USAD : UnSupervised Anomaly Detection on Multivariate Time Series

Author	Julien Audibert julien.audibert@orange.com EURECOM Biot, France Orange Sophia Antipolis, France et al.
read date	@2022년 5월 4일
Journal	KDD
PDF	[2020 KDD] USAD UnSupervised Anomaly Detection on Multivariate Time.pdf
Published Date	2020
detail	연구 배경 : 프랑스 통신기업인 Orange에서 IT 모니터링을 위해 개발한 방법론 - 구조 : GAN에서 아이디어를 얻어 적대적으로 훈련된 AE를 사용 1. Training Stage : 기존 AE 훈련 + 일종의 생성자인 AE_1, 판별자 AE_2를 각각 Loss를 나눠서 훈련 ㄴ 훈련 단계에서는 정상 데이터만을 가지고 학습함 2. Detectoin stage : Test로, \alpha & \beta 하이퍼파라미터를 통해 민감도를 정해서 이상치를 결정 - 실험 결과 : 공공데이터에 Orange데이터에서 SOTA 달성 - 코드 : test 시 with torch.no_grad(): 조차 안되어 있음 → 공식 코드는 없지만, 구조 이해에 도움이 됨
keyword	Anomaly Detection MTS Unsupervised
link	https://dl.acm.org/doi/pdf/10.1145/3394486.3403392
status	Finished!

배경 : Orange(프랑스 통신기업) 소속 인원들이 낸 논문

ABSTRACT

- IT 시스템의 automatic supervision은 Orange에서 최근 과제이다.
- IT운영에 도달하는 사이즈와 복잡성을 감안할 때, 정상 또는 비정상 행동을 추론하는데 사용하는 시간 경과에 따라 측정 값을 얻기 위해 센서의 수가 증가하여 기존의 전문가 기반 감독 방법이 느려지거나 오류가 발생하기 쉽습니다.
- 제안 : fast and stable method based on “adversely trained autoencoders”
  - UnSupervised Anomaly Detection for multivariate time series (USAD)
  - adversely trained autoencoders : 적대적으로 훈련된 오토인코더를 기반으로 함
  - AE 구조가 비지도 학습을 가능하게 한다
  - 적대적 훈련의 사용과 이것의 구조는 빠른 훈련 제공 동안 이상치를 고립시킨다.

INTRODUCTION

- 연구 배경
  - IT 시스템 모니터링은 측정 가능한 이벤트 및 시스템 결과에 대한 감독 프로세스이다.
  - 이제는 필요한 센서의 수가 급격히 증가하여 기존의 전문가가 정의한 임계값 방법은 확장할 수 없기에 사용할 수 없다.
    - Orange만의 IT system monitoring이 필요해 짐
    - 자동화된 IT 시스템 모니터링을 위해서는 센서가 수집한 다양한 측정 값을 관찰하고 이를 통해 정상 및 비정상 동작을 추론하는 방법의 개발이 필요
  - 고차원 데이터인 차원의 저주에 의해 AD에서 최적이 아닌 성능으로 어려움을 겪음
  - RNN : 계산량이 많고 학습하는 데 상당한 시간이 필요

    - GAN : 붕괴 및 비수렴 문제 有

- 제안 방법 : UnSupervised Anomaly Detection for multivariate time series (USAD)
    - GAN에서 영감을 받아 인코더-디코더가 적대적으로 학습하는 구조
    - GAN 기반으로 하는 방법에 비해 안정성을 얻으면서 비정상을 포함하는 입력의 재구성 오류를 증폭하는 방법을 배울 수 있음

- 기여점 3가지
    1. adversarial training 프레임워크 내 인코더-디코더 아키텍처를 제안
    2. 견고성, 훈련 속도, 성능 입증 위해 공공 데이터 세트에 대한 연구 수행
    3. 확장성, 안정성, 견고성, 교육 속도, 고성능에 대한 회사의 요구 사항을 충족하는지 입증하기 위해 Orange의 독점 데이터로 연구 수행

RELATED WORK

- DAGMM : The Deep Autoencoding Gaussian Mixture Model
- MSCRED : The Multi-Scale Convolutional Recursive EncoderDecoder
- LSTMVAE : LSTM with a variational autoencoder(VAE)
    - LSTM 네트워크를 통해 시계열의 시간 종속성을 모델링하고 기존 방법보다 더 나은 일반화 기능을 얻음
- ALAD : Adversarially Learned Anomaly Detection
- OmniAnomaly : MTS의 AD를 위한 stochastic RNN
    - stochastic variable connection and a planar normalizing flow를 통해 MTS 표현을 학습
    - reconstruction probabilities을 사용하여 이상을 결정
- 위의 방법들은 훈련 속도를 희생시키면서 좋은 결과를 얻음

  → 훈련 시간(i.e. energy consumption)을 고려하지 않음

METHOD

3.1 Problem formulation

$$\mathcal{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$$

- multivariate time series : x는 벡터이고, 특정 시간 t에서 특정된 프로세스의 관찰. 정상치만 포함

$$\hat{\mathbf{x}}_t,\ t > T$$

- $\hat{x_t}$: t 시점의 unseen observation = 이상치
    - anomaly score를 통해 구분되며, 임계값을 기준으로 한다.

$$W_t = \{\mathbf{x}_{t-K+1}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}.$$

- W_t = a time window of length K at given time t : t시점에서 K 길이를 가진 시간 윈도우

$$\mathcal{W} = \{W_1, \ldots, W_T\}$$

- 정상인 MTS τ 를 윈도우로 변환한 것으로, 훈련의 input으로 쓰임

$$\widehat{W}_t,\ t > T, \text{a label } y_t$$

- $\hat{W}_t$ = unseen window로 testset에 이상점수 y_t를 할당 1=이상치, 0=이상치 아님

3.2 Unsupervised Anomaly Detection

- AE : encoder E + decoder D
- E : X(input) → Z(latent variables), D : Z(latent variables)→ R(reconstruction)
- reconstruction error : X - R (L2-norm 사용 됨) = anomaly score
  - high error → 이상치

$$\mathscr{L}_{AE} = \|X - AE(X)\|_2,$$

$$AE(X) = D(Z), \quad Z = E(X)$$

- 이상 정도가 너무 적은 경우 reconstruction이 잘될 수 있음
  → GAN을 통해 입력 데이터가 이상인지 정상인지 식별을 먼저하자
- 정상 데이터를 사용하여 G를 훈련을 하고, 훈련 후 D는 이상치를 검출함. 입력 데이터가 훈련된 데이터 분포와 다르면 판별기는 이를 Fake(=이상치)라고 판단함

- USAD는 2단계의 적대적 훈련 프레임 워크 내에서 AE 구조로 공식화 됨
  1. training

     : 데이터가 이상을 포함하지 않을 때 식별할 수 있는 모델을 훈련하여 AE의 본질적 한계를 극복할 수 있도록 좋은 재구성 수행 가능
  2. detectoin stage

     : AE 구조는 적대적 훈련 동안 안정적을 얻을 수 있게 함으로, GAN에서 마주치는 collapse and non-convergence mode 문제를 해결함

- USAD의 3가지 구성요소 : E(encoder), D_1 = decoder, D_2 = decoder → AE_1, AE_2

$$AE_1(W) = D_1(E(W)), \quad AE_2(W) = D_2(E(W))$$

Training stage

Phase 1: Autoencoder training (정상데이터의 분포를 학습)

- 각 AE가 입력을 재현하도록 훈련함. Z = AE(W) 후 각 디코더에 의해 재 구성

$$\mathscr{L}_{AE_1} = \|W - AE_1(W)\|_2$$
$$\mathscr{L}_{AE_2} = \|W - AE_2(W)\|_2$$

Phase 2 : Adversarial training

- AE_1 훈련(일종의 생성자) =AE_2를 속이기 위해 훈련
  - 목표 = min $AE_2(AE_1(W))$ 와 W의 차이
- AE_2 훈련(일종의 판별자) = real data W vs $AE_1(W)$를 구별하기 위해
  - AE_2가 판별자니까, AE_1에서 나온 데이터를 넣어서 학습한다
  - 목표 = max $AE_2(AE_1(W))$ 와 W의 차이

$$\mathscr{L}_{AE_1} = + \|W - AE_2(AE_1(W))\|_2$$
$$\mathscr{L}_{AE_2} = - \|W - AE_2(AE_1(W))\|_2$$

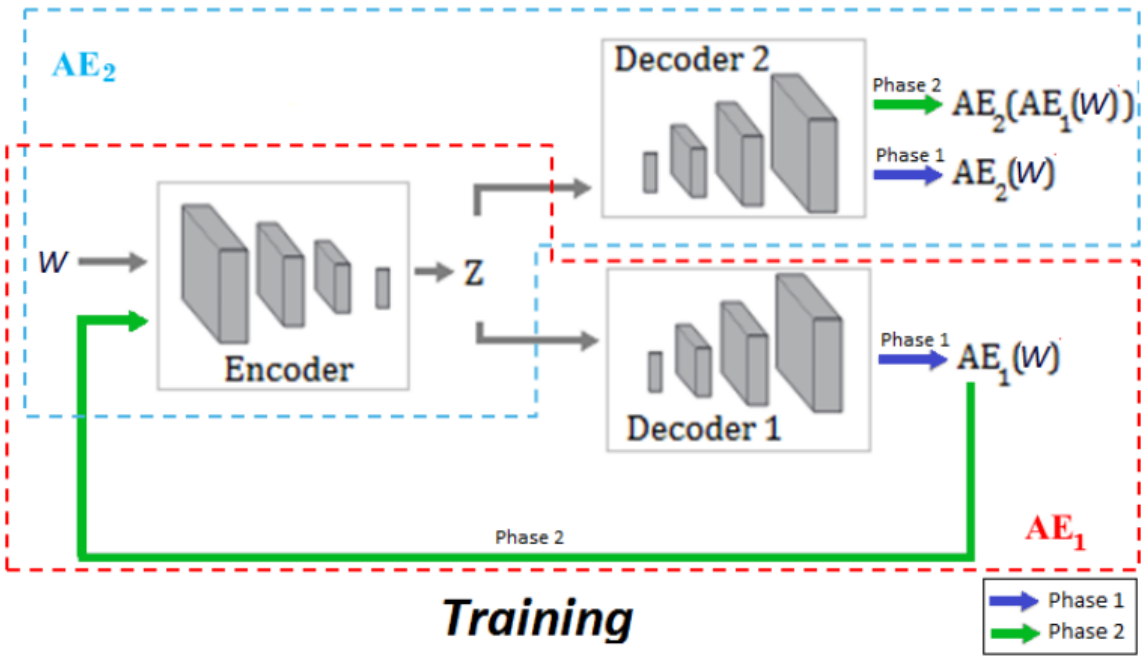
Phase 1 + 2 조합

- 데이터가 이상을 포함하지 않을 때 식별할 수 있는 모델을 훈련

  → AE의 본질적 한계를 극복할 수 있도록 좋은 recontruction 수행 가능

- Loss_AE_1 = AE_1의 재구성 오류는 작게하면서, AE_2를 속이기 위해 AE_2 output과 차이 적게
- Loss_AE_2 = AE_2의 재구성 오류는 작게하면서, 실제와 AE_2 output을 구별 위해 차이 크게
  - n = epoch, 초반에는 phase 1에 가중치를 두고, 후반에는 phase 2에 가중치를 부여함

$$\mathscr{L}_{AE_1} = \frac{1}{n} \|W - AE_1(W)\|_2 + \left(1 - \frac{1}{n}\right) \|W - AE_2(AE_1(W))\|_2 \quad (7)$$
$$\mathscr{L}_{AE_2} = \frac{1}{n} \|W - AE_2(W)\|_2 - \left(1 - \frac{1}{n}\right) \|W - AE_2(AE_1(W))\|_2 \quad (8)$$



Algorithm 1 USAD training algorithm

Input: Normal windows Dataset $\mathcal{W} = \{W_1, ..., W_T\}$, Number epochs N
Output: Trained AE_1, AE_2

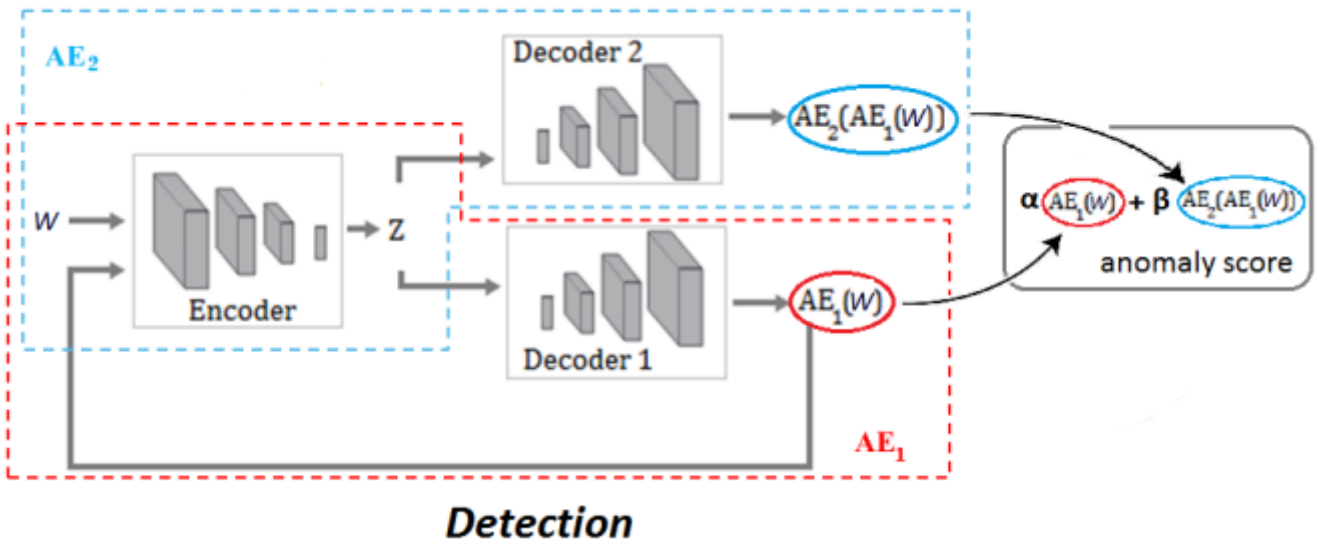
$E, D_1, D_2 \leftarrow$ initialize weights
$n \leftarrow 1$
repeat
  for $t = 1$ to T **do**
    $Z_t \leftarrow E(W_t)$
    $W_t^{1'} \leftarrow D_1(Z_t)$
    $W_t^{2'} \leftarrow D_2(Z_t)$
    $W_t^{2''} \leftarrow D_2(E(W_t^{1'}))$
    $\mathscr{L}_{AE_1} \leftarrow \frac{1}{n} \left\|W_t - W_t^{1'}\right\|_2 + \left(1 - \frac{1}{n}\right) \left\|W_t - W_t^{2''}\right\|_2$
    $\mathscr{L}_{AE_2} \leftarrow \frac{1}{n} \left\|W_t - W_t^{2'}\right\|_2 - \left(1 - \frac{1}{n}\right) \left\|W_t - W_t^{2''}\right\|_2$
    $E, D_1, D_2 \leftarrow$ update weights using $\mathscr{L}_{AE_1}$ and $\mathscr{L}_{AE_2}$
  end for
  $n \leftarrow n + 1$
until $n = N$

Detectoin stage = Test에서 사용

- training stage에서 학습된 AE_1, AE_
- Anomlay Score

$$\mathscr{A}(\widehat{W}) = \alpha \|\widehat{W} - AE_1(\widehat{W})\|_2 + \beta \|\widehat{W} - AE_2(AE_1(\widehat{W}))\|_2 \qquad (9)$$

- 이상치 점수 = 생성자에 해당하는 LOSS + 판별자에 해당하는 LOSS
- $\alpha + \beta = 1$
  - $\alpha > \beta$ = TP & FP ↓ (low detection sensitivity : 낮은 민감도)
  - $\alpha < \beta$ = TP & FP ↑ (high detection sensitivity : 높은 민감도)



Detection

Algorithm 2 USAD Detection algorithm

Input: Test windows Dataset $\widehat{W}$: $(\widehat{W}_1, ..., \widehat{W}_{T^*})$, threshold λ , parameters α and β
Output: Labels $\mathbf{y} : \{y_1, ..., y_{T^*}\}$
for $t = 1$ to T^* **do**
$\widehat{W}_t^{1'} \leftarrow D_1(E(\widehat{W}_t))$
$\widehat{W}_t^{2''} \leftarrow D_2(E(\widehat{W}_t^{1'}))$
$\mathscr{A} \leftarrow \alpha \left\|\widehat{W}_t - \widehat{W}_t^{1'}\right\|_2 + \beta \left\|\widehat{W}_t - \widehat{W}_t^{2''}\right\|_2$
if $\mathscr{A} \geq \lambda$ **then**
$y_t \leftarrow 1$
else
$y_t \leftarrow 0$
end if
end for

3.3 Implementation

- 3단계로 감지방법을 나눌 수 있음
  1. data pre-processing stage = 데이터 정규화 & K길이의 시간창으로 분할되는 훈련 및 감지
  2. training stage = MTS의 정상적 행동 포착하여 각 윈도우 창에 대한 이상 점수를 생성 목표
  3. anomaly detection stage = 2에서 훈련된 모델을 사용하여 실시간 수행. 새로운 윈도우에 대해 이상 점수를 얻고, 임계값 보다 높으면 abnormal이 선언

EXPERIMENTAL SETUP

4.1 Public Datasets & 4.2 Feasibility study: Orange's dataset

Dataset	Train	Test	Dimensions	Anomalies (%)
SWaT	496800	449919	51	11.98
WADI	1048571	172801	123	5.99
SMD	708405	708420	28*38	4.16
SMAP	135183	427617	55*25	13.13
MSL	58317	73729	27*55	10.72
Orange	2781000	5081000	33	33.72

1. Secure Water Treatment(SWaT)
   - 산업용수 처리 공장의 11일 = 7일 정상 + 4일 일부 비정상
2. Water Distribution (WADI)
   - SWaT 확장판으로 16일 = 14일 정상 + 2일 일부 비정상
3. Server Machine Dataset(SMD)
   - 인터넷 회사의 5주 길이로 28대의 서버 시스템에 각 33개의 모니터링
4. Soil Moisture Active Passive (SMAP) & Mars Science Laboratory (MSL)
   - 토양수 유무 및 화성 과학연구소 NASA 데이터 세트로, 55/27 독립체에 각 25/55 모니터링
5. Orange
   - 27개의 기술 측정값 + 6개의 비즈니스 측정 = 총 m = 33개의 연속형 변수
   - 32일 train data, 60일 test data

4.3 Evaluation Metrics

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = 2 \cdot \frac{P \cdot R}{P + R} \qquad F1^* = 2 \cdot \frac{\bar{P} \cdot \bar{R}}{\bar{P} + \bar{R}}$$

- $\bar{P}, \bar{R}$ = average precision and recall, respectively

EXPERIMENTS AND RESULTS

5.1 Overall performance

Methods	SWaT						WADI					
	Without			With			Without			With		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
AE	0.9903	0.6295	0.7697	0.9913	0.7040	0.8233	0.9947	0.1310	0.2315	0.3970	0.3220	0.3556
IF	0.9512	0.5884	0.7271	0.9620	0.7315	0.8311	0.2992	0.1583	0.2071	0.6241	0.6155	**0.6198**
LSTM-VAE	0.9897	0.6377	0.7756	0.7123	0.9258	0.8051	0.9947	0.1282	0.2271	0.4632	0.3220	0.3799
DAGMM	0.4695	0.6659	0.5507	0.8292	0.7674	0.7971	0.0651	0.9131	0.1216	0.2228	0.1976	0.2094
OmniAnomaly	0.9825	0.6497	0.7822	0.7223	0.9832	0.8328	0.9947	0.1298	0.2296	0.2652	0.9799	0.4174
USAD	0.9851	0.6618	**0.7917**	0.9870	0.7402	**0.8460**	0.9947	0.1318	**0.2328**	0.6451	0.3220	0.4296

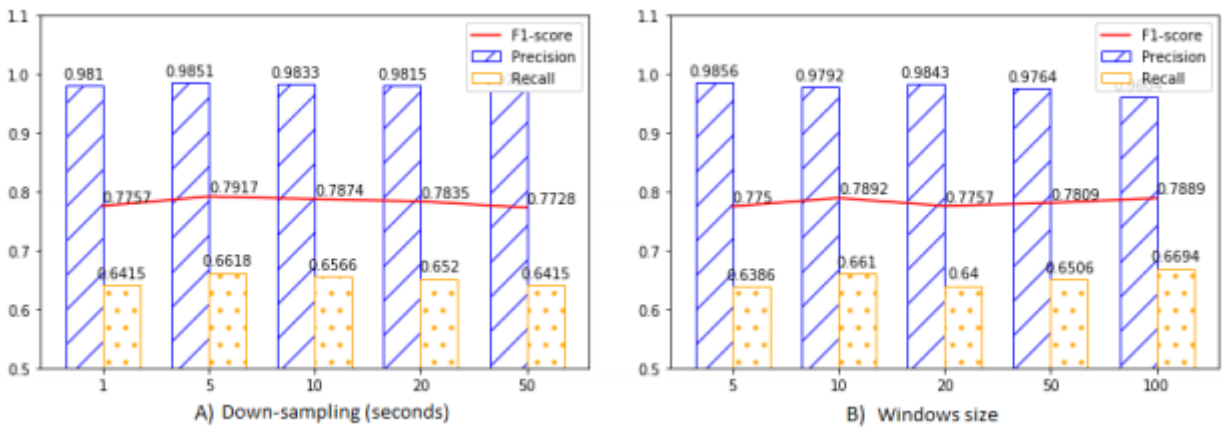
Methods	SMD				SMAP				MSL			
	P	R	F1	F1*	P	R	F1	F1*	P	R	F1	F1*
AE	0.8825	0.8037	0.8280	0.8413	0.7216	0.9795	0.7776	0.8310	0.8535	0.9748	0.8792	0.9101
IF	0.5938	0.8532	0.5866	0.7003	0.4423	0.5105	0.4671	0.4739	0.5681	0.6740	0.5984	0.6166
LSTM-VAE	0.8698	0.7879	0.8083	0.8268	0.7164	0.9875	0.7555	0.8304	0.8599	0.9756	0.8537	0.9141
DAGMM	0.6730	0.8450	0.7231	0.7493	0.6334	0.9984	0.7124	0.7751	0.7562	0.9803	0.8112	0.8537
OmniAnomaly	0.9809	0.9438	**0.9441**	**0.9620**	0.7585	0.9756	0.8054	0.8535	0.9140	0.8891	0.8952	0.9014
USAD	0.9314	0.9617	0.9382	0.9463	0.7697	0.9831	**0.8186**	**0.8634**	0.8810	0.9786	**0.9109**	**0.9272**

- IF = Isolation Forests, DAGMM → 시간 정보 반영 불가(Exploiting Temporal Information)

- point-adjust = one observation of an anomalous segment is correctly detected, all the other observation of the segment are also considered as correctly detected

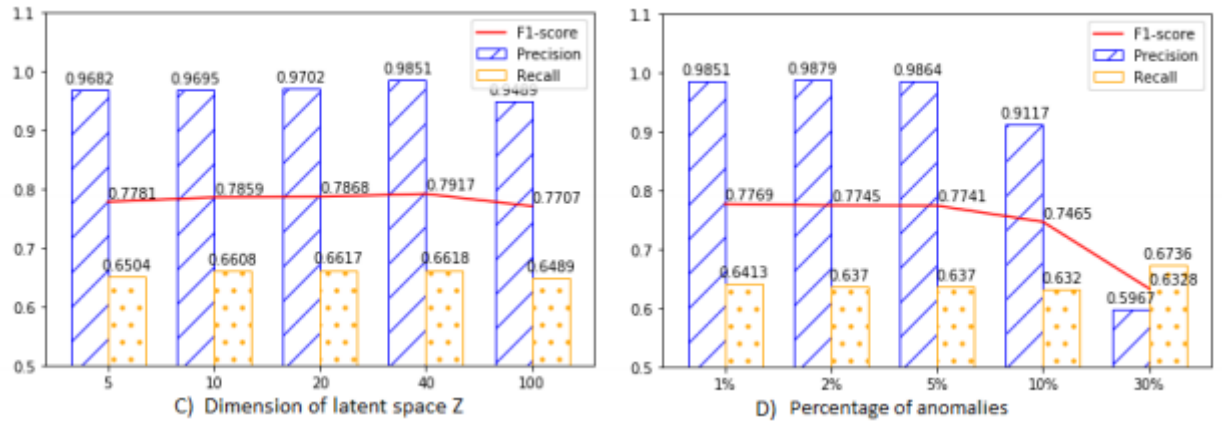
  : Window 내부의 한 시점이라도 abnormal이면, 해당 구간을 abnormal로 부여하는 방식

5.2 Effect of parameters : SWaT 기준



A) the training set’s down-sampling rate : 50% 선택. 비교적 둔감한 편

B) the window size K : 10에서 최상의 결과. 창이 작을 수록 동작 변화를 더 빨리 감지



C) the dimension of the latent space Z : 데이터 셋 별로 상이

D) the percentage of anomalies in the training set

- 실험 시 모두 정상만을 가지고 훈련 했지만, 비정상이 real-world에는 있을 것이기에 가우시안 노이즈(μ = 0, σ = 0.3)를 추가
- 10% 정도 노이즈를 줬을 때 성능이 저하되기 시작함. orange의 경우 사내에서 발생되는 이상치의 확률이 매우 적기 때문에 해당 문제를 직면할 것 같지 않음

α	β	FP	TP	F1
0.0	1.0	604	35,616	0.7875
0.1	0.9	580	35,529	0.7853
0.2	0.8	571	35,285	0.7833
0.5	0.5	548	34,590	0.7741
0.7	0.3	506	34,548	0.7738
0.9	0.1	299	34,028	0.7684

- α를 증가시키고 β를 감소시킴으로써 (0.0에서 0.9로 통과할 때 최대 50%까지) TP의 수 감소를 제한하면서 (0.0에서 0.9로) FP의 수를 줄일 수 있다는 것을 관찰 → 실험에선 0.5로 셋팅
- 관리자는 낮은 감도 수준을 선호하여 잘못된 긍정의 수를 제한하지만 중요한 사고가 발생할 경우 경고하는 반면, 기술자는 높은 감도를 선호하여 최소한의 사고를 놓칠 수 있다.

Table 5: Training Time (min) per epoch on each dataset

Methods	SWAT	WADI	SMD	SMAP	MSL
OmniAnomaly	13	31	87	48	11
USAD	0.06	0.12	0.06	0.08	0.03
Acceleration factor	216	258	1331	581	349

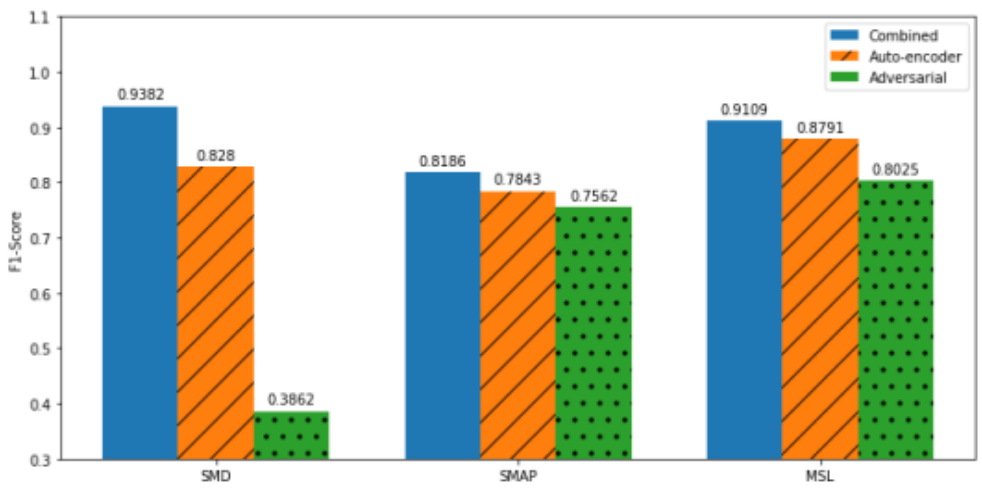
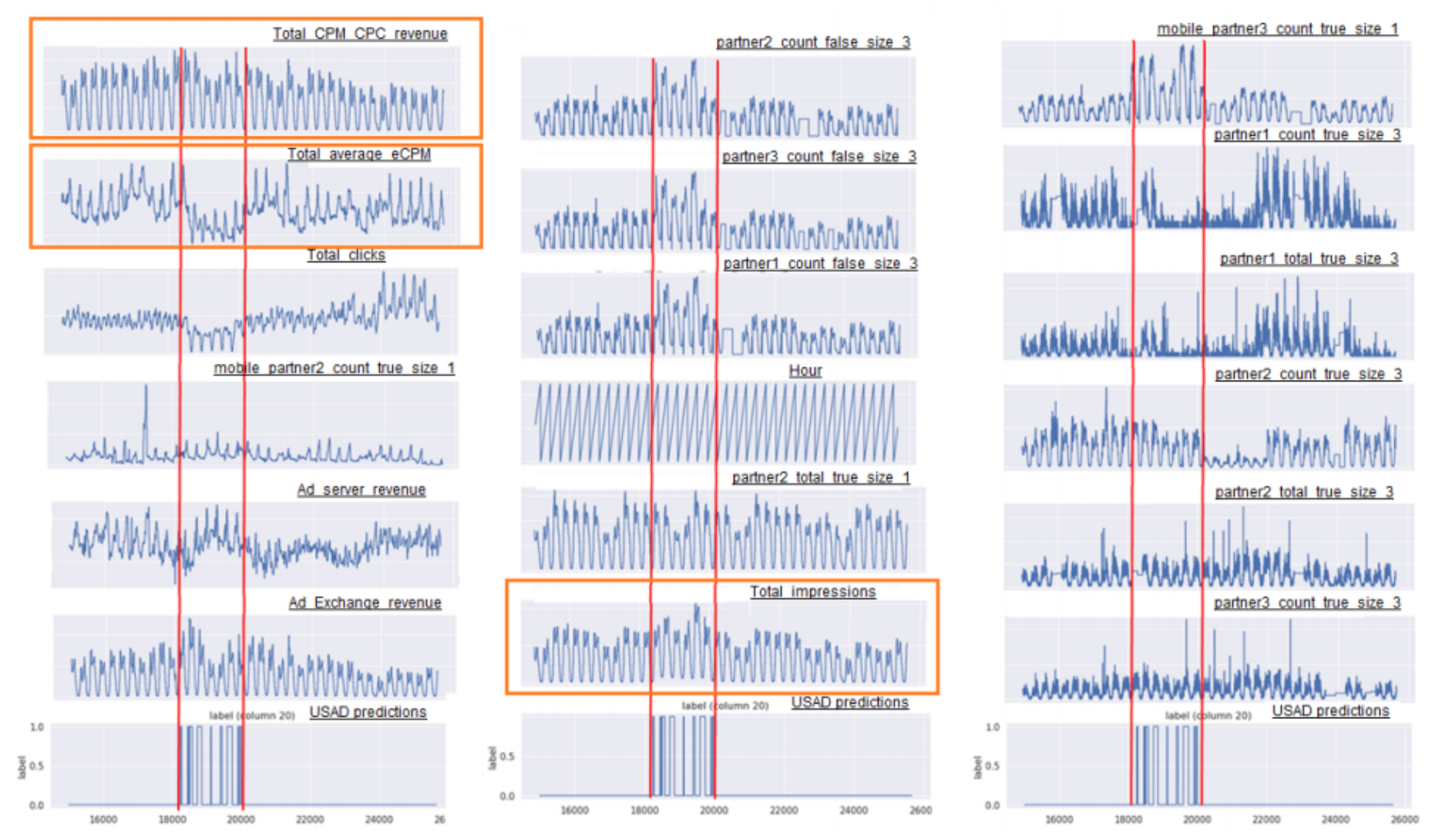


Figure 3: Impact with and without adversarial training on USAD

- 적대적 훈련이 없는 USAD = 주황색은 이상을 잘 감지 할 수 없음을 보여 줌

5.5 Feasibility study



- USAD는 두 달 동안의 테스트 데이터 기간 동안 30분 이내에 모든 중요한 사건을 탐지할 수 있었다. 예를 들어, USAD는 오렌지(Orange)에서 감독을 담당하는 운영자가 감지하는 데 24시간이 걸린 사건을 30분 이내에 감지할 수 있었다

CONCLUSIONS

- AE기반 GAN에서 영감을 받은 적대적 훈련 내의 MTS에 대한 비지도학습 제안
- 최첨단 기술보다 우수한 성능
- 민감도를 매개 변수화하고 단일 모델에서 일련의 탐지 수준을 생성할 수 있는 가능성을 제공
- 어려웠던 점
  - 많은 이상 징후를 포함하지 않는 긴 train 기간을 수집하는 것이 예상치 못한 어려움

코드 돌린 결과

> https://github.com/manigalati/usad

- 위의 작성자를 통해서 돌릴 수 있었는데, 데이터셋이 무겁지 않아서 금방 돌아갔다.
- test 시 with torch.no_grad(): 조차 안되어 있음
  → 모델이 실제로 좋은 성능을 냈는지 오피셜한 코드가 없음