

2025 01 07
발표 자료

광운대학교 로봇학과
FAIR Lab

김한서

이번 주 진행사항

- MOAT
 - 논문 리뷰

MOAT: Motif-guided Debiasing Framework for Time Series Forecasting

Li Zhang¹, Yifeng Gao¹, Mucun Sun², Shuochao Yao³, Ashley Gomez¹, Jessica Lin³

¹University of Texas Rio Grande Valley

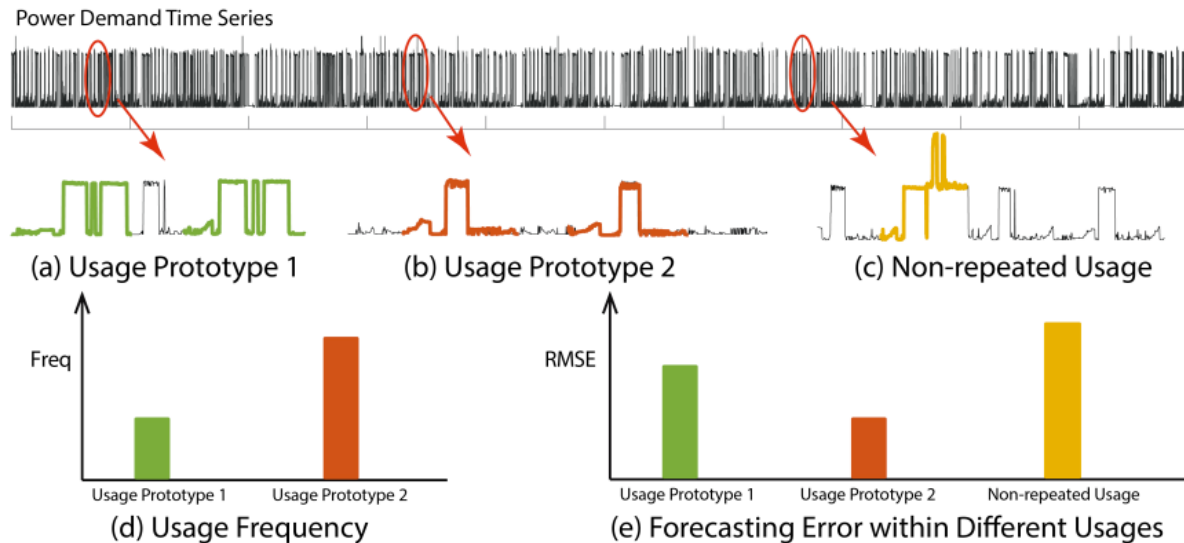
²Idaho National Lab

³George Mason University

{li.zhang,yifeng.gao, asley.gomez06}@utrgv, mucun.sun@inl.gov, {shuochao,jessica}@gmu.edu

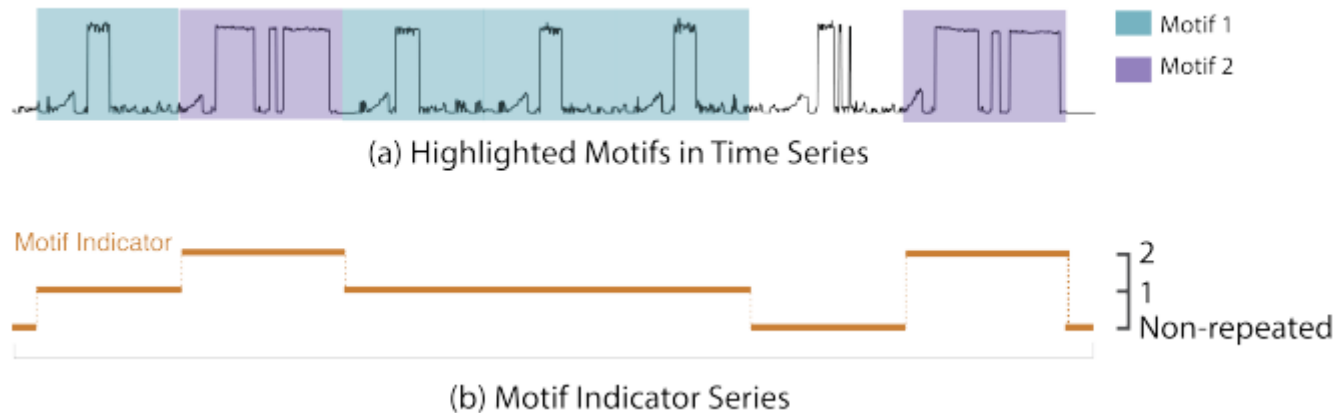
- Published at AAAI 2025
- 링크: [https://github.com/AI4TS/AI4TS.github.io/blob/main/Camera_Ready_AAAI2025/28%5CCameraReady%5CMOAT_AI4TS%20\(1\).pdf](https://github.com/AI4TS/AI4TS.github.io/blob/main/Camera_Ready_AAAI2025/28%5CCameraReady%5CMOAT_AI4TS%20(1).pdf)

Background



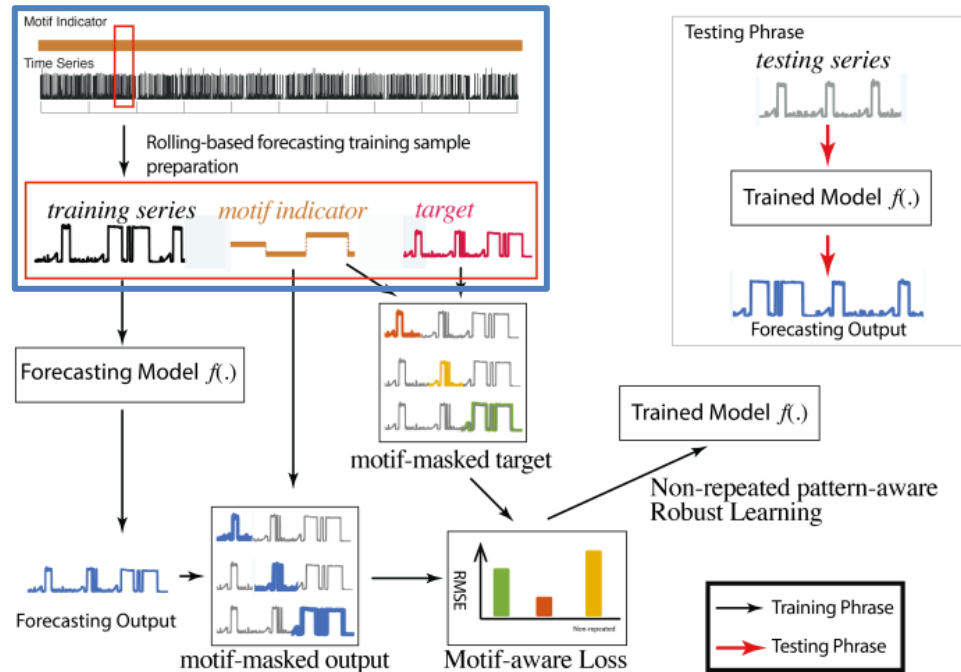
- 기존 모델은 전체 데이터의 평균 손실을 줄이는 데 집중하여 빈도가 높은 패턴에는 최적화 되지만, 복잡하거나 발생 빈도가 낮은 패턴에는 예측 신뢰도가 떨어지는 문제 발생
- MOAT는 위 문제를 해결하기 위해 시계열 모티프 정보를 활용해 취약한 패턴을 집중적으로 개선

시계열 모티프 식별



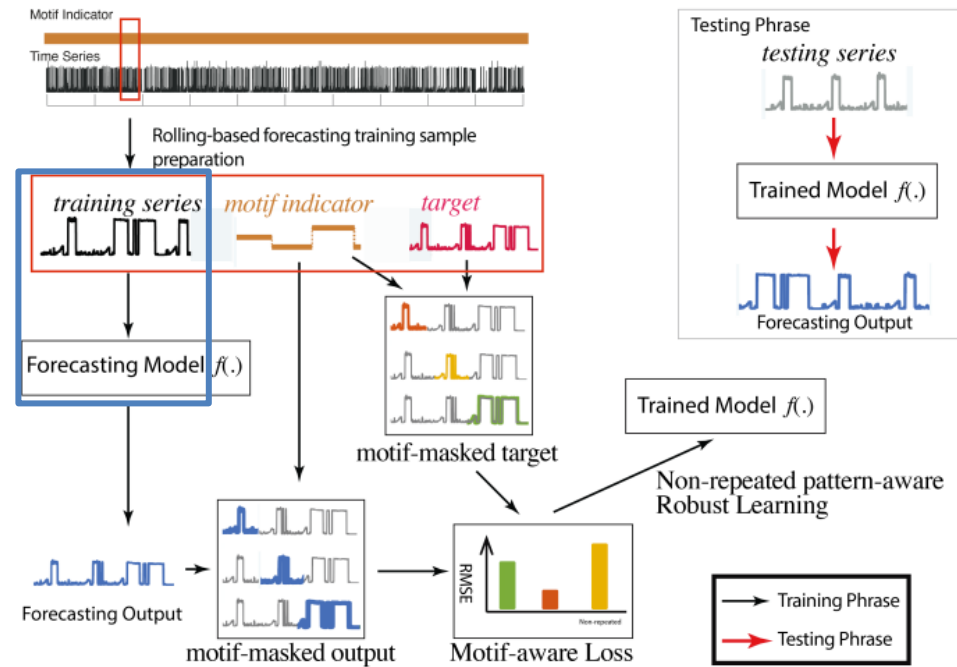
- 모티프(Motif)
 - 시계열 내에서 반복되는 패턴 영역을 Motif1과 Motif2로 강조
- Motif Indicator Series
 - 원본 시계열의 각 시점이 어떤 모티프에 속하는지 나타내는 메타 데이터
 - 각 패턴 영역을 0번인 Non-repeated와 1~ k 개의 모티프 그룹으로 변환

모델 구조



- Rolling 기반 샘플 준비
 - 고정된 길이의 슬라이딩 윈도우를 사용해 training series(X_i)와 Target Sequence(Y_i) 생성
- Motif Indicator
 - 각 원소(M_i)는 해당 시점(i)의 시계열이 어떤 그룹에 속하는지를 나타냄
→ [0번, 1번, 2번, 2번, 1번 ...]

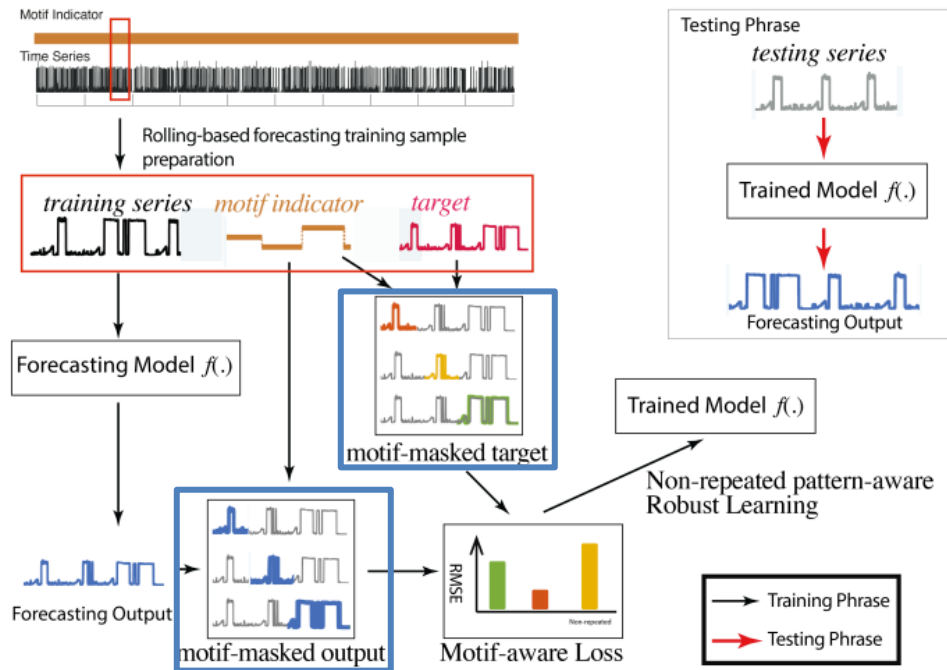
모델 구조



- Forecasting Model

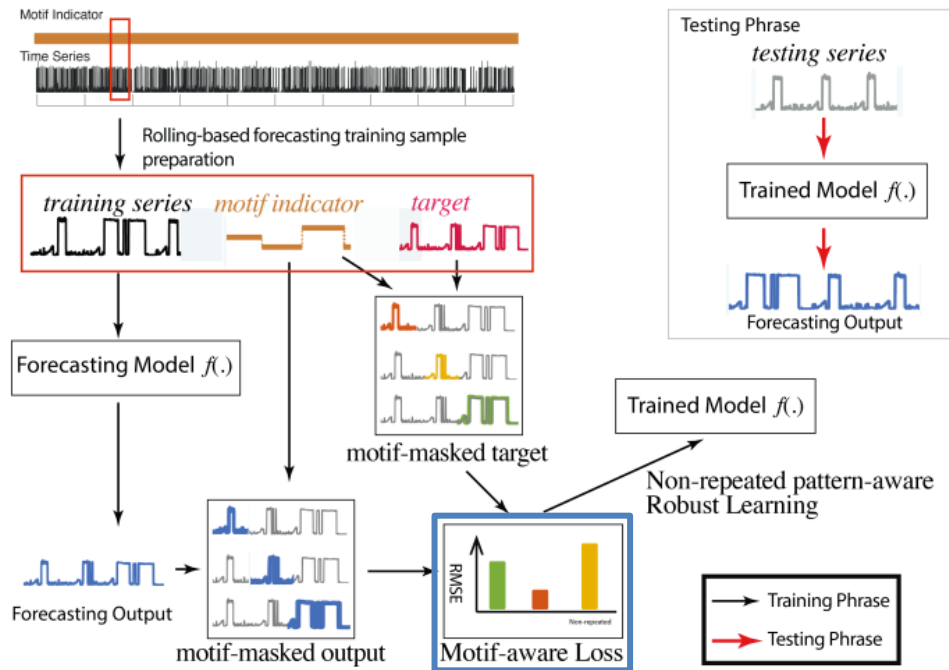
- 예측 모델 f_{θ} 의 입력으로 training series가 들어가게 되고, 이후에 예측값 \hat{y}_i 생성 $\hat{y}_i = f(X_i)$
- 예측 모델의 경우, MLP, Informer, FEDformer 등 기존 모델들을 Backbone으로 사용

모델 구조



- Motif-masked output
 - 모델 출력값인 전체 예측값(\hat{Y})에 Motif Indicator(M)를 적용해 각 모티프 영역별로 분리
 - $Mask(\hat{Y}, M, k)$ 연산을 통해 특정 모티프(k) 영역에 해당하는 위치의 값만 남기고, 나머지는 0으로 설정
- Motif-masked target
 - 전체 Target Sequence(Y)에 Motif Indicator(M)를 적용해 각 모티프 영역별로 분리
 - 나머지 과정은 동일

모델 구조

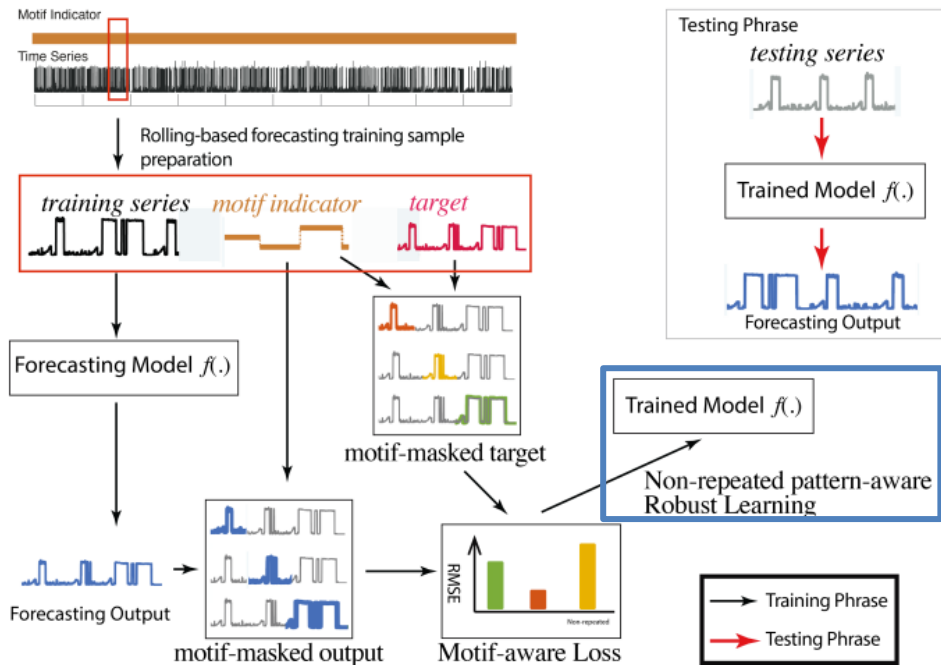


- Motif-aware Loss

- 분리된 예측값과 실제 정답값의 모티프(k) 영역별로 비교해 RMSE를 계산, 이를 통해 모델이 어떤 모티프에서 잘 예측하는지, 오차가 큰지를 수치화 하여 가장 성능이 나쁜 모티프 영역(L_{worst})을 찾아냄

$$L_{M_i} = \sum_i \|Y_{i,k}^{(m)} - \hat{Y}_{i,k}^{(m)}\|_2^2$$

모델 구조



- Robust Learning

- 가장 성능이 나쁜 모티프 영역의 손실을 최소화 하고, 비반복 영역(L_{M_0})의 노이즈로 인한 과적합 방지를 위해 학습 중 오차 비중에 따라 α 를 동적으로 업데이트($\alpha \rightarrow \alpha'$)

$$\min_{\theta} \alpha \frac{L_{\text{worst}}(X, Y, \theta)}{\text{worst motif region}} + (1 - \alpha) \frac{L_{M_0}(X, Y, \theta)}{\text{non-motif region}}, \quad \alpha' = \min \left(\alpha, \frac{L_{\text{worst}}(X, Y, \theta)}{L_{M_0}(X, Y, \theta) + L_{\text{worst}}(X, Y, \theta)} \right)$$

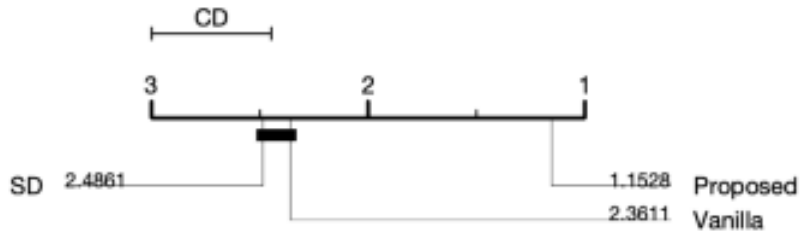
- 가장 성능이 나쁜 영역의 가중치를 높이는 Max-step과 이 가중치들을 고려해 전체 손실을 최소화하는 Min-step을 통해 Saddle point를 최적화

주요 모델 성능 비교

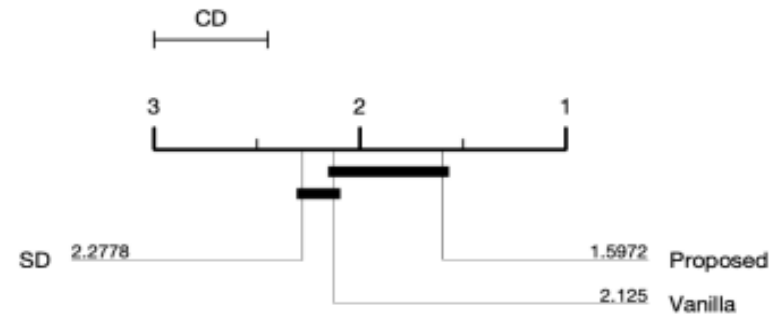
Dataset	Length	Method	Base Models											
			MLP				FEDFormer				Informer			
			bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)	bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)	bias ↓	utility ↓	improved bias ↑ (%)	improved utility ↑ (%)
9*ETTH2	L=24	Proposed	0.07869	0.09881	11.73	-1.85	0.25245	0.23163	3.23	0.02	0.0829	0.095	44.73	0
		Vanila	0.08915	0.09702	0	0	0.26088	0.23167	0	0	0.15	0.095	0	0
		SD	0.15123	0.15218	-69.64	-56.85	0.28194	0.24196	-8.07	-4.44	0.15123	0.15	-0.82	-57.89
	L=48	Proposed	0.11799	0.13251	7.18	-0.52	0.27215	0.24355	4.18	2.21	0.13325	0.15115	8.73	3.11
		Vanila	0.12712	0.13182	0	0	0.28402	0.24906	0	0	0.146	0.156	0	0
		SD	0.24416	0.236	-92.07	-79.03	0.30232	0.2565	-6.44	-2.99	0.248	0.247	-69.86	-58.33
	L=96	Proposed	0.18401	0.18008	4.41	-1.07	0.28141	0.25179	3.4	1.68	0.2045	0.211	9.91	3.21
		Vanila	0.1925	0.17817	0	0	0.29131	0.25609	0	0	0.227	0.218	0	0
		SD	0.41965	0.39465	-118	-121.5	0.31952	0.26969	-9.68	-5.31	0.361	0.344	-59.03	-57.8
9*PowerDemand	L=48	Proposed	0.238	0.138	11.85	1.43	0.89763	0.75374	-2.1	-2.81	0.27	0.188	20.12	1.57
		Vanila	0.27	0.14	0	0	0.87914	0.73316	8.11	11.1	0.338	0.191	0	0
		SD	0.238	0.2	11.85	-42.86	0.95669	0.82469	-16.87	-3.09	0.254	0.178	24.85	6.81
	L=96	Proposed	0.282	0.169	6	-7.64	0.8186	0.8	11.02	3.61	0.3885	0.234	5.01	0.85
		Vanila	0.3	0.157	0	0	0.92	0.83	0	0	0.409	0.236	0	0
		SD	0.32	0.299	-6.67	-90.45	1.13	0.933	-22.83	-12.41	0.399	0.239	2.44	-1.27
	L=192	Proposed	0.386	0.22	-2.66	-10	0.942	0.86	11.13	3.37	0.399	0.235	7.21	0.84
		Vanila	0.376	0.2	0	0	1.06	0.89	0	0	0.43	0.237	0	0
		SD	0.6	0.5	-59.57	-150	1.12	0.916	-5.66	-2.92	1.06303	0.356	-147.22	-50.21
9*TwoPattern10	L=64	Proposed	0.67733	1.30872	48.73	15.83	0.95918	1.79206	11.93	0.6	0.877	1.002	36.22	35.56
		Vanila	1.32118	1.55485	0	0	1.08915	1.80285	0	0	1.375	1.55485	0	0
		SD	0.99382	1.35392	24.78	12.92	0.97622	1.60533	10.37	10.96	1.03	1.35392	25.09	12.92
	L=96	Proposed	0.96985	1.31167	25.08	9.27	0.92499	1.77354	8.68	-7.07	0.77	1.04	40.95	28.07
		Vanila	1.29453	1.44576	0	0	1.01292	1.65646	0	0	1.304	1.44576	0	0
		SD	1.06303	1.2784	17.88	11.58	0.99334	1.45744	1.93	12.01	1.07	1.2784	17.94	11.58
	L=128	Proposed	1.08828	1.26564	13.92	5.72	0.891	1.46	7.23	1.13	0.905	1.025	25.82	0.49
		Vanila	1.26432	1.34246	0	0	0.96044	1.47676	0	0	1.22	1.03	0	0
		SD	1.04834	1.181	17.08	12.03	0.97628	1.31255	-1.65	11.12	1.13	1.181	7.38	-14.66
9*TwoPattern5	L=64	Proposed	0.42666	1.35032	38.29	9.1	0.574	1.792	28.23	-1.79	0.7675	0.8475	32.82	26.94
		Vanila	0.6914	1.48547	0	0	0.79976	1.76046	0	0	1.1425	1.16	0	0
		SD	0.65901	1.37387	4.69	7.51	0.70245	1.61965	12.17	8	0.806	1.25	29.45	-7.76
	L=96	Proposed	0.61961	1.38122	20.06	6.75	0.594	1.80337	15.95	-6.34	0.74525	0.9745	34.48	26.45
		Vanila	0.77508	1.48117	0	0	0.70673	1.69583	0	0	1.1375	1.325	0	0
		SD	0.7792	1.34203	-0.53	9.39	0.71594	1.72	-1.3	-1.42	1.03	1.31	9.45	1.13
	L=128	Proposed	0.71703	1.3818	8.75	3.1	0.706	1.73	-5.22	-1.99	0.9575	1.0275	14.7	25.81
		Vanila	0.7858	1.42606	0	0	0.671	1.69631	0	0	1.1225	1.385	0	0
		SD	0.87153	1.3038	-10.91	8.57	0.76611	1.56247	-14.18	7.89	1.08	1.386	3.79	-0.07

- 대부분의 실험에서 우수한 성능을 달성
- MLP, FEDformer, Informer 등 어떤 모델을 사용하더라도 일관되게 성능이 향상됨

Bias 및 Utility 성능 비교



① Bias 성능 비교



② Utility 성능 비교

- 기존 Vanilla 모델이나 SD 모델보다 MOAT를 사용한 모델이 편향 제거 성능과 평균 예측 성능 면에서 가장 우수한 성능을 달성하였음