CSET'21

NSR
National Security Research Institute

Two ICS Security Datasets and One Anomaly Detection Contest using HAI testbed

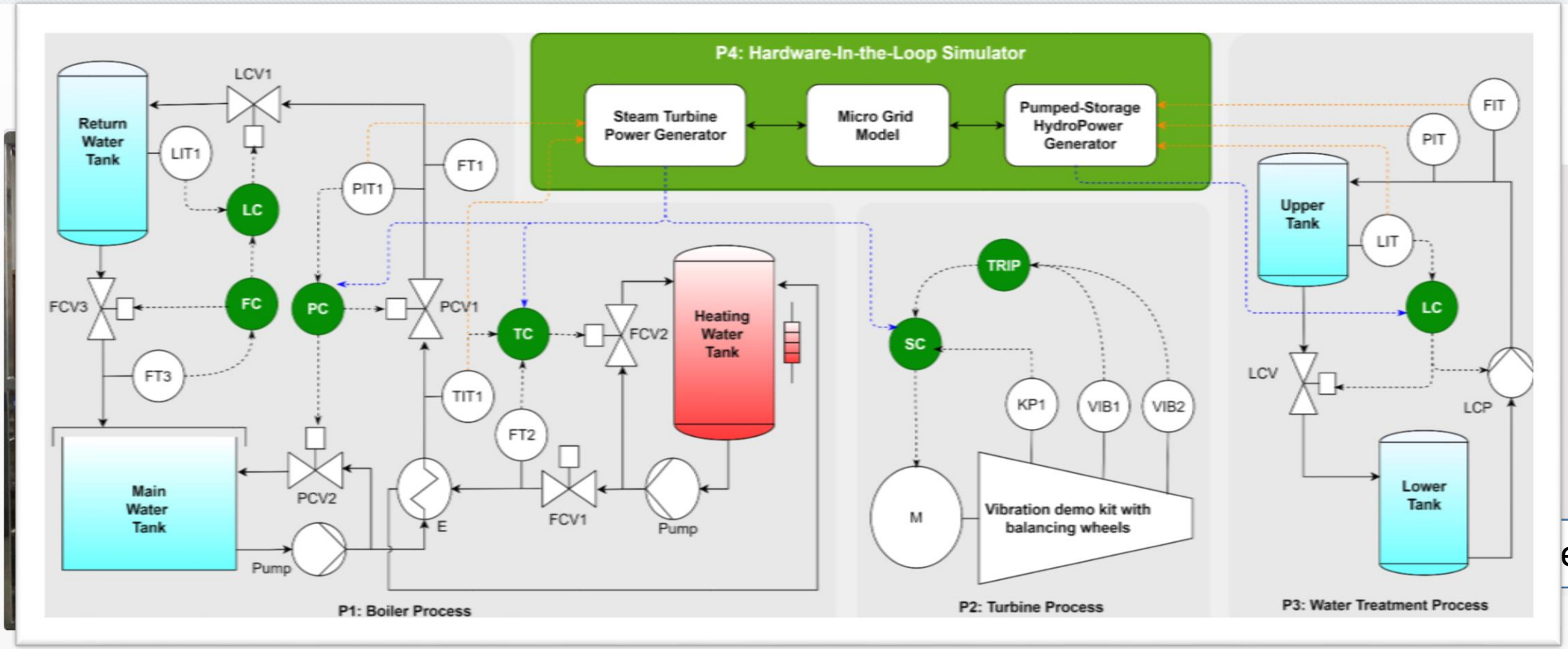
2021. 8. 9.

Hyeok-Ki Shin, Woomyo Lee, **Jeong-Han Yun**, and Byung Gil Min

CONTENTS

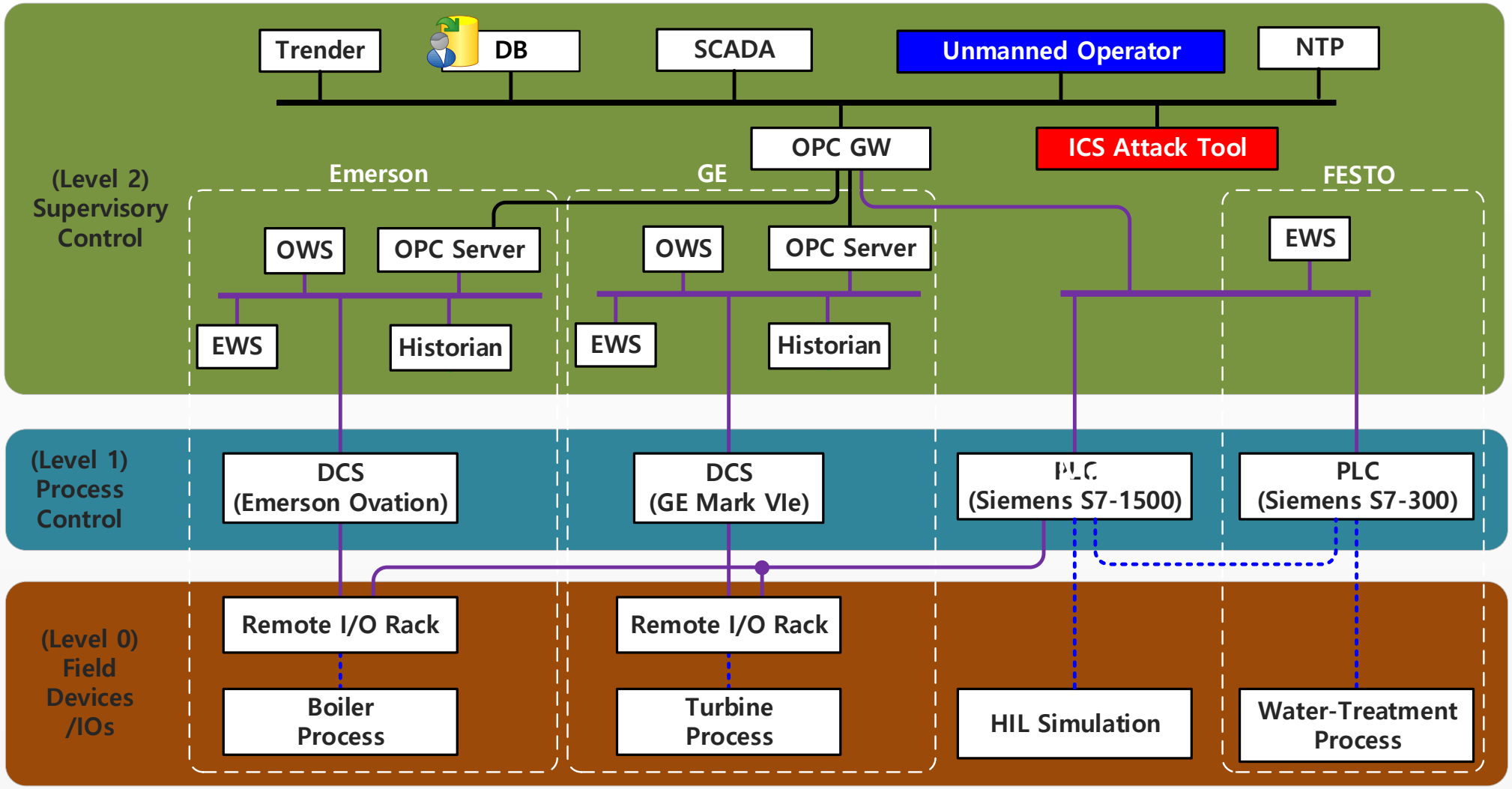
① Process Augmentation with a HIL(Hardware-In-the-Loop) simulator

- To increase the **correlation between signals**, not to get precise simulation results
- Three ICS testbeds were interconnected via a HIL simulator that simulates complex power gen. system.



HK Shin, et al, "Implementation of Programmable CPS Testbed for Anomaly Detection," CSET@USENIX Security, 2019.

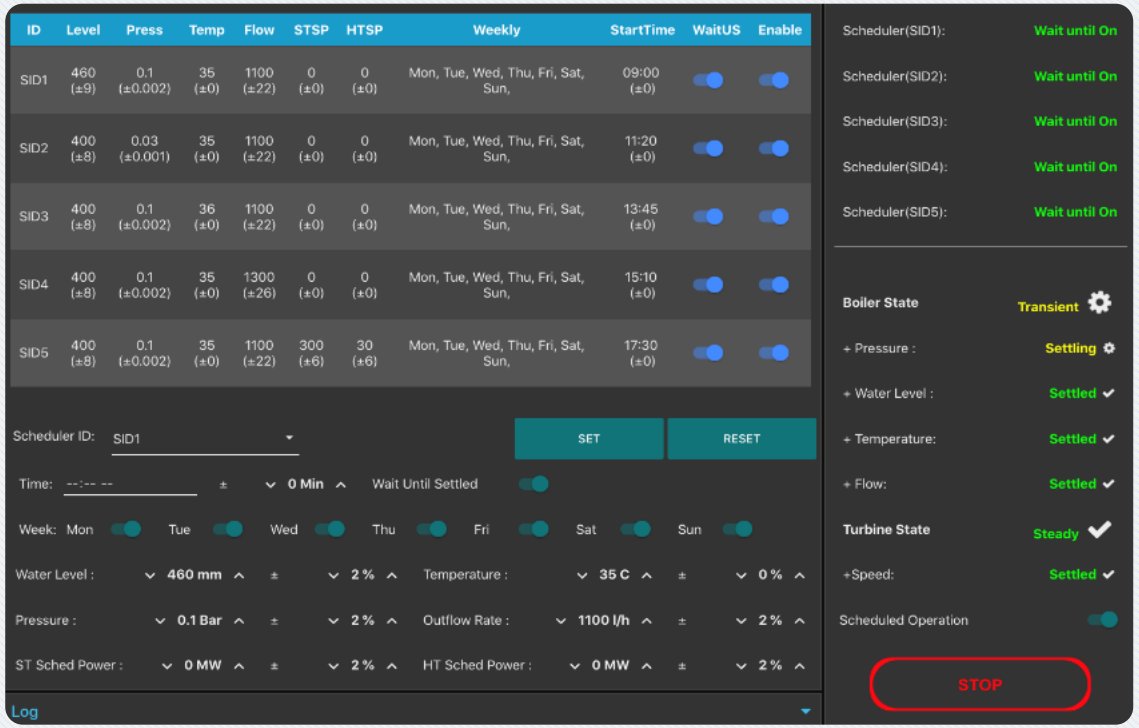
Network Structure of HAI Testbed

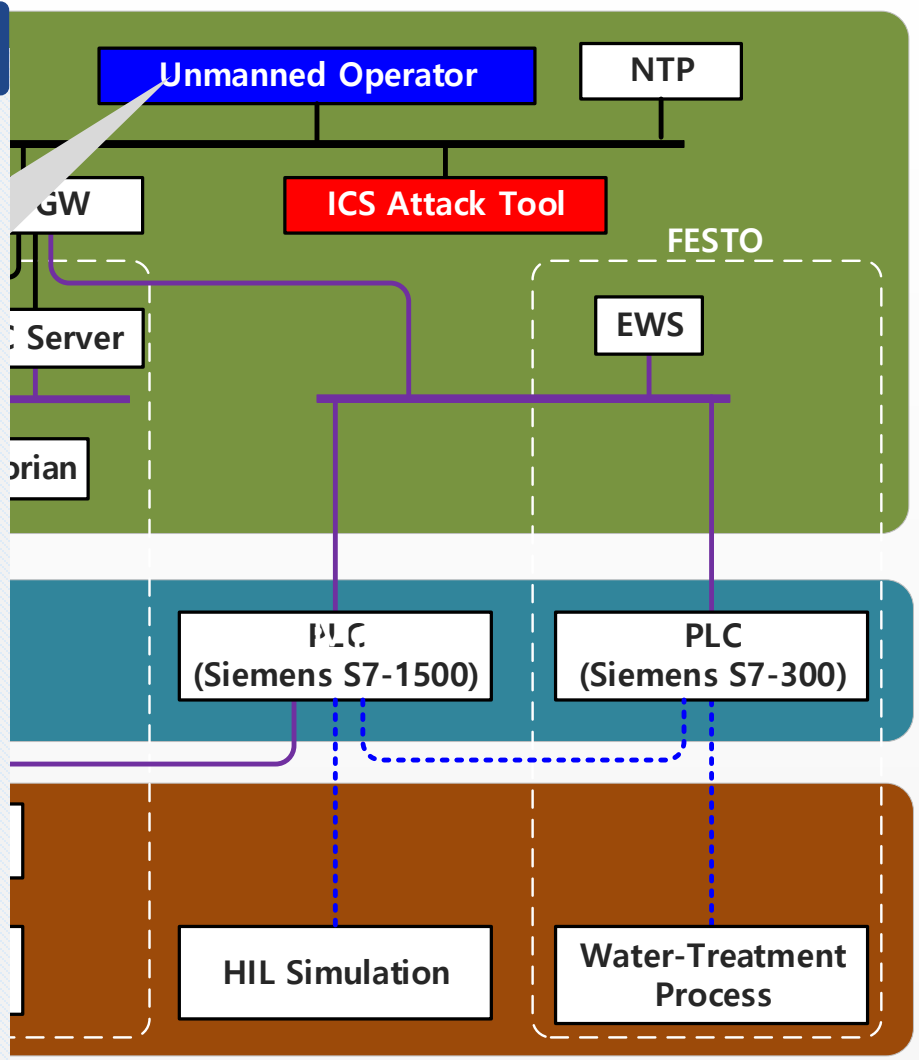


Network Structure of HAI Testbed

② Unmanned Normal Operation

- Check whether the controller is stabilized
- Command a new SetPoint within operational range

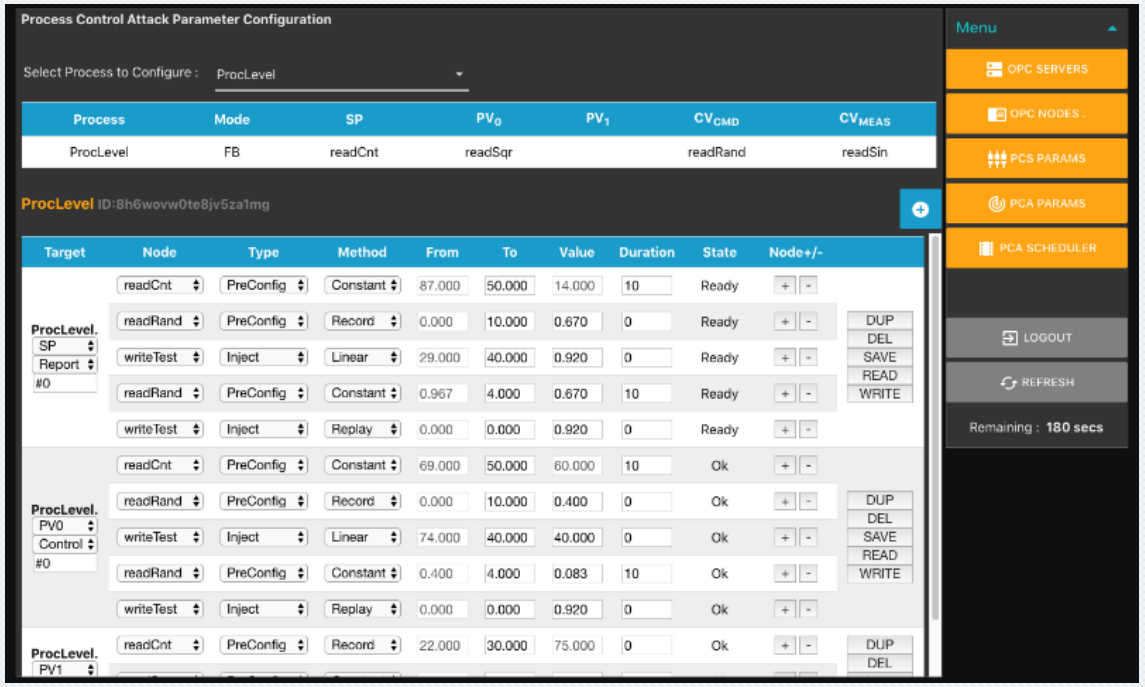


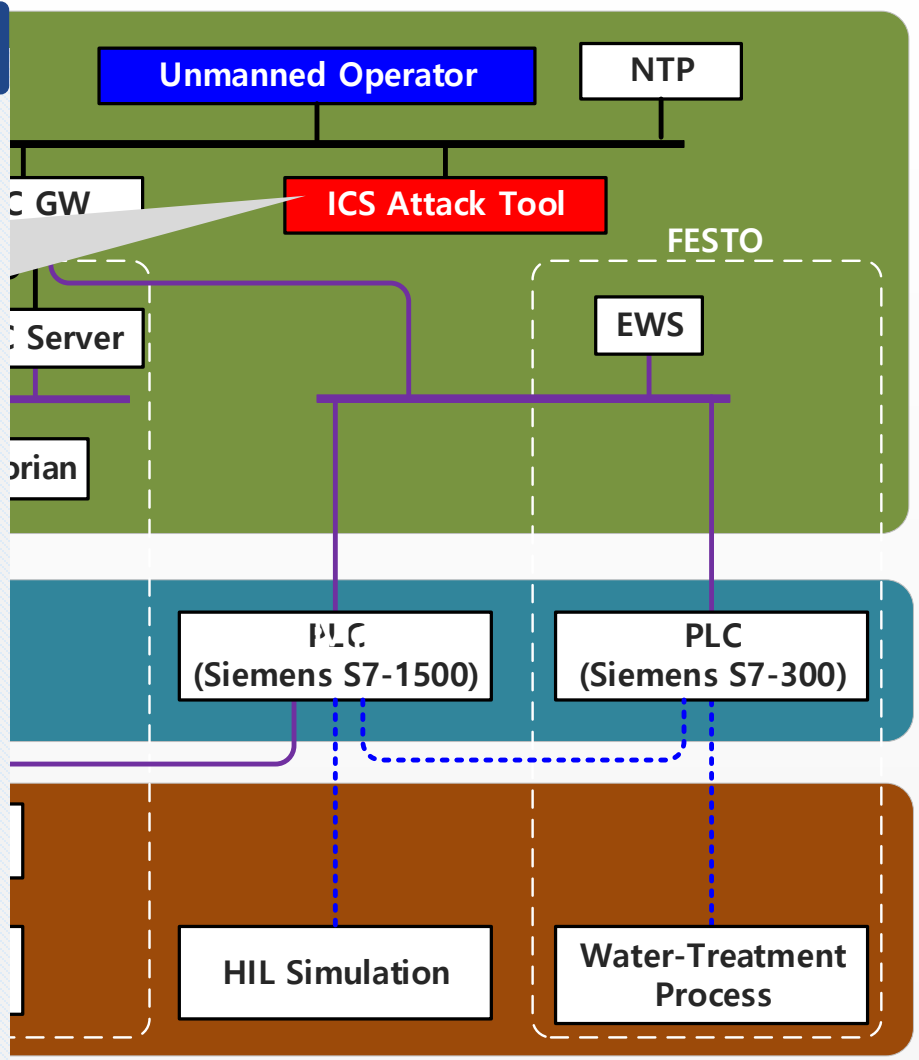


Network Structure of HAI Testbed

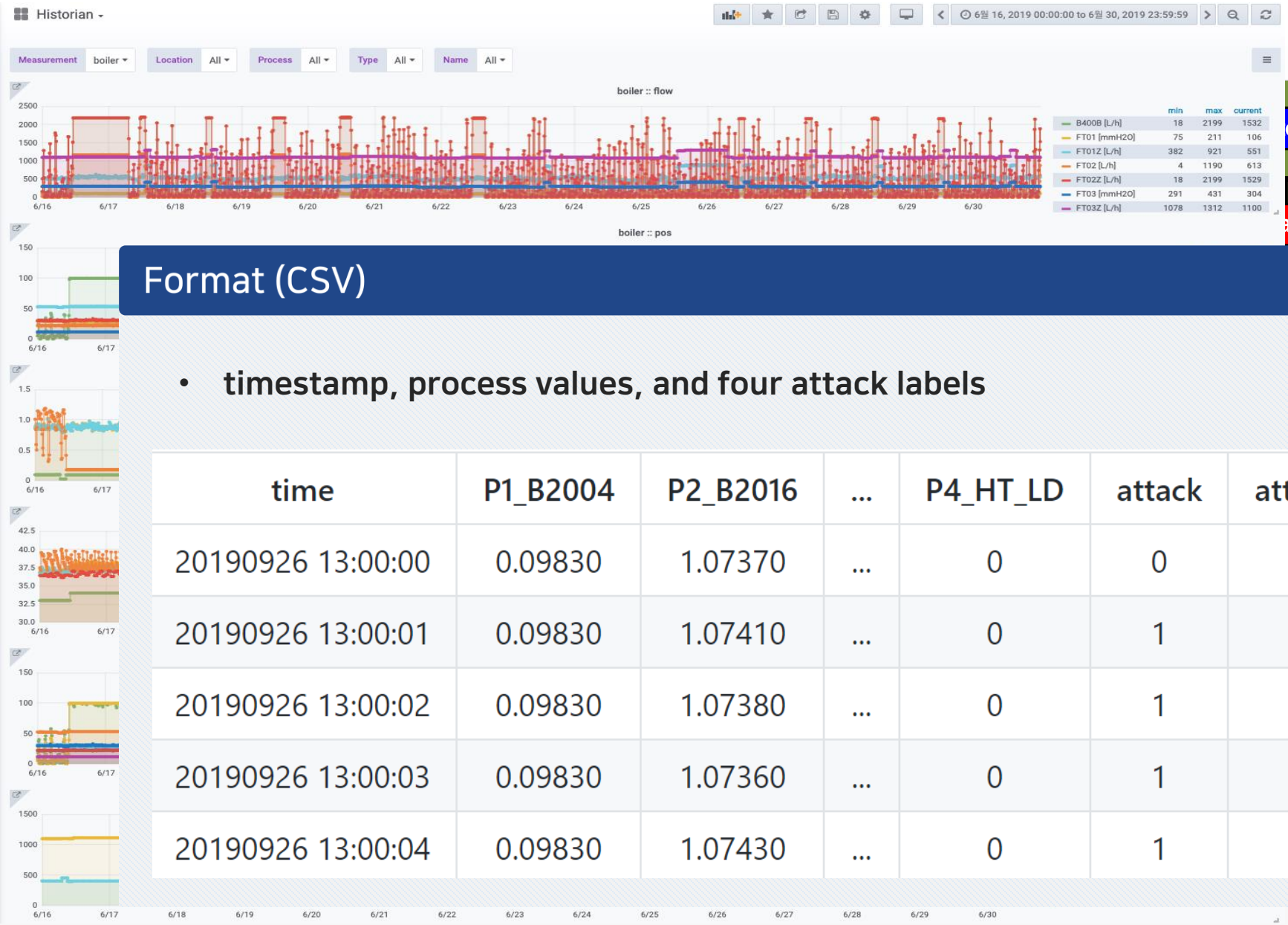
③ Scalable Attack Tool based on Process Control Loop

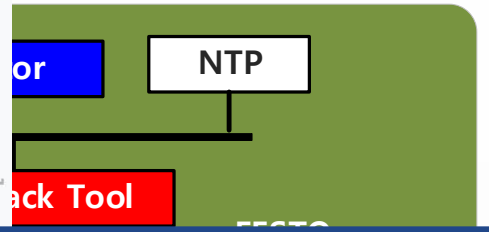
- Changing the internal values by modifying the parameters of Function Block (FB)





HAI Security Dataset





Format (CSV)

- timestamp, process values, and four attack labels

time	P1_B2004	P2_B2016	...	P4_HT_LD	attack	attack_P1	...	attack_P3
20190926 13:00:00	0.09830	1.07370	...	0	0	0	...	0
20190926 13:00:01	0.09830	1.07410	...	0	1	0	...	1
20190926 13:00:02	0.09830	1.07380	...	0	1	0	...	1
20190926 13:00:03	0.09830	1.07360	...	0	1	1	...	1
20190926 13:00:04	0.09830	1.07430	...	0	1	1	...	1

HAI Security Dataset

• GitHub - https://github.com/icsdataset/hai

	HAI 20.07		HAI 21.03
Number of process :	6		7 (TRIP)
Points/sec :	59		78
Duration(size) :	7 days(117MB)	training data	11 days(471MB)
Number of scenario :	5		10
Duration(size) :	5 days(177MB)	test data	5 days(205MB)
Number of scenario(single/multiple) :	38(19/19)		50(25/25)
Repeated attack :	X		O
SIS attack :	X		O (TRIP)

Survey : S. Choi, et.al., "A Comparison of ICS Datasets for Security Research Based on Attack Paths," CRITIS, 2018.
HAI testbed : H. Shin, W. Lee, J. Yun, S. Kim, "Implementation of Programmable CPS Testbed for Anomaly Detection," CSET@USENIX Security, 2019.
HAI 20.07 : H. Shin, W. Lee, J. Yun, H. Kim, "HAI 1.0: HIL-based Augmented ICS Security Dataset," CSET@USENIX Security, 2020.
HAI 21.03 : H. Shin, W. Lee, J. Yun, B. Min, "Two ICS Security Datasets and Anomaly Detection Contest on the HIL-based Augmented ICS Testbed," CSET, 2021

HAICon 2020

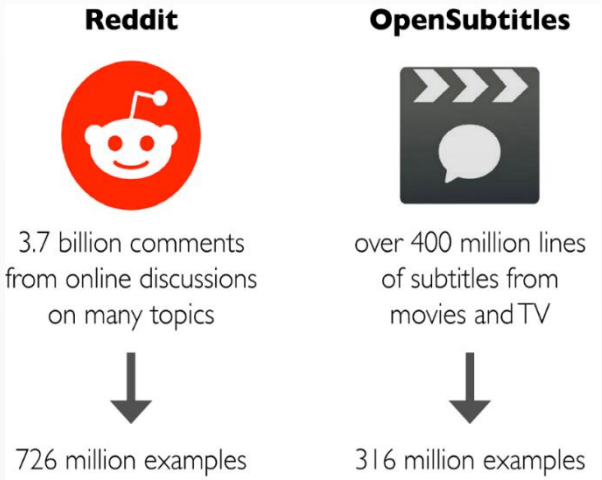
Online Anomaly Detection Competition with HAI 21.03 Dataset

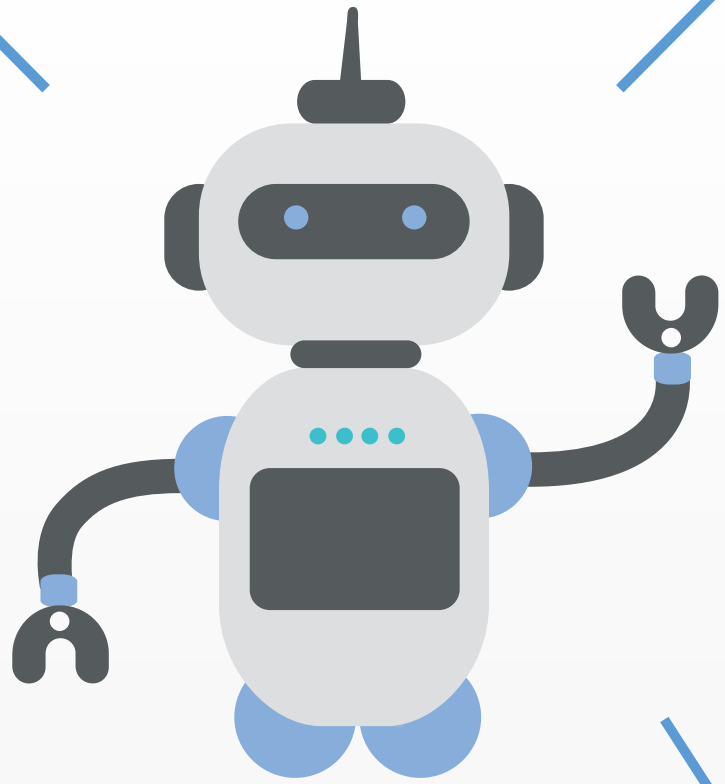
- ☑ August 17 ~ September 29, 2020
- ☑ $16,000 Prize
- ☑ http://dacon.io

  Note that foreign participants must team up with at least one Korean

Datasets

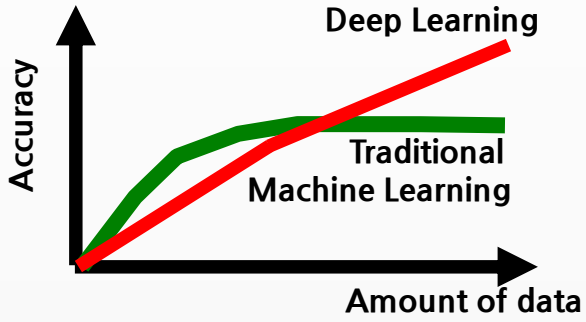
- HAI 20.07 : Warm-up
- HAI 21.03 : Competition





Baseline code

- Data preprocessing
- RNN-based anomaly detection



Performance Metric

- Which anomaly detector is effective in time-series?

Baseline model code for HAICon 2020

NSR

Data → **Preprocessing** → **Training** → **Anomaly Detection**

- HAI 20.07
- HAI 21.03

- Min-Max normalization
- Exponential Weight Avg.

- Model : Stacked RNN
- Loss function : MSE
- Optimizer : AdamW
- Skip connection

- Static threshold (MSE)

7 teams of TOP 10 used this baseline model

HAI 2.0 Baseline Model

본 문서는 HAICon을 위한 베이스라인 모델을 제공합니다.

Recurrent neural network(RNN)을 이용했으며 구현 시 라이브러리는 PyTorch를 이용했습니다.

GPU가 1개 있다고 가정하고 코드가 작성되었습니다. CPU로만 구동해보실 경우에는 모델과 batch feed 코드에서 .cuda()를 호출하지 않으시면 됩니다.

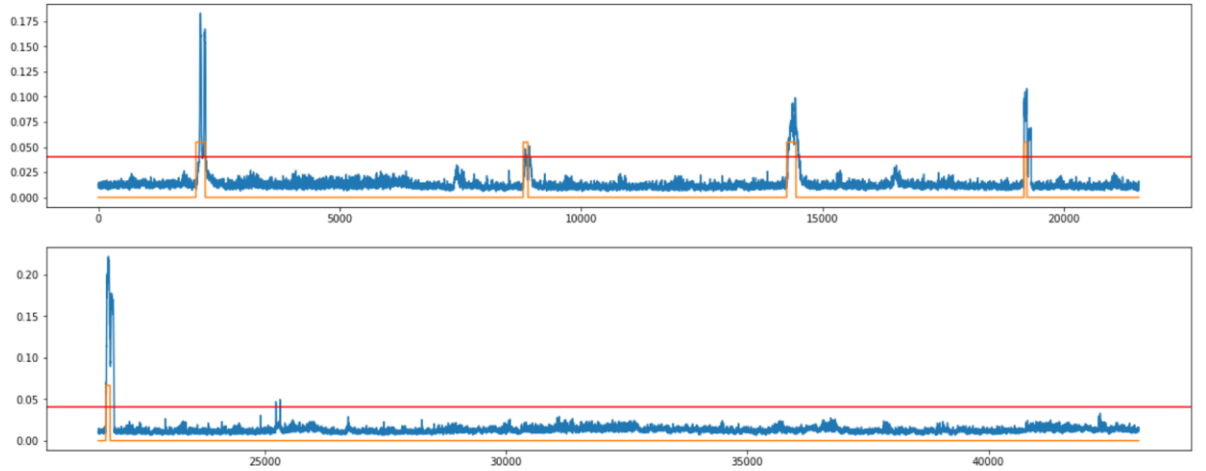
구현 편의성을 위해 전역 변수가 많이 사용되었습니다. 양해 바랍니다. 전역 변수는 모두 대문자로 명명했습니다.

```
%reload_ext watermark
#%reload_ext lab_black
```

위의 extension은 code format과 버전 출력을 위해서 사용한 것입니다. 실행하지 않아도 됩니다.

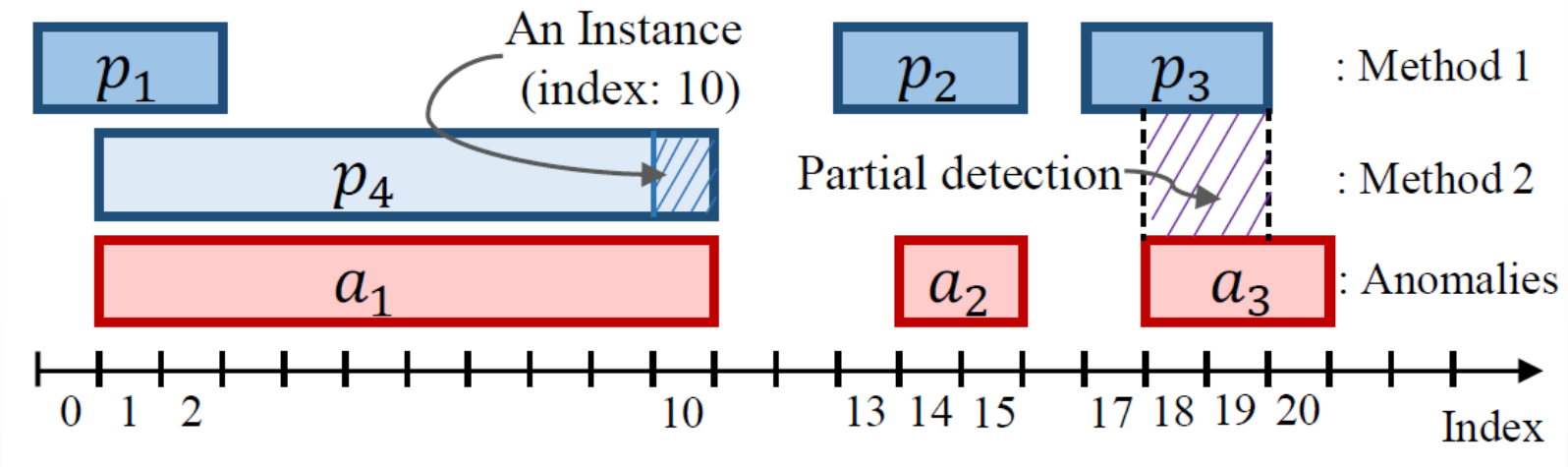
```
%watermark -v -p dateutil,numpy,matplotlib,pandas,torch,tqdm,TaPR_pkg,cv2
```

```
THRESHOLD = 0.04
check_graph(ANOMALY_SCORE, CHECK_ATT, piece=2, THRESHOLD=THRESHOLD)
```



https://dacon.io/competitions/official/235624/codeshare/1570?page=1&dtype=recent

Performance Metric for Time-series Anomaly Detection



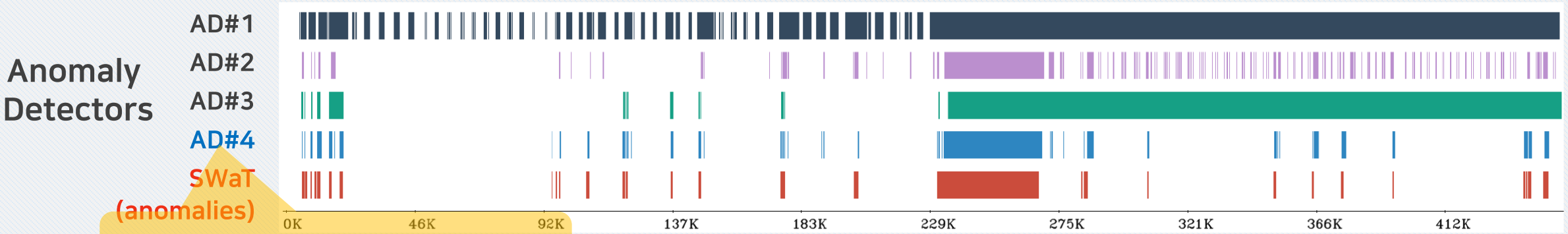
Precision/recall metric is not proper

- **Method 2 gets higher precision/recall scores than Method 1:**
  - ✓ **Method1 detects 6 seconds abnormal instances with 3 seconds false alarms.**
  - ✓ **Method 2 detects 10 seconds abnormal instances w/o any false alarm.**

- **But, Method 1 is more effective at signaling anomalies (a1, a2, and a3) than Method 2.**

Our Performance Metric for Time-series Anomaly Detection

eTaPR: Enhanced Time-Aware Precision and Recall



AD#4 is the most effective

eTaPR only picks AD#4 among metrics for time-series.

Ranking \ Metric	eTaPR[1]	TaPR[2]	TSAD[3]
1st	**AD#4 (0.73)**	AD#3 (0.72)	AD#3 (0.57)
2nd	AD#3 (0.39)	**AD#4 (0.60)**	**AD#4 (0.45)**
3rd	AD#2 (0.28)	AD#2 (0.11)	AD#2 (0.11)
4th	AD#1 (0.17)	AD#1 (0.09)	AD#1 (0.08)

[1] Won-seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim, "Enhanced Time-Series Aware Precision and Recall for Anomaly Detection", HAICon 2020 python package - https://dacon.io/competitions/official/235624/data (still updating)

[2] Won-seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim, "Time-Series Aware Precision and Recall for Anomaly Detection", *CIKM* 2019, source code - https://github.com/saurf4ng/TaPR

[3] Nesime Tatbul et al., "Precision and Recall for Time Series," *Advances in Neural Information Processing Systems*, 2018

HAICon 2020 : Competition result

Public Ranking

For 30% of test data

#	팀	팀 멤버	점수	제출수
1	통계왕		0.98065	100
2	SIlab		0.98031	105
3	j1117j		0.98031	3
4	역모를꿈꾸는자		0.97999	86
5	Dynamite		0.97934	33
6	Hady		0.97926	3
7	FDAI		0.97858	98
8	neptune		0.97675	40
9	K-Normal		0.97652	121
10	합정역 1번출구		0.97597	53

Final Ranking

For 100% of test data

#	팀	팀 멤버	최종점수	제출수
1	역모를꿈꾸는자		0.93793	86
2	SIlab		0.93614	102
3	자몽주스		0.93414	61
4	내밑은2등		0.92752	68
5	K-Normal		0.9248	121
6	csrcai		0.92365	79
7	합정역 1번출구		0.92164	53
8	APL		0.91937	55
9	초초		0.91811	22
10	parkqho		0.9177	9

Discussion - Anomaly detection

1. Baseline model works well

- **7 teams among top 10 used the baseline**
  - Min-max normalization & Exp. Weight average
  - RNN + skip connection
  - Prediction error based anomaly detection

2. Ensembles with Multiple monitoring periods

- **Each attack has different period**
  - Detects well if attack period = window size
- **We cannot guess what is the best window size**
  - Multi-window ensemble can practically improve performance

3. Threshold selection (post processing)

- **Threshold studies are needed**
  - Threshold is the most critical,
    so most participants spend most time for it
  - But most research focus on deep learning models

4. How to use abnormal data?

- **Train data has no abnormal data, but practically**
  - may get abnormal data in field test
  - may generate typical cases with domain knowledge
- **How to use such information?**

Conclusion

Summary

- **Two datasets: HAI 20.07/21.03**
  - https://github.com/icsdataset/hai
- **Online AI contest: HAICon 2020**
  - https://dacon.io/competitions/official/235624/overview/description
  - Performance metric : eTaPR
  - RNN-based baseline code

Future work

- **Update HAI datasets**
  - The Third HAI dataset is ready
    - Using some new hardware devices
    - Will be shared after HAICon 2021
  - Research for data generation framework
    - Expanding data collection area
    - Various scenarios for user purposes
- **HAICon 2021**
  - https://dacon.io/competitions/official/235757/overview/description
  - 17 Aug. ~ 22 Oct.
  - Performance metric : eTaPR
  - RNN-based baseline code

NSR
National Security Research Institute

Q & A