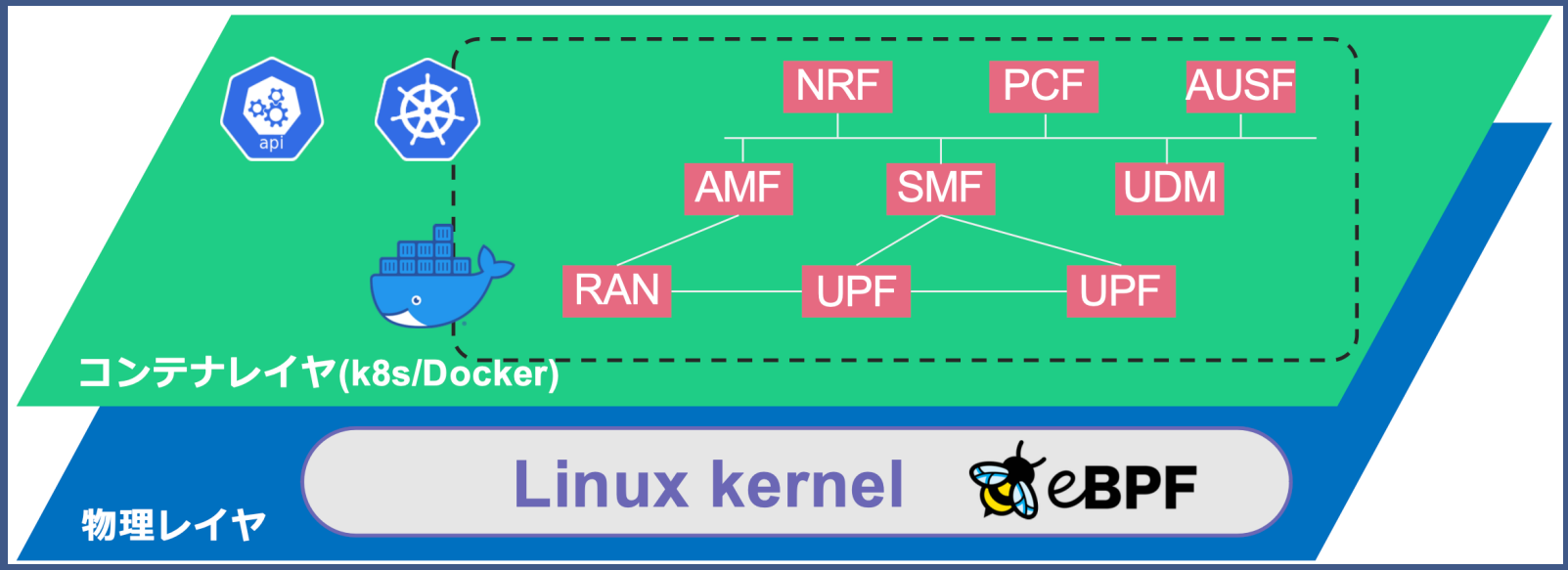
Network Failure Prediction on CNFs 5GC with Linux eBPF

UT-NakaoLab-AI

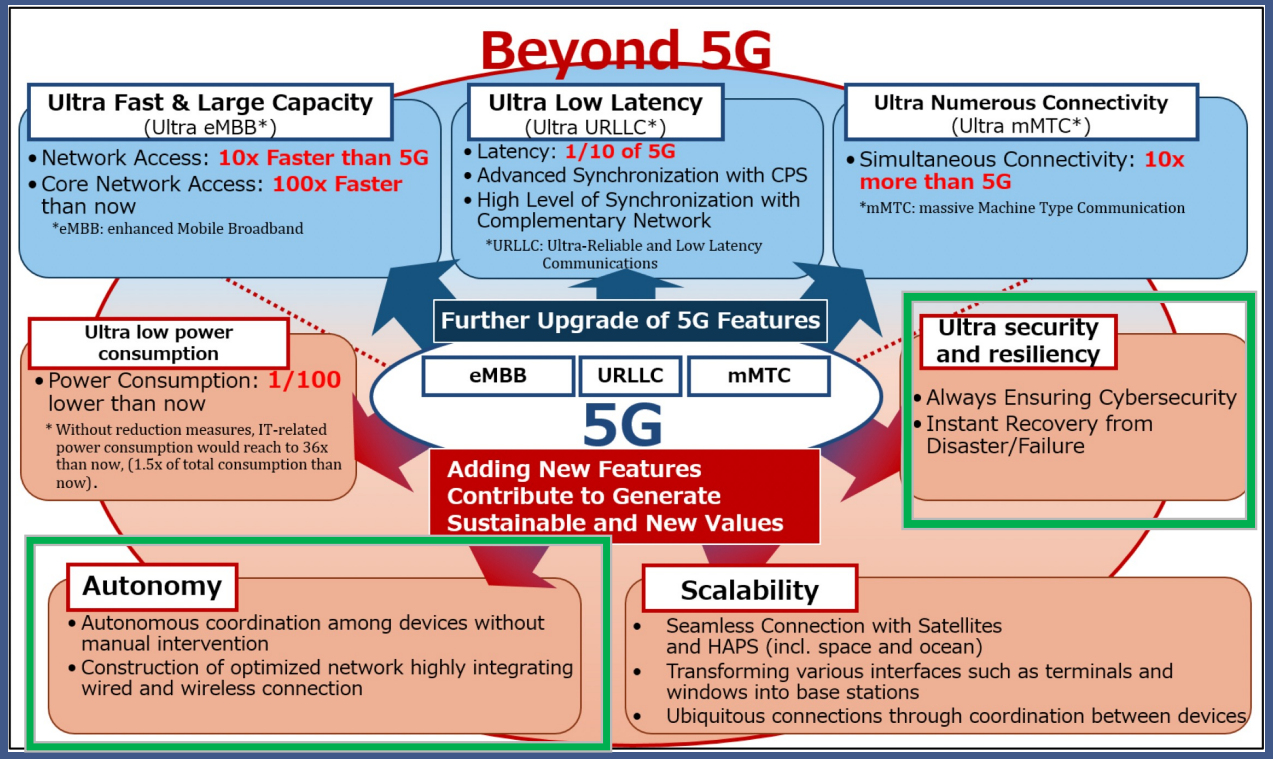
Takeru Hakii

Introduction



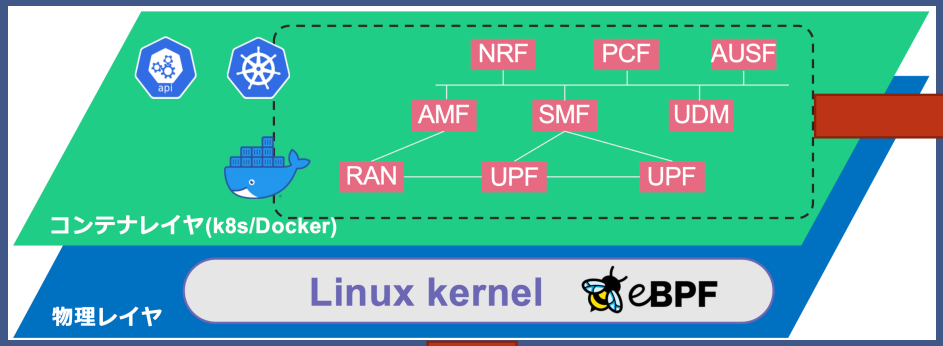
By leveraging flexible containerized network functions (NF)
Scalable networks can be rapidly deployed

Introduction



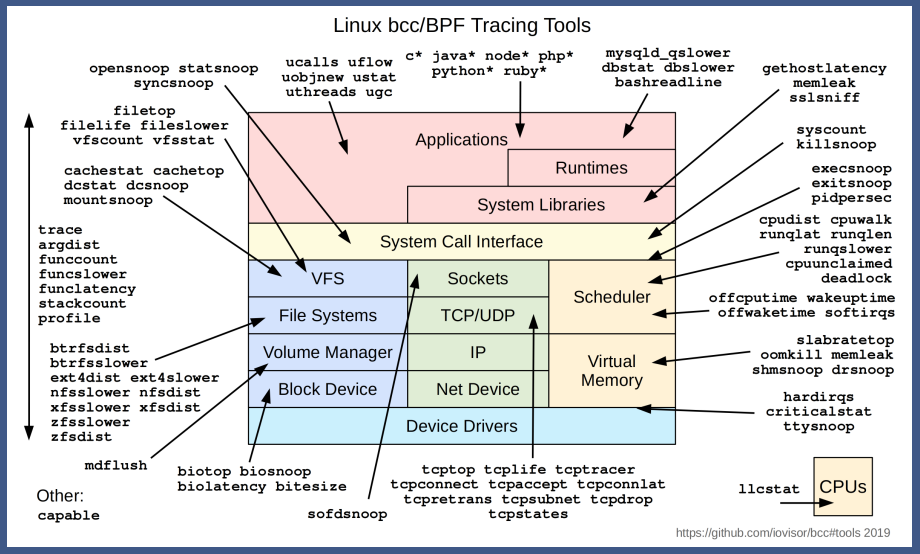
Need a system that monitors the network and automatically detects and restores failures.

Introduction

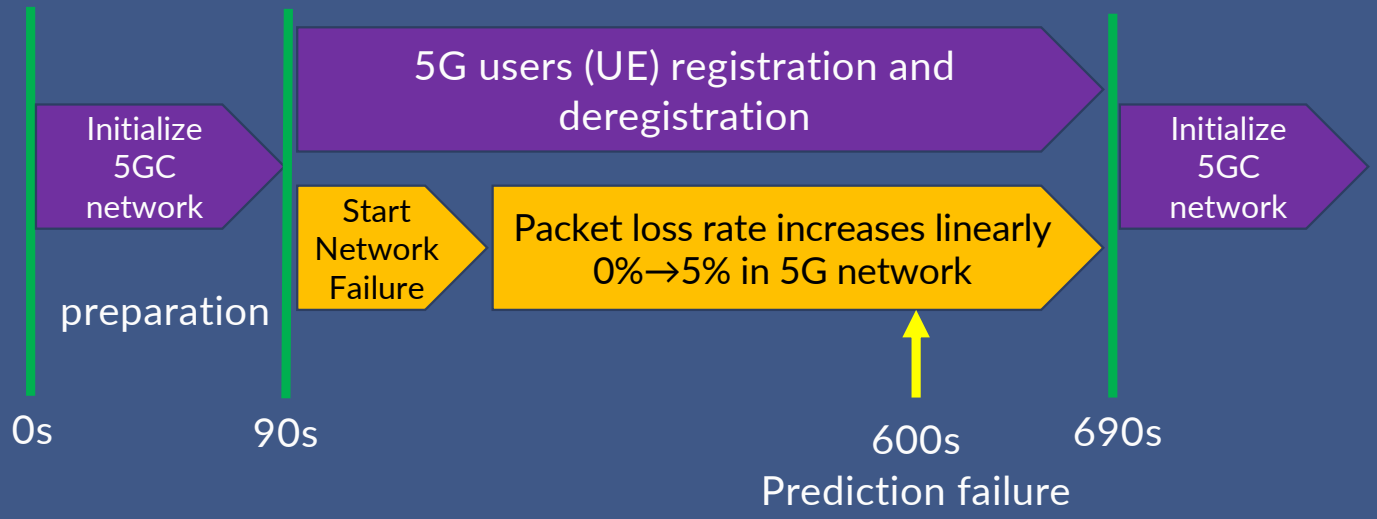


Only log data obtained from containers difficult to predict failures in advance

Using eBPF (extended Berkley Packet Filter), we get more detailed data



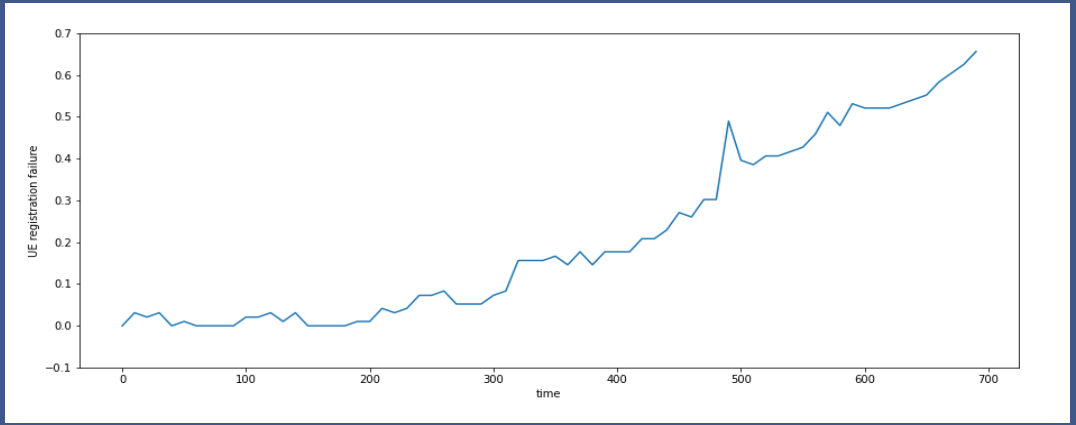
Scenario

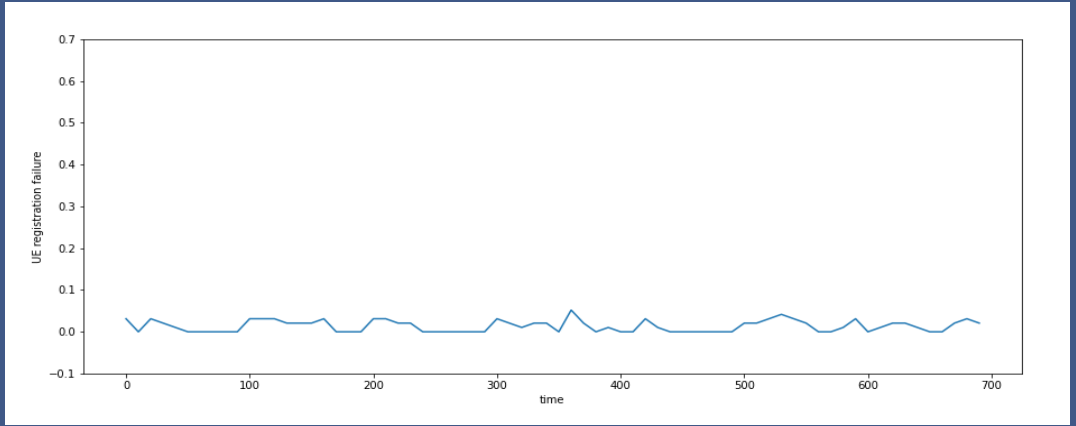


- One cycle is 690 seconds
- Logging every 10 seconds
- The first 90 seconds is a preparation period, initializing the network
- The subsequent 600 seconds (10 minutes) cause a failure, increasing the packet loss rate linearly

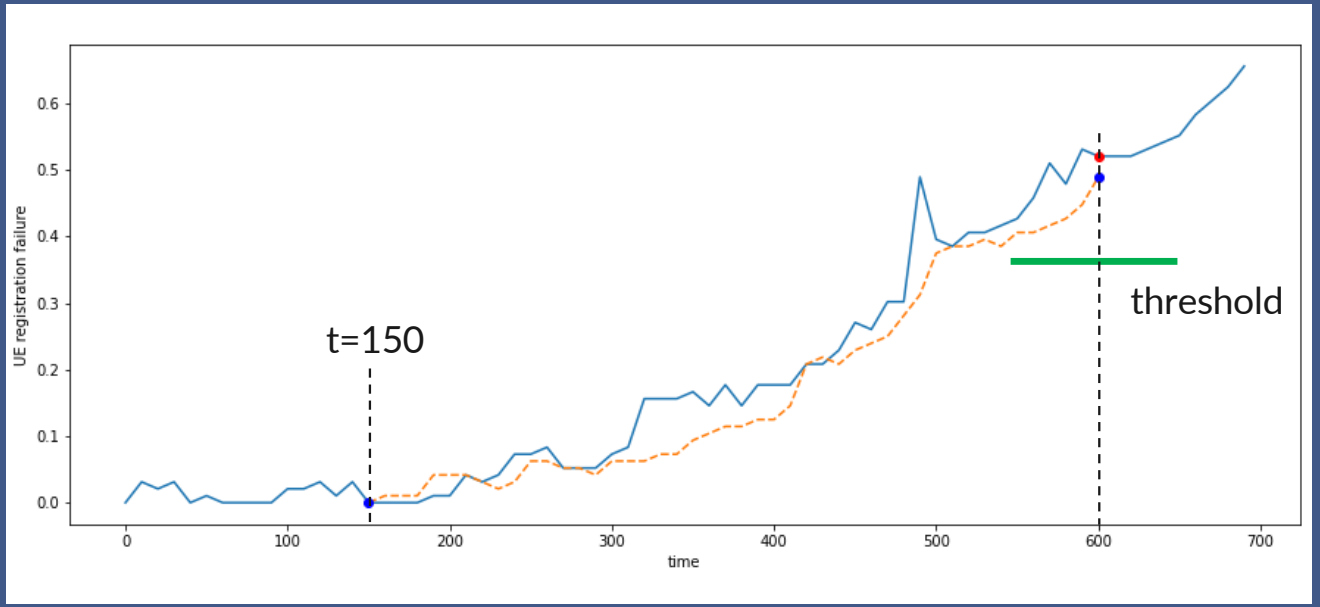
Dataset

- Train : 600 cycles
- Test : 300 cycles
- Metrics : 3326





Task1



Predicts the number of UE registration failures at 600 seconds from a certain point *t*, and determines if a failure has occurred according to a threshold value.

Make t as small as possible so that the F1 score exceeds 0.9.

Task2

```
Index(['amf.amf.app.cadvisor.container_cpu_cfs_periods',
       'amf.amf.app.cadvisor.container_cpu_cfs_throttled_periods',
       'amf.amf.app.cadvisor.container_cpu_cfs_throttled_seconds',
       'amf.amf.app.cadvisor.container_cpu_system_seconds',
       'amf.amf.app.cadvisor.container_cpu_usage_seconds',
       'amf.amf.app.cadvisor.container_cpu_user_seconds',
       'amf.amf.app.cadvisor.container_last_seen',
       'amf.amf.app.cadvisor.container_memory_cache',
       'amf.amf.app.cadvisor.container_memory_failcnt',
       'amf.amf.app.cadvisor.container_memory_failures',
       ...
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_LAST_ACK_CLOSE.usec.avg',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_LAST_ACK_CLOSE.usec.max',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_LAST_ACK_CLOSE.usec.med',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_LAST_ACK_CLOSE.usec.min',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_SYN_RECV_ESTABLISHED.count',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_SYN_RECV_ESTABLISHED.usec.avg',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_SYN_RECV_ESTABLISHED.usec.max',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_SYN_RECV_ESTABLISHED.usec.med',
       'upf.upf3.infra.tcpstates._::ffff:10.244.0.1_SYN_RECV_ESTABLISHED.usec.min',
       'upf.upf3.infra.tcptracer._192.168.13.70_connect.count'],
```

Reduce metrics as much as possible from over 3000 metrics

Method(Task1)

- Use all features (all) and remove features that are 0 for all periods (removed)
- Feature size
  - all: 3325
  - removed: 1723
- Scaling of both training and test data using minimum and maximum values of training data
- Threshold = 0.3 (28.8)
- timesteps: the input sequence size
- delay: interval between input and output

- Model: LSTM
- Loss function: MSE
- Optimizer: adam
- Epochs: 15
- Training data: 500 cycles
- Validation data: 100 cycles

Detection time *t*	timesteps	delay	Data size	Loss input
350	15	35	10000	15
400	15	40	7500	10
450	15	45	5000	5
460	14	46	5000	4
470	13	47	5000	3
480	12	48	5000	2

Result(Task1)

all	F1 score	MSE	Metrics num
250	1.00	0.00198	3325
200	1.00	0.00199	3325
150	0.91	0.00294	3325
140	0.89	0.00317	3325
130	0.80	0.00344	3325
120	0.44	0.00533	3325

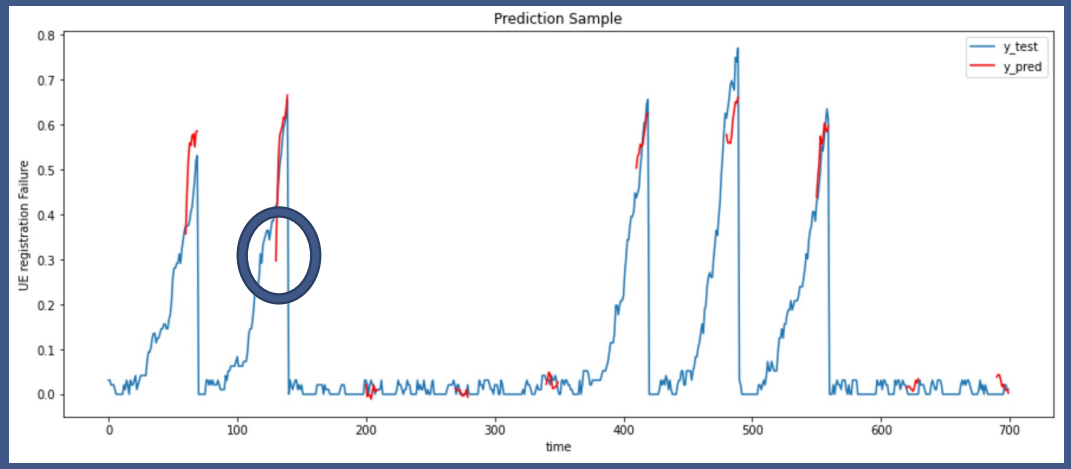
removed	F1 score	MSE	Metrics num
250	1.00	0.00196	1723
200	1.00	0.00209	1723
150	0.98	0.00268	1723
140	0.92	0.00301	1723
130	0.70	0.00498	1723
120	0.39	0.00606	1723

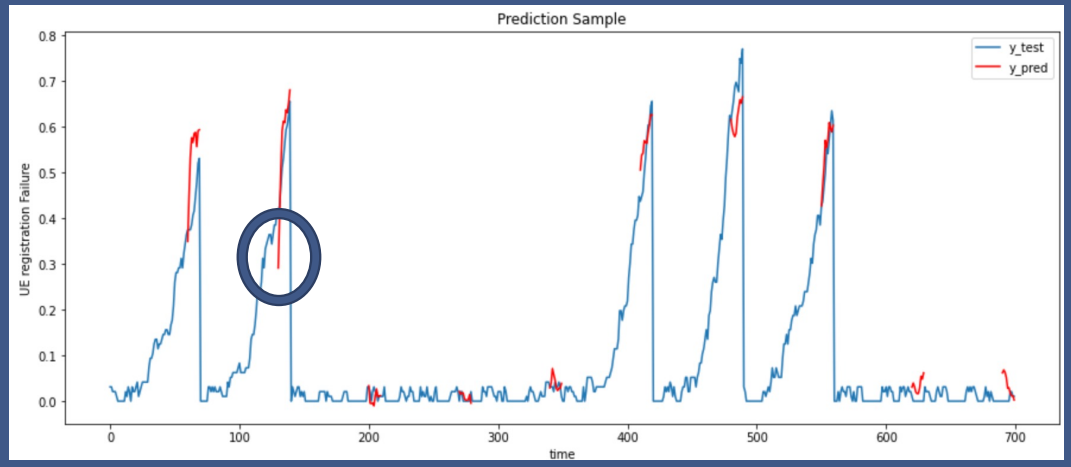
Successful prediction at detection time 150s by using all features

Successful prediction at detection time 140s by removing 0 metrics

The removed model has higher performance than the all model.

Result(Task1)

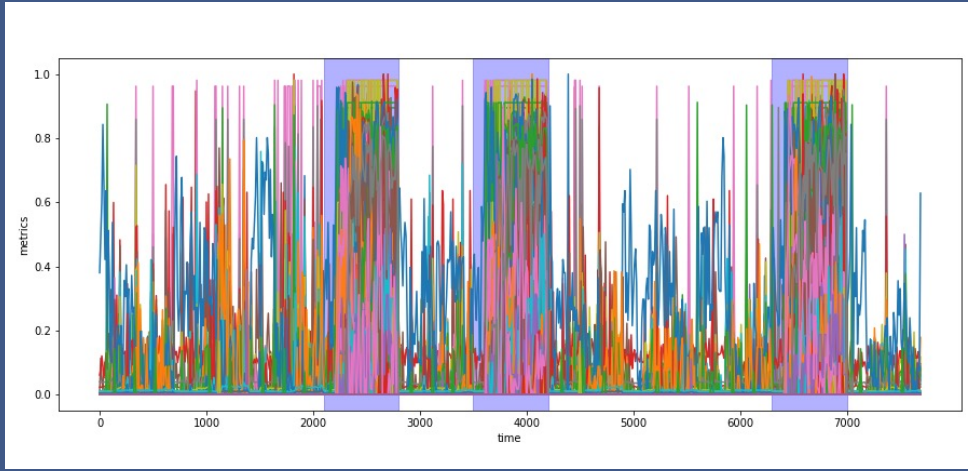


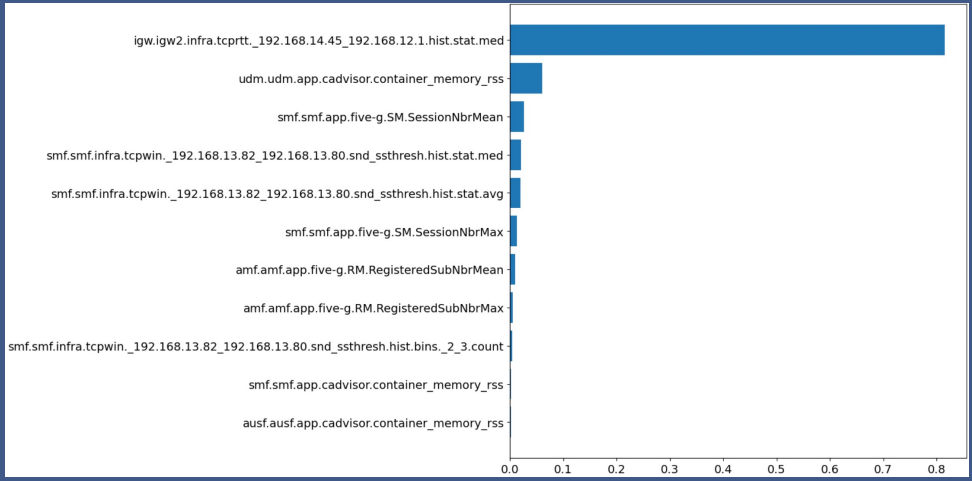


Both models may underpredict the number of UE registration failures at 600 seconds

Method(Task2)

- method1 (diff)
  - |mean(loss)|>2|mean(normal)|
  - feature size: 3326→329
- method2 (RF)
  - calculate feature importance by training random forests
  - sort features in order of importance and use for LSTM training
  - Feature size: 1723, 1500, 1000, 500, 400, 329
  - Detection time: 140s





Result(Task2 diff)

removed	F1 score	MSE	Metrics num
250	1.00	0.00196	1723
200	1.00	0.00209	1723
150	0.98	0.00268	1723
140	0.92	0.00301	1723
130	0.70	0.00498	1723
120	0.39	0.00606	1723

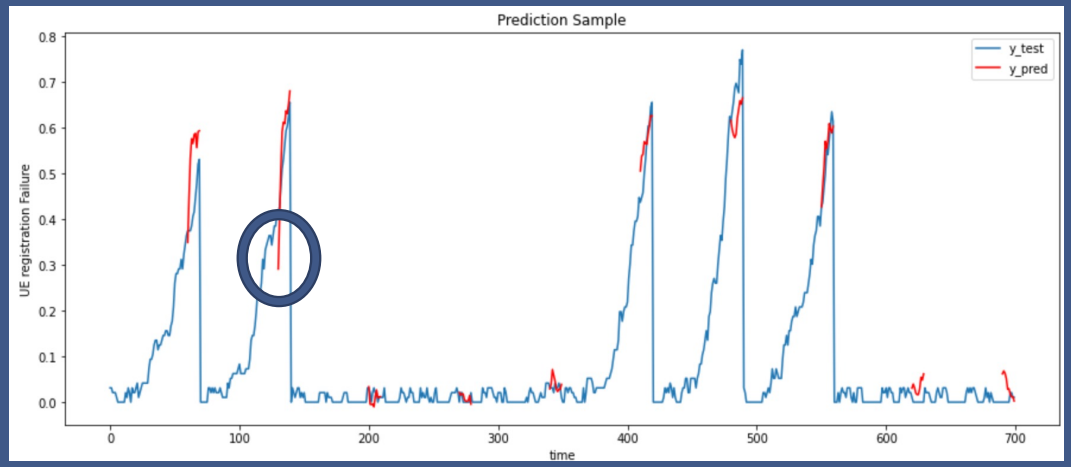
diff	F1 score	MSE	Metrics num
250	1.00	0.00249	329
200	1.00	0.00215	329
150	0.98	0.00232	329
140	0.97	0.00279	329
130	0.88	0.00449	329
120	0.50	0.00676	329

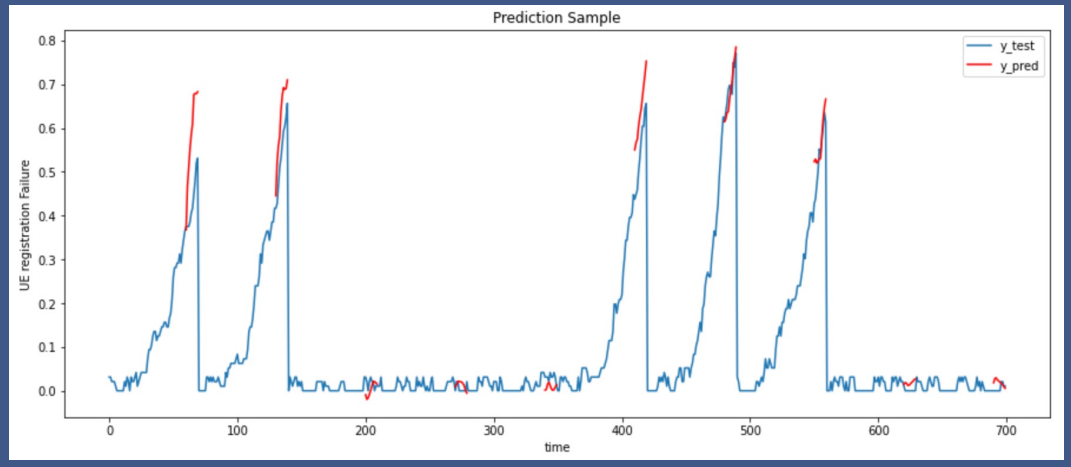
The F1 score for the diff model was 0.97 with detection time 140s.

The diff model has higher performance than the removed model.

The model with 3000 features reduced had the best performance.

Result(Task1)





The removed model may underpredict the number of UE registration failures at 600 seconds.

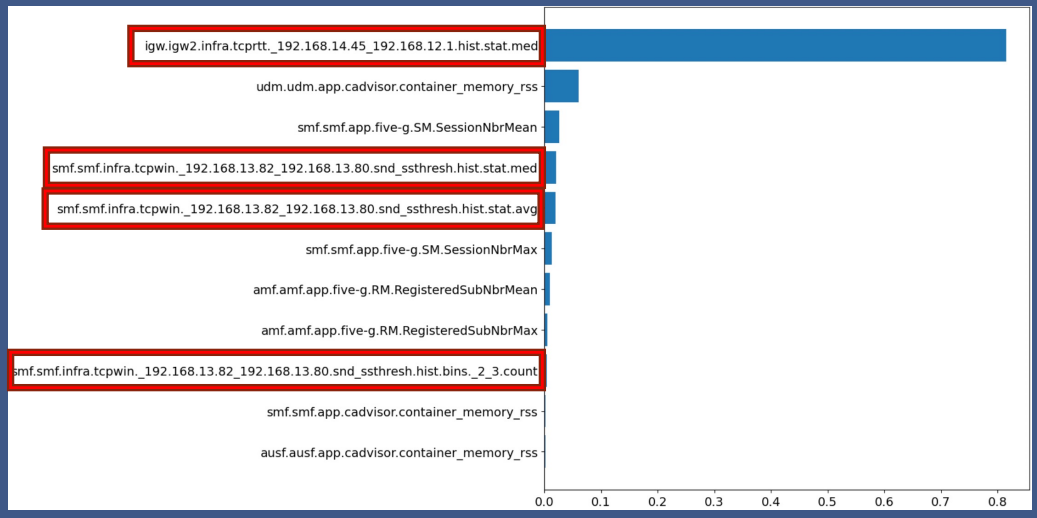
The diff model may overpredict.

Result(Task2 RF)

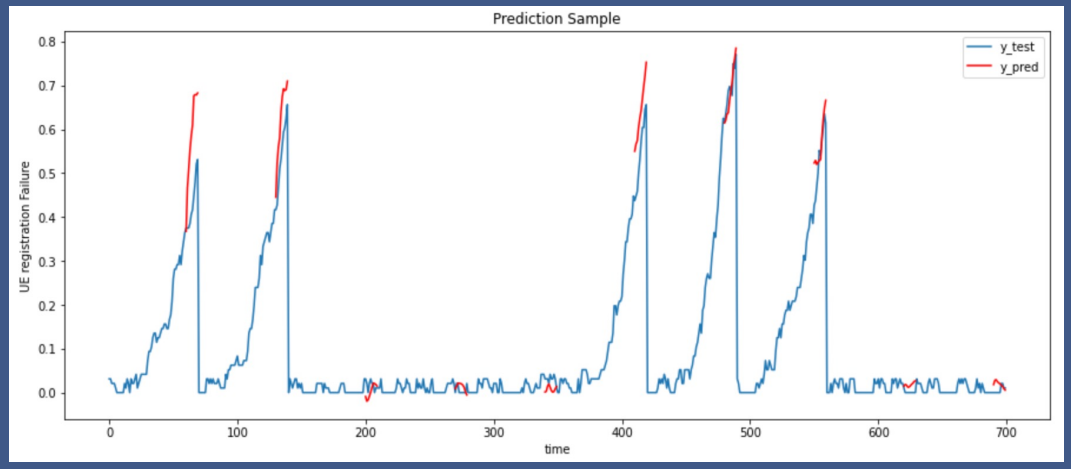
diff	F1 score	MSE	Metrics num
250	1.00	0.00249	329
200	1.00	0.00215	329
150	0.98	0.00232	329
140	0.97	0.00279	329
130	0.88	0.00449	329
120	0.50	0.00676	329

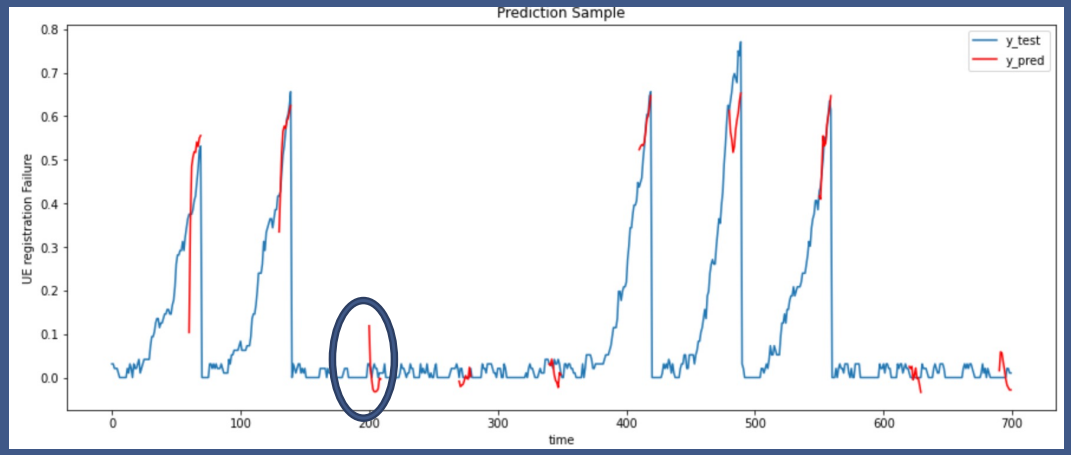
RF	F1 score	MSE	Metrics num
140	0.93	0.00444	1723
140	0.88	0.00411	1500
140	0.89	0.00326	1000
140	0.75	0.00592	500
140	0.79	0.00757	400
140	0.73	0.00626	329

TCP metrics are important in loss failure



Result(Task1)





The diff model predicts more accurately than the RF model.

The RF model does not accurately predict normal phase.

Conclusion

- Successfully predicted with a detection time of 140s
- Improved model performance by reducing the number of features
- The model performance was most improved after reducing the number of features by 3,000

Future work

- Prediction for other failure scenarios such as CPU overload
- Predictions for different user traffic characteristics
- Analysis of metrics that are important for many factors such as failure scenarios, traffic characteristics, network size, network topology etc.
- Analysis of when the model will need to be re-trained or changed

Thank you for listening

The University of Tokyo Nakao lab M1
Takeru Hakii