ITU-ML5G-PS-015: Network failure detection and root cause analysis in 5GC by NFV-based test environment

Team Kaien
2021-12-01

Team member:

阿部 海燕　Kaien ABE
abe@people.kobe-u.ac.jp

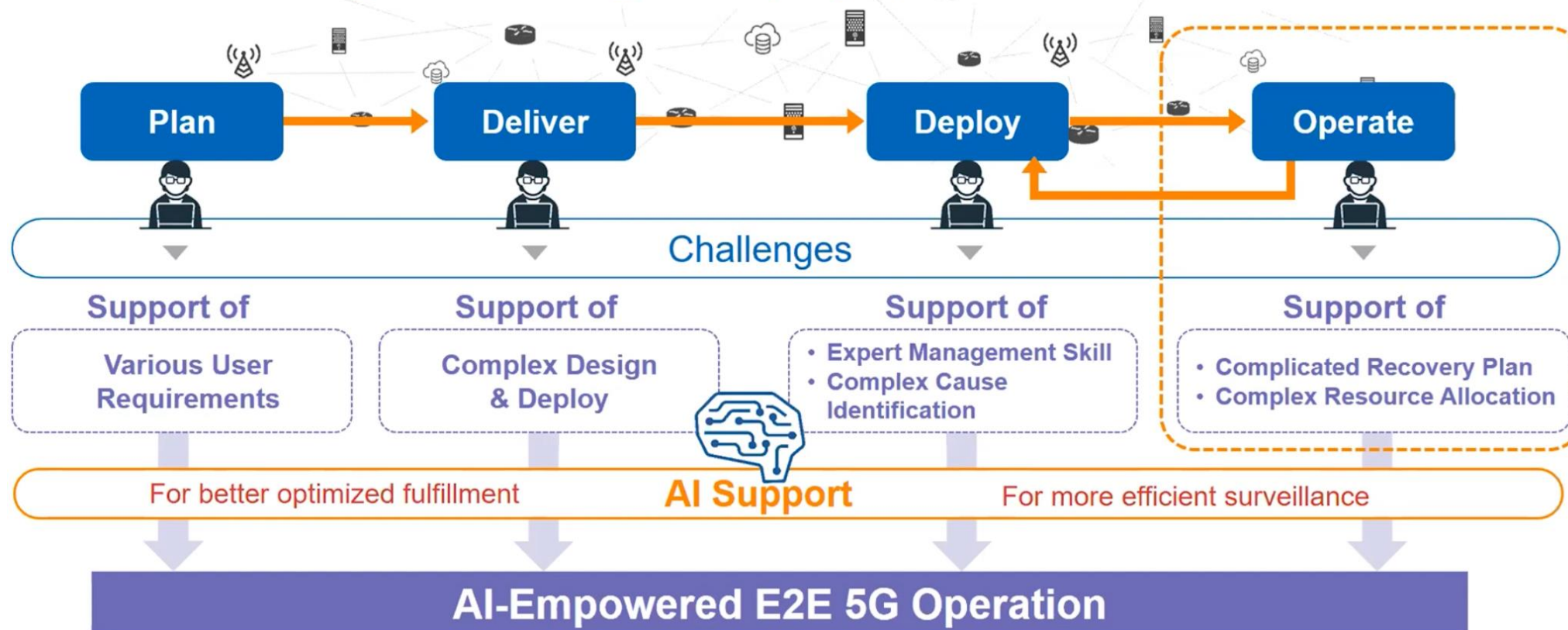**白石 善明　Yoshiaki SHIRAISHI**
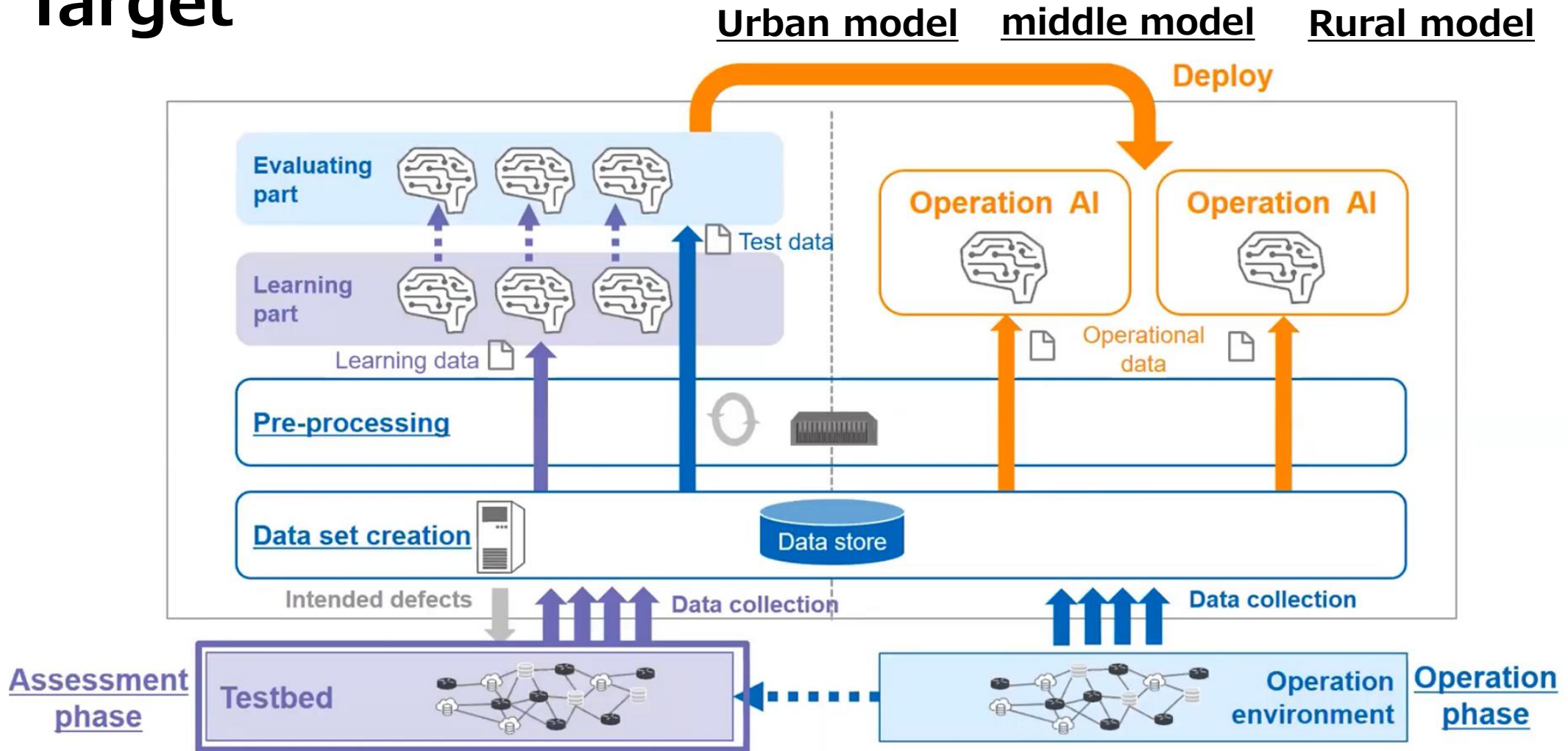zenmei@port.kobe-u.ac.jp

KOBE UNIVERSITY

Content

Project Introduction

背景

E2E 5G operational architecture empowered by AI, including **Define/Design & Deploy operations** functionalities

Plan → **Deliver** → **Deploy** → **Operate**

Challenges

Support of
Various User Requirements

Support of
Complex Design & Deploy

Support of
- **Expert Management Skill**
- **Complex Cause Identification**

Support of
- **Complicated Recovery Plan**
- **Complex Resource Allocation**

For better optimized fulfillment   **AI Support**   For more efficient surveillance

AI-Empowered E2E 5G Operation

Tomorrow, Together **KDDI**

https://www.youtube.com/watch?v=T18ivePrIUQ

Target

Urban model   middle model   Rural model

https://www.youtube.com/watch?v=T18ivePrIUQ

Data processing environment

計算機・ライブラ 環境

Computer configuration	
CPU/クロック周波数	Intel(R) Core(TM) i7-9750H/2.60GHz
メモリー（Memory）	32GB
OS	Ubuntu 18.04

Library	
Anacond3 (Environment)	Python3.8
json (process original json data)	csv (output data format)
Panadas (feature extraction)	sqlite (database for data post-processing
XGBoost (normal model)	XGBoost (increment model)

Original data is 77GB json file, it takes several days for feature extraction.

Dataset overview

Provided data: json files（77GB）	a Urban（都市）	c Rural（田舎）	b Middle（中間）
Train	98,533	98,533	12,360
Test	24,647	24,647	24,647
Failure label	2480（train）＋620（test）	2480（train）＋620（test）	310（train）＋620（test）

Extract features by python code
(One data is split to 3 data according to nodes number for more precise label)

Table data of features (101)

	a Urban（都市）	c Rural（田舎）	b Middle（中間）
Train: (101 features) [nodes: amf, ausf, udm, respectively]	295,599 （＝98,533×3） （×101 特徴量）	295,599 （＝98,533×3） （×101 特徴量）	37,080 （＝12,360×3） （×101 特徴量）
Test: (101 features) [nodes: amf, ausf, udm, respectively]	73,886 （≒24,647×3） （×101 特徴量）	73,889 （≒24,647×3） （×101 特徴量）	73,941 （＝24,647×3） （×101 特徴量）
Abnormal data (from failure label)	4,645	4,011	1,212
Abnormal /Total (%)	13%	10%	10%

Data investigation
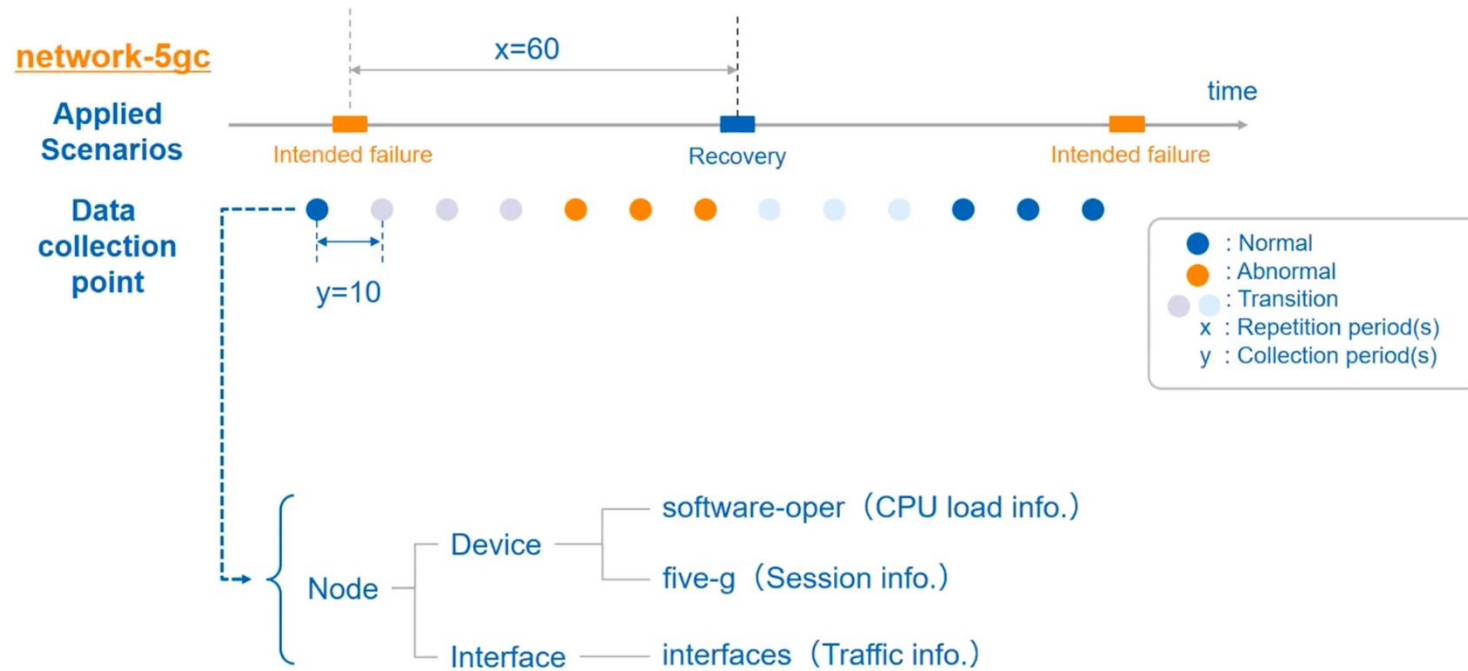
Nodes	Failure
amf	○ 障害発生
ausf	○ 障害発生
udm	○ 障害発生
gnb	× 障害なし
smf	× 障害なし
upf	× 障害なし
nrf	× 障害なし
dn	× 障害なし

8 types nodes are included in the dataset. but failures are happened in 3 types nodes according to Failure label. <u>In order to reduce calculation cost this time</u>, only these 3 types nodes are processed. In real scenario, all of nodes should be considered to process.

Failure label information

Data processing (1) データ分割のアイデアとメリット

In the provided dataset, failures happened in three nodes (total nodes=8) so that we only processing 3 nodes information for decreasing data processing work.

	障害なし （No failure）	障害発生 （Failure）	障害なし （No Failure）	Positive
Data_1	amf	ausf	udm	Label=1

101 Features   101 Features   101 Features

One data is split to 3 data

Data_1_amf	amf	Label=0
Data_1_ausf	ausf 101 Features	Label=1
Data_1_udm	udm 101 Features	Label=0
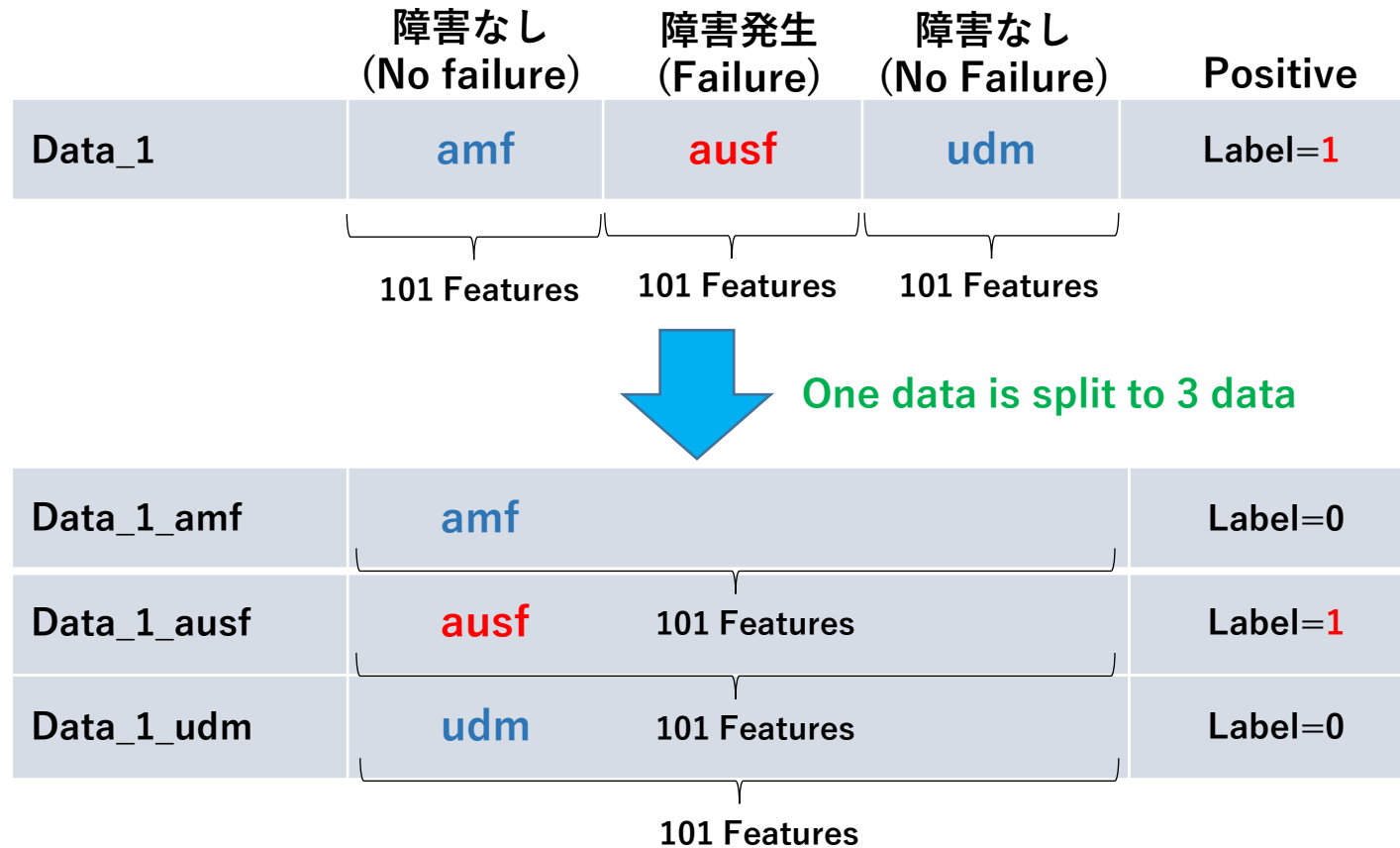
101 Features

障害発生したノートを特定できる ＆ 過学習を抑制できる
Failure nodes can be located precisely and feature number is decreased to prevent overfitting in ML

Data processing (2)-how to select features

特徴量の選抜基準

Nodes	障害	抽出項目数
amf	障害発生	204（a,c） 154（b）
ausf	障害発生	175(a,c,b)
udm	障害発生	177(a,c,b)
gnb	障害なし	
smf	障害なし	
upf	障害なし	
nrf	障害なし	
dn	障害なし	

■**Delete category info.**
{interface_type:softwareloopback,tunnel,healthy,unknown
interface_name,
interface_if_index,
Interface_if_name
platform-fru-rp,
phys-address,
Healthy, critical, unknow,up, down,
Null…
}

■**Remain digital info.**
{memory-states-,
per-core-states-,
interface-parameter…
}

■**Common features among all of nodes**

datetime	failure_type	ifname	network	node
20210209154552	interface-up	ens4	*NULL*	udma1
20210205234627	bridge-delif	*NULL*	nw-cpa1	ausfa1
20210205085657	interface-down	ens5	*NULL*	amfa1
20210209044653	memory-stress-stop	*NULL*	*NULL*	ausfa1
20210205122320	vcpu-overload-stop	*NULL*	*NULL*	amfa1
20210205141413	vcpu-overload-stop	*NULL*	*NULL*	amfa1
20210209115810	normal	*NULL*	*NULL*	*NULL*
20210205235154	vcpu-overload-stop	*NULL*	*NULL*	ausfa1
20210206041959	bridge-delif	*NULL*	nw-cpa1	ausfa1
20210205022336	interface-down	ens5	*NULL*	amfa1
20210205125623	interface-loss-start	ens5	*NULL*	amfa1
20210210014614	memory-stress-stop	*NULL*	*NULL*	amfa1
20210205205932	interface-loss-start	ens4	*NULL*	udma1
20210209055829	bridge-delif	*NULL*	nw-cpa1	amfa1
20210206020246	bridge-delif	*NULL*	nw-cpa1	udma1
20210205143538	interface-loss-stop	ens5	*NULL*	amfa1
20210205221325	memory-stress-stop	*NULL*	*NULL*	udma1
20210209225842	interface-loss-stop	ens4	*NULL*	udma1

get features (101)

**This selection method is not perfect and presice.
More deeply data observation to select more precise
and informative features is needed.**

Data processing (3)- how to label data

Form the failure information (failure start-time,failure type,node,ifname), we traced the train&test data, then found the following info. strongly related to failures, so that we label the training and test data according these info. There might be other info also related failures. Further more data investigation should be carried out.

Info & Features　特徴量	Abnormal　障害あり(label=1)	Normal　障害なし(label=0)	
Memory_status：(148)(4512) software-oper_cisco-platform-software_control-processes_control-process_memory-stats_memory-status	'Critical' 'Warning'	'Healthy'	label
Minute_status_condition: 16 software-oper_cisco-platform-software_control-processes_control-process_load-avg-minutes_load-avg-minute_status_condition	'Critical'	'Healthy'	
Operation_status: 1(203)2(4996) interfaces_interfaces-state_interface_oper-status	'down' (start at, proc) (interface_down)	'up'	
memory-stats_used-percent： software-oper_cisco-platform-software_control-processes_control-process_memory-stats_used-percent	>90% (memory_stress_stop)	<=90	Verify by some features
software-oper_cisco-platform-software_control-processes_control-process_memory-stats_free-percent	<65 (memory_stress_stop)	>=65	
software-oper_cisco-platform-software_control-processes_control-process_memory-stats_free-number	<250000 (memory_stress_stop)	>=250000	
software-oper_cisco-platform-software_control-processes_control-process_per-core-stats_per-core-stat_idle	0 (<0.5) (Bridge_delif)	>0.5	
software-oper_cisco-platform-software_control-processes_control-process_per-core-stats_per-core-stat_nice	0 (<0.5) (vcpu_overload_stop)	>0.5	

Feature Profile 1 （features name）

Feature Profile (2)(values)

Data clustering using dimension reduce model (unsuppervised)
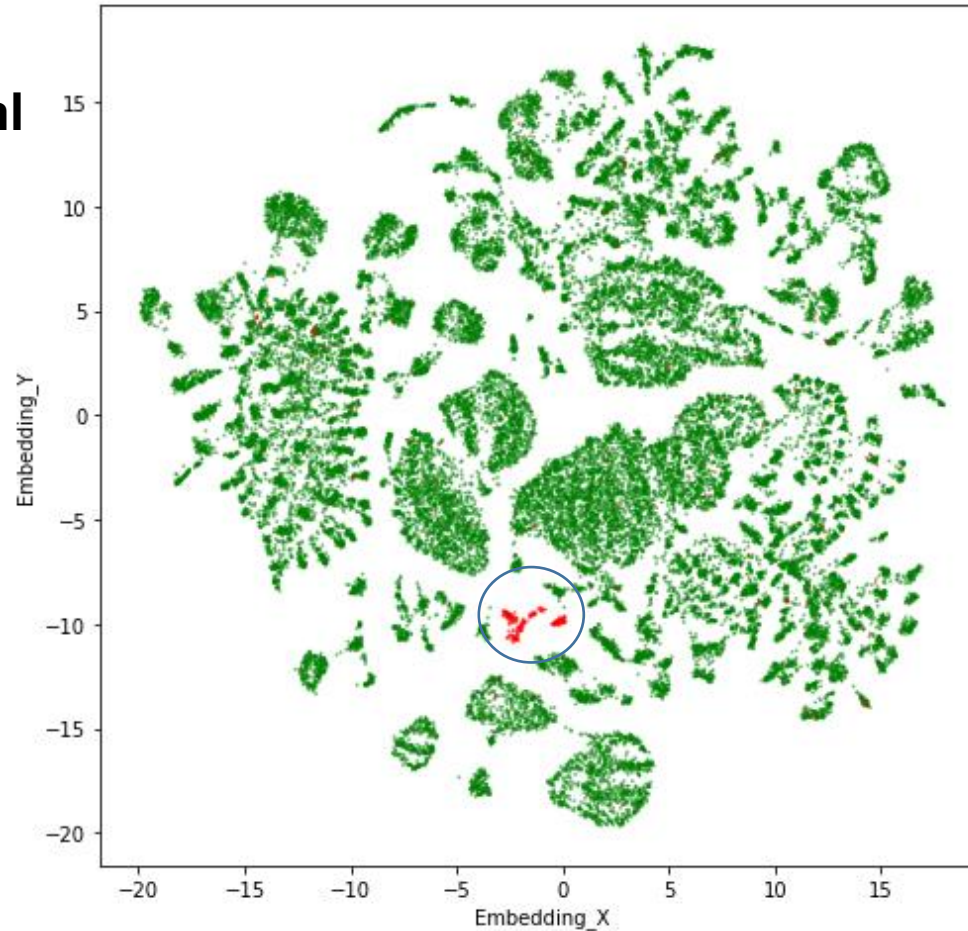
Machine Learning by Xgboost 2-class model

Task-1 課程#1

Training by a (urban) :   (Normal: 292,306; Abnormal: 3,293)
Test   by c (rural) :   (Normal: 73,082; Abnormal: 807)

	Precision	Recall	F-1
Xgboost	**100%**	**100%**	**100%**

Confusion Matrix	73,082	0(FP)
	0	807(TP)

Training by c (rural) :   (Normal:292,395; Abnormal: 3,204)
Test   by a (urban) :   (Normal: 72,534; Abnormal: 1,352)

	Precision	Recall	F-1
Xgboost	**100%**	**100%**	**100%**

Confusion Matrix	72,534	0(FP)
	0	1,352(TP)

Increment Learning by XGB Increment model

Task-2 課程#2

Training by argumentation new data b (middle)

1/X {
1/8
Limited Data

Already trained model in task-1 by data a(urban)
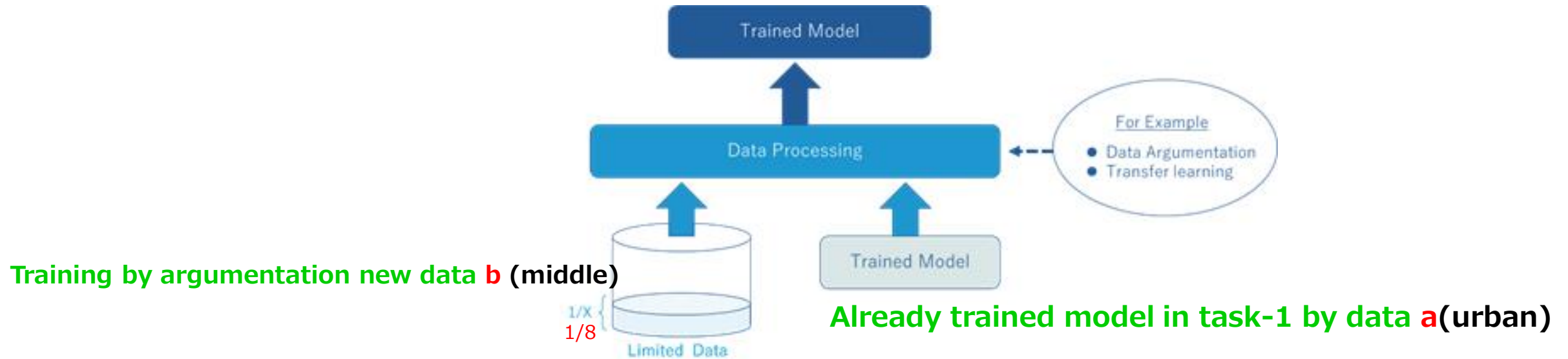
◆ **Trained model by a (urban) (Normal: 292,306; Abnormal: 3,293)** **+** **Training by b (middle)(Nomal:36,668; Abnormal: 412)**

■ **Test by b (middle) :** **(Normal: 73,129; Abnormal: 812)**

	Precision	Recall	F-1
Xgboost	**100%**	**83%**	**91%**

Confusion Matrix	73,129	0(FP)
	135	677(TP)

Increment Learning by XGB Increment model

Task 2 課程#2

Training by agumentation data b (middle)

1/8

Already trained model in task-1 by data c(rural)
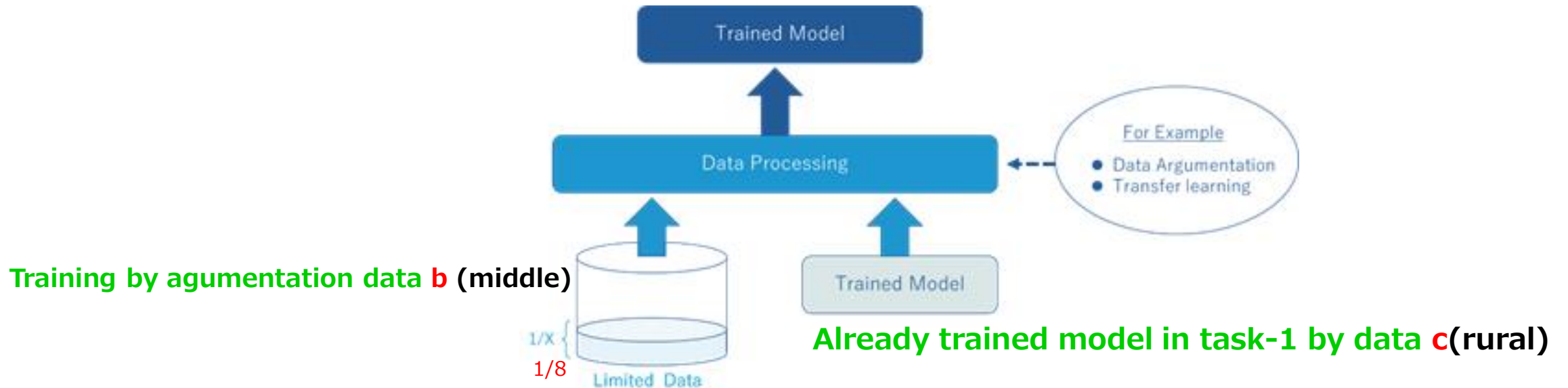
◆ **Trained model by c (urban) (Normal: 292,395; Abnormal: 3,204) ＋ Training by b (middle)(Nomal:36,668; Abnormal: 412)**

■ **Test by b (middle) :   (Normal: 73,129; Abnormal: 812)**

	Precision	Recall	F-1
Xgboost	**87%**	**100%**	**93%**

Confusion Matrix	73,005	124(FP)
	0	812(TP)

SHAP for Feature analysis

障害原因分析

(SHapley Additive exPlanations)

Machine learning based on tree models is comprehensive. When tree based models are applied in applications, we expect models can be interpretable, which mea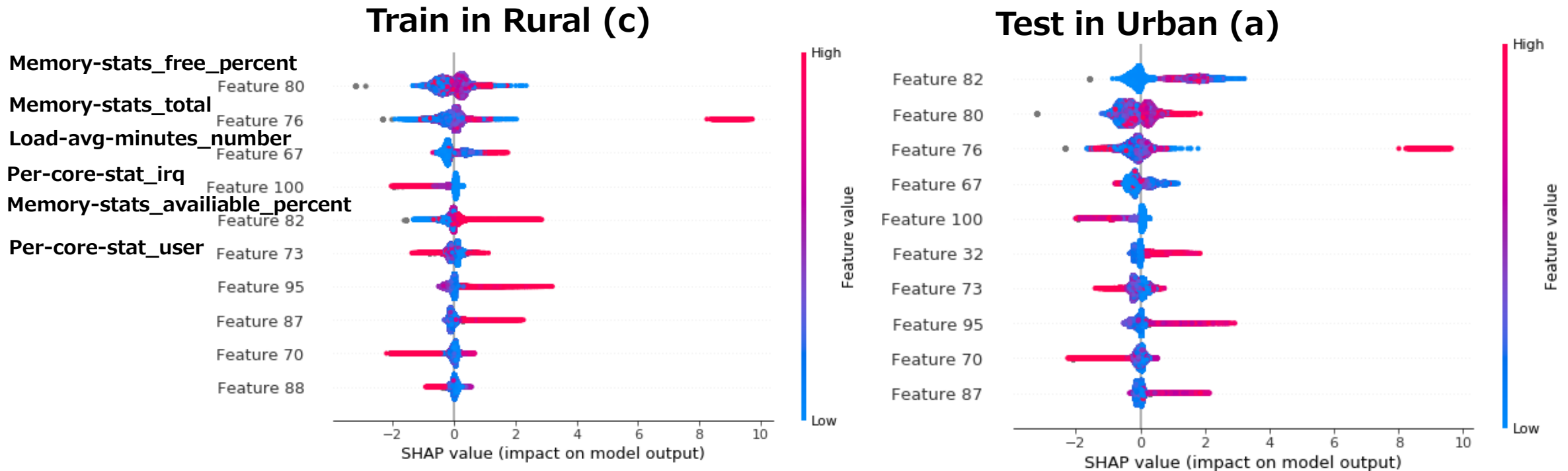ns we can understand how the model uses input features to make predictions. ShapTreeExplainer bridges theory to practice based on classic game theoretic Shapley values. It makes possible to evaluate features impact when network failure happened.

Train in Rural (c)

Memory-stats_free_percent
Memory-stats_total
Load-avg-minutes_number
Per-core-stat_irq
Memory-stats_availiable_percent
Per-core-stat_user

Test in Urban (a)

By calculate the train or test data's SHAP values, we can know how the features influence the predictions results. So when the failure is detected, we can infer the most important failure reasons by SHAP value rank.

Improvement proposal 改善点など

◆ Data clustering or mapping is helpful to confirm the feature's effectiveness

◆ More deep analysis : how different information (CPU load info, traffic info, five-g session info) is related failure by training AI models respectively.

◆ How to choose features should be investigated more deeply.

◆ Comparing neural network models with tree models for more flexible deployment.