ML5G-PS-005: Network failure prediction on CNFs 5GC with Linux eBPF

Regression-based Practical Network Failure Prediction on 5G Core Network Using AutoML

Team MLAB-NFP
Member: Jiwon Lee, Bojian Du, Kentaro Matsuura, and Ryoma Kondo
Morikawa Narusue Laboratory, the University of Tokyo

Background

Challenge

How early and accurately (f1 score ≥ 0.9) the future network failures can be predicted using 3 types of metrics: Basic (cAdvisor), Fine-grained (eBPF), 5G metric

- Target value for prediction: the number of registration failure at 10 min

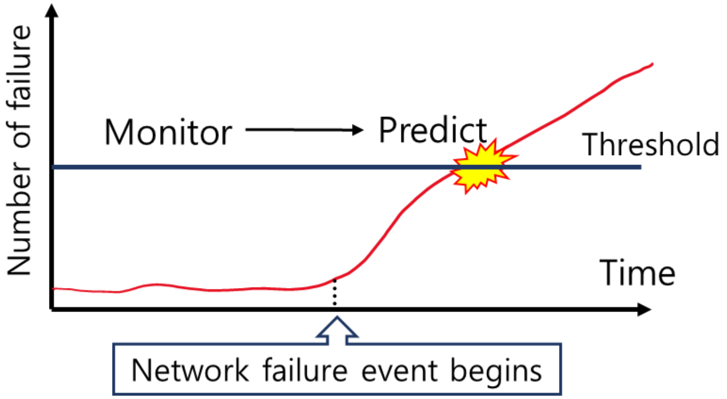
Task

Task 1

- **Use all (3326) metrics** for prediction

Task 2

- **Select a part of metrics** for prediction



Dataset

- 10 seconds interval x 70 rows per cycle
- Normal cycle (75%) / Abnormal cycle (25%)
- Failure event starts at 1.5 min (90 sec)

Task1: Objective

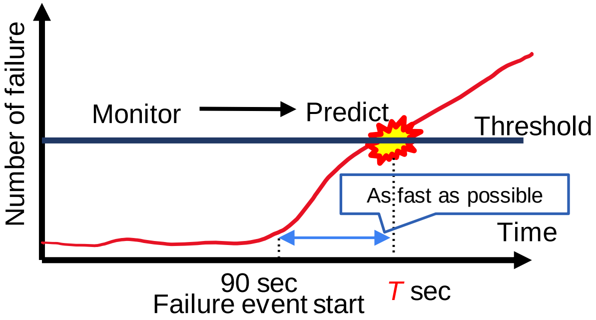
Our Concept

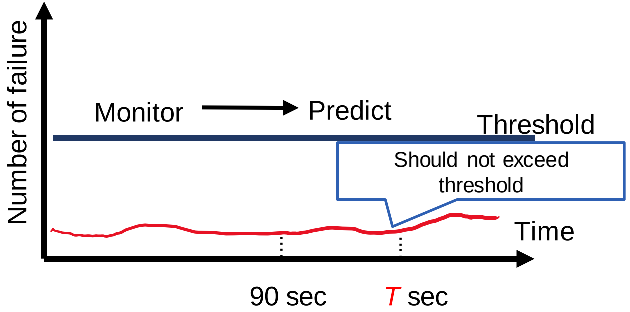
In practice, the start time of failure events should be **unknown**

- It is **NOT appropriate** to predict the condition of cycles **only at a specific time *T***
  - ➞ The start time of failure events can be later than *T*
  - ➞ Can't determine the cycle label when model raise alarm after *T*

We aim to provide a method which can apply failure prediction **for all data by time order** up to 600sec in every cycle

- For abnormal cycles, the prediction values **should exceed threshold as early as possible**
- For normal cycles, the prediction values **should not exceed threshold for all data up to 600sec**





Task1: Methodology Summary

Model

AutoGluon-Tabular (**regression model with stacking Ensemble**)

Training Data Tuning

- Normal cycles:
  Downsampling for the whole cycle
- Abnormal cycles:
  Using the data at early stage of failure only

Output Tuning

Number of registration failures at 10min for the cycle corresponding to every input

Threshold Tuning

Maximizing the margin between the prediction result of normal and abnormal cycles

Task1: Methodology Details

Model

AutoGluon-Tabular[1]

An open-source AutoML (Automated Machine Learning) framework

Strengths

- **Ensemble** multiple models and **stack** them in multiple layer
- Can robustly take raw data and deliver **high-quality predictions without any user input**
  - Beat 99% of the participating data scientists in Kaggle competitions [1]

Model training setting

- problem_type='regression'
- presets='good_quality'
- time_limit=14400 [sec]

Auto selected submodels

- LightGBMLarge
- LightGBM
- LightGBMXT
- CatBoost
- XGBoost
- Random forest
- ExtraTrees

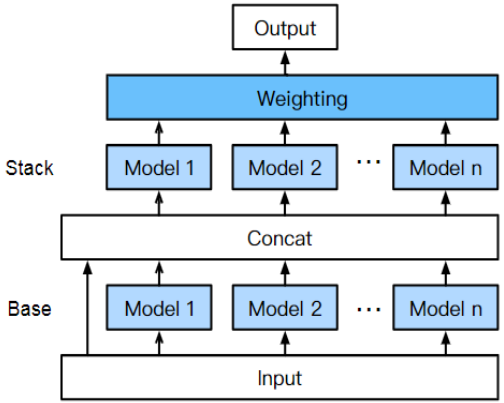


Fig. AutoGluon's multi-layer stacking strategy

[1] Erickson N, Mueller J, Shirkov A, et al. Autogluon-tabular: Robust and accurate automl for structured data[J]. arXiv preprint arXiv:2003.06505, 2020.

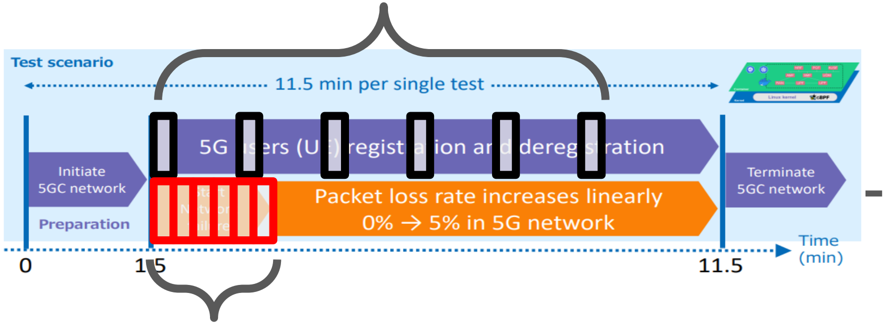
Task1: Methodology Details

Training Data Tuning Different strategies for normal and abnormal cycles

Normal cycles (450 cycles × 6 rows)

- **Downsampling for the whole cycle**
  - Constant interval of 100 sec
  - i.e. data with row index [0,10,20,30,40,50]

- **Purpose**
  - **Reduce false alarm after the prediction time** by learning the data from the whole cycle
  - **Avoid data imbalance** in training phase



Abnormal cycles (150 cycles × 6 rows)

- **Using the data at early stage of failure only**
  - i.e. data with row index [10,11,12,13,14,15]

- **Purpose**
  - **Achieve early prediction** by learn the indication of abnormal at the early stage of failure event only

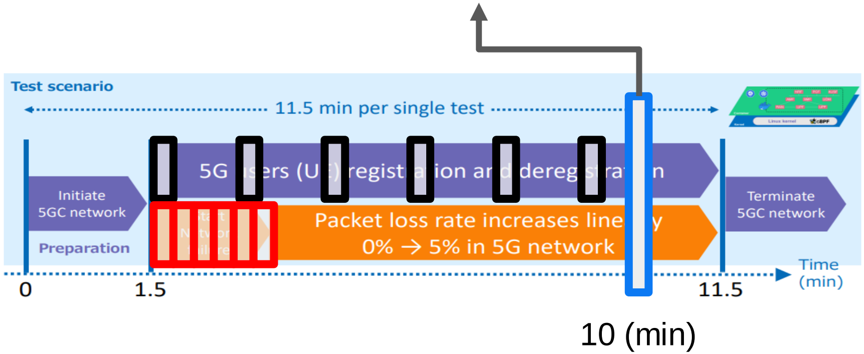
Task1: Methodology Details

Output Tuning

Number of registration failures at 10min for the cycle corresponding to every input

Output data for each input

- **Number of registration failures at 10min,** instead of at current time, **of the same cycle as the input**
  - i.e., 6 same value for 6 inputs from the same cycle



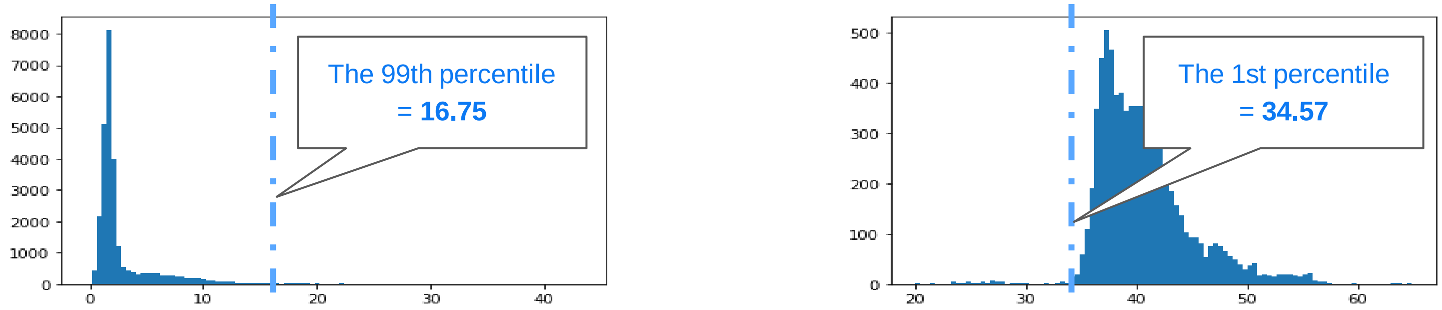
- **Purpose**
  - Let the model to correctly predict number of registration failures at 10min **at any time** in a cycle
    - Because the number of registration failures at the current time of input is not concerned in this task

Task1: Methodology Details

Threshold Tuning

Maximizing the margin between the prediction result of normal and abnormal cycles

- **Based on the prediction of training dataset (600 cycles × 60 rows)**



Histograms of **the predicted number of registration failures at 10min** in training cycles

a) Normal cycles b) Abnormal cycles (after the start of failure events)

Threshold

- The **mean value** of the 99th percentile of predicted failures in normal cycles (16.75) and the 1st percentile of predicted failures in abnormal cycles (34.57): **25.66**
  - **The percentiles are used for excluding 1% outliers**

Task1: Result Summary

Prediction time

120sec

F1 score

At 120sec

Precision: 0.866
Recall: 0.947

0.904

Up to 600sec

Precision: 0.834
Recall: 1.000

0.904

- **Achieve 100% detection rate**
- **Keep a low false alarm rate of 7.1%**

Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	71	4
Actual Negative	11	214

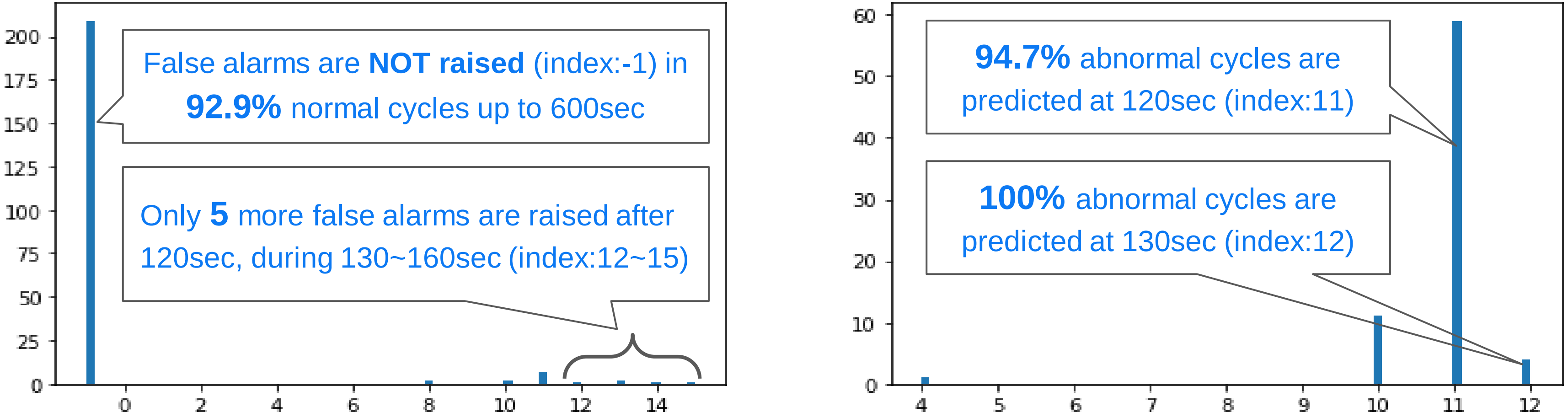
	Predicted Positive	Predicted Negative
Actual Positive	75	0
Actual Negative	16	209

Task1: Result Details

Our model not only achieves a high F1 score (0.904) at the prediction time (120sec), but also keeps high score (0.904) after that time up to 600sec

Confusion matrix before / after 120sec

F1:0.904 / 0.904		Predicted	
		Positive	Negative
Actual	Positive	**71 / 75**	**4 / 0**
	Negative	**11 / 16**	**214 / 209**



Histograms show **the row index (0~59)** of **the first time the prediction value exceeds threshold** on test dataset

a) Normal cycles b) Abnormal cycles

Background

Challenge

How early and accurately (f1 score ≥ 0.9) the future network failures can be predicted using 3 types of metrics: Basic (cAdvisor), Fine-grained (eBPF), 5G metric

- The target value for prediction: the number of registration failure at 10 min

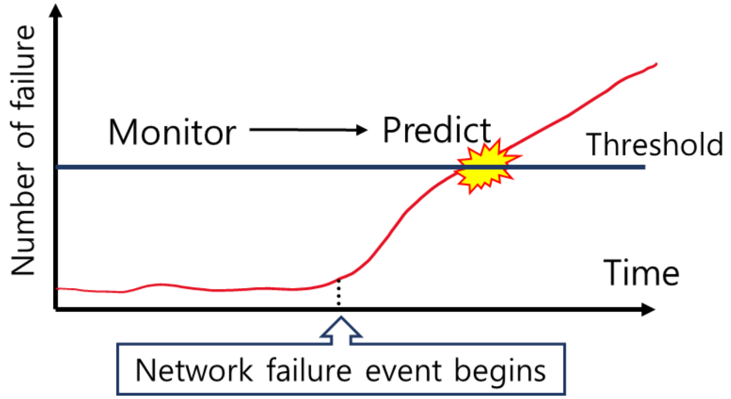
Task

Task 1

- **Use all (3326) metrics** for prediction

Task 2

- **Select a part of metrics** for prediction



Dataset

- 10 seconds interval x 70 rows per cycle
- Normal cycle (75%) / Abnormal cycle (25%)
- Failure event starts at 1.5 min (90 sec)

Task2: Feature selection Details

Random Forest Feature Importance of the Model in Task1

※ Due to the limitation of computation resource, we calculated feature importance twice by randomly selecting 400 rows from training data (keeping normal:abnormal =3:1) for each time
※ Features whose importance are lower than "**index**" are considered less important

- Features referring to UDM-TCP RTT and cAdvisor-container memory RSS tend to be important

➞ 86 columns referring to UDM-TCP RTT and 10 columns referring to cAdvisor-container memory RSS are selected

	importance
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.hist.bins._64_127.count	1.076865
smf.smf.app.cadvisor.container_memory_rss	0.931215
ausf.ausf.app.cadvisor.container_memory_rss	0.757510
amf.amf.app.cadvisor.container_memory_rss	0.749704
upf.upf3.app.cadvisor.container_memory_rss	0.593211
amf.amf.infra.tcpwin._192.168.13.80_192.168.13.82.snd_cwnd.hist.bins._4_7.count	0.559771
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.stat.avg	0.444564
udm.udm.app.cadvisor.container_memory_rss	0.405376
amf.amf.infra.tcpwin._192.168.13.80_192.168.13.82.snd_cwnd.hist.stat.min	0.351711
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.82.stat.avg	0.306957
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.stat.max	0.261798
upf.upf1.infra.runqlat.hist.bins._256_511.count	0.250903
amf.amf.infra.tcpwin._192.168.13.80_192.168.13.72.snd_cwnd.hist.stat.min	0.243320
nrf.nrf.app.cadvisor.container_memory_working_set_bytes	0.176476
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.82.stat.max	0.161863
index	0.149615

	importance
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.hist.bins._64_127.count	1.274811
smf.smf.app.cadvisor.container_memory_rss	1.253748
amf.amf.app.cadvisor.container_memory_rss	1.195922
ausf.ausf.app.cadvisor.container_memory_rss	1.081871
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.stat.avg	0.653811
udm.udm.app.cadvisor.container_memory_rss	0.536127
upf.upf3.app.cadvisor.container_memory_rss	0.515652
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.82.stat.avg	0.500319
amf.amf.infra.tcpwin._192.168.13.80_192.168.13.82.snd_cwnd.hist.bins._4_7.count	0.386906
upf.upf2.app.cadvisor.container_memory_rss	0.378289
amf.amf.infra.tcprtt._192.168.13.80_192.168.13.70.stat.avg	0.313949
amf.amf.infra.tcpwin._192.168.13.80_192.168.13.82.snd_cwnd.hist.stat.min	0.290665
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.82.stat.max	0.286260
upf.upf1.app.cadvisor.container_memory_rss	0.265179
igw.igw2.app.cadvisor.container_memory_working_set_bytes	0.250817
upf.upf1.infra.runqlat.hist.bins._256_511.count	0.244296
nrf.nrf.app.cadvisor.container_memory_working_set_bytes	0.224415
igw.igw2.infra.tcprtt._192.168.14.45_192.168.12.1.hist.bins._8192_16383.count	0.204181
udm.udm.infra.tcprtt._192.168.13.70_192.168.13.80.stat.max	0.192238
smf.smf.infra.tcprtt._192.168.13.82_192.168.13.70.stat.min	0.188466
amf.amf.infra.tcprtt._192.168.13.80_192.168.13.70.stat.max	0.178001
upf.upf1.app.cadvisor.devices._/dev/sdb2.container_fs_writes_merged	0.170604
upf.upf3.infra.runqlat.hist.bins._128_255.count	0.167914
index	0.167500

Task2: Feature selection Details

Thesis on Important Features

Definition

UDM TCP RTT

- UDM : Unified Data Management
- RTT : Round-Trip Time
  - Time for a signal/data to be sent to the other end of the communication until a response is returned.

Hypothesis

- UDM – subscriber management, user identification processing
  → **related to registration**
- RTT – depends on physical distance, number of devices, processing time
  → **Increase when failure occur**

Definition

Container memory RSS

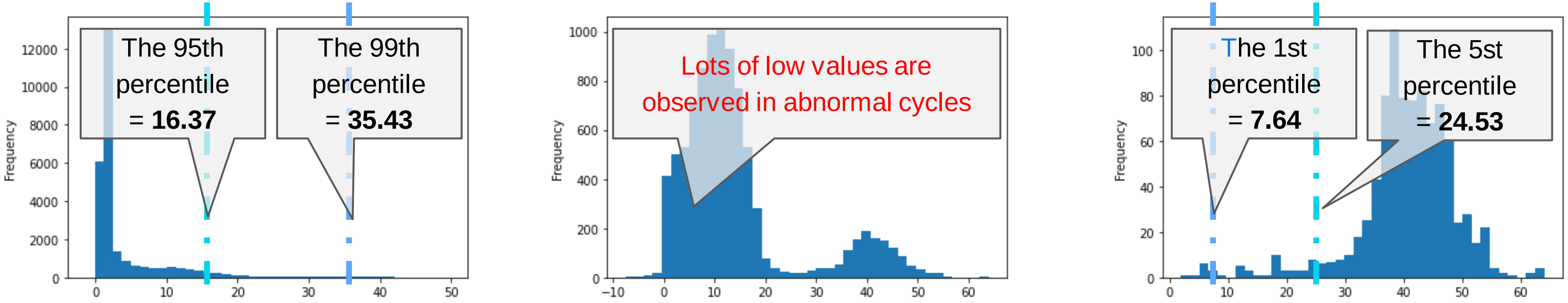
- RSS : Resident set size
  - Memory usage
  - Physical memory consumption

Hypothesis

- **Failure event will consume more physical memory than usual**

Task2: First Try

Threshold Tuning



Histograms of **the predicted number of registration failures at 10min** in training cycles

a) Normal cycles

b) Abnormal cycles (after the start of failure events, index:10~59)

c) Abnormal cycles (only the early stage of failure events, index:10~15)

Modification from Task1

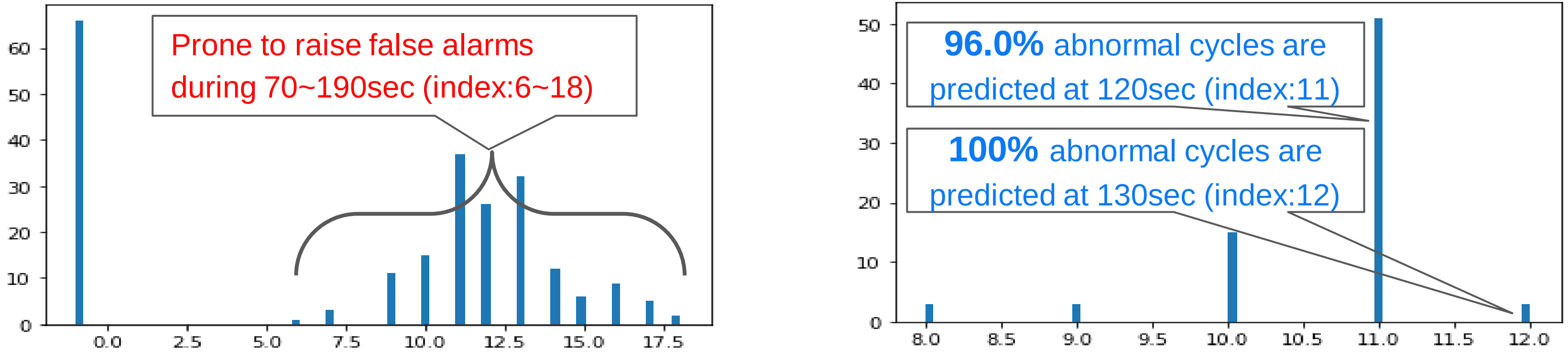
- **Use only the early stage of failure events for abnormal cycles**
  - As the prediction of the later stage of failure events are not accurate (too low)
    - It won’t affect prediction performance, since the prediction values should exceed threshold at the early stage
- **Exclude 5% outliers instead of 1% outliers in the calculation of threshold**
  - As the distribution of the prediction values becomes more dispersed for both normal and abnormal cycles

Task2: First Try

The new model keeps a high prediction rate from 120sec however, it tends to raise more false alarms during 70~190sec

Confusion matrix before / after 120sec

F1:0.673 / 0.485		Predicted	
		Positive	Negative
Actual	Positive	**72 / 75**	**3 / 0**
	Negative	**67 / 159**	**158 / 66**



Histograms of **the row index (0~59)** of **the first time the prediction value exceeds threshold** on test dataset

a) Normal cycles b) Abnormal cycles

Disscussion

- Features referring to **UDM-TCP RTT** and **cAdvisor-container memory RSS** **shows the potential in predicting abnormal cycles**
- False alarms in normal cycles are tend to occur in **the same time of the training periods of abnormal cycles**, as the beginning time of failure events are almost the same in this dataset
  - **Adding more training samples** or **providing more variety in failure event** might help reduce this problem

Task2: Methodology Details

Modification of training data

- Separate training data (600 cycles) by 500:100 ratio and create validation data
  - Use training data (500 cycles) for training data tuning
    - Double the rows of training data
  - Use validation data (100 cycles) for threshold tuning
    - Find the value that maximize f1 score

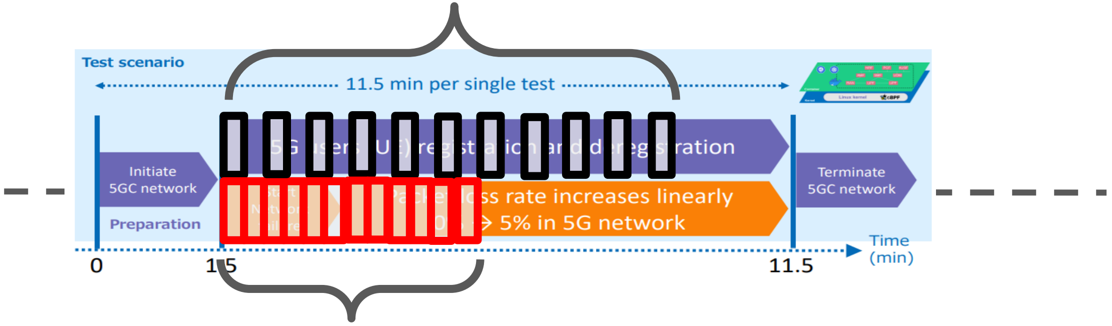
Task2: Methodology Details

Training Data Tuning Different strategies for normal and abnormal cycles

Normal cycles (375 cycles × 12 rows)

- **Downsampling for the whole cycle**
  - Constant interval of 50 sec
  - i.e. data with row index [0,5,10,15,20,25,30,35,40,45,50,55]

- **Purpose**
  - Improve prediction on normal cycles by adding more training samples



Abnormal cycles (125 cycles × 12 rows)

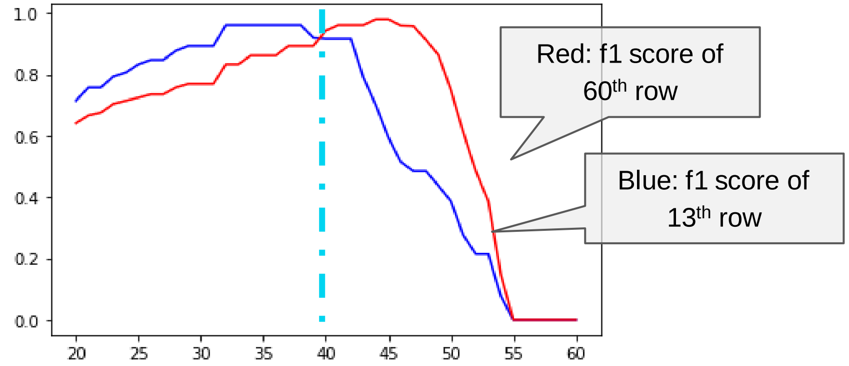
- **Using the data at early stage of failure only**
  - i.e. data with row index [10,11,12,13,14,15,16,17,18,19,20,21]

Task2: Methodology Details

Threshold Tuning

Crossing point of the f1 score of 13th row and 60th row

- **Based on the validation dataset of training data (100 cycles × 70 rows)**



F1 scores of threshold from 20 to 60 at 13th and 60th row

- **Change the threshold from 20 ~ 60 for both 13th row and 60th row**
  - Check the point where the two graphs cross each other
  - 13th row: threshold = 40, f1 score = 0.912
  - 60th row: threshold = 40, f1 score = 0.943

Threshold

- The **crossing point** of the f1 score of the 60th row (red) and the f1 score of the 13th row (blue): **40.00**
  - **The crossing point used for setting threshold to predict with high accuracy throughout the whole cycle**

Task2: Result Summary

Selected Features

~~3326~~ → 96 columns

Features referring to UDM-TCP RTT (86 columns)
and cAdvisor-container memory RSS (10 columns)

Prediction time

96.0% abnormal cycles are predicted at

130sec

F1 score

At 130sec

Precision: 0.907
Recall: 0.907

0.907

Up to 600sec

Precision: 0.915
Recall: 1.000

0.955

Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	68	7
Actual Negative	7	218

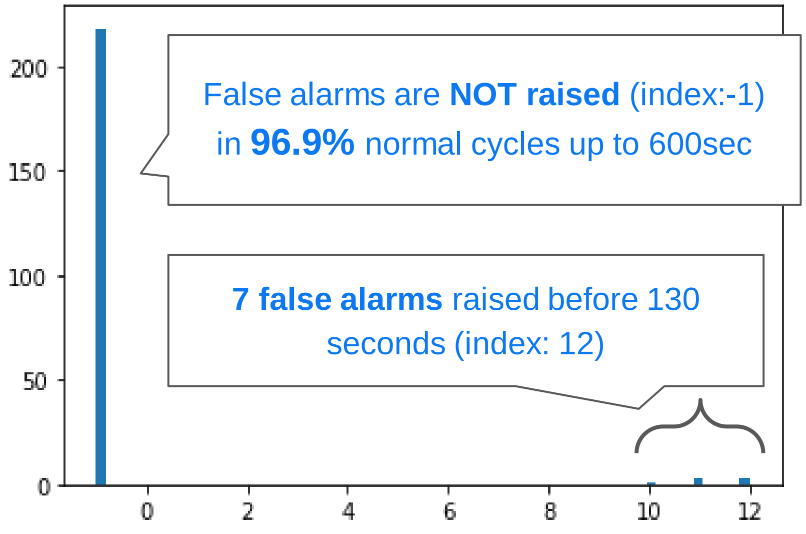
	Predicted Positive	Predicted Negative
Actual Positive	75	0
Actual Negative	7	218

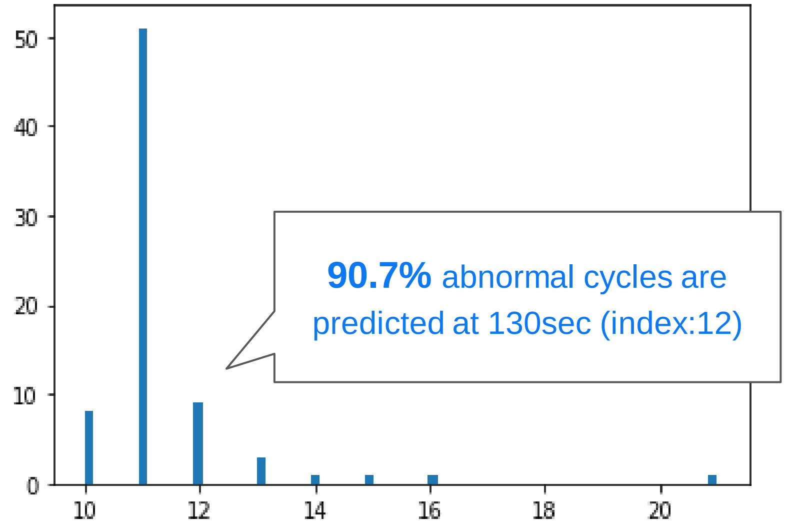
Task2: Result Details

Our model achieved a high F1 score both at the prediction time 130sec (0.907), and after that time up to 600sec (0.970).

Confusion matrix before / after 130sec

F1:0.907 / 0.955		Predicted	
		Positive	Negative
Actual	Positive	**68 / 75**	**7 / 0**
	Negative	**7 / 7**	**218 / 218**





Histograms of **the row index (0~59)** of **the first time the prediction value exceeds threshold** on test dataset

a) Normal cycles

b) Abnormal cycles

Summary

Task 1

Method

- Apply prediction for all data by time up to 600 sec in every cycle
- Model: AutoGluon-Tabular
  - Training data: downsample to 6 rows per cycle
  - Output data: failures at 10min

Result

- Prediction time: **120 sec**
- F1 score:
  - At 120 sec: **0.904**
  - Up to 600 sec: **0.904**

Task 2

Method

- Feature selection: UDM-TCP RTT (86 columns) and cAdvisor-container memory RSS (10 columns)
- Model: AutoGluon-Tabular
  - Training data: downsample to 12 rows per cycle
  - Output data: failures at 10min

Result

- Prediction time: **130 sec**
- F1 score:
  - At 130 sec: **0.907**
  - Up to 600 sec: **0.955**