# TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks

## Seminar in 2023, Paper Review

Samsung Software Developer Community
Korea Vision & Robotics
Taeuk Chu
2023.09.02

# Background

## 시계열 데이터란?

- 일정 시간 간격으로 측정된 데이터의 시간적 순서를 나타내는 데이터

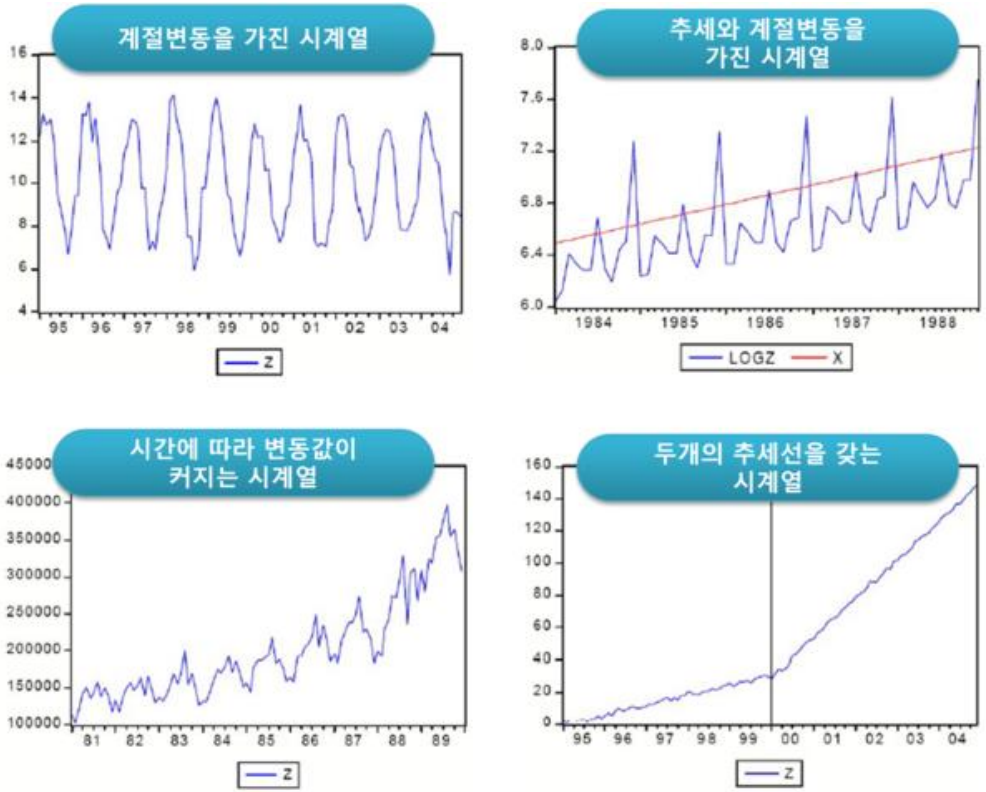
## 시계열 데이터 구성 요소

1.추세요인(Trend factor)
- 자료가 plot 일 때 오르거나 내리는 형태의 추세 존재
- 선형뿐만 아니라 비선형 형태도 존재

2.계절요인(Seasonal factor)
- 요일/월별/분기별/년별 자료에서 각 특정 고정 주기를 따라 자료가 변함

3.순환요인(Cyclical factor)
- 경제적/자연적 이유가 없이 알려지지 않은 주기를 갖고 변화

4.불규칙요인(Irregular factor)



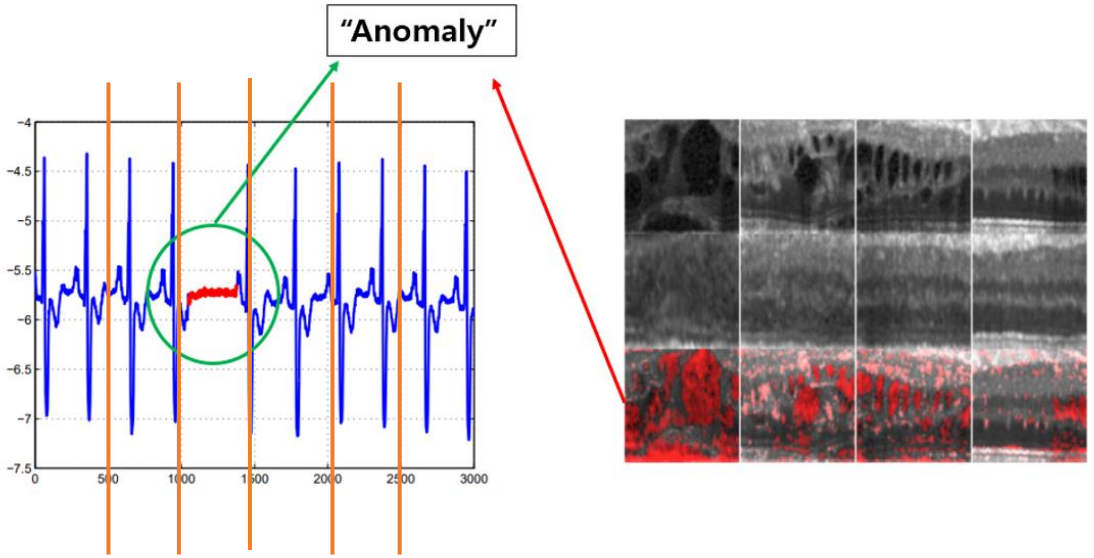
# Background

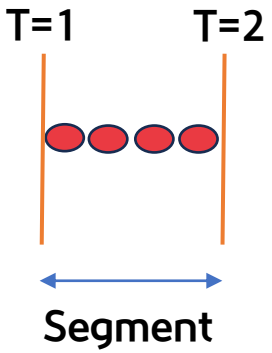
## 시계열 데이터에서 이상현상(Anomaly)이란?

정의 : 시스템이 비정상적으로 동작하는 시점(point) 또는 기간(collective)

구성 요소

- 입력 데이터 $X = (x^1, x^2, \ldots, x^T)$
- Data Points (점)
- Collective Data (집단)
- Segment (임의의 시간길이로 나눈 공간)



# Background

## 시계열 데이터에서 이상현상(Anomaly) 탐지 방법

| 구분 | 모델 종류 | 특징 |
|---|---|---|
| 임계값 | - | 시간적 상관관계 알 수 없음 |
| 군집화 | KNN | 이상 지속 시간과 이상 지속 개수 알아야 함 |
| 예측 | 통계적 (ARIMA) | 매개 변수에 따라 값이 크게 바뀜<br>광범위한 도메인 지식 필요 |
| | 인공지능 (LSTM, RNN) | 현재-과거 데이터 비교 후 미래 예측<br>이상치 데이터를 잘 학습해서 과적합될 가능성 큼 |

# Background

## 시계열 데이터에서 이상현상(Anomaly) 탐지 시 고려 사항

- 이상현상에 대한 사전지식이 없음.

= 최종 사용자가 경험을 통해 패턴 결과를 검수해야 함. => 비지도 학습으로 해결

- 기준선은 상황에 따라 변할 수 있음.

= 지도학습은 과거 데이터와 현재 데이터의 편차만을 고려 함, 미래 데이터에 대한 정보가 없음.
=> 현재 데이터를 재생성 => 딥러닝으로 해결

- 명확한 의미 있는 세그먼트를 분할하는 기준이 없음

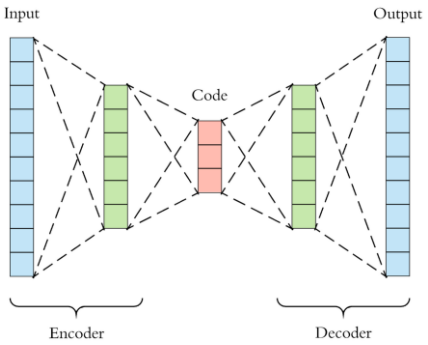
= 수많은 주기적인 시계열 신호로 분할 가능, 세그먼트 클러스터는 다양한 패턴을 나타낼 수 있음.

⇒ 알고리즘 + 인간 전문가 → *"확정된 이상치"*를 검토해 효용성을 검증해야 함.

# TadGAN

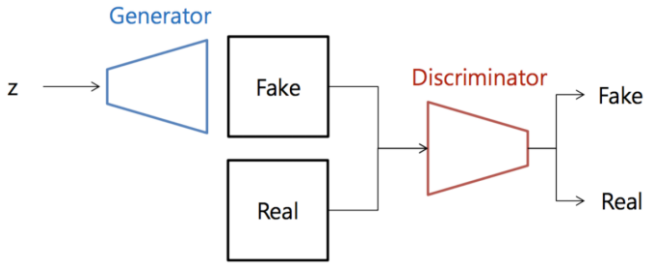
## AutoEncoder + GAN = “TadGAN”

**AE 특징**
- 저차원 공간에서 정보 손실
- 이상치에 민감
- Decoder에서 이상치는 복구 되지 않음!




AutoEncoder

+




GAN

**GAN 특징**
- 생성기(Generator) 학습 시 숨겨진 분포를 알아내기 어려움,
- 원본과 유사하게 만들 수 있음.

예측

재생성

TadGAN

AE → 저차원 공간을 활용해 숨겨진 분포 생성
GAN → 데이터를 복원/학습
=> 패턴을 복원하고 학습해 새로운 데이터 기반 예측.

# TadGAN

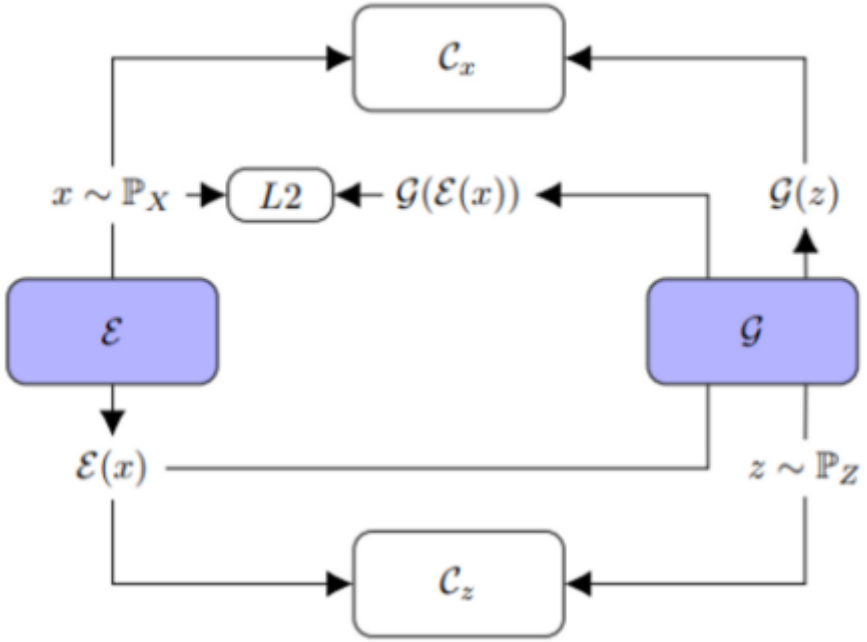
## TadGAN 구성 요소 및 아키텍처

- TadGAN 구성 요소
  - Input Data : X
  - Latent Data : Z
  - Train Data : x1,x2,… / z1,z2,…

  - 생성기 (E,G)
    - E : 시계열 Data → Latent Data
      = Encoder : 매핑을 잘하도록
    - G : Latent Data → 시계열 Data
      = Decoder : 유사 데이터를 생성하도록

    => x → E(x) → G(E(x)) → $\hat{x}$
  - 판별기 (Cx,Cz)
    - Cx : 실제 데이터 X와 G(z)의 구분,
      = G(z)가 원본 데이터를 잘 생성 했는지?
    - Cz : Mapping E 평가
      = E가 Latent를 잘 생성했는지?

백색소음(white noise)에 G(생성기) → C(식별기) ⇒ GAN 학습

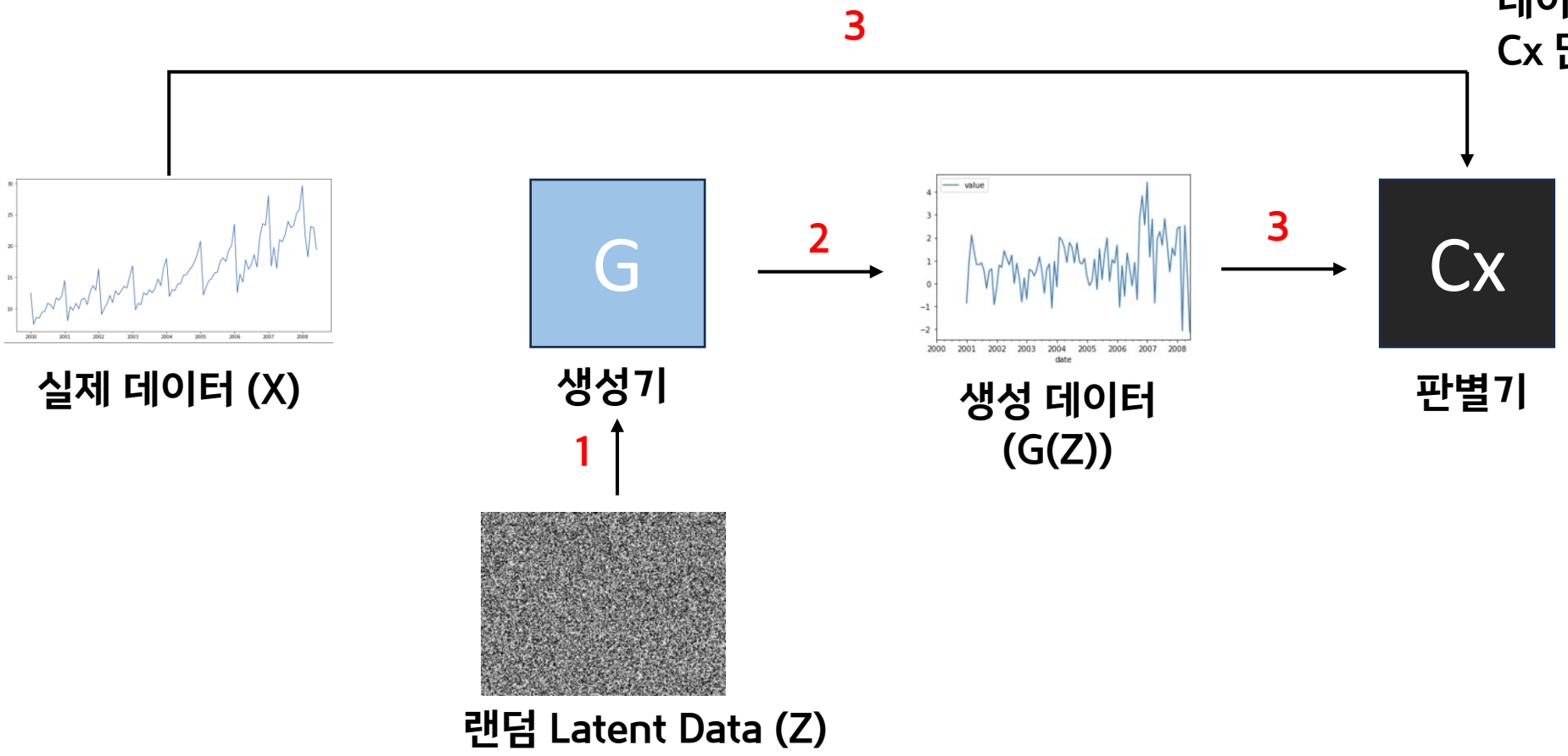



TadGAN 아키텍처

# TadGAN 동작 원리 (4단계)

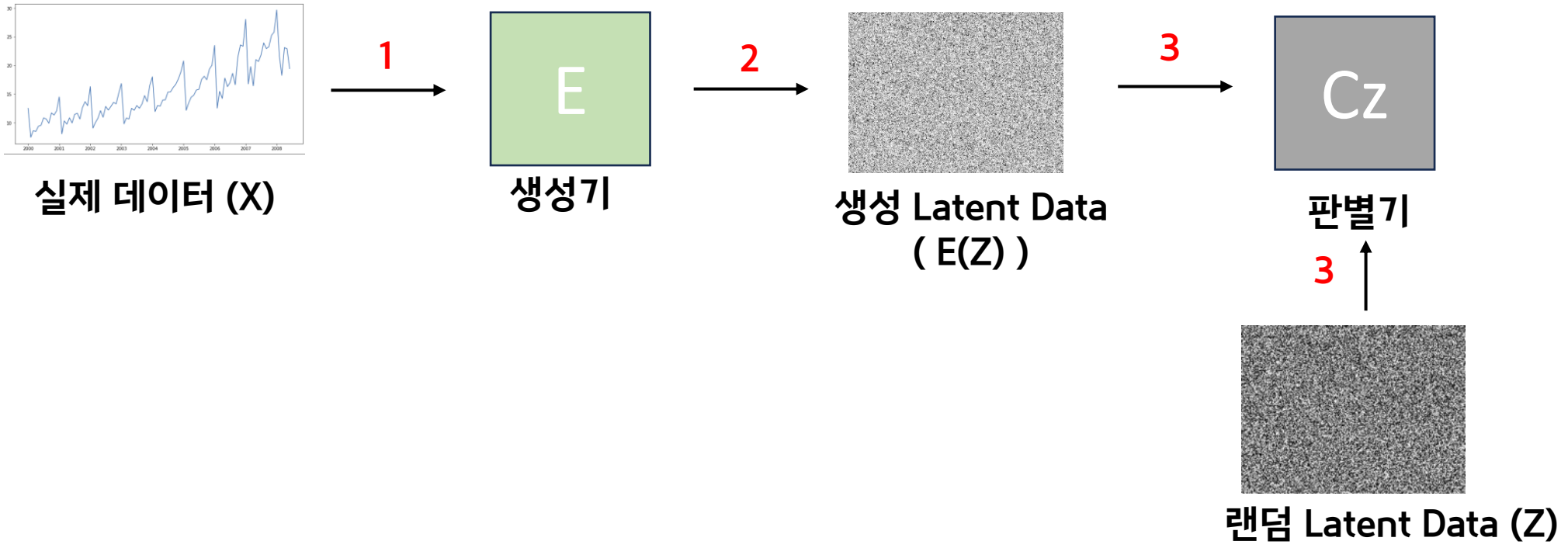
## 1단계 : 생성기 (G) 학습

생성 데이터가 실제 데이터와 같아질수록 Cx 많이 틀림



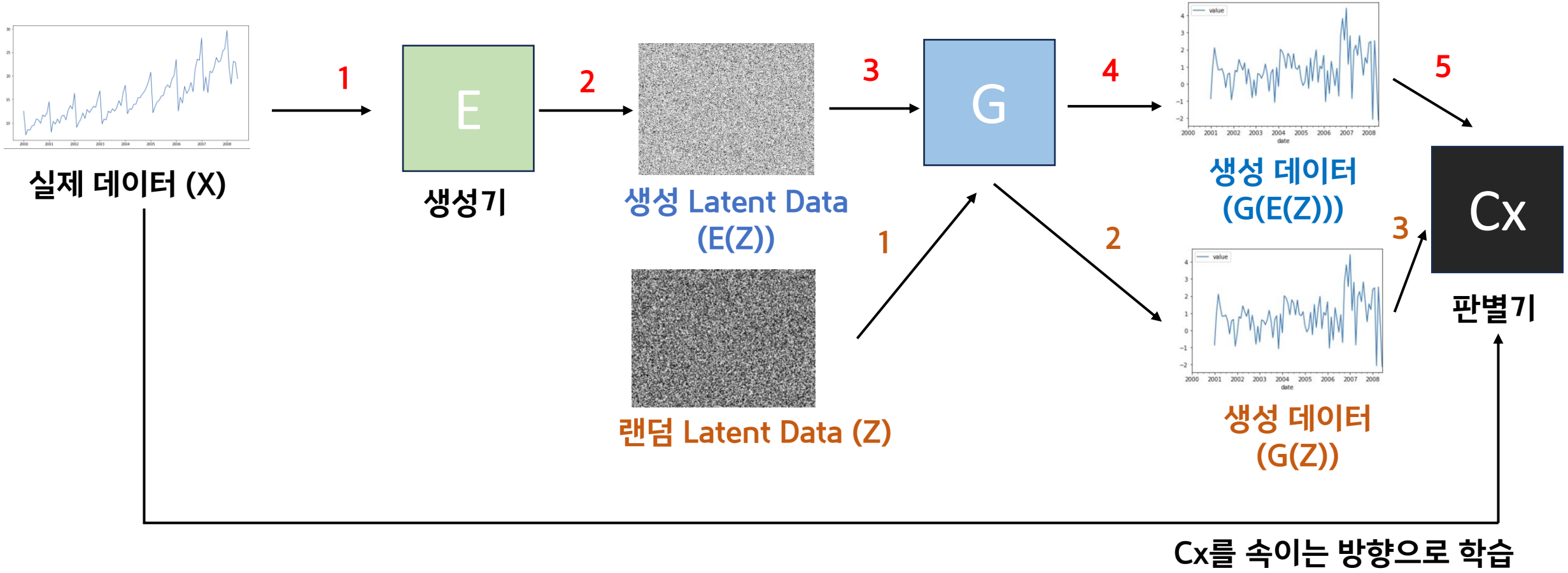
# TadGAN 동작 원리 (4단계)

## 2단계 : 생성기 (E) 학습



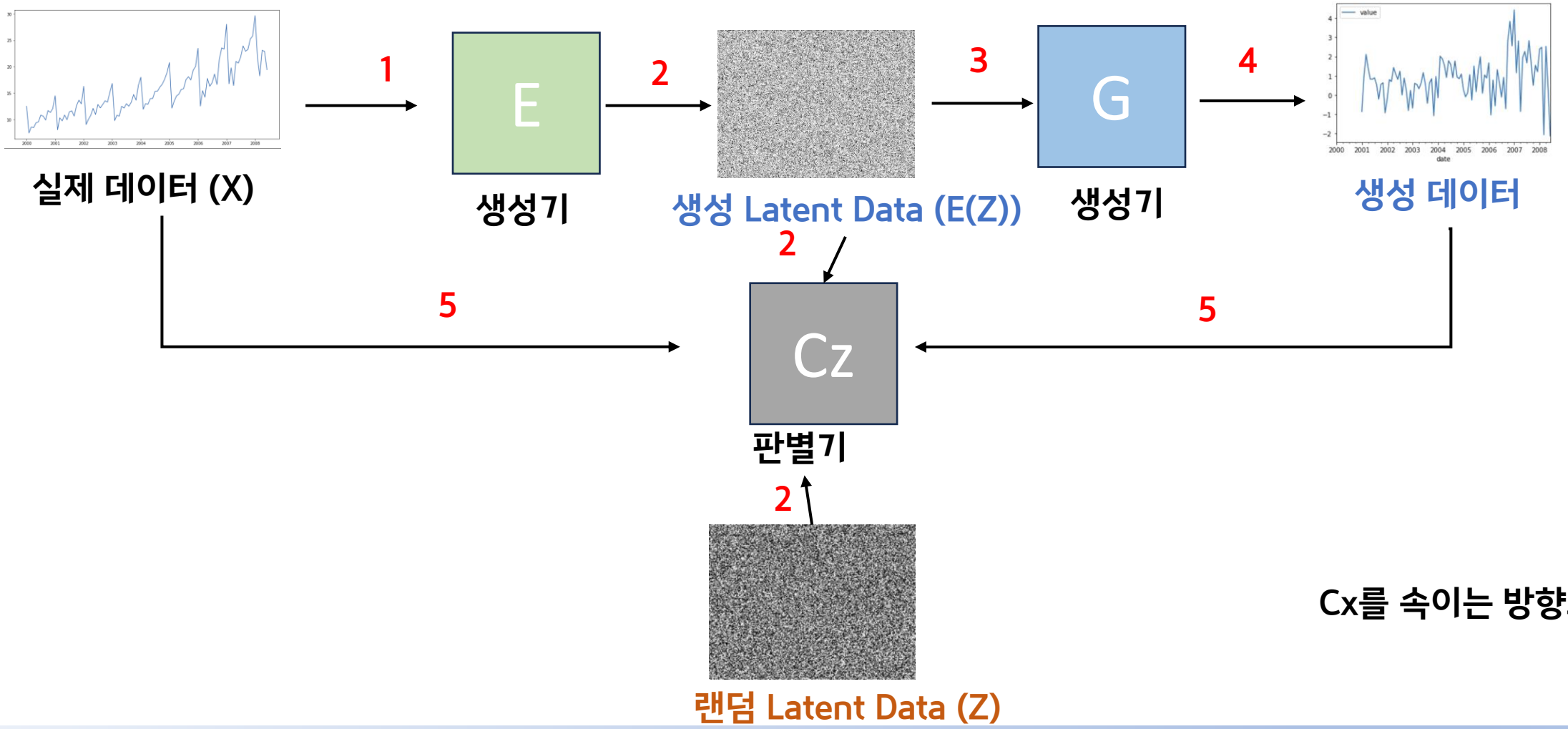
# TadGAN 동작 원리 (4단계)

## 3단계 : 판별기 (Cx) 계산



# TadGAN 동작 원리 (4단계)

## 4단계 : 판별기 (Cz) 계산

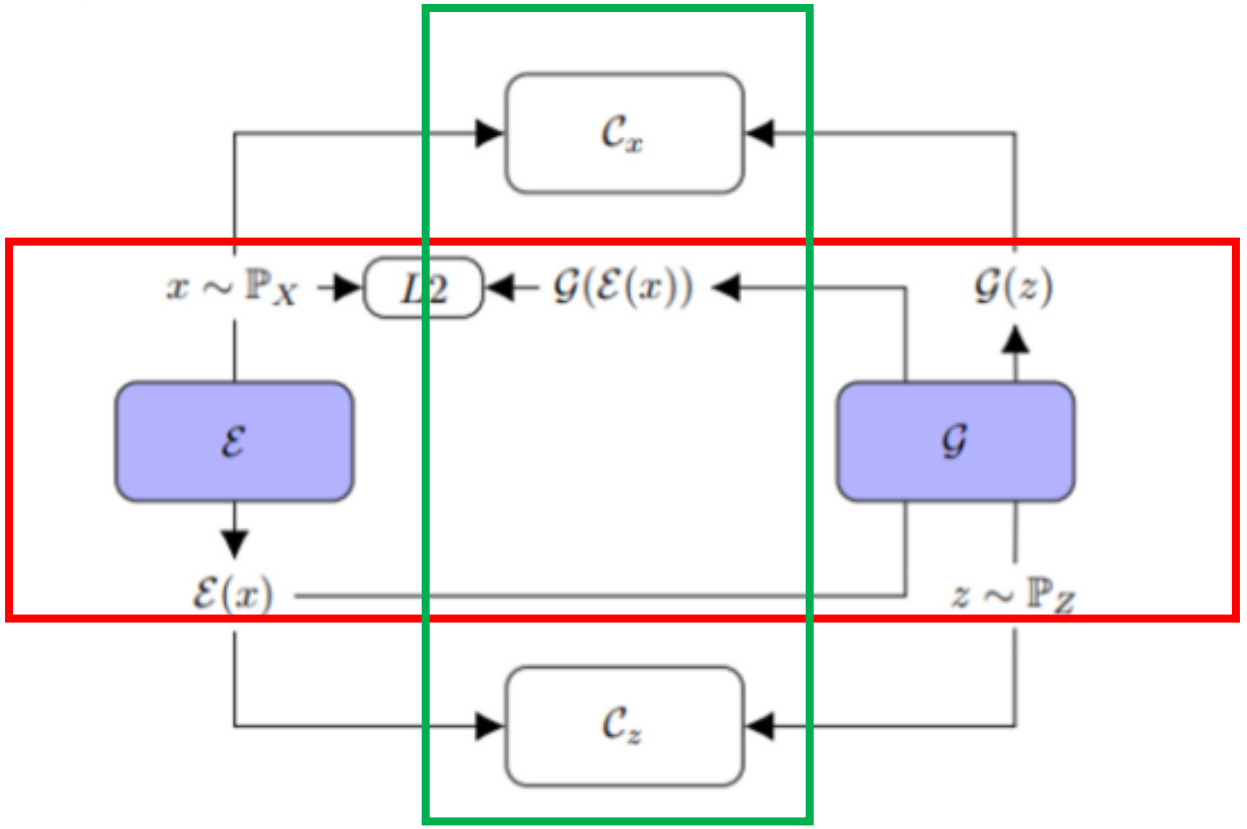



Cx를 속이는 방향으로 학습

# TadGAN 아키텍처

TadGAN 아키텍처를 다시 보면..

원본 vs 노이즈
데이터로 판별기 학습




교차 검증 느낌!

생성기 로 생성한
결과의 차이를 최소화
하는 방향으로 학습

# TadGAN Loss

Wasserstian Loss + Cyclic Consistency Loss → Mapping Space 제한

↓

Mode Collapse 문제 해결

$$L = \mathbb{E}_{x\sim\mathbb{P}_x}[\log \mathcal{C}_x(x)] + \mathbb{E}_{z\sim\mathbb{P}_z}[\log 1 - \mathcal{C}_x(\mathcal{G}(z))]$$

↓ 변경 후

$$L = \mathbb{E}_{x\sim\mathbb{P}_x}[\mathcal{C}_x(x)] - \mathbb{E}_{z\sim\mathbb{P}_z}[\mathcal{C}_x(\mathcal{G}(z))],\ \mathcal{C}_x \in \mathcal{C}_X$$

↓ (Cx, G), (Cz, E), (E,G) 고려

**Full Objective**

$$\min_{\{\mathcal{E},\mathcal{G}\}} \max_{\{\mathcal{C}_x\in\mathbf{C}_x,\mathcal{C}_z\in\mathbf{C}_z\}} V_X(\mathcal{C}_x,\mathcal{G}) + V_Z(\mathcal{C}_z,\mathcal{E}) + V_{L2}(\mathcal{E},\mathcal{G})$$

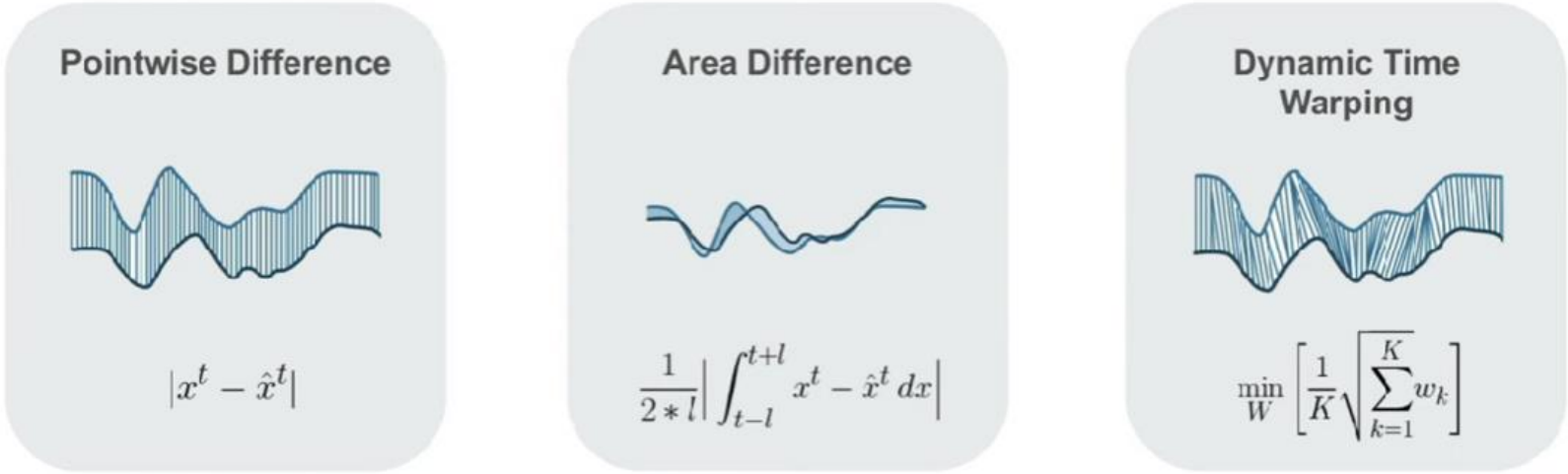
Wasserstian Loss  Cyclic Consistency Loss

# TadGAN 이상치 측정 지표

## RE(x) : Reconstruction Error

생성한 신호와 원본 신호, 두 신호 간의 차이

- Point Difference : 맨하탄 거리 차이 계산
- Area Difference : 특정 영역 차이 계산
- Dynamic Time Warping (DTW) : 원본과 생성한 시퀀스 간의 유사도 측정




얼마나 실제 신호와 같은 분포를 유지하는지?

# TadGAN 이상치 측정 지표

## Cx(x) : Critic Outputs

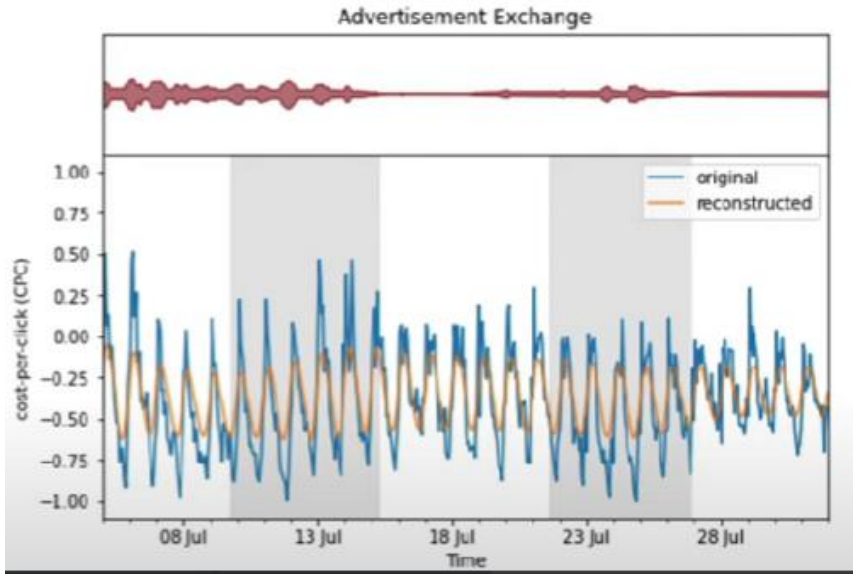
실제일 것이라는 신뢰성을 보장

Critic 구하는 방법
1. 특정 구간을 Segment로 나눔

2. Time Step 마다Cx(x)의 출력값을 이용해 시퀀스 크기를 정상 영역과 비교 후 score로 저장

3. Segment별로 KDE(커널 밀도 추정)해 최대 score를 Anomaly score set에 저장

4. 내림차순 정렬 후 이상치 감소율 구해서 이상치 찾기.




얼마나 실제 신호랑 비슷한지?

# TadGAN 이상치 측정 지표

## Combining Scores (RE(x), Cx(x))

2가지 이상치 측정 지표를 산술
- RE는 높을수록, Cx는 낮을수록 이상치일 가능성이 높음.
- 모두 Z 정규화 이 후 계산

Combine 방법
1. 더하기

$$\boldsymbol{a}(x) = \alpha Z_{RE}(x) + (1-\alpha) Z_{C_x}(x)$$

2. 곱하기

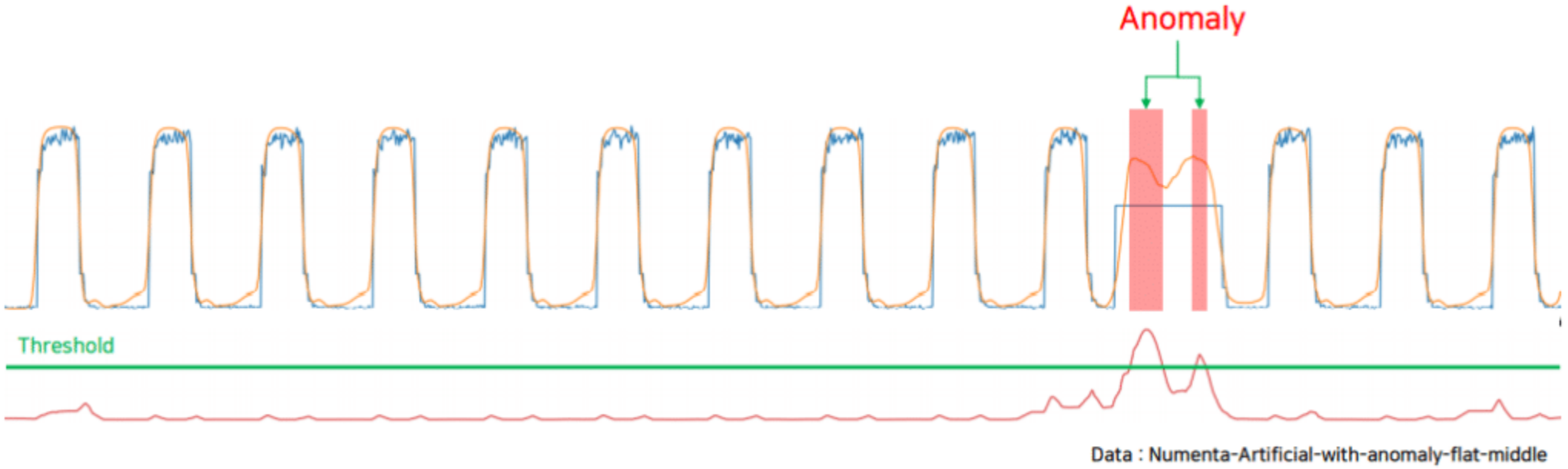
$$\boldsymbol{a}(x) = \alpha Z_{RE}(x) \odot Z_{C_x}(x)$$

알파($\alpha$)는 상대적 가중치

# TadGAN 이상치 판단

## Threshold로 판단

- Threshold = Sliding Window 내의 $\mu \pm 4\sigma$ 를 기준으로 함 (4 표준편차)
- Window size 는 과거 몇 개의 데이터로 Anomaly를 판단할 것인지를 의미



# 최종 정리

**Algorithm 1:** TadGAN

**Require:** $m$, batch size.
$epoch$, number of iterations over the data.
$n_{critic}$, number of iterations of the critic per epoch.
$\eta$, step size.

**for** *each epoch* **do**
  **for** $\kappa = 0, \ldots, n_{critic}$ **do**
    Sample $\{(x_i^{1\ldots t})\}_{i=1}^m$ from real data.
    Sample $\{(z_i^{1\ldots k})\}_{i=1}^m$ from random.
    $g_{w_{\mathcal{C}_x}} = \nabla_{w_{\mathcal{C}_x}}[\frac{1}{m}\sum_{i=1}^m \mathcal{C}_x(x_i) - \frac{1}{m}\sum_{i=1}^m \mathcal{C}_x(\mathcal{G}(z_i)) + gp(x_i, \mathcal{G}(z_i))]$
    $w_{\mathcal{C}_x} = w_{\mathcal{C}_x} + \eta \cdot \texttt{adam}(w_{\mathcal{C}_x}, g_{w_{\mathcal{C}_x}})$
    $g_{w_{\mathcal{C}_z}} = \nabla_{w_{\mathcal{C}_z}}[\frac{1}{m}\sum_{i=1}^m \mathcal{C}_z(z_i) - \frac{1}{m}\sum_{i=1}^m \mathcal{C}_z(\mathcal{E}(x_i)) + gp(z_i, \mathcal{E}(x_i))]$
    $w_{\mathcal{C}_z} = w_{\mathcal{C}_z} + \eta \cdot \texttt{adam}(w_{\mathcal{C}_z}, g_{w_{\mathcal{C}_z}})$
  **end**
  Sample $\{(x_i^{1\ldots t})\}_{i=1}^m$ from real data.
  Sample $\{(z_i^{1\ldots k})\}_{i=1}^m$ from random.
  $g_{w_{\mathcal{G},\mathcal{E}}} = \nabla_{w_{\mathcal{G}}, w_{\mathcal{E}}}[\frac{1}{m}\sum_{i=1}^m \mathcal{C}_x(x_i) - \frac{1}{m}\sum_{i=1}^m \mathcal{C}_x(\mathcal{G}(z_i)) + \frac{1}{m}\sum_{i=1}^m \mathcal{C}_z(z_i) - \frac{1}{m}\sum_{i=1}^m \mathcal{C}_z(\mathcal{E}(x_i)) + \frac{1}{m}\sum_{i=1}^m \|x_i - \mathcal{G}(\mathcal{E}(x_i))\|_2]$
  $w_{\mathcal{G},\mathcal{E}}, = w_{\mathcal{G},\mathcal{E}} + \eta \cdot \texttt{adam}(w_{\mathcal{G},\mathcal{E}}, g_{w_{\mathcal{G},\mathcal{E}}})$
**end**
$X = \{(x_i^{1\ldots t})\}_{i=1}^n$
**for** $i = 1, \ldots, n$ **do**
  $\hat{x}_i = \mathcal{G}(\mathcal{E}(x_i))$;
  $RE(x_i) = f(x_i, \hat{x}_i)$;
  $score = \alpha Z_{RE}(x_i) + (1-\alpha) Z_{\mathcal{C}_x}(\hat{x}_i)$
**end**

## TadGAN : GAN + AutoEncoder

- 시계열 데이터 이상치 탐지에 특화
- 패턴 복원 학습을 반복해 최대한 원본 데이터를 모방시킴
- 저차원 공간 옮긴 후 고차원 공간 이동 시 이상치 특징 사라진다고 가정

구조
- 2개의 생성기(Generator)와 2개의 판별기(Critics)가 존재
  - Latent/Time Data 종류

Loss
- RE(x)와 Cx(x)가 있음.
- 각각 거리별 유사성, 맥락적 유사성을 의미

장점
- 비지도학습 : 라벨 없어도 됨
- 순환 일관성 : 자기 자신을 학습하기에 잡음 및 변형에 강함

# 참고 자료

TadGAN Github : https://github.com/sintel-dev/Orion

TadGAN Pytorch 구현 : https://github.com/arunppsg/TadGAN