

시계열 데이터란?

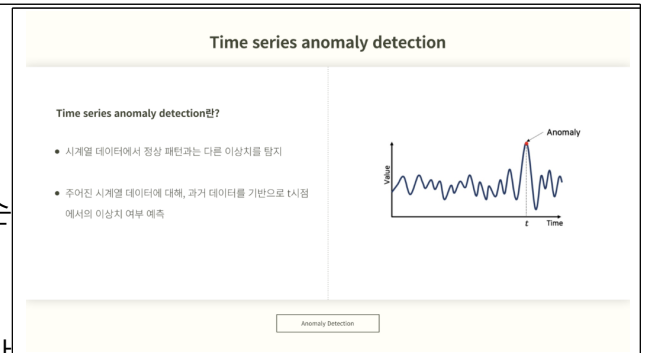
관측치가 시간적 순서를 가진 데이터 세트의 집합

- 단변량 시계열 : 동일한 간격의 시간의 증가에 대해 순차적으로 기록된 한 개의 변수 관측치로 구성된 시계열

- 다변량 시계열 : 동일한 간격의 시간의 증가에 대해 순차적으로 기록된 두 개 이상의 변수 관측치로 구성된 시계열

- 시계열 데이터에 대한 이상 탐지의 어려움

1. 이상 유형이 다양함
2. 정상 비정상 데이터를 정확히 구분하여 라벨링하기가 어려움
3. 정상 데이터에 비해 비정상 데이터가 적기 때문에 데이터 불균형 문제 발생



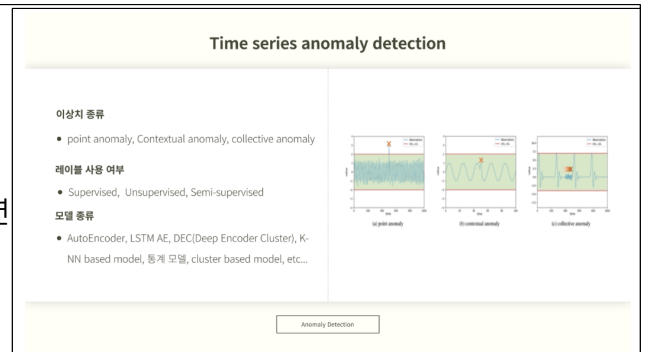
- 시계열 데이터의 이상 현상 정의

시스템이 비정상적으로 작동하는 시점 또는 기간으로 정의

1. 시점 이상 : 비정상적인 값에 도달한 단일 시점 또는 연속적인 시점의 집합
2. 기간 이상 : 특정 구간에서 이상 현상 발견

시점 이상 : Point Anomaly

구간 이상 : Contextual Anomaly, Collective Anomaly



- AutoEncoder 개념

입력 데이터를 재복원하는 비지도학습 방법론

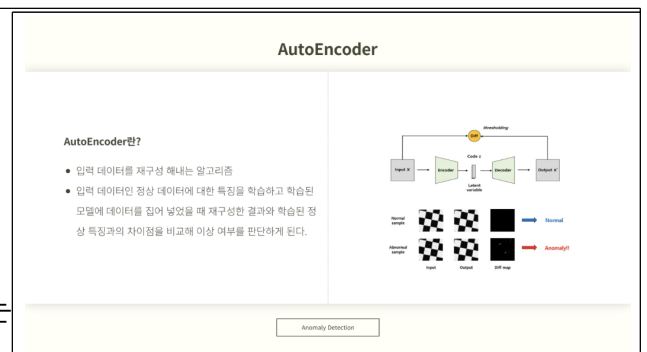
입력된 값과 비슷하게 출력되도록 재구성하는 것이 목표

두 개의 구조로 구성 : Encoder + Decoder

Encoder : 데이터를 압축하여 중요한 정보만 살리는 feature로 변환

Decoder : 압축된 feature를 이용하여 입력 데이터를 최대한 원본과 가깝게 복원

- 비정상 데이터는 학습되지 않았기에 복원된 데이터가 입력과 많은 차이 발생
- 복원 에러 = 입력 데이터 - 복원 데이터
- 임계치를 정하고, 복원 에러가 임계치를 넘게 되면 이상 탐지



- LSTM-AutoEncoder

sequence 데이터를 위한 LSTM 구조를 사용하는 Autoencoder

시간적인 특성을 고려, 이전 정보를 활용하는 LSTM 네트워크 사용

- AutoEncoder 모델의 한계

정상데이터만으로 학습하기 때문에, 다른 데이터가 들어와도 Training set과 비슷하게 만드는 과적합 문제 발생

LSTM AutoEncoder

LSTM AE

- 기존의 AutoEncoder 네트워크 셀을 LSTM셀로 대체한 것

LSTM AE의 장점

- LSTM셀로 대체함으로써 Time Serise Data에서의 시간 종속성을 보존할 수 있다.

Anomaly Detection

Error > threshold 이면 이상치라고 판단,
 Error < threshold 이면 정상치라고 판단한다.

LSTM AutoEncoder

LSTM AE Anomaly Detection

- LSTM AE 모델의 데이터 입력
- 입력 데이터가 Encoder와 Decoder를 거치면서 재구성된다.
- 재구성된 데이터와 실제 데이터의 차이(Error)를 구하고

Error > threshold 이면 이상치라고 판단,
 Error < threshold 이면 정상치라고 판단한다.

Anomaly Detection

- Best FI search

Test 데이터셋에서 가장 높은 성능을 내는 threshold를 찾는 방법

-> unsupervised training에서는 성능을 측정하기 어려움

- Quantile

데이터의 분포에서 특정 백분위수(quantile) 값을 기준으로 이상치를 탐지하는 것을 목표로 한다.

- peaks over threshold(POT)

1. 임계값 설정: POT 방법에서는 분석하려는 데이터에서 임계값을 설정합니다. 이 임계값은 데이터의 특성과 목적에 따라 결정됩니다.

2. 극값 탐지: 임계값을 초과하는 극값들을 탐지합니다. 극값은 데이터의 꼬리 부분에 위치한 값으로, 데이터 분포의 극단적인 값들을 의미합니다.

3. 극값 모델링: 탐지한 극값들을 활용하여 확률 분포를 모델링합니다. 주로 극값들은 일반적인 분포 가정을 만족하지 않으므로, 극단값 모델링에 적합한 통계 모델을 사용합니다.

4. 극단값 분석: 모델링된 분포를 기반으로 극단값의 특성을 분석합니다. 이를 통해 극단값의 빈도, 확률, 특정 사건의 발생 가능성 등을 추정할 수 있습니다.

- Dynamic error thresholds

threshold ϵ selected from the set:

$$\epsilon = \mu(\mathbf{e}_s) + z\sigma(\mathbf{e}_s)$$

Where ϵ is determined by:

$$\epsilon = \operatorname{argmax}(\epsilon) = \frac{\Delta\mu(\mathbf{e}_s)/\mu(\mathbf{e}_s) + (\Delta\sigma(\mathbf{e}_s)/\sigma(\mathbf{e}_s))}{|\mathbf{e}_a| + |\mathbf{E}_{seq}|^2}$$

Such that:

$$\begin{aligned}\Delta\mu(\mathbf{e}_s) &= \mu(\mathbf{e}_s) - \mu(\{e_s \in \mathbf{e}_s | e_s < \epsilon\}) \\ \Delta\sigma(\mathbf{e}_s) &= \sigma(\mathbf{e}_s) - \sigma(\{e_s \in \mathbf{e}_s | e_s < \epsilon\}) \\ \mathbf{e}_a &= \{e_s \in \mathbf{e}_s | e_s > \epsilon\} \\ \mathbf{E}_{seq} &= \text{continuous sequences of } \mathbf{e}_a \in \mathbf{e}_a\end{aligned}$$

$$\mathbf{e}_s$$

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score	0.25	0.76	0.33	0.86	0.22	0.65	0.56	0.43	0.73	0.23	0.73	0.95	0.23	0.46	0.82	0.93	0.78	0.63	0.23	0.23

$$\{e_s \in \mathbf{e}_s | e_s < \epsilon\}$$

Index	0	2	4	7	9	12	13	18	19
Score	0.25	0.33	0.22	0.43	0.23	0.23	0.46	0.23	0.23

$$\mathbf{e}_a = \{e_s \in \mathbf{e}_s | e_s > \epsilon\}$$

Index	1	3	5	6	8	10	11	14	15	16	17
Score	0.76	0.86	0.65	0.56	0.73	0.73	0.95	0.82	0.93	0.78	0.63

$$\mathbf{E}_{seq}$$

Index			5	6			10	11		14	15	16	17
Score			0.65	0.56			0.73	0.95		0.82	0.93	0.78	0.63

입실론의 값을 z값을 변형해가면서 가장 best의 입실론값(threshold)를 찾는 방법론

예시) 2번째 그림

초기 threshold의 값을 0.5로 했을때의 예시이다.

e(a) 집합은 threshold 보다 큰 error들의 집합이다.

E(seq)는 e(a) 집합에서 길이가 2이상인 연속된 e(a)인 집합이다. 하지만 본 논문에서는 E(seq)집합은 원소의 갯수로 보며, 해당 예시에서 E(seq)는 3이다.

error의 표준편차와 평균을 구해서 z값에 따른 입실론중에서 가장 큰 입실론이 최종 입실론이 된다. 그리고 본 논문 저자의 말에 따르면 z값은 2.5에서 10사이에서 가장높은 성능을 보인다고 언급하였다.

Threshold Search

Threshold Search 기법

- Best FI search
 - Test 데이터셋에서 가장 높은 성능을 내는 threshold를 찾는 방법
 - > unsupervised training에서는 성능을 측정하기 어려움
- Quantile
 - Anomaly score의 상위 p 퍼센트를 anomaly로 표시하는 분위수 접근 방식을 사용하는 방법
- peaks over threshold(POT)
 - Heavy-tailed distribution로 anomaly score의 tail을 별도로 모델링하고 특정 p-value를 사용하는 방법
- Dynamic error thresholds
 - non-parametric approach로 smoothed errors의 기초 통계량을 활용해서 extrem value를 찾는 방법

Anomaly Detection

dynamic threshold 논문에서는 Exponential weight average smoothing된 error값을 요구한다.

- Exponential weight average smoothing

$$EWMA_t = \alpha * r_t + (1 - \alpha) * EWMA_{t-1}$$

alpha : 사용자가 정한 가중치값

r : 현재 기간의 값

본 논문의 official code에서는 alpha의 값을 pandas ewm의 default 값인 0.5를 사용한것을 확인했다.

Exponential weight average smoothing은 이전 시점에 가중치는 덜 부여하기 위해 고안된 방법이다.

Exponential weight average smoothing

Exponential weight average smoothing

$$EWMA_t = \alpha * r_t + (1 - \alpha) * EWMA_{t-1}$$

- alpha : 사용자가 정한 가중치
- r : 현재 기간의 값

$$EWMA_t = \alpha * r_t + (1 - \alpha) * (EWMA_{t-1} + (1 - \alpha) * (EWMA_{t-2} + (1 - \alpha) * (EWMA_{t-3} + \dots)))$$

- 이전 가중치로 다시 계산하여 식을 확장할 수 있다.
- 해당 계산식은 EWMA(0)까지 계산한다.

따라서 $r(t-k)$ 의 가중치는 다음과 같이 계산될 수 있다.

$$\alpha * (1 - \alpha)^k$$

alpha의 값이 0 ~ 1사이 이므로 k값이 커지면 가중치의 값이 작아진다.
따라서 더 오래된 이전 값일 수록 더 낮은 가중치가 부여된다.

Anomaly Detection

dynamic threshold

dynamic threshold 식

threshold ϵ selected from the set:

$$\epsilon = \arg\min_{\epsilon \in \{e_1, e_2, \dots, e_n\}} \text{err}(\epsilon)$$

Where ϵ is determined by:

$$\epsilon = \arg\min_{\epsilon \in \{e_1, e_2, \dots, e_n\}} \left(\frac{\sum_{t=1}^n |e_t - \mu|}{n} + \frac{\sum_{t=1}^n |e_t - \mu|}{n} \right)$$

Such that:

$$\begin{aligned} \mu &= \frac{1}{n} \sum_{t=1}^n e_t \\ \sigma &= \sqrt{\frac{1}{n} \sum_{t=1}^n (e_t - \mu)^2} \\ e_1 &= \mu + \sigma * \epsilon \\ e_2 &= \mu - \sigma * \epsilon \end{aligned}$$

- ϵ : threshold
- e_1 : smoothing 된 error data
- e_2 : 이상치 데이터
- $E[\text{seq}]$: 연속된 이상치의 경우

threshold를 0.5로 설정했을 때의 예시

Anomaly Detection

해당 모델의 구성은
구본근 외 (2022) LSTM 오토인코더를 이용한 이상 탐지의
임계치 결정 방법

논문의 모델 테스트 환경을 참고하여 구현하였습니다.

NYC DataSet Anomaly Detection

데이터셋의 구성

NYC Taxi data

	count
mean	15137
std	6939
min	0
25%	10262
50%	16778
75%	19838
max	39197

- 학습데이터 구성
 - TrainDataSet : 처음 ~ 80230
 - TestDataSet : 80230 ~ 마지막
- 학습 파라미터 구성
 - LSTM window size : 30
 - latent dim : 20
 - learning rate : 0.001
 - optimizer : Adam
 - loss : mean squared error
 - batch size : 32
- Threshold 기법
 - Quantile : 상위 1%를 Threshold로 참고 Threshold보다 큰 value들을 이상치로 판단

Anomaly Detection

NYC DataSet Anomaly Detection

결과 분석

- point anomaly 과 Contextual anomaly의 대해서 탐지를 잘한 것 같다.
- 백분위수 상위1%이상과 하위 1%이하의 시점이 이상치로 판단되었다
 - 이러서만 골라낸 이상치가 아니지만 이상치라고 판단할 수 밖에 없는 경우가 있다고 판단하였다.

→ 따라서 실제 이상치만을 찾기 위해서는 백분위수가 적절하지 않다고 판단하였다
그리고 real time training에서는 사용하기가 어렵다..

예측 결과

Anomaly Detection

Real Time Training Simulation

결과

- Real Time에서도 백분위수를 사용하여 예측
 - 데이터가 추가됨으로 백분위수, 즉 Threshold를 계속해서 업데이트함
- error에 절대값을 취하고 상위 1%의 값만 이상치로 판단했음에도 많은 이상치가 탐지됨
- 해결책
 - dynamic threshold를 사용하여 상한마다 동적으로 threshold를 변환하여 상한의 맞는 anomaly를 찾아내자



예측 결과

Anomaly Detection

Real Time Training Simulation

Real Time에서 dynamic threshold의 문제점

- 학습 초기에 학습이 덜 된 모델의 error가 너무 크다
 - dynamic threshold 계산식에는 전체 error의 평균값과 표준편차를 사용하므로 초반에 큰 error값은 이후 이상치를 판별할 수 없게 만드는 요소가 된다.
- 해결책 : dynamic threshold를 계산할 때 전체 error를 계산하는 것이 아닌 $t - \text{window size} - t$ 까지의 데이터를 가지고 계산한다.
단, window size는 한 주기의 길이로 가환한다.



error(실제값 - 예측값) 계산

Anomaly Detection

Real Time Training Simulation

결과 분석

- Quantile 보다는 더 정확한 Point Anomaly 탐지
- Contextual anomaly 패턴에서의 심각한 문제
 - 이전 window size 데이터만으로 threshold를 구하므로 Point 패턴이 아닌 급격한 Contextual 패턴에서는 모델이 탐지해버린다.
- 해결책
 - error가 큰 구간은 초반에만 사용하고 그 이후에는 사용하지 않는다.
 - error가 안정화되는 구간에서 사용 종료
- 하지만 처리가 안정화 되는 시점을 어떻게 찾을 것인가?



error(실제값 - 예측값) 계산

Anomaly Detection

Real Time Training Simulation

분석 견해

- window size가 해당 모델에서 중요한 하이퍼미터로 생각되었다.
 - 해신의 주기를 대략적으로 알고 주기를 window size "alpha"로 한다면 더 좋은 결과가 나올 것 같다는 생각이 들었다.
 - 반면 알 수 없다면 데이터 분석 방법으로 추측할 수 있는 방법을 찾으면 좋다는 생각을 해보았다.
- 해당 실험에서는 10000개의 데이터만을 사용하였는데 더 많은 데이터에서도 dynamic하게 예측이 잘지는 모르겠다.
- 이해 낮은 집중된 환경이지만 10000개의 데이터를 real time simulation 하는데도 LSTM 특성상 오랜시간이 걸린다. 과연 이 모델을 실시간 하이퍼터에 적용할 수 있을지 미궁이다.

Anomaly Detection

Autoencoder 모델의 한계가 명확하기 때문에 더 발전된 모델이 필요하다고 느낌

- AutoEncoder에서 더 향상된 모델 제시

AutoEncoder within An Adversarial Training Framework

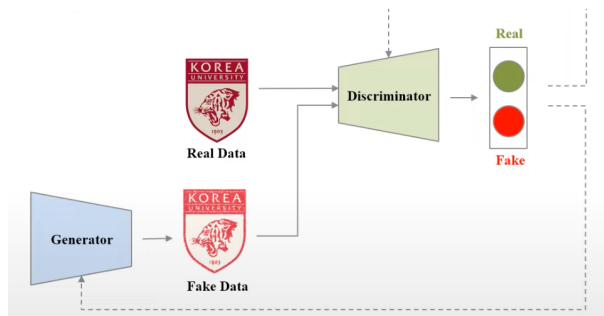
AUDIBERT, Julien, et al. (2020) USAD : unsupervised anomaly detection on multivariate time series

- GAN 기본 구조 : 생성기 + 판별기

생성기 : 원본 데이터와 유사한 가짜 데이터 생성

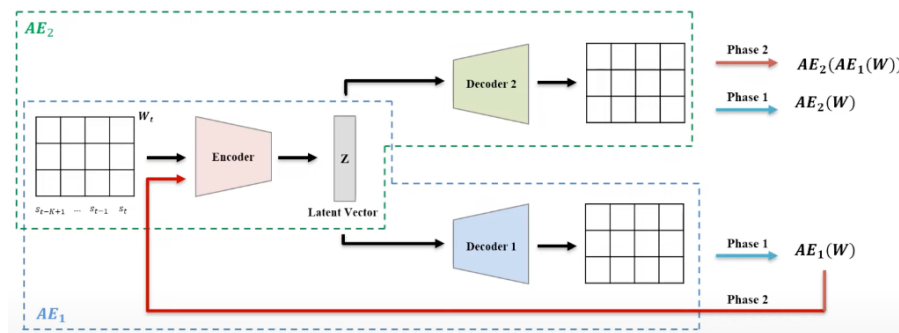
판별기 : 원본 데이터와 가짜 데이터를 구별

두 네트워크가 경쟁하며 서로의 역할을 발전시킴



- 제안 모델의 이점

AutoEncoder 한계 극복



AE1을 학습하여 AE2를 속일 수 있는 데이터 생성, 즉 실제와 비슷한 데이터를 생성하도록 학습
AE2를 학습하기 위해 입력값으로 AE1의 output을 사용 -> AE2가 실제 데이터와 가상 데이터인 AE1을 구분하도록 학습

AE1은 생성기, AE2는 판별기 역할을 수행한다.

- AE1

실제 데이터와 AE1의 생성데이터의 error를 최소화하도록 학습을 진행

- AE2

실제 데이터와 가상 데이터를 이용한 AE2의 생성 데이터의 error가 되도록 학습을 진행

Real Time Training Simulation

분석 견해

- window size가 해당 모델에서 중요한 파라미터로 생각되었다.
- 해당 window size를 대략적으로 알고 주기를 window size "alpha"로 만든다면 더 좋은 결과 가 나올 것 같다는 생각이 들었다.
- 만약 알 수 없다면 데이터 분석 방법론으로 추측할 수 있는 방법을 찾아보면 좋다는 생각을 해보았다.

- 해당 상황에서는 10000개의 데이터를 사용하였는데 더 많은 데이터에서도 dynamic하게 적용이 될지는 모르겠다.

- 이후 낮은 컴퓨팅 환경이지만 10000개의 데이터를 real time simulation 하는것도 LSTM 특성상 오랜시간이 걸렸다. 과연 이 모델을 실시간 빅데이터에 적용할 수 있을지 미지수이다.

Anomaly Detection