TadGAN: Time series Anomaly Detection Using Generative Adversarial Networks.

(A. Gieger, et al., IEEE 2020)

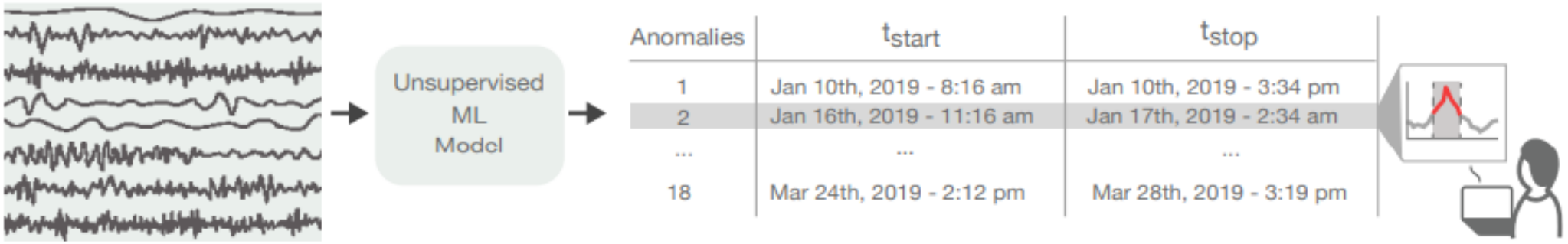
권휘성

Time Series Anomaly Detection

Unsupervised Time Series Anomaly Detection

- Input data: 시계열 $X = (x^1, x^2, x^3, \dots, x^T), where\ x^i \in R^{m \times 1}$

- 목적: 이상을 보이는 time segment 찾기 $A_{seq} = \{a^1_{seq}, a^2_{seq}, a^3_{seq}, \dots, a^k_{seq}\}$



Time Series Anomaly Detection

Methodology	Model	Characteristics
Out-of-limit method	Setting threshold	Inflexible & cannot detection contextual anomalies
Proximity	KNN, LOF	Prior knowledge (anomaly duration, # of anomalies) needed
Prediction	Statistical Method (ARIMA, Holt Winters)	Sensitive to params & strong assumption required
	Machine Learning (LSTM)	Detect through current - previous hidden state comparison

Time Series Anomaly Detection

Methodology	Model	Characteristics
Reconstruction	AE, VAE, LSTM-AE	Lost information at low dim & Easily overfitted w/o regularization
Reconstruction (GANs)	GAN, BeatGAN	May be ineffective to capture hidden distribution via generator

TadGAN: AE 방식과 GAN 방식을 혼합하여 각각의 한계점들을 보완.

Time Series Anomaly Detection

Overview

- 주요 내용

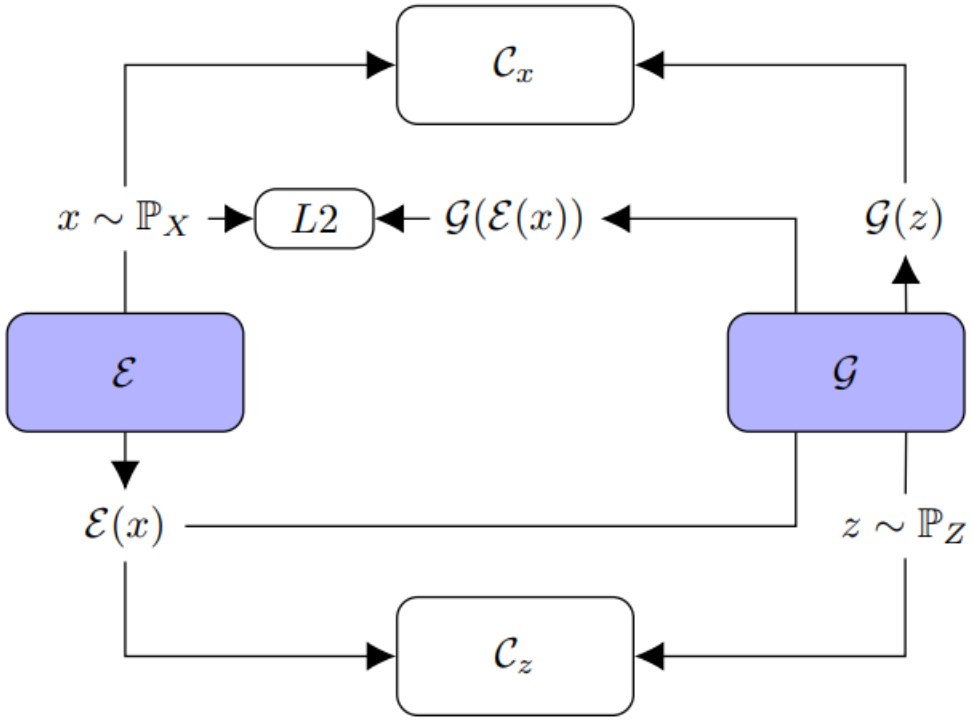
1. Cycle consistent GAN architecture for time-series to time-series mapping 제안
- 생성기 & 분류기 2개씩 사용
- Wassertein loss로 mode collapse 해결
- Cycle- consistent loss로 mapping function search space 제한

2. Reconstruction error 계산 시 Context similarity 평가에 적합한 척도 두 개 사용
- Point difference & Area difference & Dynamic Time Warping
- Anomaly score 계산 시, robust한 점수 도출 가능

3. 시계열 벤치마크 데이터셋 11개 중 6개에 대해 baseline보다 높은 f1 score 기록

Time Series Anomaly Detection

Architecture



- Generators
  - ① $\mathcal{E} : X \rightarrow Z$ (Encoder)
  - ② $\mathcal{G} : Z \rightarrow X$ (Decoder)
  - $\Rightarrow x_i \rightarrow \mathcal{E}(x_i) \rightarrow \mathcal{G}(\mathcal{E}(x_i)) \approx \hat{x}_i$

- Discriminators (=Adversarial Critics)
  - ① $\mathcal{C}_x$: 실제 데이터 x_i 와 $\mathcal{G}(z)$ 구별자
  - ② $\mathcal{C}_z$: latent domain으로의 mapping ($\mathcal{E}$) 성능 평가

Time Series Anomaly Detection

Wassertein Loss

- Mode collapse를 해결하기 위해 도입한 adversarial loss
  - 함수의 upper bound에 대한 제약이 추가됨.
  - 효과: 함수가 smoothing 되어 gradient descent 될 때, gradient explosion 가능성 낮추어 학습 안정화

$\mathcal{C}_x$, $\mathcal{G}$ 학습

$\mathcal{G}$ 는 x와 유사한 데이터를 생성하도록, $\mathcal{C}_x$는 생성된 데이터와 실제 데이터를 잘 구분하도록 학습

$\mathcal{C}_x$: 1d conv layer 활용하여 local temporal feature를 잡아내고자 함

$\mathcal{G}$: bidirectional 2-layer LSTM 구조 활용 (hidden units: 64)

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

⇩ 변경 후

$$\min_{\mathcal{G}} \max_{\mathcal{C}_x\in\mathcal{C}_X} V_X(\mathcal{C}_x, \mathcal{G})$$

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

Time Series Anomaly Detection

Wassertein Loss

- Mode collapse를 해결하기 위해 도입한 adversarial loss
  - 함수의 upper bound에 대한 제약이 추가됨.
  - 효과: 함수가 smoothing 되어 gradient descent 될 때, gradient explosion 가능성 낮추어 학습 안정화

$\mathcal{C}_z$, $\mathcal{E}$ 학습

$\mathcal{E}$는 latent domain Z 으로 mapping 잘하도록, $\mathcal{C}_z$는 z로부터의 sample과 $\mathcal{E}(x)$ 잘 구분하도록 학습

$\mathcal{C}_z$: 1d conv layer 활용하여 local temporal feature를 잡아내고자 함

$\mathcal{E}$: 1-layer LSTM 구조 활용 (hidden units: 100)

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

변경 후

$$\min_{\mathcal{E}} \max_{\mathcal{C}_z\in\mathcal{C}_Z} V_Z(\mathcal{C}_Z, \mathcal{E})$$

$$V_Z(\mathcal{C}_z, \mathcal{E}) = \mathbb{E}_{z\sim\mathbb{P}_Z}[\mathcal{C}_z(z)] - \mathbb{E}_{x\sim\mathbb{P}_X}[\mathcal{C}_z(\mathcal{E}(x))]$$

Time Series Anomaly Detection

Cycle-consistent loss

- $x_i \rightarrow \mathcal{E}(x_i) \rightarrow \mathcal{G}(\mathcal{E}(x_i)) \approx \hat{x_i}$ 를 만족하기 위한 L2 norm
- Wasserstein loss 단독 사용 시, 위 수식이 반드시 성립된다는 보장 X

⇒ 가능한 mapping function search space를 줄이기 위해 도입한 loss

- Anomalous value 강조 위해 L1 대신 L2 사용

$$V_{L2}(\mathcal{E}, \mathcal{G}) = \mathbb{E}_{x \sim \mathbb{P}_X}\left[\left\|x - \mathcal{G}(\mathcal{E}(x))\right\|_2\right]$$

Time Series Anomaly Detection

Full Objective

- Wasserstein loss와 Cycle Consistent loss 활용하여 generator와 discriminator들 학습함.
- TadGAN 구조의 장점: Anomaly score 계산 시, 2가지 방법 활용 가능

$\mathcal{E}, \mathcal{G}$ 로 실제와 복원한 시퀀스 간의 차이 $\rightarrow RE(x)$

$\mathcal{C}_x$로 실제와 생성된 시퀀스 간의 차이 $\rightarrow \mathcal{C}_x(x)$

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

Time Series Anomaly Detection

Reconstruction Errors Reconstruction Errors (RE(x))

- Point-wise difference

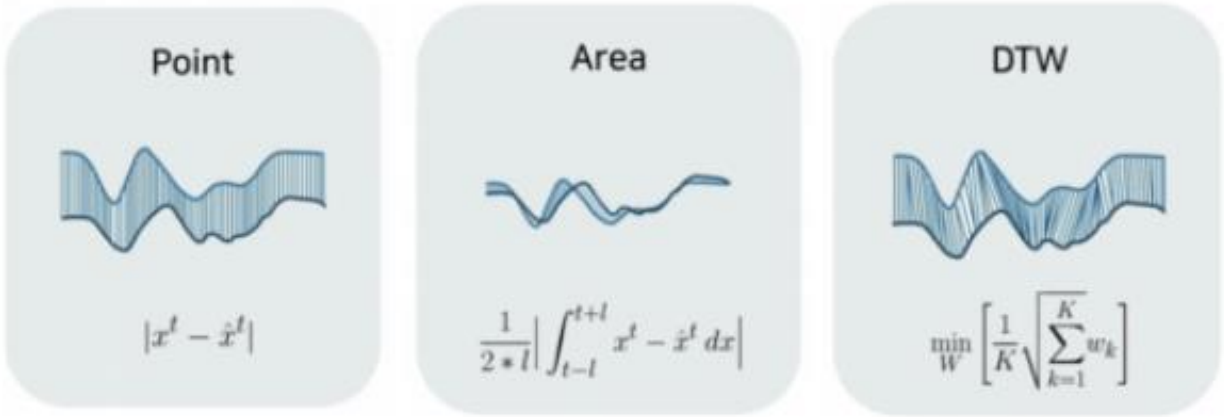
$$s_t = \left| x^t - \hat{x}^t \right|$$

- Area difference

$$s_t = \frac{1}{2 * l} \left| \int_{t-l}^{t+l} x^t - \hat{x}^t \, dx \right|$$

- Dynamic time wraping

$$s_t = W^* = \mathrm{DTW}(X, \hat{X}) = \min_W \left[\frac{1}{K} \sqrt{\sum_{k=1}^{K} w_k} \right]$$



Time Series Anomaly Detection

Critic Outputs

- Cx를 통해 생성된 시퀀스와 원본 시퀀스가 얼마나 real 같은지에 대한 score 도출
- 특정 구간 내 스코어에 KDE 적용 후, max value를 anomaly score로 설정

Time Series Anomaly Detection

Full Objective

- Wasserstein loss와 Cycle Consistent loss 활용하여 generator와 discriminator들 학습함.
- TadGAN 구조의 장점: Anomaly score 계산 시, 2가지 방법 활용 가능

$\mathcal{E}, \mathcal{G}$ 로 실제와 복원한 시퀀스 간의 차이 $\rightarrow RE(x)$

$\mathcal{C}_x$로 실제와 생성된 시퀀스 간의 차이 $\rightarrow \mathcal{C}_x(x)$

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

Time Series Anomaly Detection

Combining both score

- Anomaly score 계산 시 reconstruction error & Critic output을 모두 사용
- 2가지 기준으로 anomaly score 계산함으로써 robust한 탐지 가능
- RE는 높을 수록 Cx는 낮을수록 anomaly일 가능성 높음.
  - 각각 Z 정규화 시킨 후 2가지 criterion 결합

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

$$\boldsymbol{a}(x) = \alpha Z_{RE}(x) \odot Z_{\mathcal{C}_x}(x)$$

Time Series Anomaly Detection

Identifying Anomalous Sequence

- Time step마다 anomaly score들이 도출되는 상황
- Threshold를 정하기 위해 sliding window 내 $\mu \pm 4\sigma$ 기준으로 적용
- 설정방법: window $size = T$, $step\ size = \frac{T}{3*10}$ (경험적으로,,)

  ** Widow size는 과거 몇 개 데이터로 anomaly를 판단할 것인지를 의미함*

Mitigating false positives(FP)

- Sliding window로 발생하는 False Positive 증가 문제에 대한 솔루션
- 현재 시퀀스 $\boldsymbol{a}^i$의 최댓값과 이전 시점 시퀀스 $\boldsymbol{a}^{i-1}$ 의 최댓값으로 계산한 통계량이 θ 를 넘지 못하면, 현재 시퀀스 전체를 normal로 다시 분류

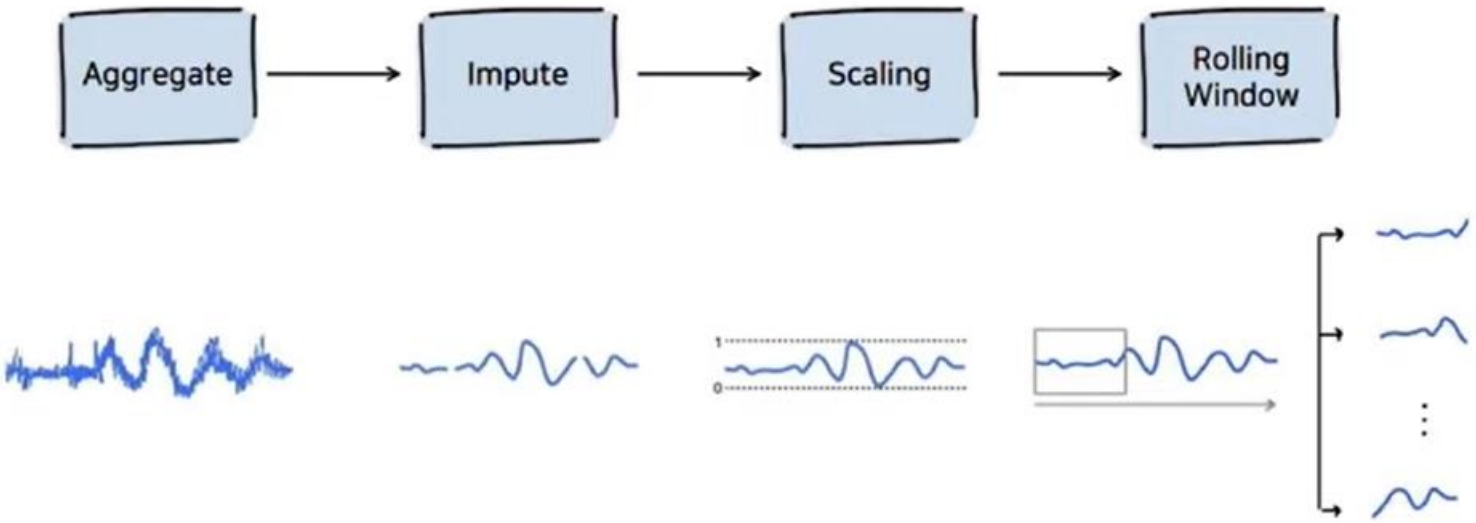
$$p_i < \theta$$

$$where\ p_i = (\boldsymbol{a}^{\mathrm{i-1}}_{\max} - \boldsymbol{a}^{\mathrm{i}}_{\max})/\boldsymbol{a}^{\mathrm{i-1}}_{\max}$$

Time Series Anomaly Detection

Data Preprocessing

- [-1,1] 사이로 정규화
- Window size = 100, step size = 1



Evaluation metric: Precision, Recall, F1-score

Baseline: ARIMA, LSTM, Autoencoder, MAD-GAN,, etc

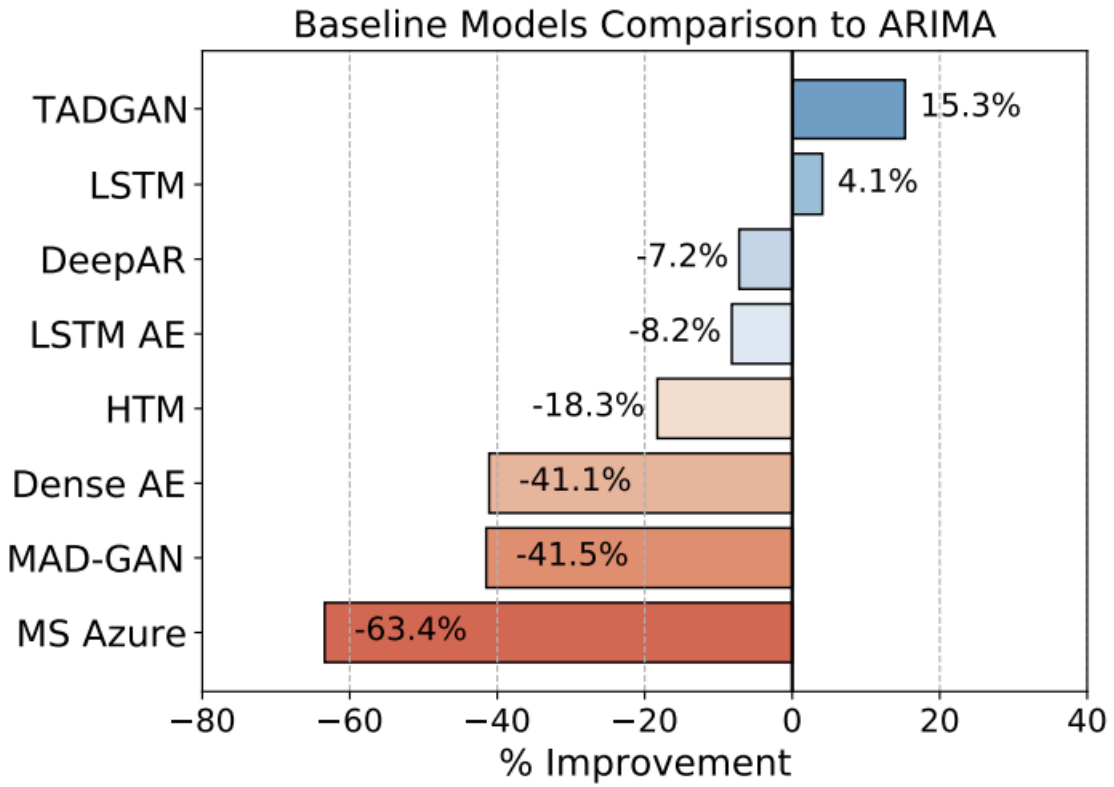
Time Series Anomaly Detection

Results

	NASA		Yahoo S5				NAB					
Baseline	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traf	Tweets	Mean±SD
TadGAN	**0.623**	**0.704**	**0.8**	0.867	0.685	0.6	**0.8**	**0.8**	0.644	0.486	**0.609**	**0.700±0.123**
(P) LSTM	0.46	0.69	0.744	**0.98**	0.772	0.645	0.375	0.538	0.474	**0.634**	0.543	0.623±0.163
(P) Arima	0.492	0.42	0.726	0.836	**0.815**	**0.703**	0.353	0.583	0.518	0.571	0.567	0.599±0.148
(C) DeepAR	0.583	0.453	0.532	0.929	0.467	0.454	0.545	0.615	0.39	0.6	0.542	0.555±0.130
(R) LSTM AE	0.507	0.672	0.608	0.871	0.248	0.163	0.545	0.571	**0.764**	0.552	0.542	0.549±0.193
(P) HTM	0.412	0.557	0.588	0.662	0.325	0.287	0.455	0.519	0.571	0.474	0.526	0.489±0.108
(R) Dense AE	0.507	0.7	0.472	0.294	0.074	0.09	0.444	0.267	0.64	0.333	0.057	0.353±0.212
(R) MAD-GAN	0.111	0.128	0.37	0.439	0.589	0.464	0.324	0.297	0.273	0.412	0.444	0.35±0.137
(C) MS Azure	0.218	0.118	0.352	0.612	0.257	0.204	0.125	0.066	0.173	0.166	0.118	0.219±0.145

Time Series Anomaly Detection

Results



Time Series Anomaly Detection

Results

	NASA		Yahoo S5				NAB					
Variation	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traf	Tweets	Mean+SD
Critic	0.393	0.672	0.285	0.118	0.008	0.024	0.625	0	0.35	0.167	0.548	0.290±0.237
Point	0.585	0.588	0.674	0.758	0.628	**0.6**	0.588	0.611	0.551	0.383	0.571	0.594±0.086
Area	0.525	0.655	0.681	0.82	0.567	0.523	0.625	0.645	0.59	0.435	0.559	0.602±0.096
DTW	0.514	0.581	0.697	0.794	0.613	0.547	0.714	0.69	0.633	0.455	0.559	0.618±0.095
Critic×Point	0.619	0.675	0.703	0.75	**0.685**	0.536	0.588	0.579	0.576	0.4	0.59	0.609±0.091
Critic+Point	0.529	0.653	**0.8**	0.78	0.571	0.44	0.625	0.595	**0.644**	0.439	0.592	0.606±0.111
Critic×Area	0.578	**0.704**	0.719	**0.867**	0.587	0.46	**0.8**	0.6	0.6	0.4	0.571	0.625±0.131
Critic+Area	0.493	0.692	0.789	0.847	0.483	0.367	0.75	0.75	0.607	0.474	0.6	0.623±0.148
Critic×DTW	**0.623**	0.68	0.667	0.82	0.631	0.497	0.667	0.667	0.61	0.455	0.605	**0.629±0.091**
Critic+DTW	0.462	0.658	0.735	0.857	0.523	0.388	0.667	**0.8**	0.632	**0.486**	**0.609**	0.620±0.139
Mean	0.532	0.655	0.675	0.741	0.529	0.438	0.664	0.593	0.579	0.409	0.580	
SD	0.068	0.039	0.137	0.211	0.182	0.154	0.067	0.209	0.081	0.087	0.02	