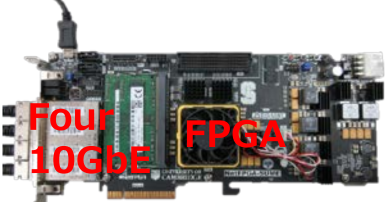
Today’s talk: Online learning



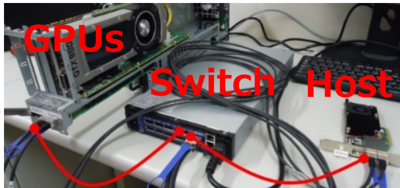


I/O intensive
Compute intensive
Message queuing middleware (Apache Kafka)
Stream processing (Apache Spark Streaming)
Inference (DNN, CNN, RNN)
Document DB (MongoDB)
Batch processing (Apache Spark)
Serialization (Apache Thrift)
KVS / Column DB (Cassandra, HBase)
Online learning (SGD, ChangeFinder, OS-ELM)
Graph DB (Neo4j), graph processing
Batch learning (Apache Spark MLlib)
MPSoC17
MPSoC16
MPSoC16
MPSoC17
Today’s talk

Tight integration of I/O and compute →FPGA



Massive parallelism → Networked GPU cluster



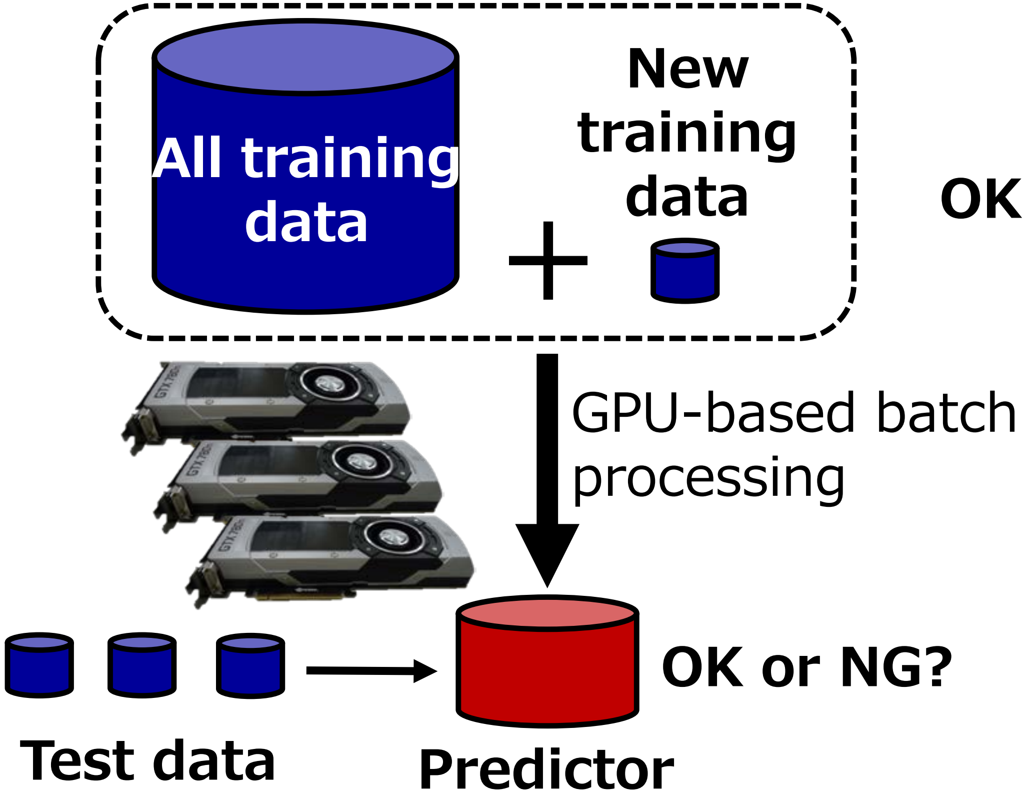
Offline vs. Online learning

- ✓ Examples: DNN, CNN, …
- ✓ Learning cost is high
- ✓ Predictor updated infrequently

- ✓ Learning cost is low
- ✓ Predictor updated frequently
- ✓ Not very versatile

Offline learning

Online learning



All training data

+

New training data

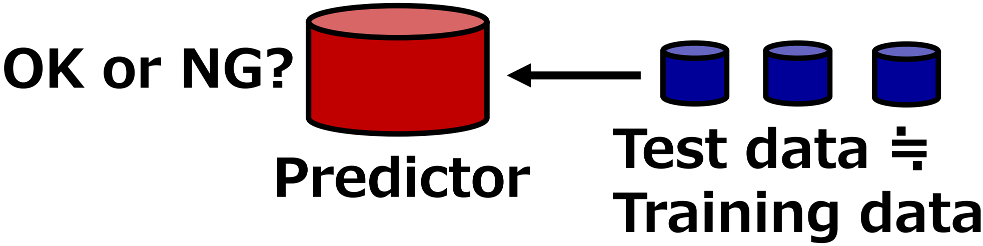
GPU-based batch processing

Test data

Predictor

OK or NG?

Sequential learning + Inference



OK or NG?

Predictor

Test data ≒ Training data

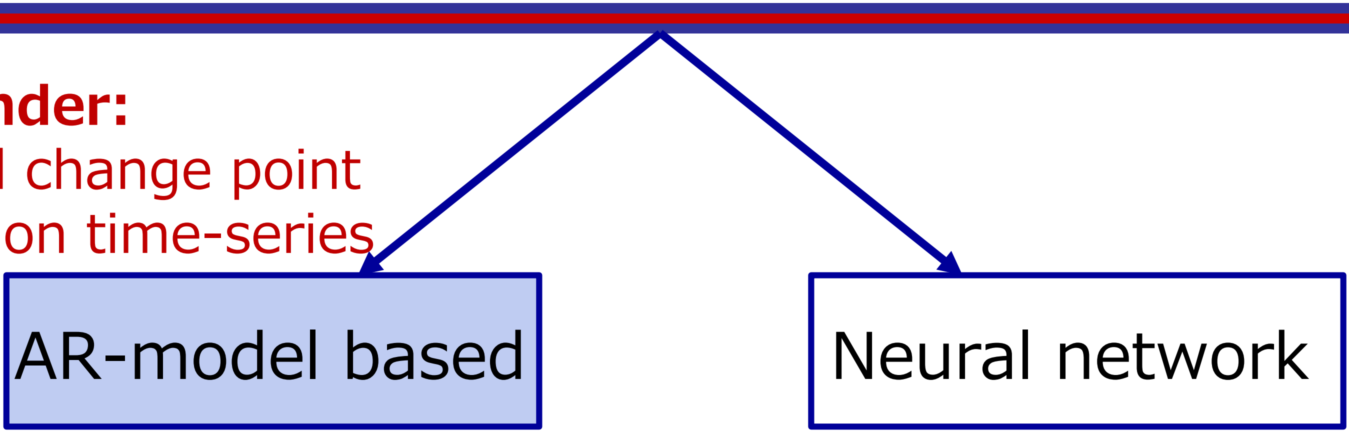
FPGA-based stream processing

10GbE x4

FPGA NIC/Switch

Online learning approaches

ChangeFinder:
Outlier and change point detections on time-series data

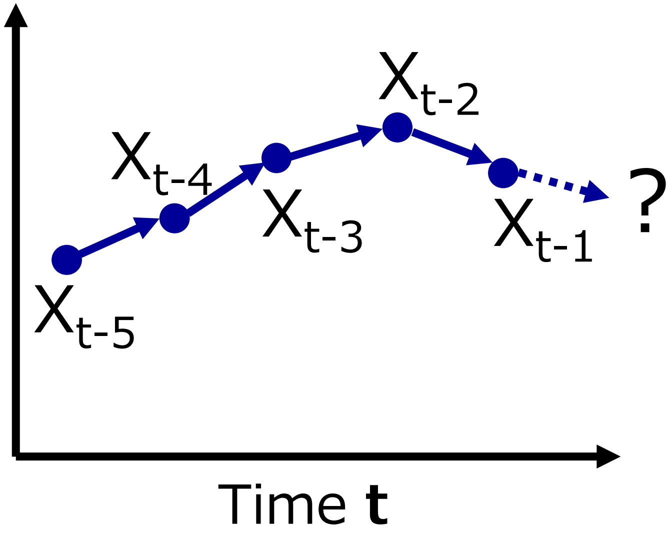


AR-model based

Neural network

Next value X_t is predicted based on recent p values

Online sequential learning for SLFN (input, hidden, and output layers)



$$X_t = \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots$$

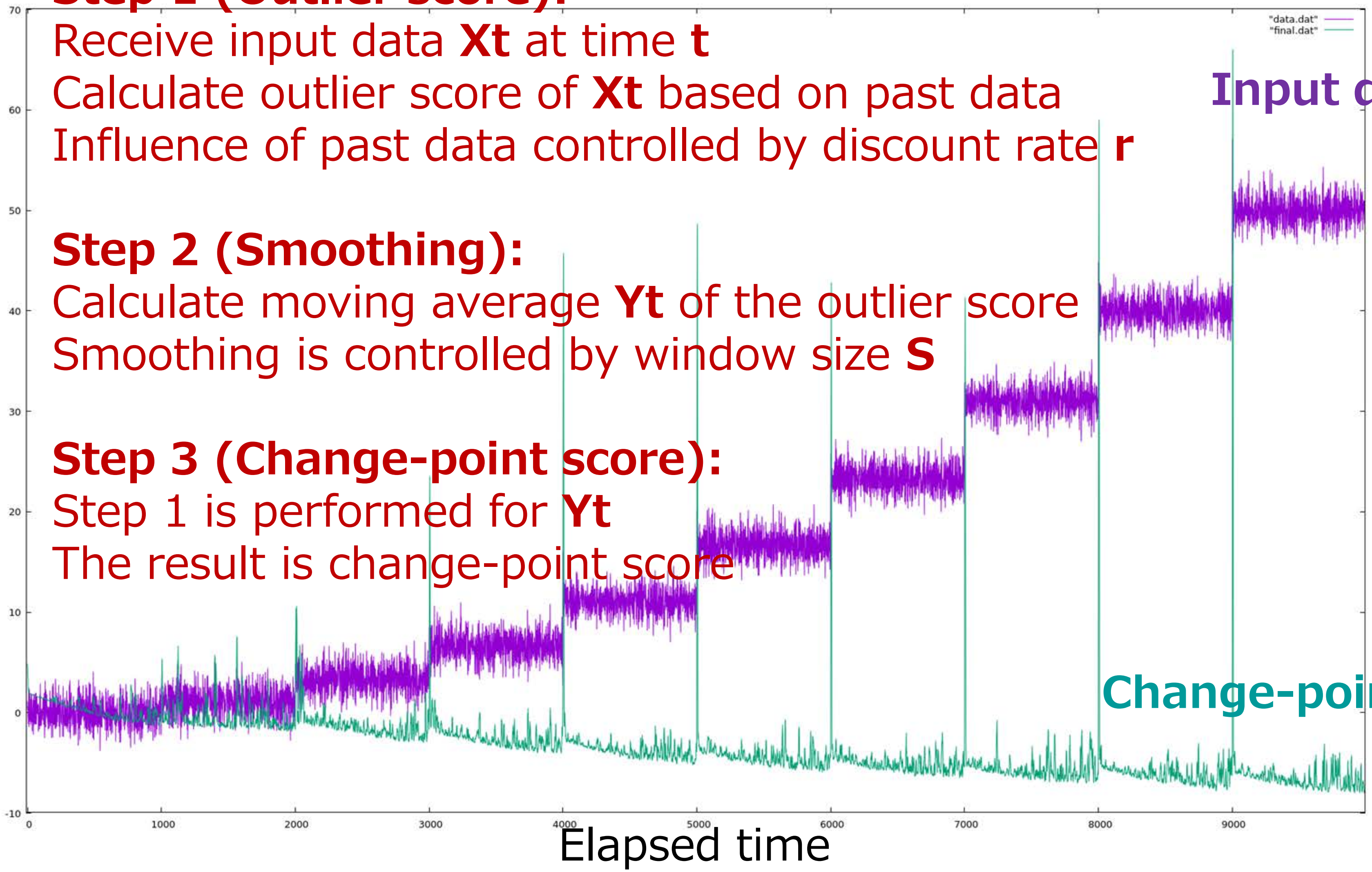
ChangeFinder on 10GbE FPGA

- ChangeFinder algorithm [J.Takeuchi, IEEE TKDE'06]

Step 1 (Outlier score):
Receive input data **Xt** at time **t**
Calculate outlier score of **Xt** based on past data
Influence of past data controlled by discount rate **r**

Step 2 (Smoothing):
Calculate moving average **Yt** of the outlier score
Smoothing is controlled by window size **S**

Step 3 (Change-point score):
Step 1 is performed for **Yt**
The result is change-point score

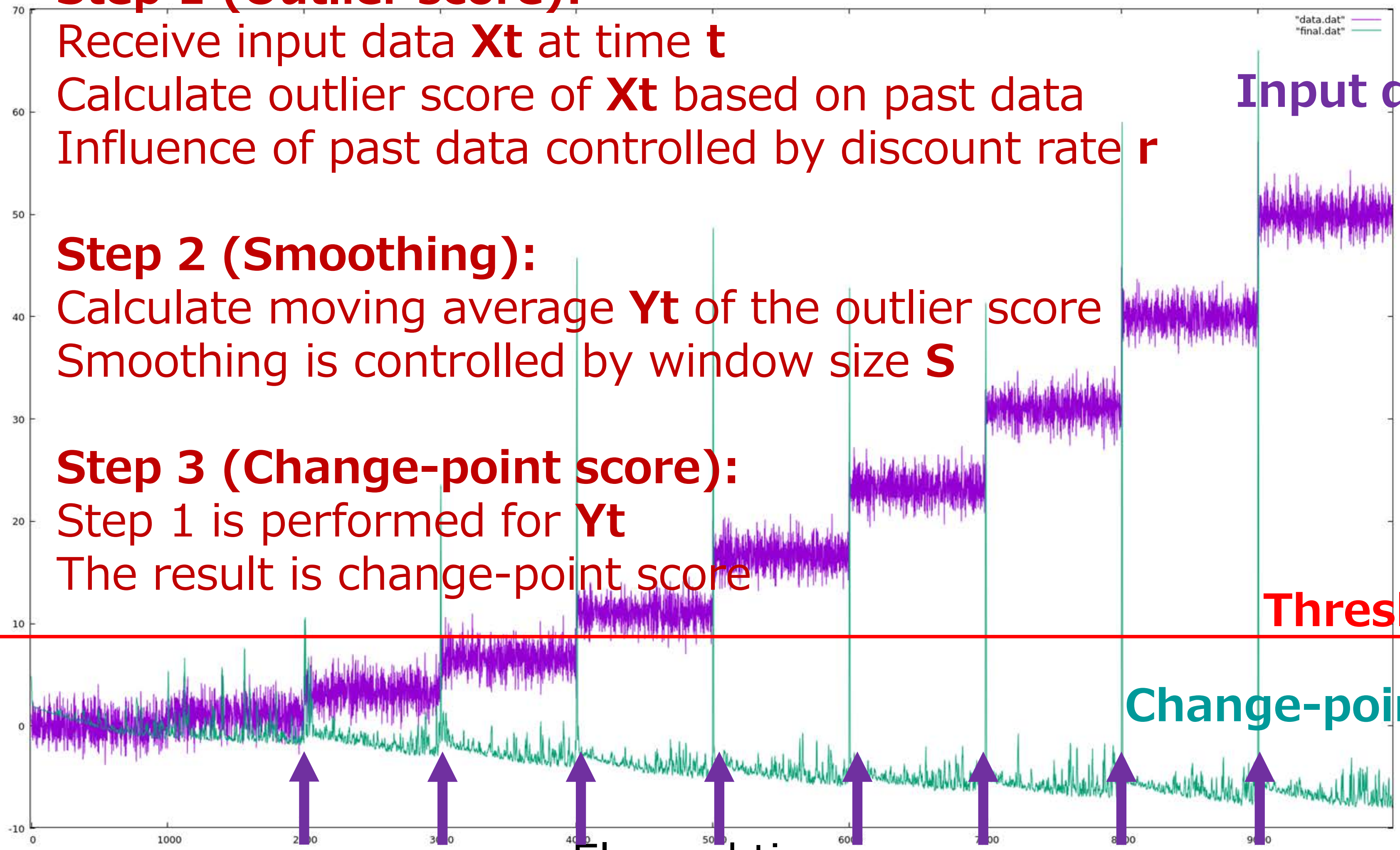


Input data Xt

Change-point score

ChangeFinder on 10GbE FPGA

- ChangeFinder algorithm [J.Takeuchi, IEEE TKDE'06]



Step 1 (Outlier score):
Receive input data **Xt** at time **t**
Calculate outlier score of **Xt** based on past data
Influence of past data controlled by discount rate **r**

Step 2 (Smoothing):
Calculate moving average **Yt** of the outlier score
Smoothing is controlled by window size **S**

Step 3 (Change-point score):
Step 1 is performed for **Yt**
The result is change-point score

Input data Xt

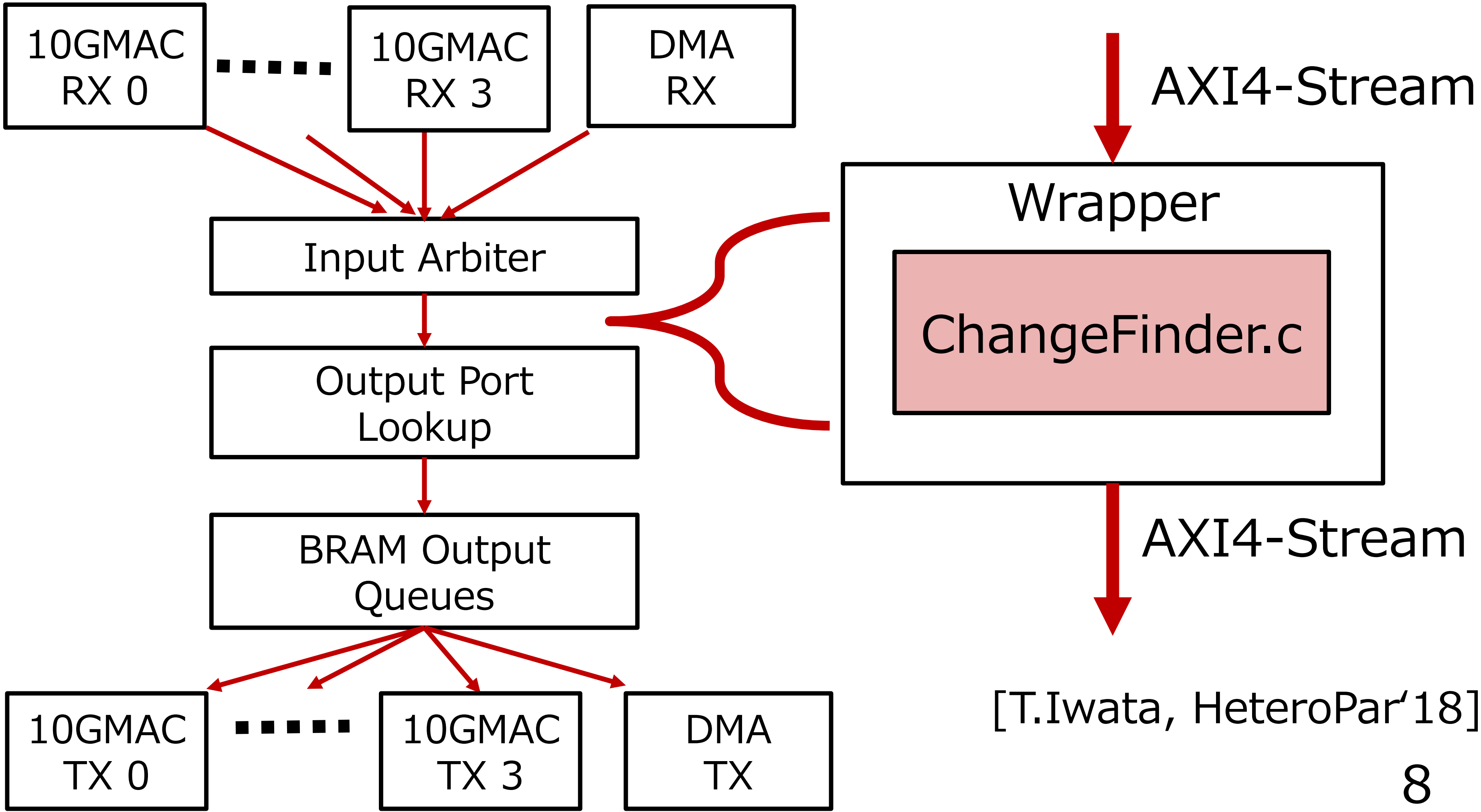
Threshold

Change-point score

Elapsed time

ChangeFinder on 10GbE FPGA

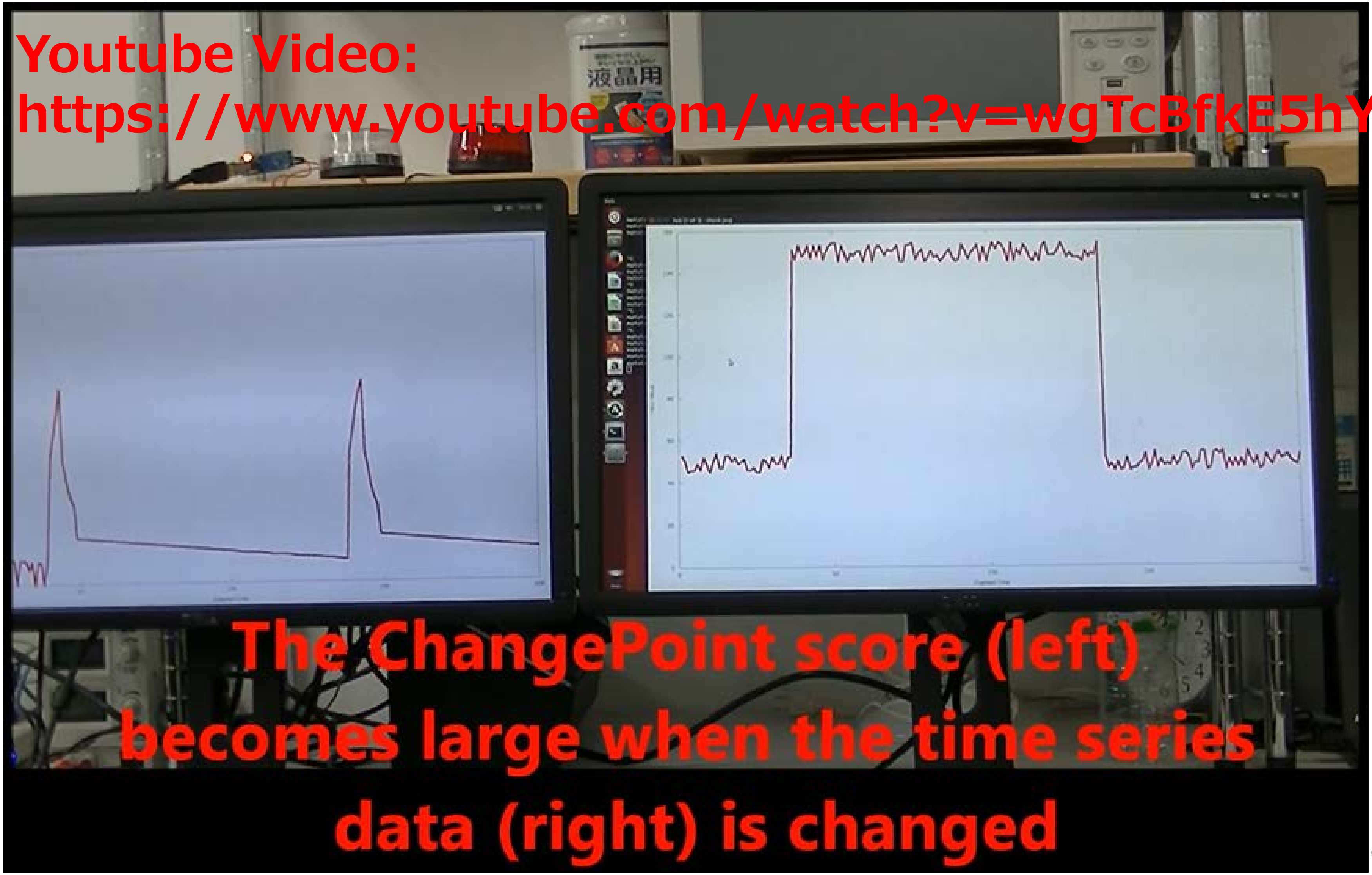
- 10GbE NIC datapath by Verilog HDL
- Application logic in wrapper in HLS



[T.Iwata, HeteroPar'18]

ChangeFinder on 10GbE FPGA

- Throughput: 83.4% of 10GbE line rate



Youtube Video:
https://www.youtube.com/watch?v=wgTcBfkE5hY

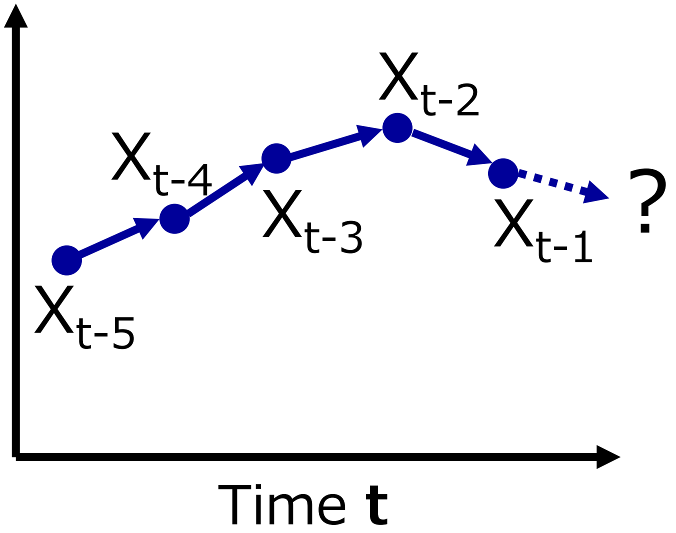
The ChangePoint score (left) becomes large when the time series data (right) is changed

Online learning approaches

ChangeFinder:
Outlier and change point detections on time-series data

AR-model based

Next value X_t is predicted based on recent p values



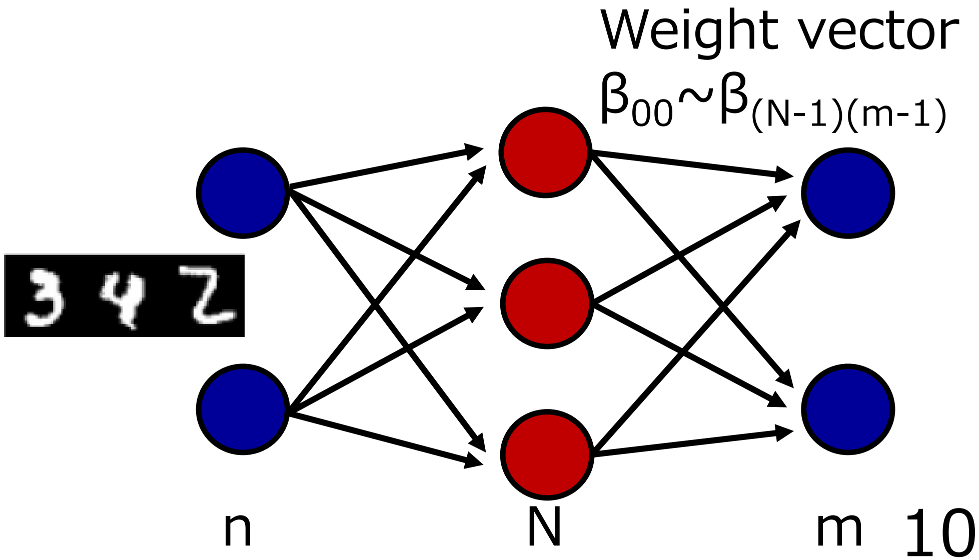
$$X_t = \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots$$

OS-ELM :
Single hidden layer neural network (SLFN)

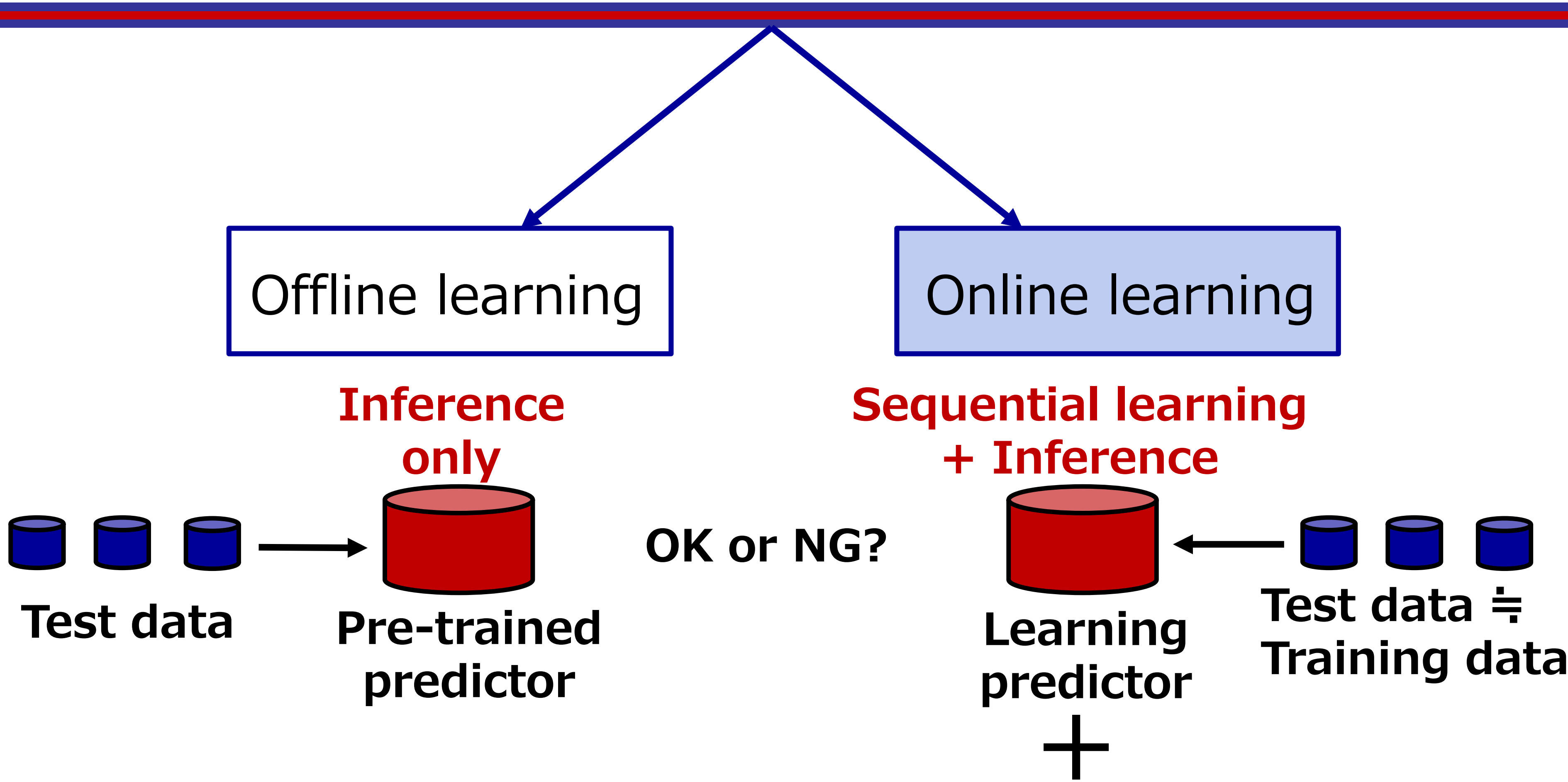
Neural network

Online sequential learning for SLFN (input, hidden, and output layers)

[N. Liang, TNN 2006]



Online learning + unsupervised



Offline learning

Online learning

Inference only

Sequential learning + Inference

Test data

Pre-trained predictor

OK or NG?

Learning predictor

Test data ≒ Training data

＋

Unsupervised anomaly detection (No training data needed)

[M.Tsukada, HeteroPar'18]

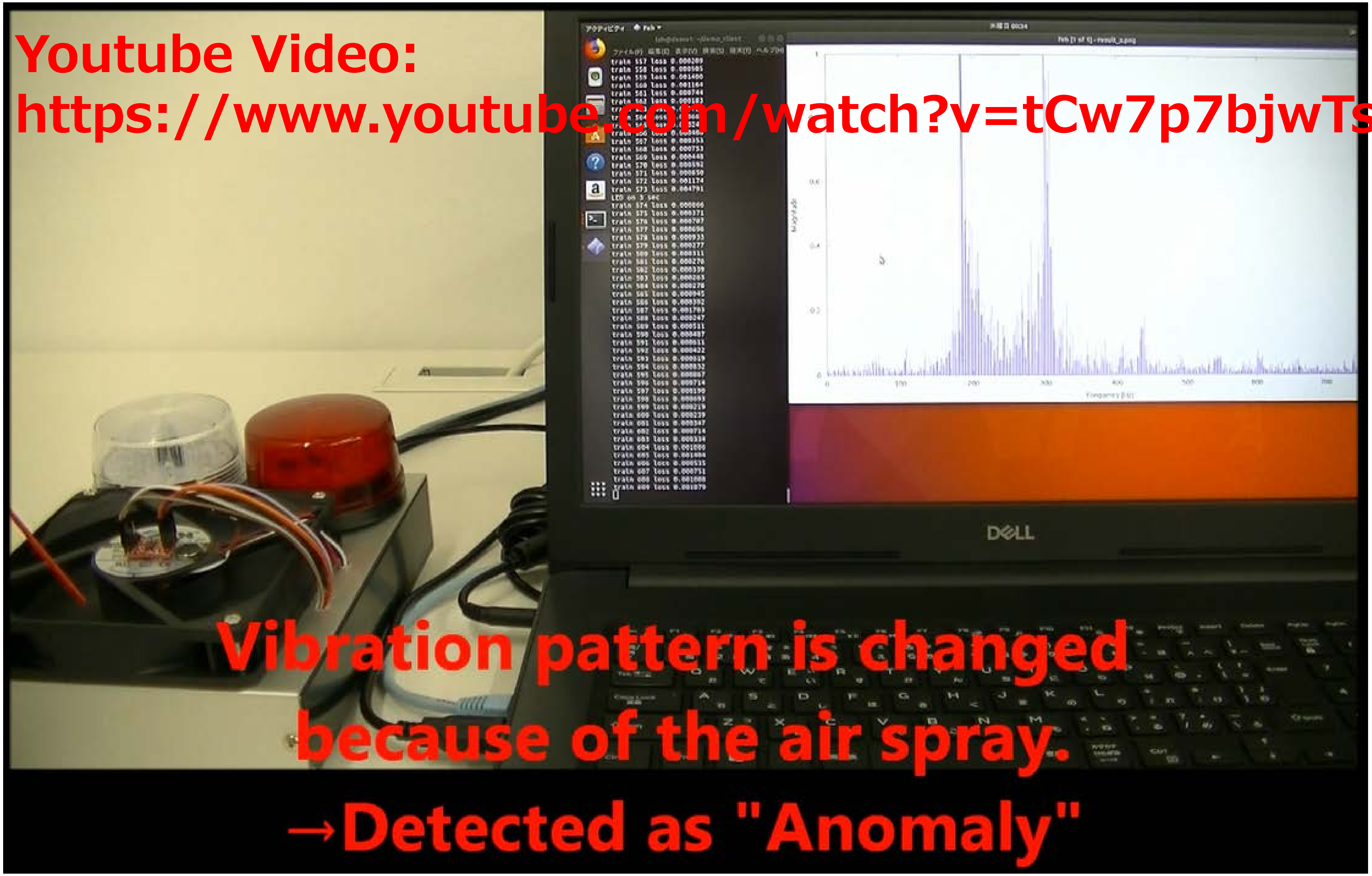
*Collaboration with Prof. M.Kondo (UTokyo)

Normal values (incl. noise) are learned after the deployment
→ Anomaly detection adapted to a given environment

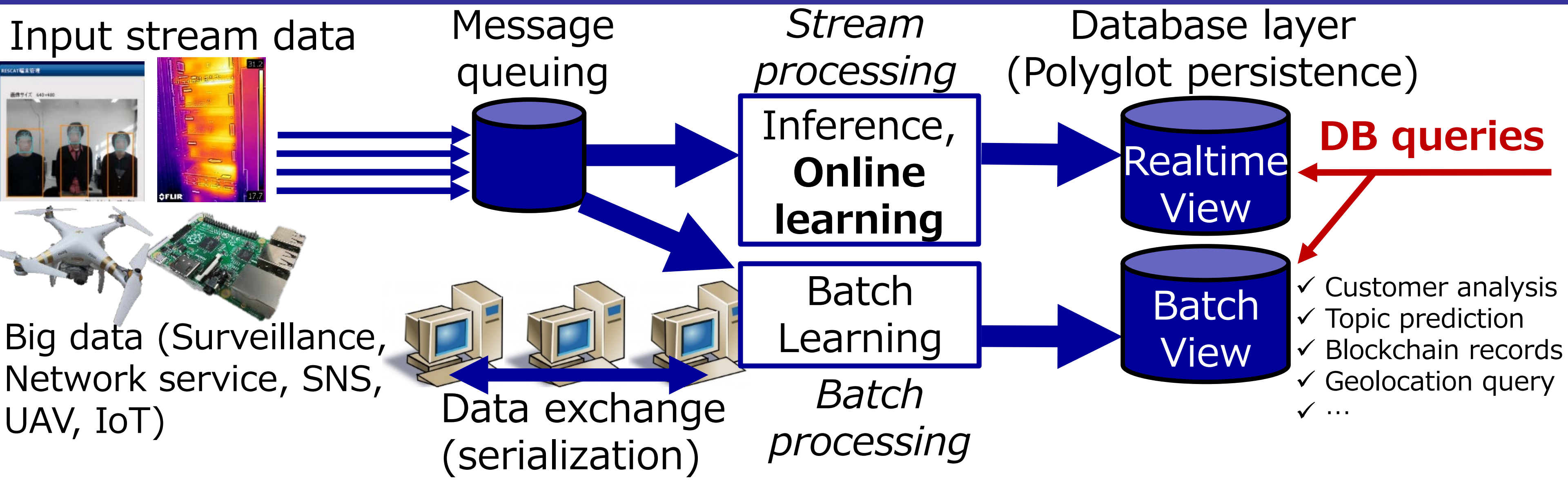
Online learning + unsupervised

- Learn vibration pattern of fan **+ noise**

Youtube Video:
https://www.youtube.com/watch?v=tCw7p7bjwTs



Summary: Online learning FPGA



Input stream data
Message queuing
Stream processing
Database layer (Polyglot persistence)
Inference, **Online learning**
Realtime View
DB queries
Batch Learning
Batch View
Big data (Surveillance, Network service, SNS, UAV, IoT)
Data exchange (serialization)
Batch processing

- ✓ Customer analysis
- ✓ Topic prediction
- ✓ Blockchain records
- ✓ Geolocation query
- ✓ …

I/O intensive ⟷ **Compute intensive**

Message queuing middleware (Apache Kafka)	Stream processing (Apache Spark Streaming)	Inference (DNN, CNN, RNN)	Document DB (MongoDB)	Batch processing (Apache Spark)
Serialization (Apache Thrift)	KVS / Column DB (s, HBase)	**Online learning (SGD, ChangeFinder, OS-ELM)**	Graph DB (Neo4j), graph processing	Batch learning (Apache Spark MLlib)

MPSoC17

MPSoC16

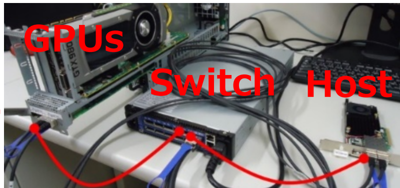
MPSoC16

MPSoC17

Today's talk

Tight integration of I/O and compute →FPGA

Massive parallelism → Networked GPU cluster



References (1/2)

- Outlier detection on 10GbE FPGA NIC
  - Ami Hayashi, et.al., "An FPGA-Based In-NIC Cache Approach for Lazy Learning Outlier Filtering", PDP 2017.
  - Ami Hayashi, et.al., "A Line Rate Outlier Filtering FPGA NIC using 10GbE Interface", ACM Comp Arch News (2015).
- Change-point detection on FPGA NIC
  - Takuma Iwata, et.al., "Accelerating Online Change-Point Detection Algorithm using 10GbE FPGA NIC", HeteroPar 2018.

References (2/2)

- Online sequential unsupervised anomaly detector on FPGA
  - Mineto Tsukada, et.al., "OS-ELM-FPGA: An FPGA-Based Online Sequential Unsupervised Anomaly Detector", HeteroPar 2018.

Thank you for listening

Acknowledgement:
This work is supported by JST CREST JPMJCR1785