

# OmniAnomaly : Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network

≡ Author	Ya Su et al.
📅 read date	@2022년 6월 24일
📖 Journal	KDD
📎 PDF	<a href="#">[2019 KDD] Robust Anomaly Detection for Multivariate Time Series.pdf</a>
≡ Published Date	2019
≡ detail	
≡ keyword	Anomaly Detection MTS
🔗 link	<a href="https://dl.acm.org/doi/pdf/10.1145/3292500.3330672">https://dl.acm.org/doi/pdf/10.1145/3292500.3330672</a>
📌 status	Finished!

## Introduction

- 관측치가 정규 패턴과 다를수록 이상으로 간주될 가능성이 높음 → “2가지 해결필요 과제”
  - MTS의 temporal dependency(GRU)과 stochastic(VAE)을 모두 고려  
→ 정상적인 패턴을 포착하기 위한 로버스트한 z representation을 학습
  - 확률적 딥러닝 접근 방식(VAE) → 탐지된 개체-수준 이상에 대한 해석을 제공
- 기여점 정리
  - a novel stochastic RNN for MTS AD (VAE를 사용해서 stochastic 이라고 하는 듯)
    - 산업 장치 모니터링에 필요한 입력 데이터의 로버스트한 표현을 학습하기 위해 확률 variables간 explicit temporal dependence를 처리할 수 있는 최초의 MTS AD 알고리즘
  - 제안 모델 뿐만 아니라 다른 알고리즘과 함께 작동하는 확률 기반 MTS AD 알고리즘을 위한 **최초의 이상 해석 접근 방식 제안 (해석 정확도는 최대 0.89)**
  - 4가지 핵심 기술에서 큰 효과를 보여 줌
    - GRU
    - planar NF
    - stochastic variable connection(확률적 변수 연결)  
by Linear Gaussian State Space Model (SSM)
    - 자동 이상치 임계 값 선택을 위한 adjusted Peaks-Over-Threshold 방법

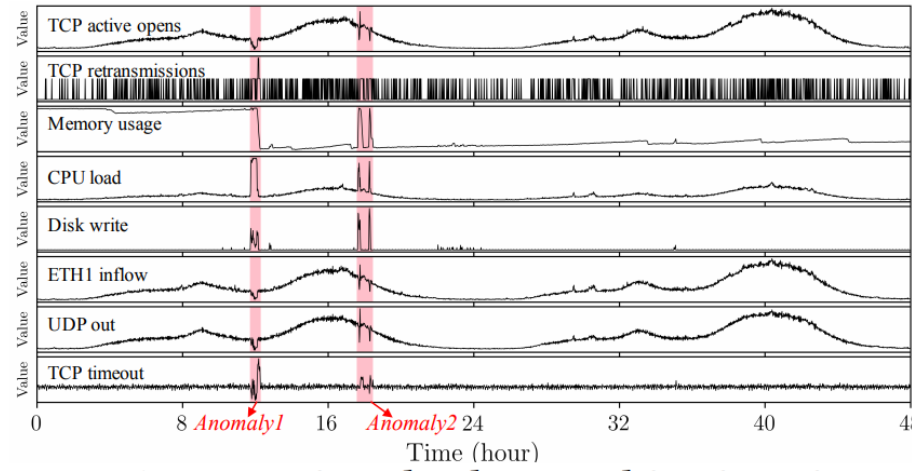
## RELATED WORK

- 제안 모델
  - 시계열의 시간적 의존성과 확률성을 명시적으로 모델링할 수 있도록 VAE와 GRU를 결합하는 확률적 RNN
  - 확률 변수 연결과 같은 기술(SSM)을 적용하여 확률 변수 사이의 시간적 의존성을 모델링
    - 확률 변수는 과거 확률 변수로부터 더 많은 정보를 포착 & 입력 데이터를 더 잘 나타낼 수 있다

## PRELIMINARIES(예비)

### 3.1 Problem Statement

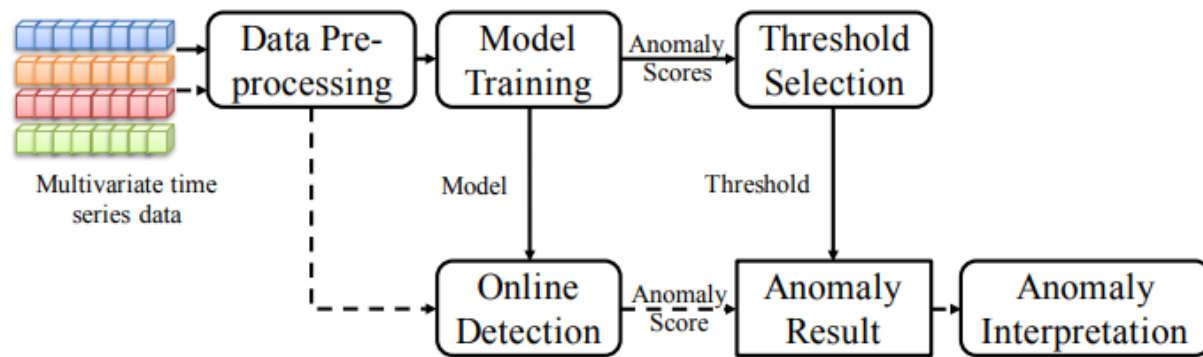
- MTS:  $x = \{x_1, x_2, \dots, x_N\}$ ,  $x_t = [x_t^1, x_t^2, \dots, x_t^M]$   $R^{M \times N}$  (M개의 변수, N개의 샘플)



**Figure 1: An 8-metric 2-day-long multivariate time series snippet from the server machine dataset, with two anomalous regions highlighted in pink.**

- M=8 values, N=48 hours인 데이터로 빨간색으로 칠해진 만큼이 time window 크기이고, 2개의 이상치 관측되는 샘플
- $N = 2\text{days} * 24\text{hours} * 60\text{ minutes}$  (분단위로 변경),  $T = \text{window size} \rightarrow x_{t-T:t}$

### 3.2 Overall Structure



- Fig 2. 설명
  - 직선 : 오프라인 훈련 / 점선 : 온라인 훈련 / 실행 단위 : 주 or 월마다
  - Data Preprocessing
    - : 전처리 과정에서 데이터는 표준화에 의해 변환된 다음 길이 T+1의 슬라이딩 윈도우를 통해 시퀀스로 분할 됨
  - Model training
    - : 전처리 된 데이터가 보내지며, MTS의 정규 패턴을 캡처하고, 각 관측치에 대한 이상 점수를 출력하는 모델을 학습함
  - Threshold Selection
    - : 이상치 점수는 해당 모듈에서 POT방법에 따라 자동적 임계값을 선택하는데 사용 됨
  - Online Detection
    - : 전처리 후 새로운 관측치가 나오면 이상치 점수를 가질 수 있게 input된다.
- $x_t$ 의 이상치 점수가 이상치 임계치보다 낮은 경우,  $x_t$ 는 이상치로 판별되고 그렇지 않으면 정상

### 3.3 Basics of GRU, VAE and Planar NF

- GRU
  - 매개변수가 LSTM보다적고 구조가 단순해서 데이터셋이 크지않은 경우 적합
  - 시계열에서 복잡한 시간 의존성을 포착 (complex temporal dependence)
- VAE
  - 계절성(빈도가 일정) 단별량 시계열 AD에 성공적으로 적용 되었었음(이전 연구)

- 명시적 확률적 모델링 가능
- 계산어려운 사후(Posterior)확률분포  $p(z|x)$ 를  $q(z|x)$  즉, 다루기 쉬운 일정 분포로 근사
  - $q(z|x) = N(\mu_p(x), \sum_q(x))$  (가우시안 분포를 따름)
  - 구조
    - $q_\phi(\cdot)$  = q-net = 인코더(inference net)
    - $p_\theta(\cdot)$  = p-net = 디코더(generative net)

- planar NF (invertible mapping functions)

- 목적 : To learn the non-Gaussian posterior density  $q_\phi(z_t|x_t)$

$$z_t^K = f^K(f^{K-1}(\dots f^1(z_t^0))), \text{ where } f^k(k = 1, 2 \dots K)$$

$$(f^k(z_t^{k-1}) = z_t^{k-1} + u \tanh(w^T z_t^{k-1} + b)) \quad u, w \text{ and } b \text{ are the parameters}$$

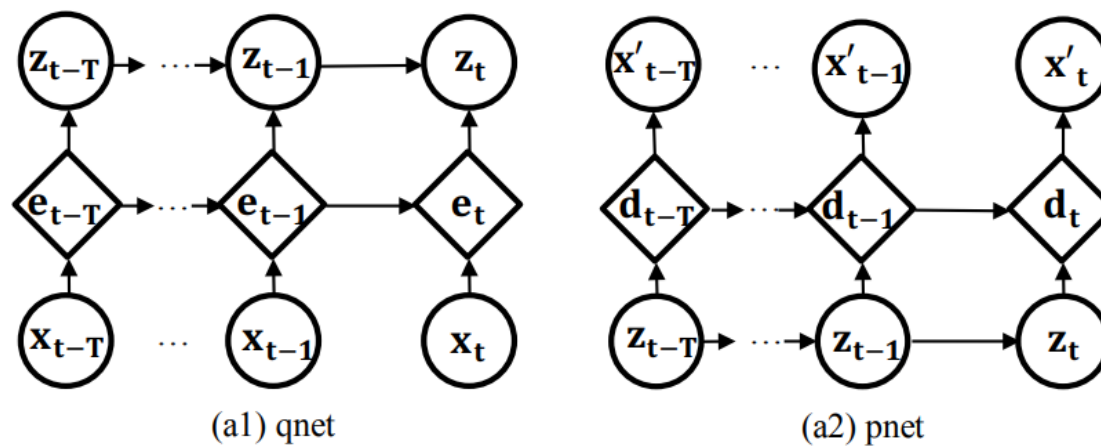
:  $z_0$  에 특정한 형태의 함수  $f^k$  들을 반복 적용하여 모델이 더욱 복잡한 형태의 잠재변수를 표현하게끔 만들어 줌

- q-net에서 planar NF의 최종 output인  $z_t = z_t^K$  을 확률변수로 사용
- 복잡하고 유연한 분포를 캡처해서 모델의 성능을 향상

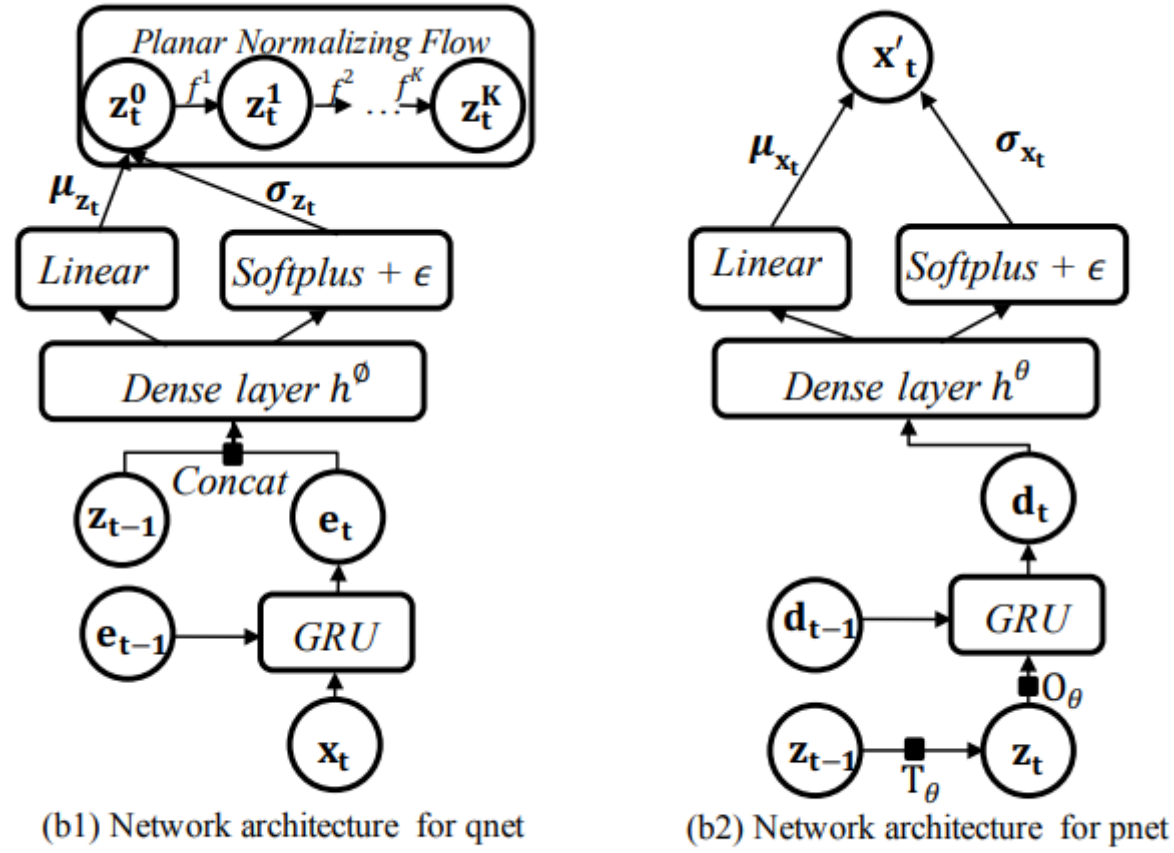
## DESIGN OF OMNIANOMALY

### 4.1 Network Architecture

1. GRU : x-space에 있는 MTS 관측치 사이 간 complex temporal dependence를 포착
2. VAE : 표현학습을 위해, x-space에 있는 관측치를 stochastic variables로(z-space) 맵핑
3. Linear Gaussian State Space Model (SSM) : 확률변수 연결
  - : latent space에서 확률 변수간의 temporal dependence을 명시적으로 모델링 함
  - stochastic variables간의 연결, stochastic variable과 GRU hidden variable의 연결
4. planar NF : qnet의 stochastic variables의 복잡한 분포 표현에 도움 되기 위해 사용



- $e_t, d_t$ : memory variables in GRU cells which are deterministic(일정한 결과)
- $z_t$  : z-space variable which is stochastic(랜덤한 결과, 불확실성)
- *edges*: represent the dependence between variables



(b) Detailed network architecture of *OmniAnomaly* at time  $t$ .

- qnet :  $x_t$ 를 input하여  $z_t$  생성
  - t시점의 input data  $x_t$ 와 이전 qnet의 hidden state  $e_{t-1}$ 를 GRU에 넣어서  $e_t$  산출
    - $e_t$  :  $x_t$ 와 이전 x-space 관측치 사이의 long-term complex temporal 포착
  - 이전 qnet에서 최종 산출 된  $z_{t-1}$ 과  $e_t$ 를 concat
    - dense layer h 에 넣어서 stochastic variable  $z_t$ 를 위한  $\mu_{z_t}, \sigma_{z_t}$ 를 산출
  - Gaussian  $N(\mu_{z_t}, \sigma_{z_t}^2 I) \rightarrow z_t^0$ 를 planar NF를 사용하여  $f^k$ 를 K번 변환
    - non-Gaussian을 따르는 확률 잠재 변수  $z_t^K = z_t$  산출
- pnet :  $z_t$ 를 사용하여  $x_t$ 와 같이 재구성한  $x'_t$ 를 얻고자 함
  - linear Gaussian SSM를 사용하여 z-space variables를 연결하고 z-space variables에 temporally dependent을 만들  $\rightarrow z_t$

$$z_t = O_\theta(T_\theta z_{t-1} + v_t) + \epsilon_t$$

- $O_\theta, T_\theta$  = transition, observation matrices
- $v_t, \epsilon_t$  = transition, observation noises
- qnet의  $z_{t-1}$ 를 활용하여  $z_t$ 를 만들어냄
- 왜  $z_t$ 를 새로 만들었을까?  $z_{t-1}$ 에 dependence하게 하려고!
  - 확률 변수 연결(SSM)을 적용하여 확률 변수 사이의 시간적 의존성을 모델링
  - 확률 변수는 과거 확률 변수로부터 더 많은 정보를 포착 & 입력 데이터를 더 잘 나타낼 수 있다
- 생성된  $z_t$ 와 이전 디코더의 hidden state  $d_{t-1}$ 를 GRU에 넣어서 현재 시점의  $d_t$  산출
- dense layer h layer에 넣어서 각각  $\mu_{x_t}, \sigma_{x_t}$ 를 산출
  - $N(\mu_{z_t}, \sigma_{z_t}^2 I)$  가우시안 분포를 따르는  $x'_t$ 를 복원 함

- 이상치라면  $x'_t$ 와  $x_t$ 는 크게 다름  $\rightarrow x_t$ 의 reconstruction 확률 기반으로 이상치 탐지

## 4.2 Offline Model Training

- 훈련은 offline에서 하며, 이때 GRU의 네트워크 파라미터를 업데이트한다.

- qnet and pnet은 파라미터를 조정하여 동시에 훈련됨
- Object Function : VAE에서 사용하는 ELBO objective Function 최적화

$$\tilde{\mathcal{L}}(\mathbf{x}_{t-T:t}) \approx \frac{1}{L} \sum_{l=1}^L [\log(p_{\theta}(\mathbf{x}_{t-T:t} | \mathbf{z}_{t-T:t}^{(l)})) + \log(p_{\theta}(\mathbf{z}_{t-T:t}^{(l)})) - \log(q_{\phi}(\mathbf{z}_{t-T:t}^{(l)} | \mathbf{x}_{t-T:t}))] \quad (5)$$

- 1st term : Negative reconstruction error (pnet 디코더)
- 2nd + 3rd term = regularization
  - 2nd term : Linear Gaussian SSM을 통해 계산 (pnet 디코더)
  - 3rd term : true posterior distribution 추정을 위한 loss term (qnet 인코더)

### 4.3 Online Detection

- 정상인지 이상인지를 확인하는 단계

$$S_t = \log(p_{\theta}(\mathbf{x}_t | \mathbf{z}_{t-T:t}))$$

- $S_t$  = anomaly score of  $x_t$ 
  - ↑(잘 재구성) = 정상과 가까움 (임계치 보다 높은 경우)
  - ↓(잘 재구성 못함) = 이상에 가까움 (임계치 보다 낮은 경우)

### 4.4 Automatic Threshold Selection

자동으로 임계치를 정하는 방법 : 되게 복잡한 통계적 기법 (별도 논문 多)

$$\{S_1, S_2, \dots, S_{N'}\}$$

: Anomaly Score  $S$ 를 EVT를 통해  $th_F$ 를 설정

- Extreme Value Theory (EVT) : 극한값의 법칙을 찾는 것이 목표인 통계 이론
  - 이상치의 점수가  $S_t$ 의 분포의 꼬리 부분에 위치할 것이라는 점을 기반으로 활용
- Peaks-Over-Threshold (POT) : EVT의 한 부분. 임계점(threshold)을 초과하는 값을 극단치로 정의
  - 모수를 사용하여 GPD(파레토 분포)에 의한 추정 통해 확률 분포의 꼬리 부분을 맞춤
  - **GPD function**

$$\bar{F}(s) = P(th - S > s | S < th) \sim (1 + \frac{\gamma s}{\beta})^{-\frac{1}{\gamma}}$$

: ( $th - S > s$  |  $th$  보다  $S$ 가 낮을 때) 확률 ~

- $th$  = 초기 설정 임계 값
- $th - S$  = 임계값 보다 낮은 부분 (낮은 분수 like 7% 이하) → 조절 필요
- $\gamma$  &  $\beta$  = GPD의 shape & scale 파라미터
  - $\hat{\gamma}, \hat{\beta}$  : Maximum Likelihood Estimation (MLE)로 추정

- **Final threshold**

$$th_F \simeq th - \frac{\hat{\beta}}{\hat{\gamma}} ((\frac{qN'}{N'_{th}})^{-\hat{\gamma}} - 1)$$

- $q = S < th$  를 관측하기 위한 원하는 확률 (like  $10^{-4}$ ) → 조절 필요



- $N' =$  관측치 수
- $N'_{th} = S_i(S_i < th \text{를 만족하는})$ 의 갯수

## 4.5 Anomaly Interpretation

$$S_t = \sum_{i=1}^M S_t^i, \text{ where } S_t^i = \log(p_{\theta}(x_t^i | z_{t-T:t}))$$

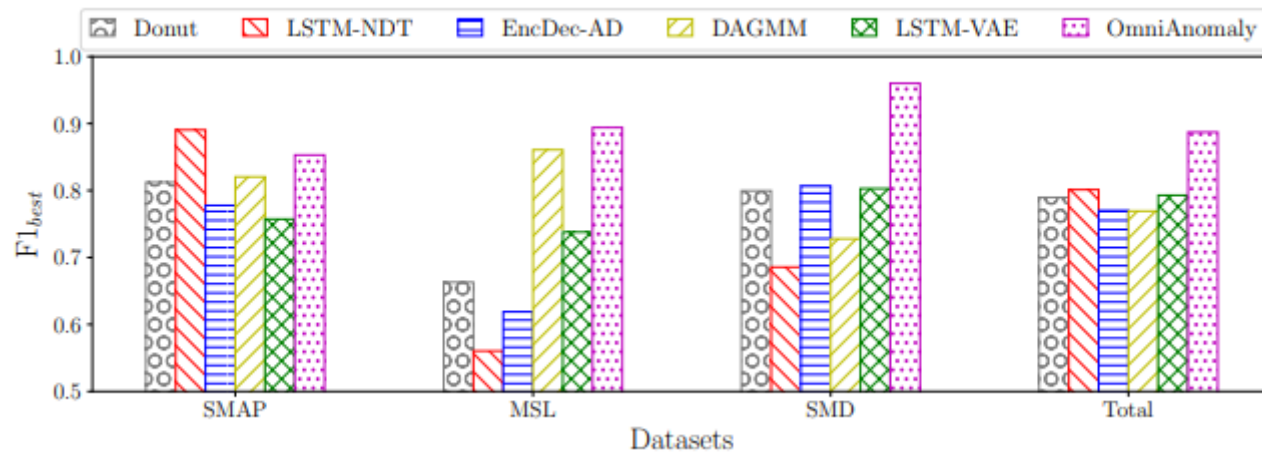
- 앞에서 계산했던 anomaly score는 위와 같이 표현할 수 있다.
- $S_t^i =$  Anomaly score of  $x_t^i$  ( $x_{t-T:t-1}$ 로 부터의 rich information을 갖고 있음)
- $i$ 부터  $M$ 까지의  $S_t^i$ 를 오름 차순으로 sorting  $\rightarrow$  list  $AS_t$ 를 얻음
  - 작은  $S_t^i =$  높은  $AS_t$  순위 =  $x_t^i$ 가  $x_t$ 에 높은 기여

## 5 EVALUATION

### 5.1 Datasets and Performance Metrics

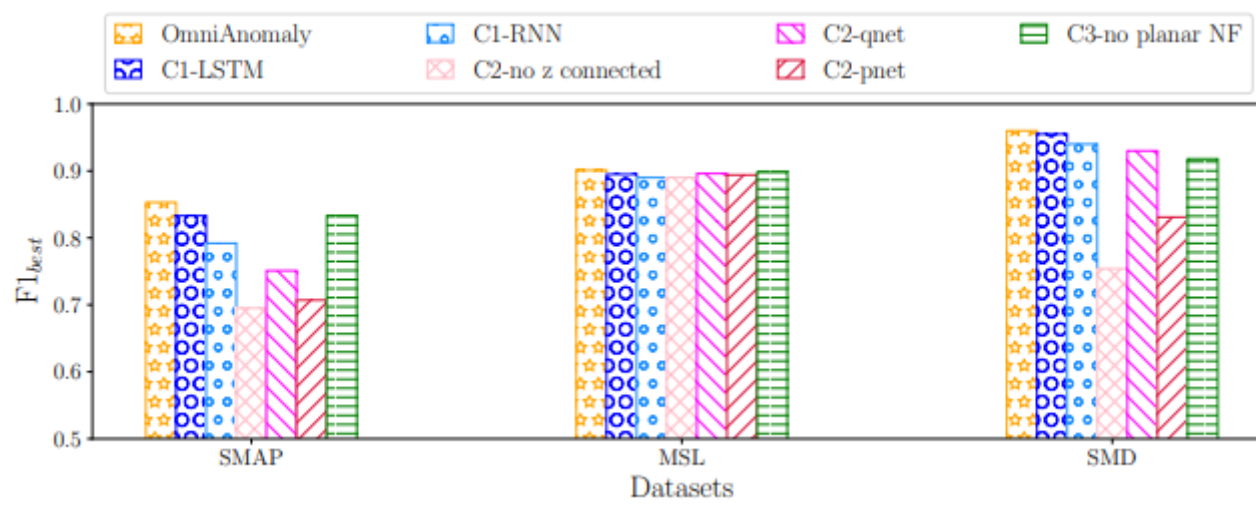
- three datasets
  - SMD (Server Machine Dataset)
  - SMAP (Soil Moisture Active Passive satellite) and MSL (Mars)
  - Science Laboratory rover)
- Performance Metrics
  - Precision, Recall, F1-Score (denoted as F1)

### 5.2 Results and Analysis



Methods	SMAP			MSL			SMD			Total		
	P (Precision)	R (Recall)	F1	P	R	F1	P	R	F1	P	R	F1
LSTM-NDT [6]	0.8965	0.8846	<b>0.8905</b>	0.5934	0.5374	0.5640	0.5684	0.6438	0.6037	0.7598	0.7794	<b>0.7694</b>
DAGMM [27]	0.5845	0.9058	0.7105	0.5412	0.9934	<b>0.7007</b>	0.5951	0.8782	0.7094	0.5835	0.9042	0.7093
LSTM-VAE [16]	0.8551	0.6366	0.7298	0.5257	0.9546	0.6780	0.7922	0.7075	<b>0.7842</b>	0.7782	0.7075	0.7411
OmniAnomaly	0.7416	0.9776	<b>0.8434</b>	0.8867	0.9117	<b>0.8989</b>	0.8334	0.9449	<b>0.8857</b>	0.7797	0.9586	<b>0.8599</b>

- 제안 모델이 대체적으로 가장 높은 점수를 보인다



- Ablation Study : C1~ C4
  - (1) GRU, (2) z-공간 변수 연결, (3) 평면 NF, (4) 자동 이상 임계값 선택을 위한 POT 방법
  - C2 SSM 사용이 많은 기여를 하고 있음을 볼 수 있다.

## 코드 실행 결과

<https://github.com/thuml/Anomaly-Transformer>

버전 맞추기 넘 힘들었다.... python down해야 함

<https://github.com/carrtesy/TSAD/blob/master/models/OmniAnomaly.py>

→ 해당 코드로 공부하는게 좀 더 직관적

모델을 살펴 본 결과 POT 부분은 이해 하기 정말 어려운 코드로 만들어져 있다.

qnet & pnet은 pytorch인 2번째 코드로는 간략하게 설명되어 있음