\Orchestrating a brighter world NEC

No More Cumbersomeness: Automatic Predictive Modeling on Apache Spark

Masato Asahara and Ryohei Fujimaki

NEC Data Science Research Labs.

Jun/07/2017 @Spark Summit 2017

Who we are?

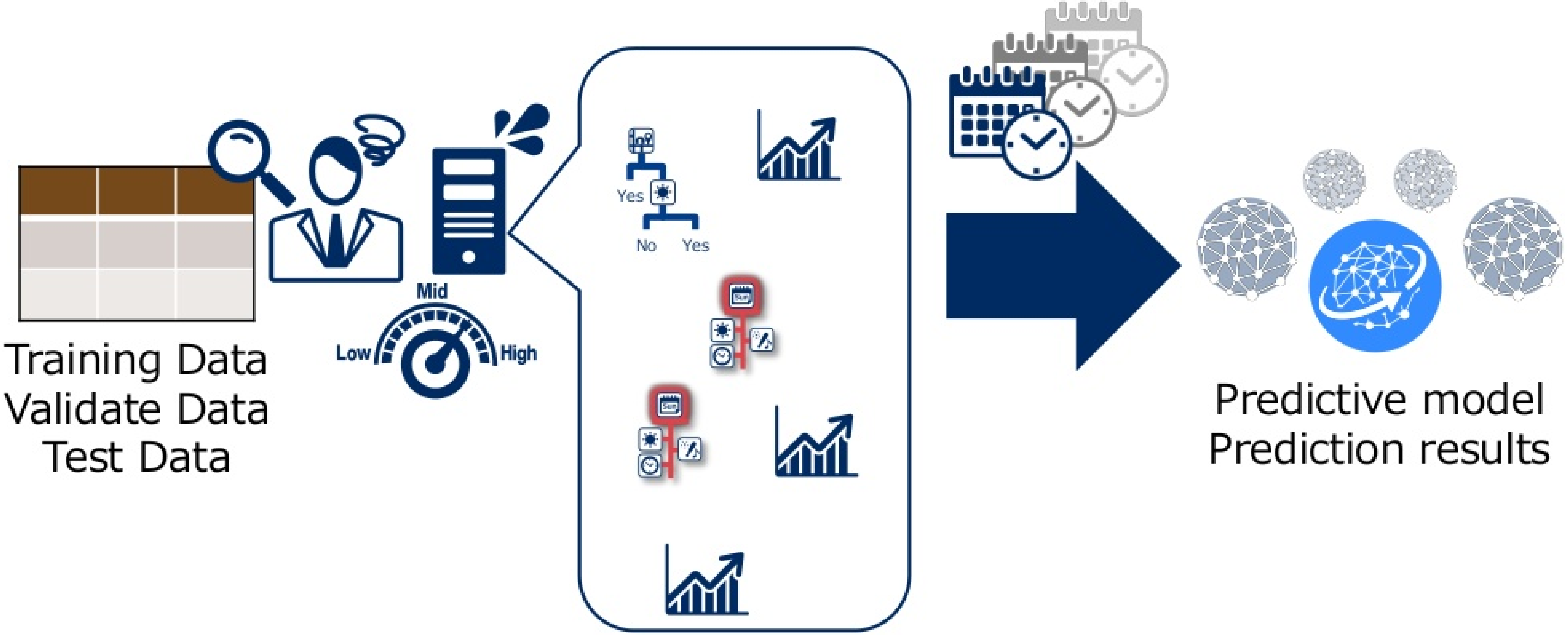
Masato Asahara (Ph.D.)
Researcher, NEC Data Science Research Laboratory

- Masato is currently leading developments of Spark-based machine learning and data analytics systems, which fully automate predictive modeling.
- Masato received his Ph.D. degree from Keio University, and has worked at NEC for 7 years as a researcher in the field of distributed computing systems and computing resource management technologies.

Ryohei Fujimaki (Ph.D.)
Research Fellow, NEC Data Science Research Laboratory

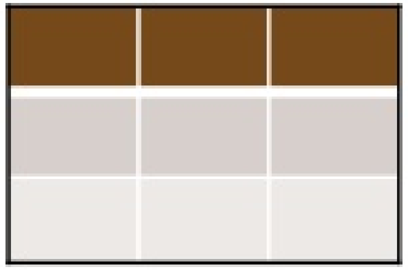
- Ryohei is research fellow, data science research laboratories, for NEC Corporation, a leading provider of advanced analytics technologies based on artificial intelligence.
- In addition to technology R&D, Ryohei is also heavily involved with co-developing cutting-edge advanced analytics solutions with NEC's global business clients and partners.
- Ryohei received his Ph.D. degree from the University of Tokyo, and became the youngest research fellow ever in NEC Corporation's 117-year history.

Agenda



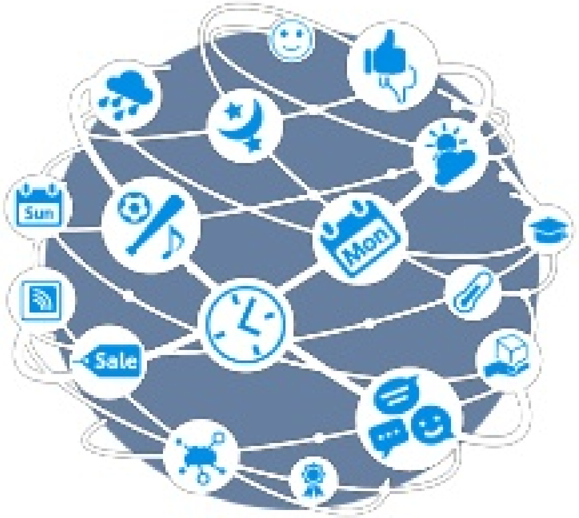
Training Data
Validate Data
Test Data
Mid
Low
High
Yes
No
Yes
Predictive model
Prediction results

Agenda



Training Data
Validate Data
Test Data

Input

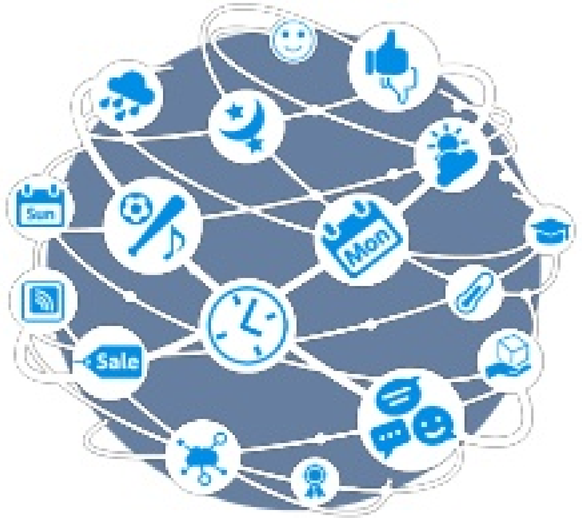


NEC Automatic Predictive Modeling

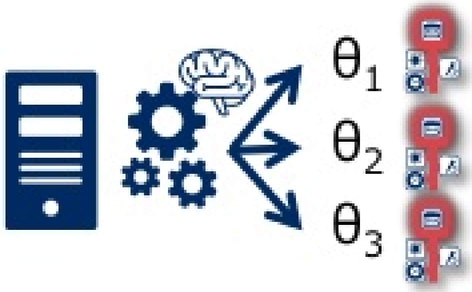
High-speed Generation

High accurate
predictive model
&
prediction results

Agenda



Mini-gapped Integration



Parameter-aware Scheduling

High Scalability

Automatic Predictive Modeling

Enterprise Applications of Predictive Analysis

Energy/Water Operation Mgmt.

Sales Optimization

Product Price Optimization

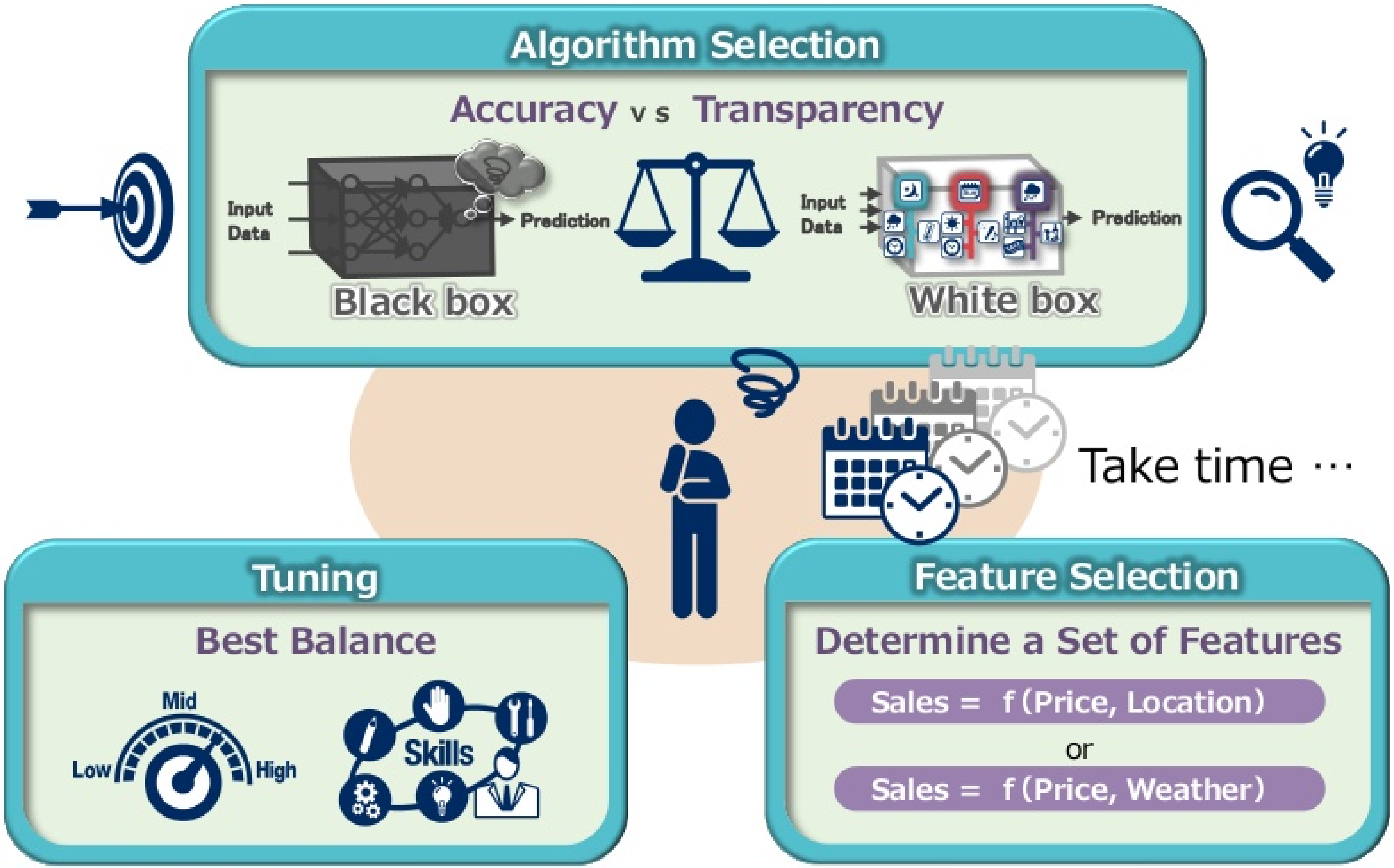
Driver Risk Assessment

Predictive Maintenance

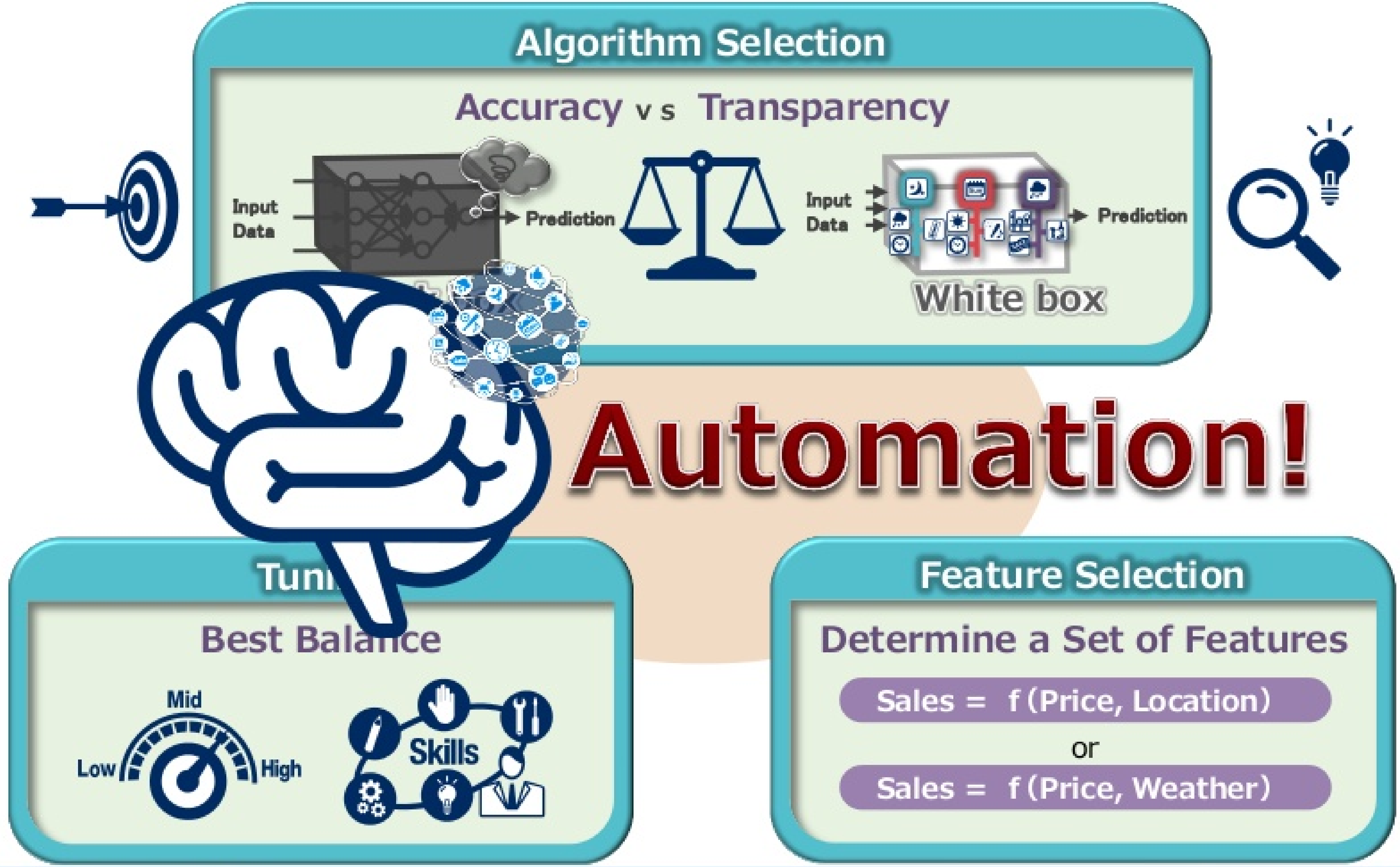
Churn Retention

Inventory Optimization

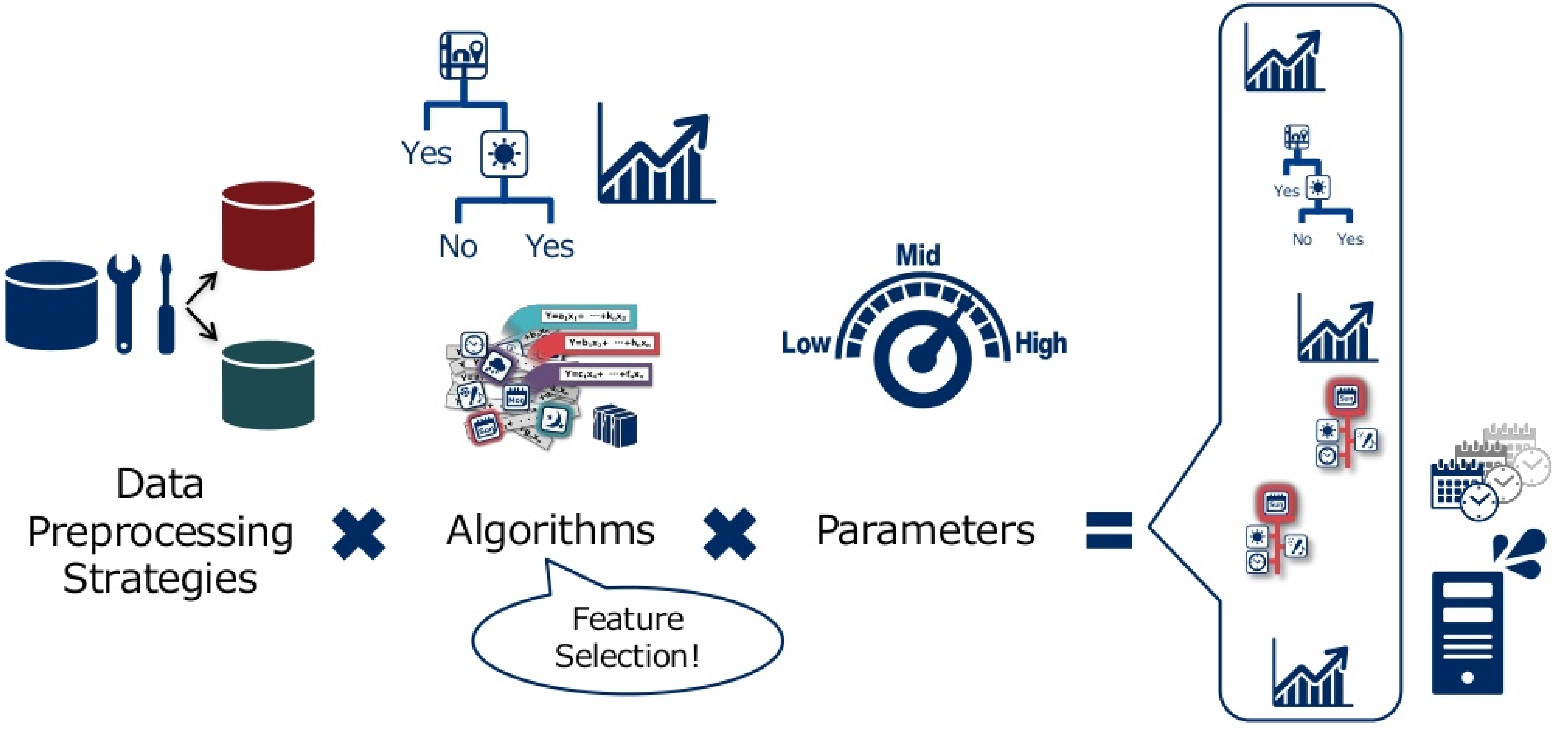
Predictive Model Design is "Black-Arts"



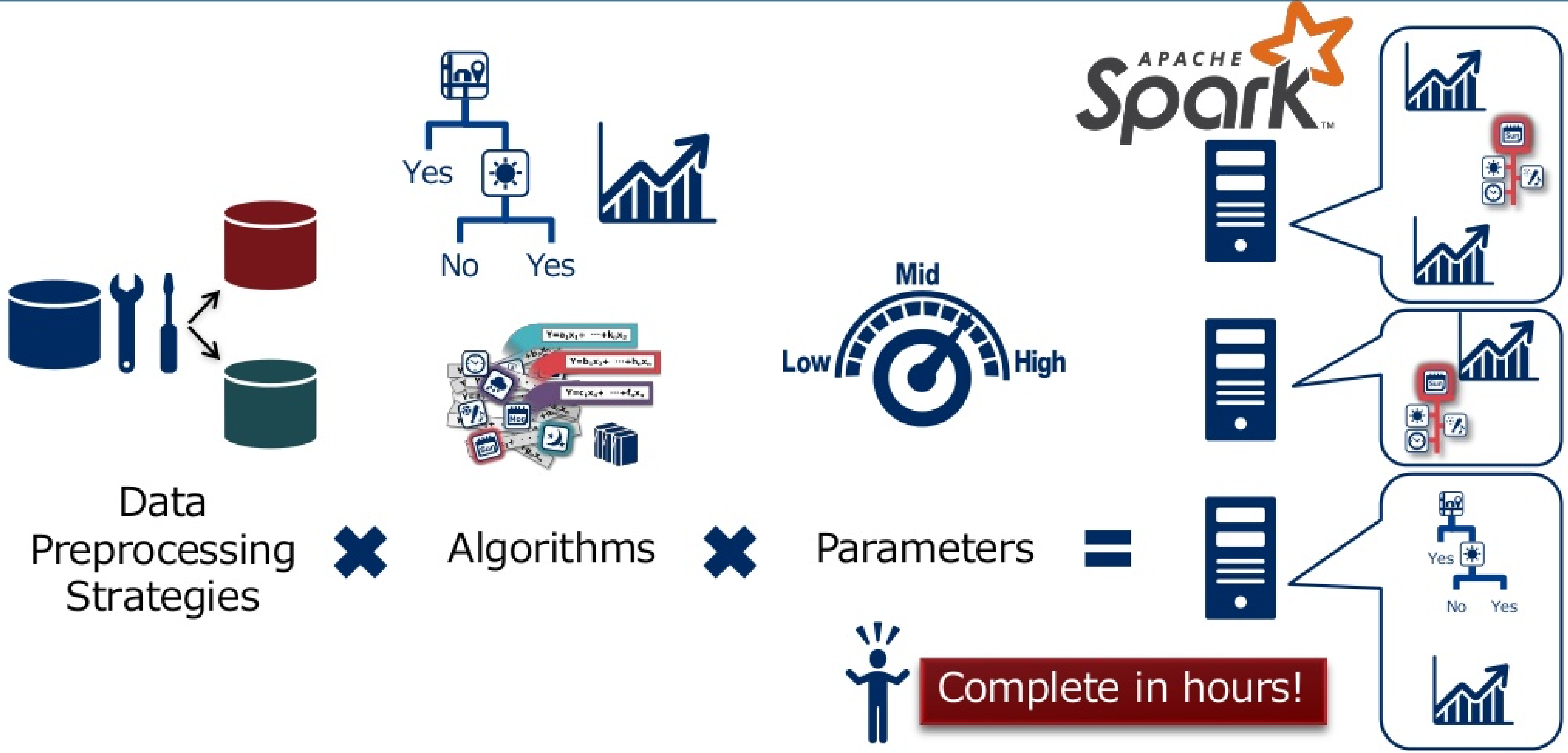
NEC Automatic Predictive Modeling



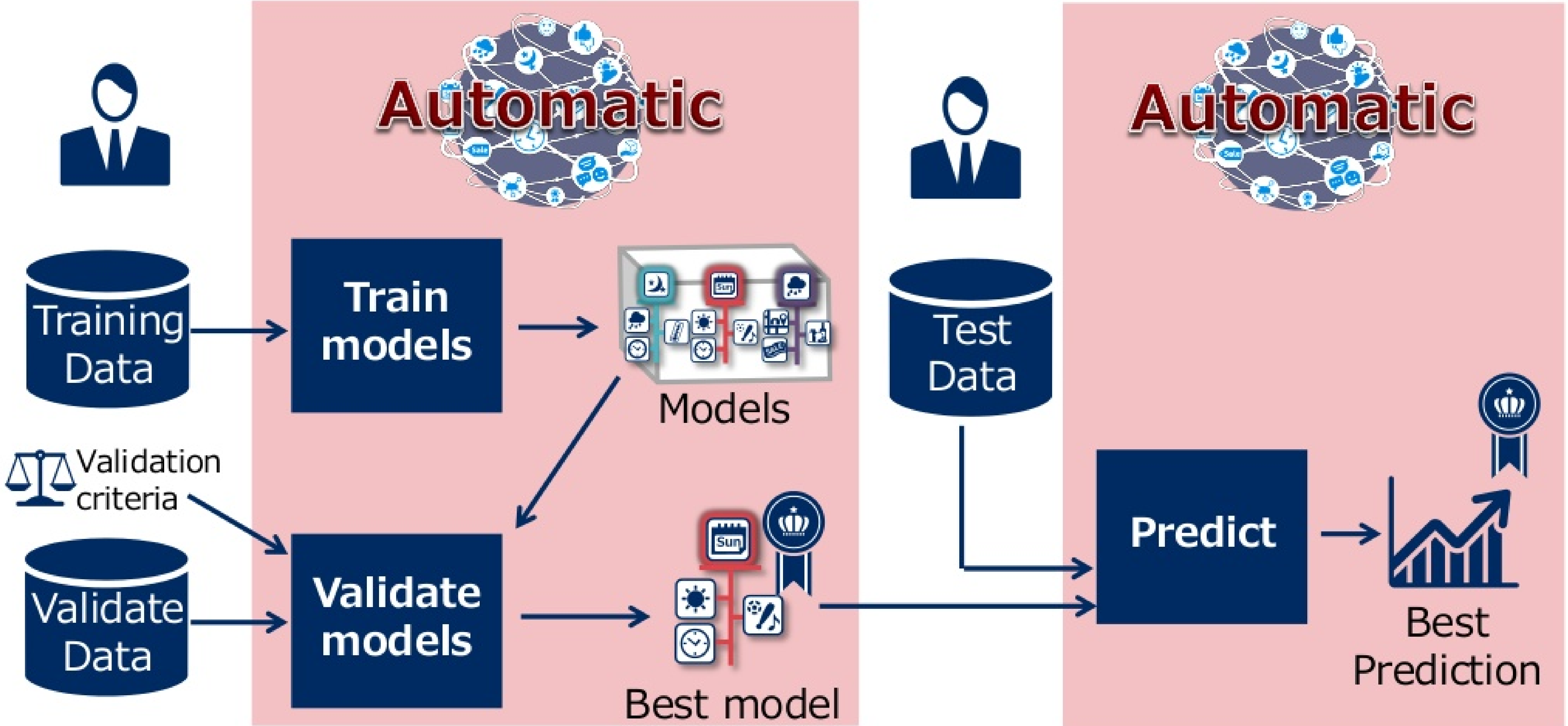
Explore Massive Modeling Possibilities



Automate and Accelerate by Apache Spark!

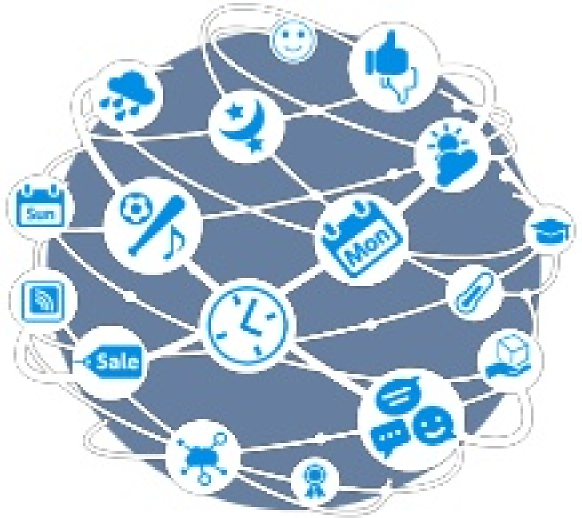


Modeling and Prediction Flow

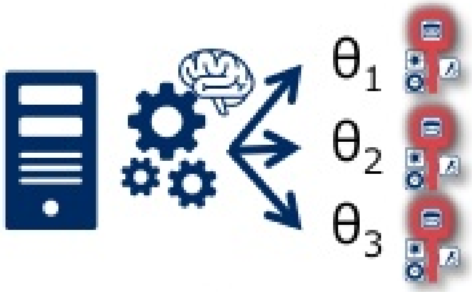


Design Challenges and Solutions

3 Design Challenges



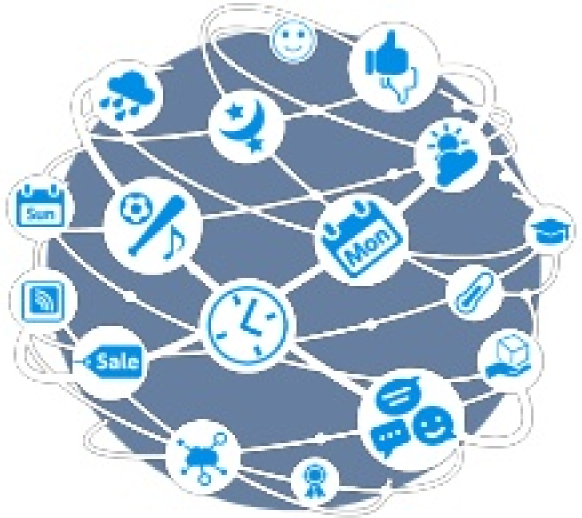
Mini-gapped Integration



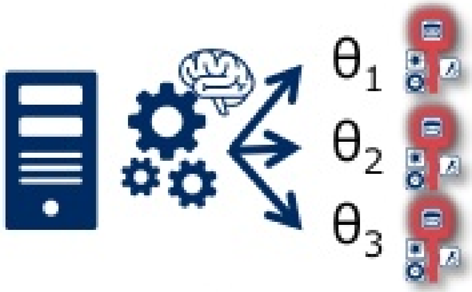
Parameter-aware Scheduling

High Scalability

3 Design Challenges



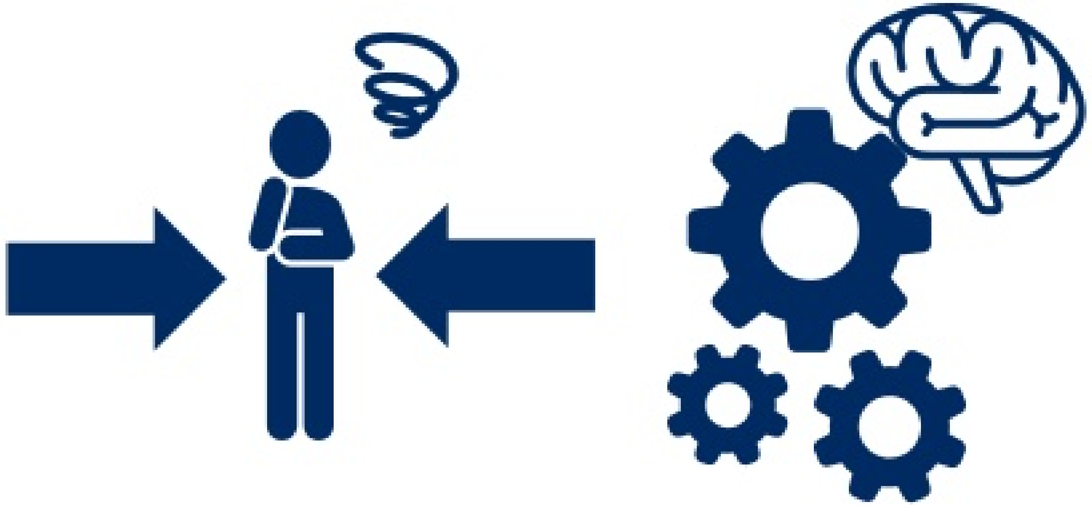
Mini-gapped Integration



Parameter-aware Scheduling

High Scalability

Implementation Gap between Spark and ML engines

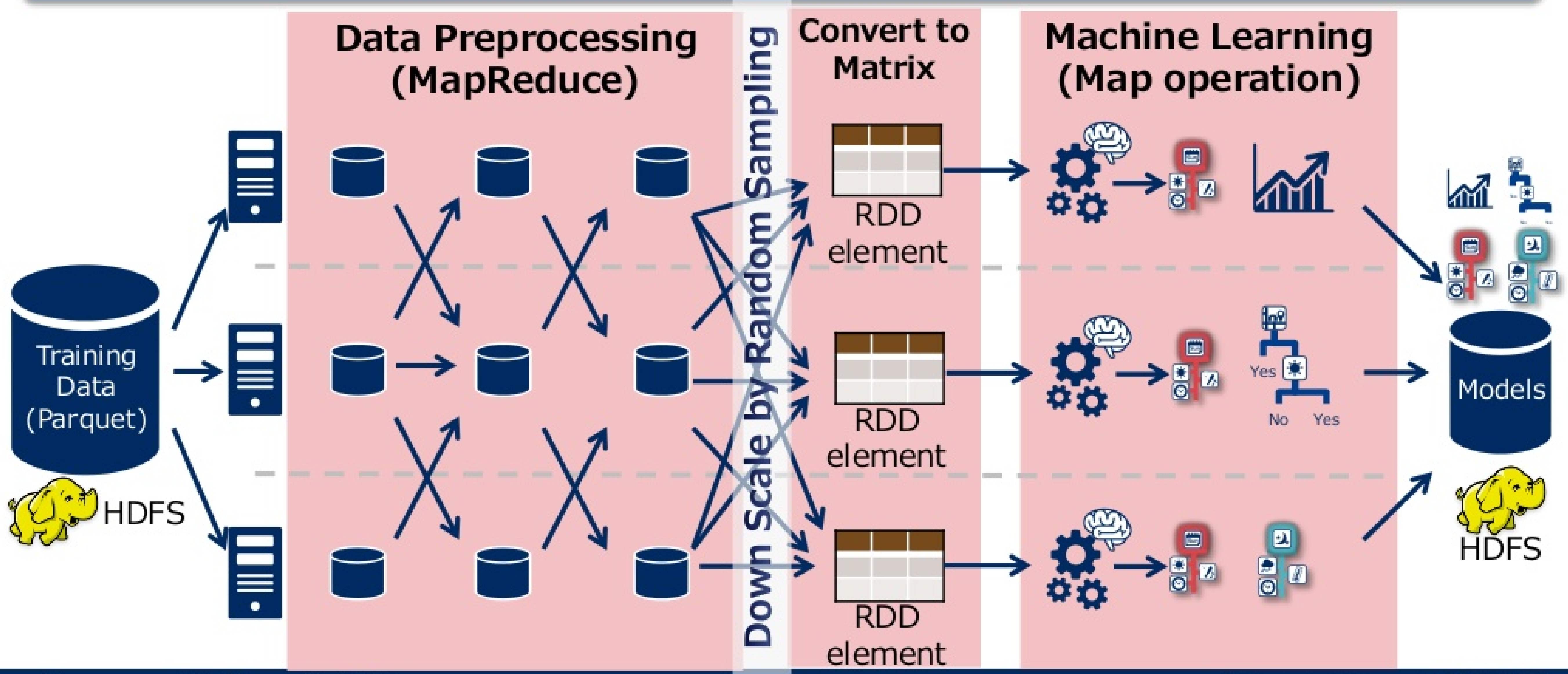


- **Distributed** memory architecture
- **MapReduce** computation model
- **Scala on JVM**

- **Single or shared** memory architecture
- **Standard** computing framework
- **High-speed native binary code**

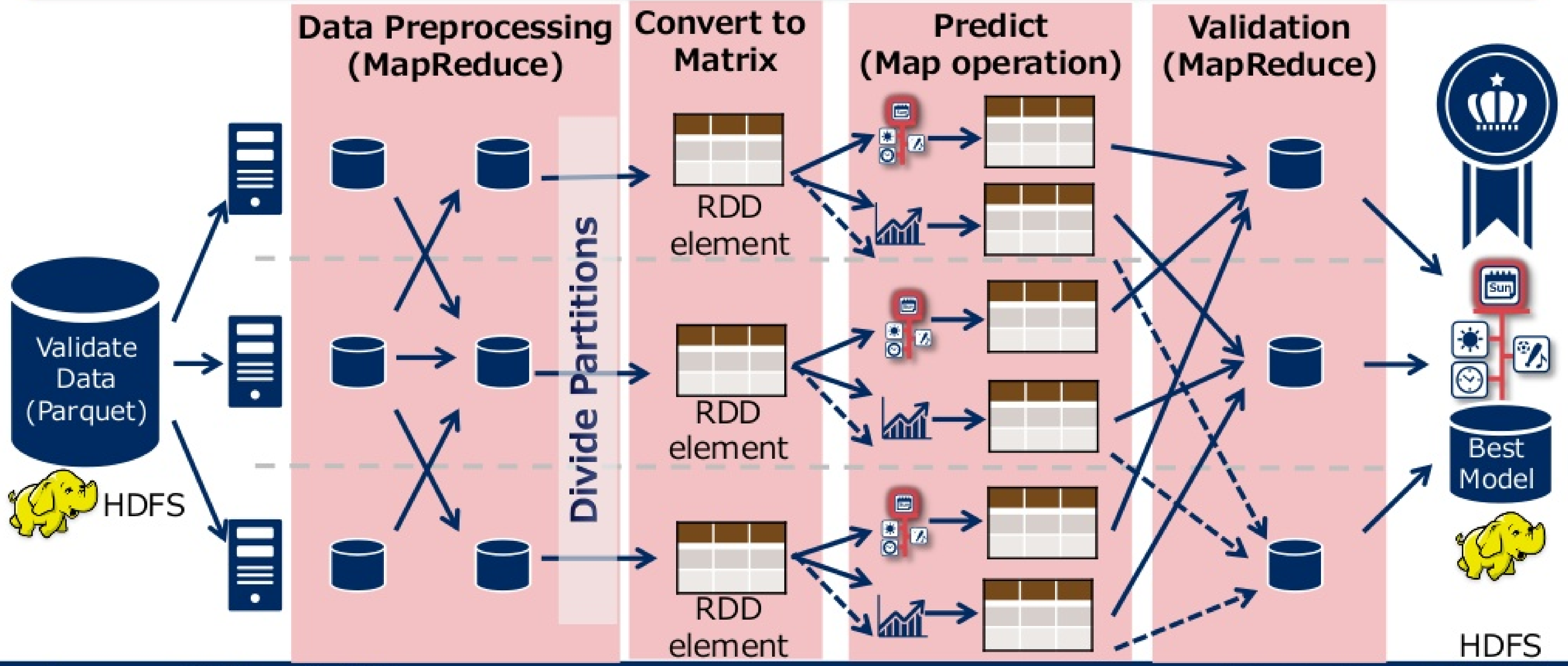
Mini-gapped Integration

Smoothly bridge 'distributed' preprocessing and 'parallel' execution of ML engines



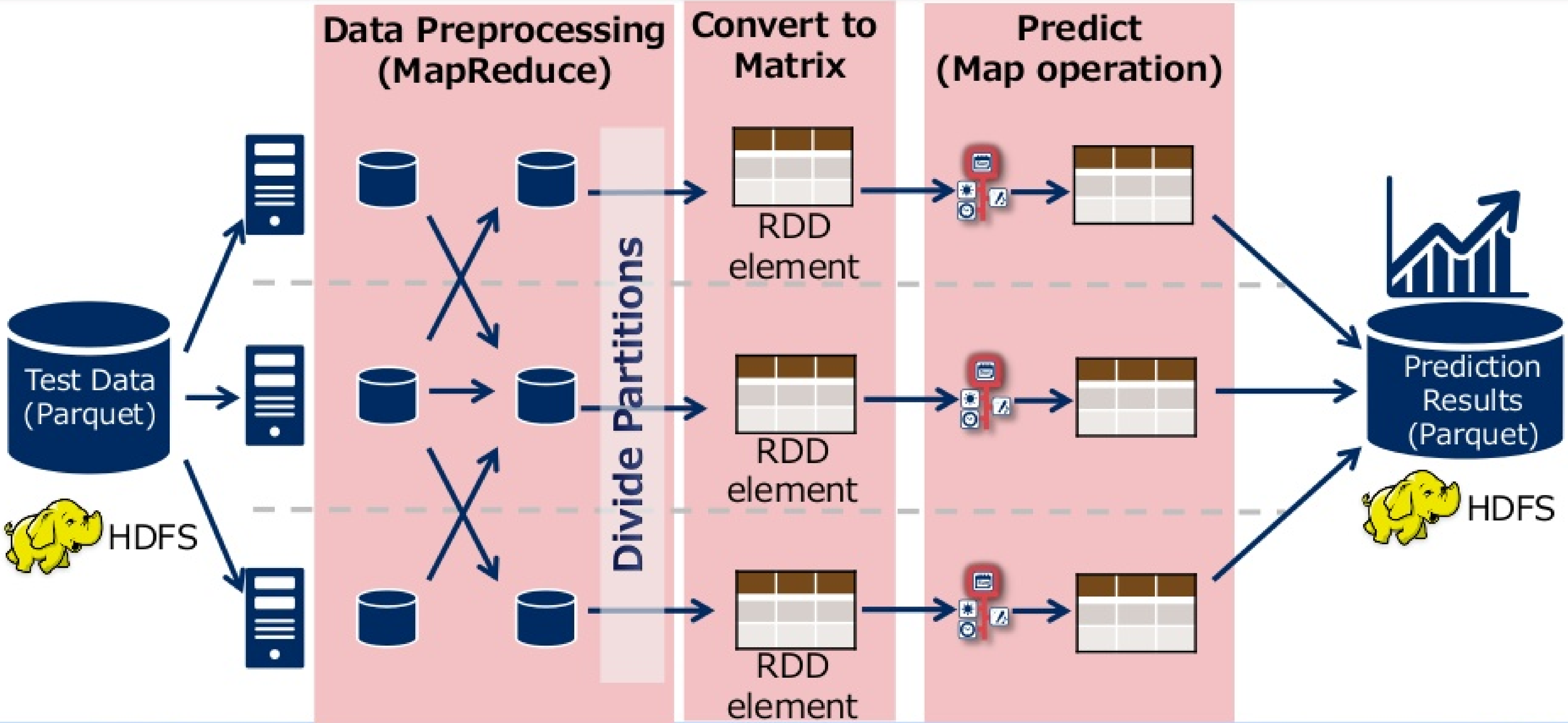
Mini-gapped Integration

Smoothly bridge 'distributed' preprocessing and 'parallel' execution of ML engines

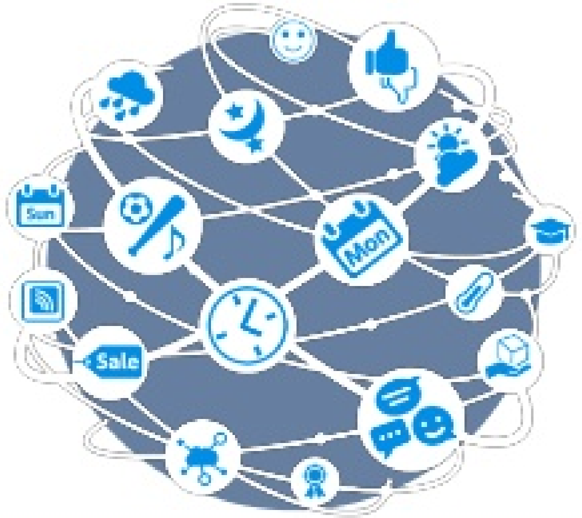


Mini-gapped Integration

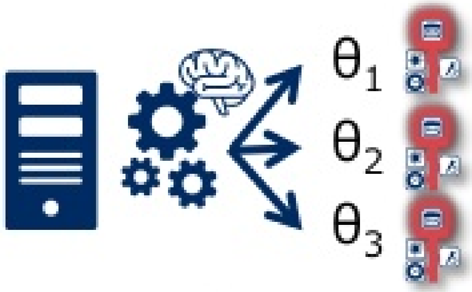
Smoothly bridge 'distributed' preprocessing and 'parallel' execution of ML engines



3 Design Challenges



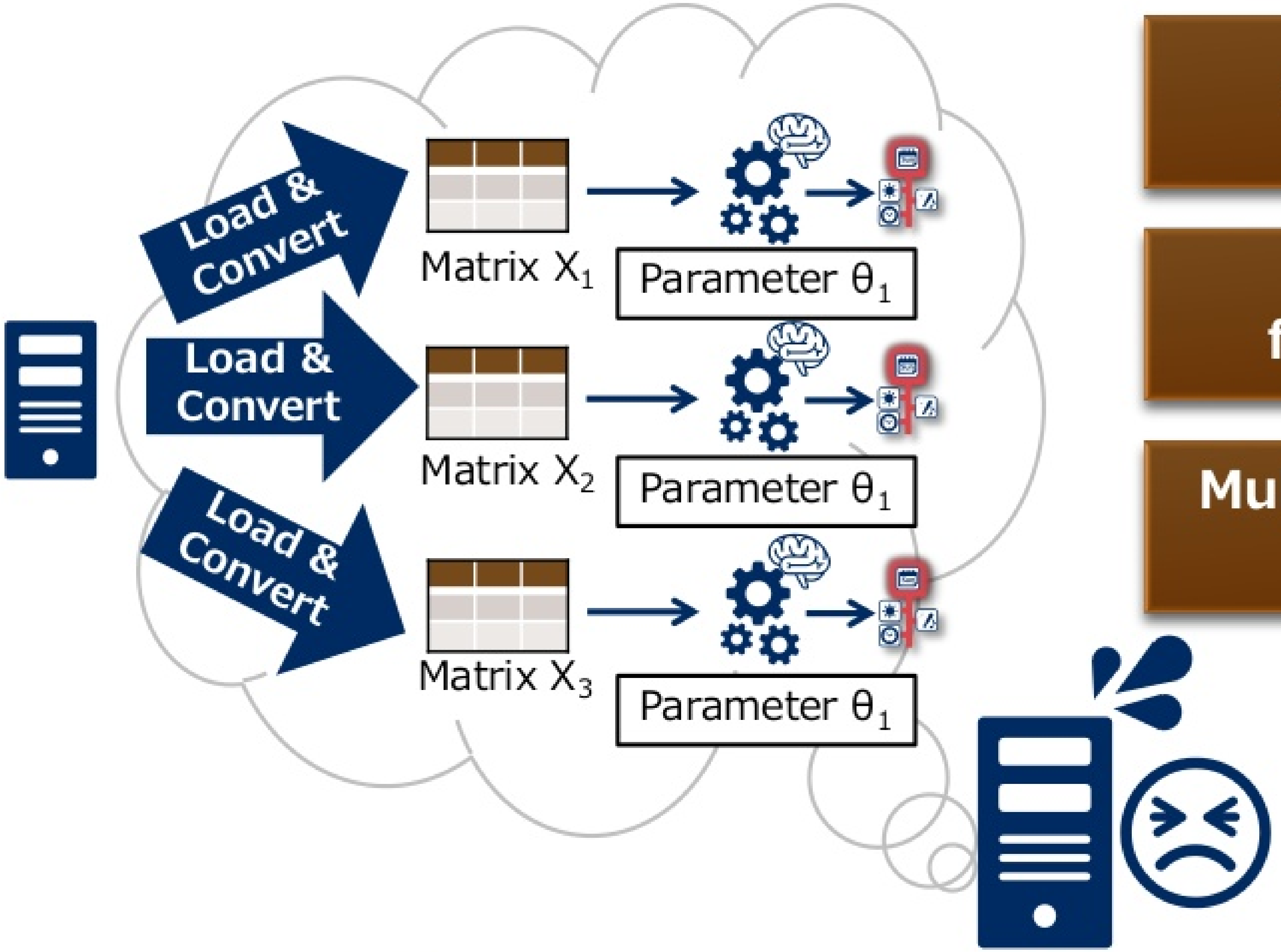
Mini-gapped Integration



Parameter-aware Scheduling

High Scalability

Naive Implementation requires Multiple Data Load & Convert

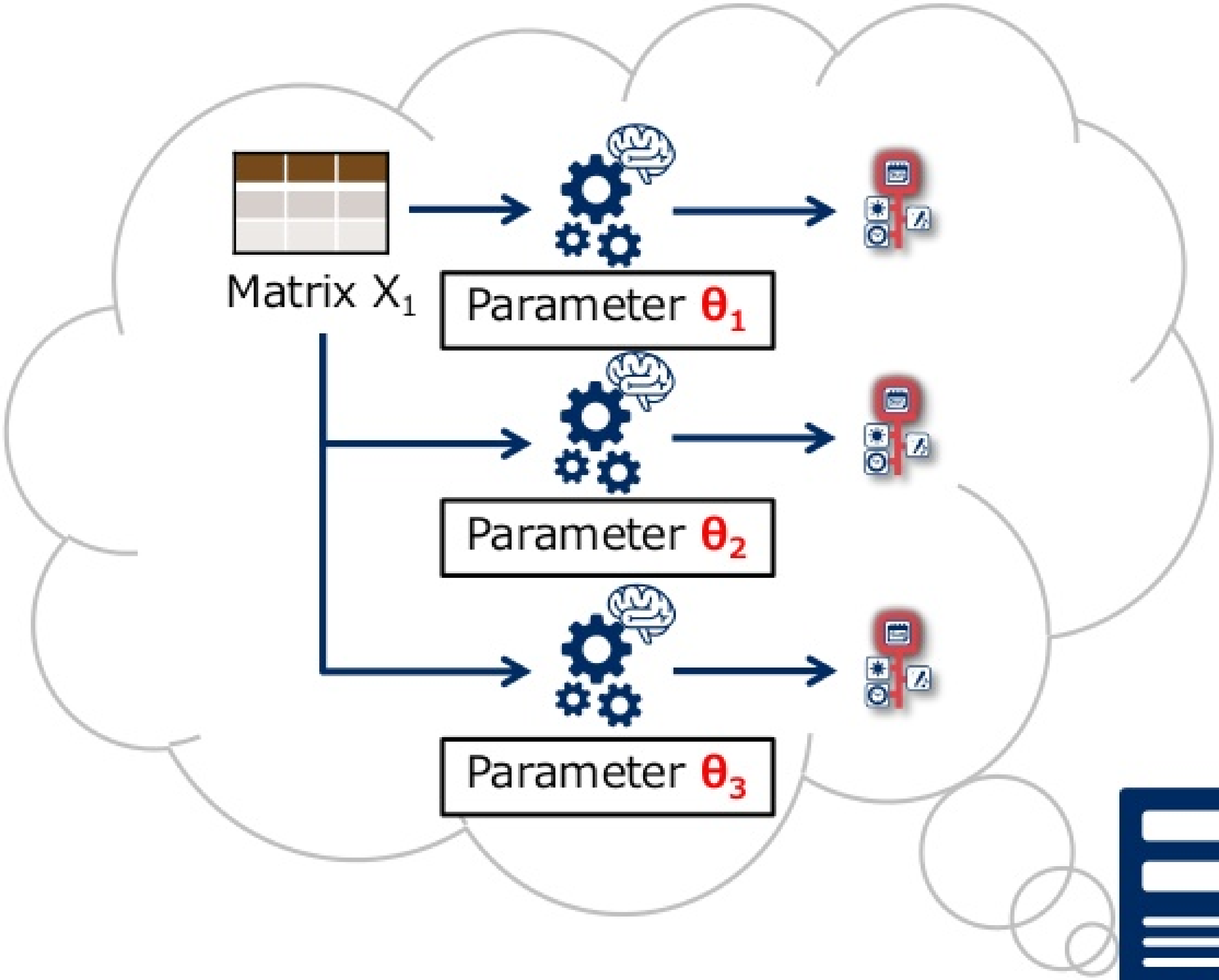


Waste of memory

Load of data from other servers

Multiple data convert to matrix

Parameter-aware Scheduling

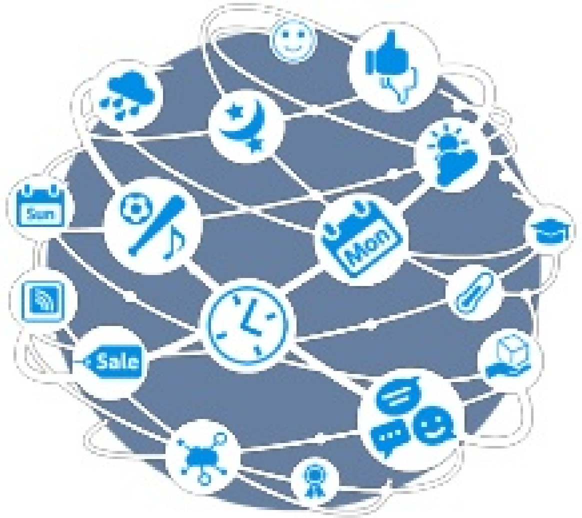


Efficient memory usage

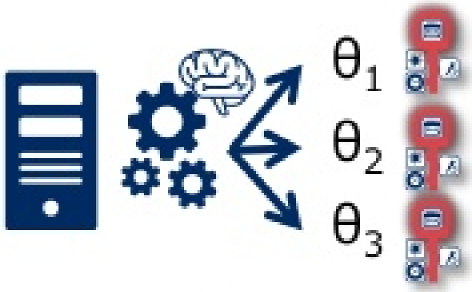
Low load of data from other hosts

Low frequency of data convert to matrix

3 Design Challenges



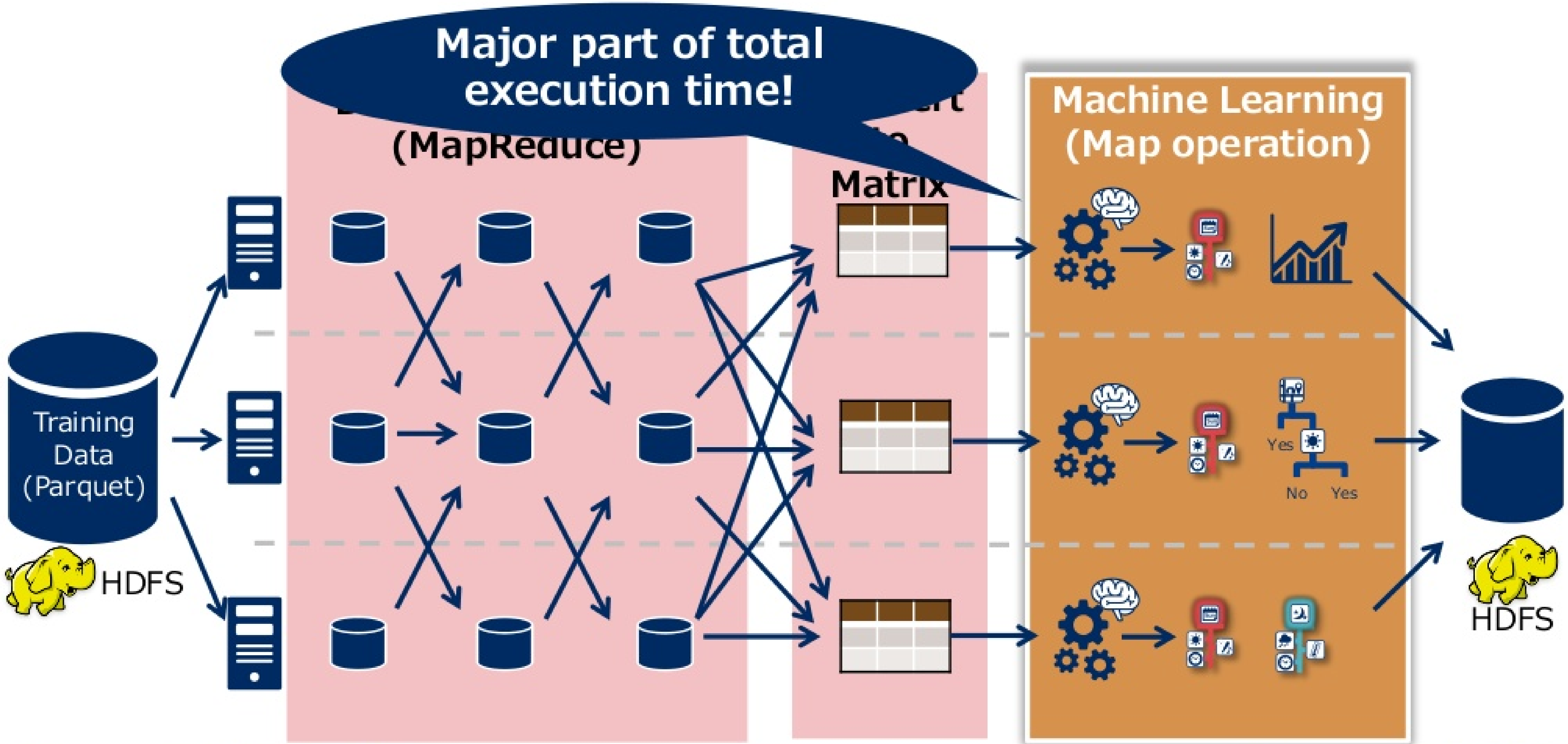
Mini-gapped Integration



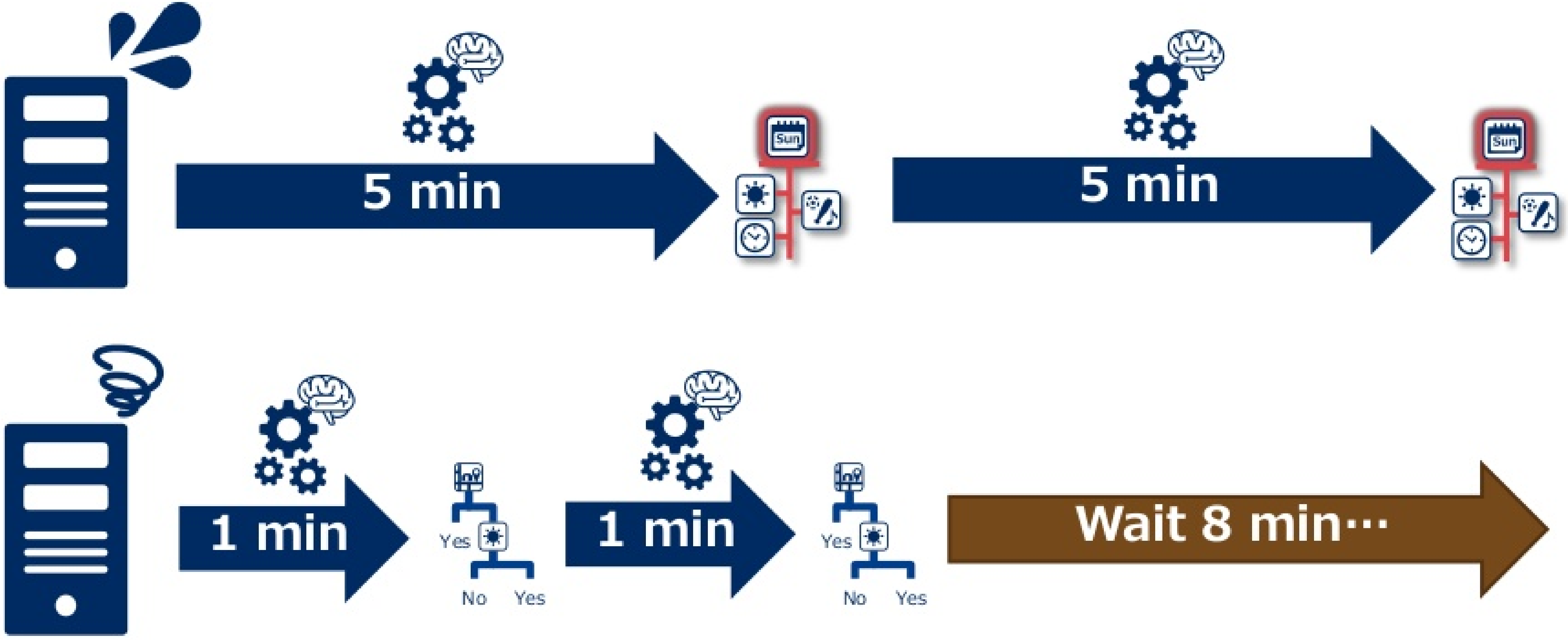
Parameter-aware Scheduling

High Scalability

Major Part of Computing is Machine Learning

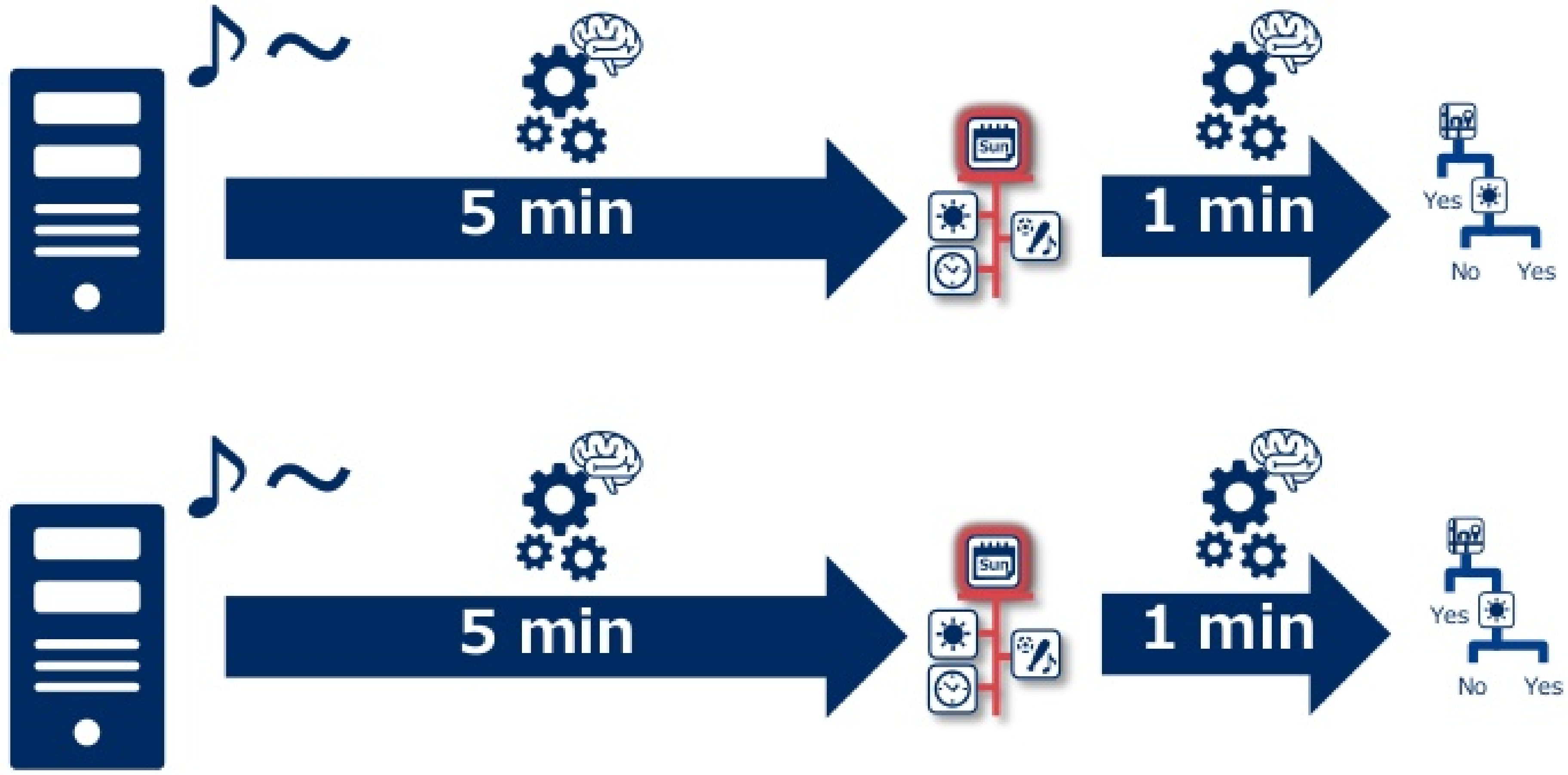


Bad Scheduling degrades Execution Efficiency



Predictive Scheduling increases Performance Scalability

Balance complex model learning and simple one



Tip of Spark on YARN for Stable Execution

Default Configuration sometimes fails Execution w/ Enough Memory…

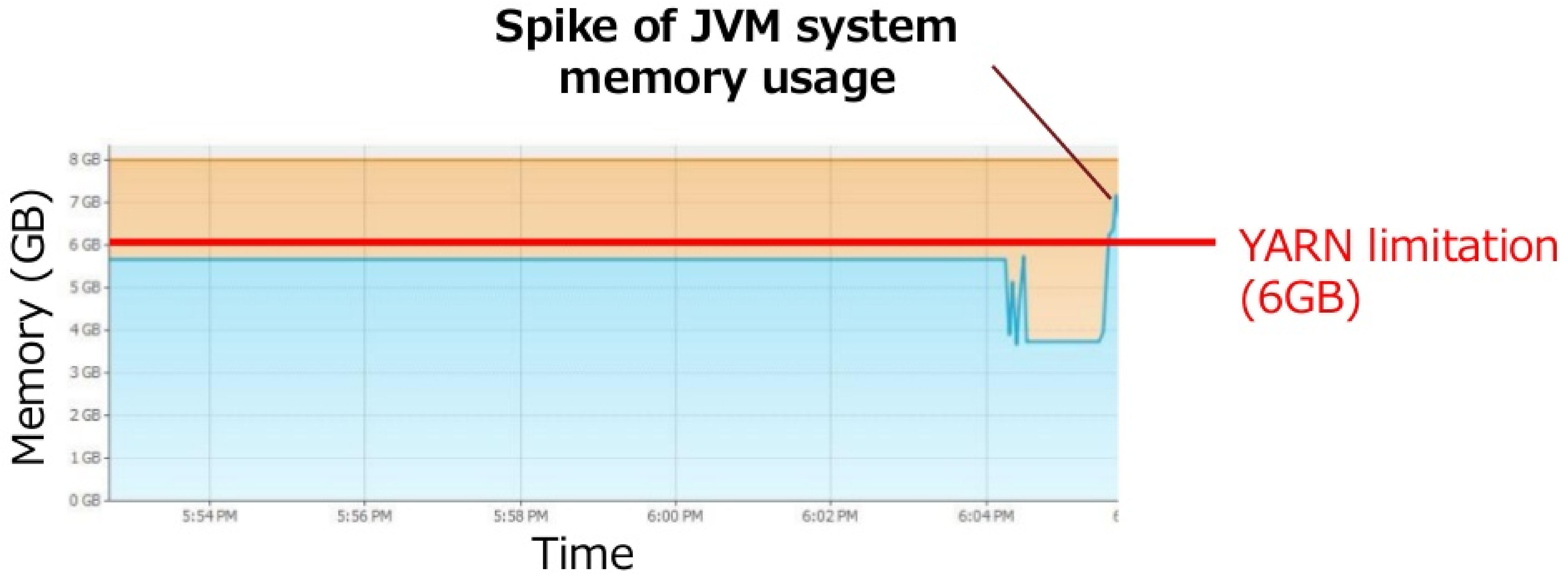
Spark WebUI says …

ExecutorLostFailure (executor 23 exited caused by one of the running tasks) Reason: Container killed by YARN for exceeding memory limits. 9.0 GB of 9 GB physical memory used. Consider boosting spark.yarn.executor.memoryOverhead.

We serve much much memory to Spark but it continues failing. Why!?!?

Spark on YARN Tips for Stable Execution

… Because JVM system memory spikes over than YARN limitation suddenly (*)



(*) Shivnath and Mayuresh. “Understanding Memory Management In Spark For Fun And Profit,” Spark Summit 2016.

Spark on YARN Tips for Stable Execution

Tips: We have to carefully configure 'spark.yarn.*.memoryOverhead'

spark.yarn.executor.memoryOverhead	executorMemory * 0.10, with minimum of 384	The amount of off-heap memory (in megabytes) to be allocated per executor. This is memory that accounts for things like VM overheads, interned strings, other native overheads, etc. This tends to grow with the executor size (typically 6-10%).

(http://spark.apache.org/docs/2.1.1/running-on-yarn.html)

15% overhead was required in our case…

Optimized overhead memory configuration is our future work

Evaluation

Evaluation Setup

- Compare accuracy with manual predictive modeling
- Measure execution time

Prediction problem

- Targeting top-10% of potential positive sample

Manual predictive modeling

- Done with scikit-learn v0.18.1
- All parameters are set with default values
  - except RandamForest (n_estimators = 200)

Data set

- KDDCUP 2014 competition data
  - 557K records for training and validate data
  - 62K records for test data
  - Features: 500
- KDDCUP 2015 competition data
  - 108K records for training and validate data
  - 12K records for test data
  - Features: 500
- IJCAI 2015 competition data
  - 87K records for training, validate and test data
  - Features: 500

Cluster Spec.

Scalable Modular Server (DX2000)

- Size: **3U!**
- # Server modules: **34**
- CPU: **272** Intel **Xeon-D 2.1GHz** cores
  - (128 cores used in the evaluation)
- MEM: **2TB**
  - High memory access: marked **16x score** of Terasort of HiBench v5 compared to 3 x 1U servers!
- Storage: **34TB SSD**
- Network: **10GbE** for internal network

- Spark v1.6.0, Hadoop v2.7.3

http://www.nec.com/en/global/prod/dxseries/dx2000/product/index.html

Precision and Execution Time

NEC Automatic Predictive Modeling produces competitively-accurate models automatically in hours!

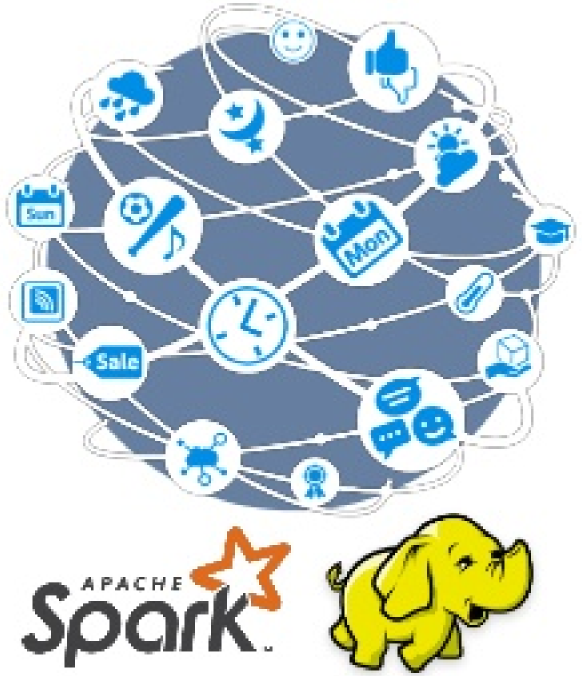
Top-10% Precision

data	NEC	Logistic Regression	SVM	Random Forests
KDDCUP 2014	**15.6%**	13.5%	12.0%	14.8%
KDDCUP 2015	**97.1%**	95.5%	93.1%	97.2%
IJCAI 2015	**8.2%**	8.3%	8.1%	8.2%

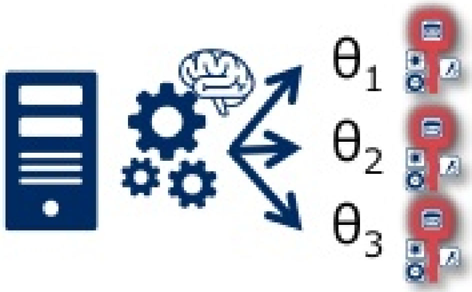
Execution time

data	NEC
KDDCUP 2014	**172** minutes
KDDCUP 2015	**45** minutes
IJCAI 2015	**36** minutes

Summary



Mini-gapped Integration



Parameter-aware Scheduling

High Scalability

Future work

Extend to other models (e.g., DeepLearning)

Reduce YARN memory overhead

Speed up by FPGA, Vector processors

\Orchestrating a brighter world

NEC

Appendix

Software Stack

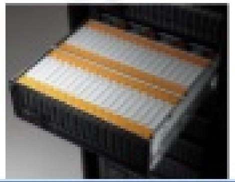
Web UI

Automation Engine

Native-speed Machine Learning Libraries

In-Memory Distributed Computing

Data Store / Resource Management



Computing Infrastructure

Scalable Modular Server (DX2000) Spec.: Server Module

Form factor	Server module that plugs into the Module Enclosure
Number of Processors	1
Processors	Intel® Xeon® Processor D-1527(2.20GHz/4-core/6MB) Intel® Xeon® Processor D-1541(2.10GHz/8-core/12MB) Intel® Xeon® Processor D-1571(1.30GHz/16-core/24MB)
Memory type	DDR4-2133 ECC SO-DIMM
Memory slots	4
Memory capacity	16 GB / 32 GB / 64 GB
Storage type	M.2 SATA SSD
Internal storage capacity	128 GB / 256 GB / 512 GB / 1TB
Network	2 10GbE links to switch modules 2 additional 10GbE links to switch modules with an optional 10G LAN module (Occupies one server module slot)
Systems management	EXPRESSSCOPE Engine 3
Operating systems and virtualization software	Red Hat® Enterprise Linux® 7.2 / 6.8 VMware ESXi™ 6.0 Microsoft® Windows® Server 2012 R2 CentOS 7.2 Ubuntu14.04LTS / 16.04LTS

Scalable Modular Server (DX2000) Spec.: Module Enclosure

Form factor / height	3U Rack
Server module slots	44 (22 slots can be used for 10G LAN modules and 8 slots can be used for PCIe cards) * The number of installable modules may change according to configuration.
Network switches	2 network switch modules (L2)
Redundant cooling fan	Standard, hot plug
Power supplies	3 Hot plug power supply 1600 Watt 200-240 VAC ± 10% 50 / 60 Hz ± 3 Hz
Redundant power supply	2+1 redundant, hot plug
Temperature and humidity conditions (non-condensing)	Operating: 10 to 35* °C/ 50 to 95* °F , 20 to 80% Non-operating: -10 to 55°C/14 to 131°F, 20 to 80% * In specific configurations, the operable ambient temperature is up to 40°C/104°F
Dimensions (W x D x H) and maximum weight	448.0 x 769.0 x 130.0 mm / 17.6 x 30.3 x 5.1 in, 48 kg / 105.82 lbs

Scalable Modular Server (DX2000) Spec.: Network Switch Module (L2)

Form factor	Network Switch Module that plugs into the Module Enclosure
Network	Up link: 8 40G QSFP+ ports plus 1 1000BASE-T for management Down link: 44 10GbE