



2. Are Transformers Effective for Time Series Forecasting?

0. Abstract

- 이 연구는 Transformer를 사용한 장기 시계열 예측(LTSF) 모델에 대한 의문을 제기
- Transformer는 시퀀스 내 요소 간 의미적 관계를 추출하는 데 효과적이지만, 시계열 모델링에서는 순서 정보가 중요
- 실험 결과에서는 단순한 선형 모델인 LTSF-Linear이 기존 Transformer 기반 모델보다 우수한 성능을 보임
- 이 연구는 LTSF 작업에 대한 새로운 연구 방향을 제안하며, 앞으로 다른 시계열 분석 작업에 대한 Transformer 기반 솔루션의 타당성을 다시 고려할 필요가 있음을 주장

1. Introduction

- 시계열 예측(TSF)은 오늘날 많은 분야에서 활용되고 있음
 - 과거 데이터를 기반으로 함
 - 교통 흐름 추정, 에너지 관리, 금융 투자 등
- Transformer는 가장 성공적인 시퀀스 모델링 아키텍쳐로 평가됨
 - 주요 작동 원리: 멀티헤드 셀프 어텐션 메커니즘
⇒ 장기 시퀀스 내 요소 간 의미적 상관 관계를 추출
 - 그러나, self-attention은 순서와 무관함(= 순서와 상관 x)
→ 일부는 반대 순서까지 가짐
- 시계열 분석 시에는 데이터의 ‘순서’ 자체가 중요한 역할을 함
⇒ Transformer는 정말로 장기 시계열 예측에 효과적일까?
- 해당 연구의 기여
 - 장기 시계열 예측 작업에 대해 긍증하는 Transformer의 효과에 도전하는 첫 연구

- LTSF-Linear 라고 불리는 매우 간단한 단층 선형 모델 세트를 소개하고 기존의 Transformer 기반 LTSF 솔루션과 아홉 가지 벤치마크에서 비교
- 기존 Transformer 기반 솔루션의 다양한 측면에 대한 포괄적인 경험적 연구 수행
- 연구 결과를 통해 앞으로 시계열 분석 작업에 대한 Transformer 기반 솔루션의 타당성을 재고할 것을 주장

2. 사전 준비 사항: TSF 문제 정의

- **C Variates (가변량)**
 - 시계열 데이터는 여러 가변량(예: 여러 센서에서 수집된 여러 데이터 포인트)을 포함할 수 있음
→ 다양한 관측치나 데이터 유형을 나타냄
- **Historical Data(역사적 데이터)**
 - χ 는 주어진 시계열 데이터를 나타냄
 - 해당 데이터는 특정 시간 단계 t_1 에서 t_C 까지의 여러 가변량 값들로 구성되어 있음
- **Look-back Window Size(창 크기, L)**
 - 시계열 데이터에서 사용되는 과거 데이터의 창 크기
⇒ 이전 L개의 데이터 포인트를 고려하여 예측 모델을 구축
- **Time Series Forecasting Task(시계열 예측 작업)**
 - T 미래 시간 단계에 대한 가변량 값을 예측하는 것을 목표로 함
⇒ 현재까지의 데이터를 기반으로 T 시간 단계 뒤의 값을 예측
- **Iterated Multi-Step(IMS) Forecasting vs Direct Multi-Step(DMS) Forecasting**
 - T가 1보다 큰 경우, 두 가지 예측 접근 방식이 있음
 - **IMS 예측**
 - 단일 단계 예측 모델을 학습하고 이를 여러 번 반복하여 다중 단계 예측을 얻음
 - 자동 회귀 추정 과정으로 인해 분산이 작아지지만 오차 누적 효과가 발생
→ IMS 예측은 T가 상대적으로 작고 단일 단계 예측 모델이 매우 정확한 경우에 선호되는 방법
 - **DMS 예측**
 - 한 번에 T개의 다중 단계 예측 목표를 최적화

- 편향되지 않은 단일 단계 예측 모델을 얻기 어려운 경우나 T 가 큰 경우에 더 정확한 예측을 생성

3. Transformer 기반 LTSF 방법들

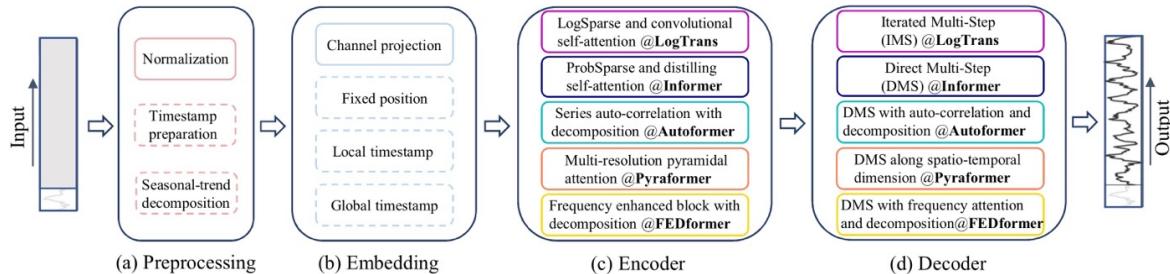


Figure 1. The pipeline of existing Transformer-based TSF solutions. In (a) and (b), the solid boxes are essential operations, and the dotted boxes are applied optionally. (c) and (d) are distinct for different methods [16, 18, 28, 30, 31].

- **Transformer** 기반 모델들은 자연어 처리 및 컴퓨터 비전 분야에서 뛰어난 성과를 이루어 냈으므로써 많은 인공 지능 작업에서 엄청난 성과를 창출함
⇒ 멀티헤드 셀프 어텐션 메커니즘의 효과적인 작동 덕분
 - 이로 인해 **Transformer** 를 기반으로 하는 시계열 모델링 기술에 대한 연구 관심이 크게 높아지고 있음
- 특히, **Transformer** 기반의 시계열 모델링 기법에는 많은 연구 작업이 **LTSF(장기 시계열 예측)** 작업에 중점을 두고 진행되고 있음
 - 장거리 종속성을 캡처할 수 있는 능력을 고려하여, 주로 장기 예측 문제에 초점을 맞추고 있음
- 그러나 기존의 Transformer 모델을 LTSF 문제에 적용할 때는 몇 가지 제한 사항이 있음
 - 셀프 어텐션 구조에 따른 이차 시간/메모리 복잡성
 - 자동 회귀 디코더 설계로 인한 오차 누적
- **Informer** 와 같은 모델은 이러한 문제를 극복하기 위해 새로운 Transformer 아키텍처와 DMS(직접 다중 단계) 예측 전략을 제안
 - 이후 다른 Transformer 변형 모델들은 성능 및 효율성 개선을 위해 다양한 시계열 특성을 모델에 통합시킴

시계열 분해

- 데이터 전처리 단계에서 시계열 예측은 일반적으로 평균을 0으로 하는 정규화를 수행
- **Autoformer** 은 각 신경 블록 뒤에 계절성-추세 분해를 적용

- 원시 데이터를 예측 가능하게 만듦
- **FEDformer** : 다양한 커널 크기로 추출된 추세 구성 요소를 섞는 전략을 제안
- 입력 시퀀스에 이동 평균 커널을 사용하여 시계열의 추세-주기성 구성 요소를 추출

입력 임베딩 전략

- Transformer 아키텍처의 셀프 어텐션 레이어는 시계열의 위치 정보를 보존하지 못함
 - 시계열 입력의 시간적 맥락을 강화하기 위해 고정된 위치 인코딩, 채널 프로젝션 임베딩 및 학습 가능한 시간 임베딩과 같은 여러 임베딩을 입력 시퀀스에 주입(\Rightarrow 시간적 임베딩)

셀프 어텐션 스키마

- Transformer는 쌍을 이루는 요소 간의 의미적 종속성을 추출하기 위해 셀프 어텐션 메커니즘을 사용
 - **Vanilla Transformer** 의 경우 $O(L^2)$ 의 시간 및 메모리 복잡성을 가짐
 - 최근 연구에서는 이를 개선하기 위해 두 가지 효율적인 전략을 제안하고 있음
 1. 셀프 어텐션 스키마에 희소성 편향을 도입
 - **LogTrans** : 연산 복잡성을 $O(LlogL)$ 로 줄이기 위해 로그 희소 마스크를 사용
 - **Pyraformer** : 계층적 다중 스케일 시간 종속성을 포착하기 위한 피라미털 어텐션을 $O(L)$ 의 시간 및 메모리 복잡성으로 도입
 2. 셀프 어텐션 행렬에서 저랭크 속성을 활용
 - **Informer** : $O(LlogL)$ 의 복잡성으로 확률적 희소 셀프 어텐션 메커니즘과 셀프 어텐션 축소 작업을 사용하여 연산 복잡성을 감소시킴
 - **FEDformer** : $O(L)$ 복잡성을 얻기 위해 주파수 어텐션 블록을 도입하고 임의로 웨이블릿 어텐션 블록을 선택
 3. 시리즈별 자동 상관 메커니즘을 도입
 - **Autoformer**
- \Rightarrow **Vanilla Transformer** 의 계산 복잡성을 줄이면서도 시계열 모델링에서 효율성을 향상시키기 위해 고안됨

디코더

- **Vanilla Transformer** 디코더는 자기 회귀적 방식으로 시퀀스를 출력
 - 특히 장기 예측의 경우 추론 속도가 느리고 오차 누적 효과가 발생
- **Informer** : DMS(직접 다중 단계) 예측을 위한 생성 스타일 디코더를 디자인

- 다른 Transformer 변형 모델도 유사한 DMS 전략을 채택
 - Pyraformer : 디코더로 공간-시간 축을 연결하는 완전 연결 레이어를 사용
 - Autoformer : 추세-주기성 구성 요소 및 계절성 구성 요소의 쌍인 자동 상관 메커니즘에서 개선된 분해된 특성 두 개를 더하여 최종 예측을 얻음
 - FEDformer : 주파수 어텐션 블록을 사용하여 디코딩 된 최종 결과를 얻기 위한 분해 스키마를 사용
- Transformer 모델은 요소 간의 의미적 상관 관계를 고려하는 것을 전제로 함
 - 하지만 셀프 어텐션 메커니즘은 순열에 무관
→ 입력 토큰과 함께 시간적 관계를 모델링하는 것은 주로 위치 인코딩에 의존
- 시계열 데이터의 원시 숫자(ex> 주식 가격, 전기 값)는 대부분 서로 간의 의미적 상관 관계를 가지지 않음
 - 연속된 데이터 포인트 간의 시간적 관계가 더 중요
 - 위치 인코딩과 하위 시리즈 임베딩은 일부 순서 정보를 보존하는 데 도움이 되지만, 순열 무관한 셀프 어텐션 메커니즘의 본질로 인해 시간 정보 손실이 발생할 수밖에 없음

4. Baseline

- 기존의 Transformer 기반 LTSF 솔루션 실험에서는 오류 누적 효과가 있는 IMS 예측 기술과 비교됨
 - 이 가설을 확인하기 위해 우리는 가장 간단한 DMS 모델인 LTSF-Linear를 도입하여 비교 기준으로 사용
- LTSF-Linear은 역사적 시계열(→ 과거 데이터)을 미래 예측을 위해 선형 레이어를 사용하여 직접 회귀

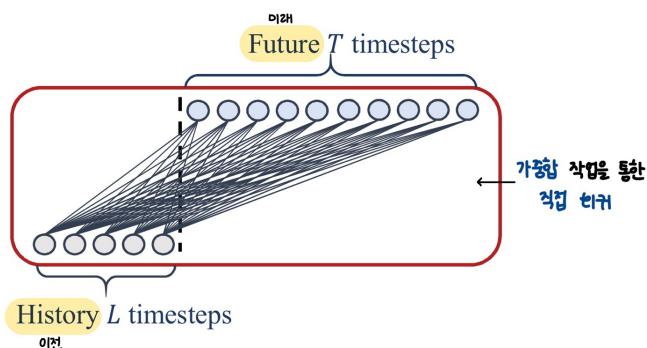


Figure 2. Illustration of the basic linear model.

- 다양한 도메인의 시계열을 처리하기 위해 두 가지 변형(= **DLinear**, **NLinear**)을 도입

- **DLinear**

추세 구성 요소와 계절 구성 요소로 시계열을 분해하고 선형 레이어를 적용하여 추세를 명확하게 처리

- **NLinear**

데이터셋에서 분포 변화가 있는 경우 입력을 정규화하기 위해 빼고 더하는 간단한 절차를 추가

5. 실험

5-1. 실험 환경

데이터셋

- ETT (Electricity Transformer Temperature/ ETTh1, ETTh2, ETTm1, ETTm2), Traffic, Electricity, Weather, ILI, Exchange-Rate와 같은 아홉 가지 널리 사용되는 실제 데이터셋에서 광범위한 실험을 수행
- 모든 데이터는 다변량 시계열 데이터

평가 지표

- Mean Squared Error(MSE)와 Mean Absolute Error(MAE)를 핵심 지표로 사용

비교 대상 메서드

- FEDformer, Autoformer, Informer, Pyraformer, 그리고 LogTrans와 같은 다섯 가지 최근 Transformer 기반 방법을 포함
- 루백 윈도우에서 마지막 값을 반복하는 단순한 DMS 방법인 Closest Repeat (Repeat)를 또 다른 간단한 기준으로 포함
- 정확도가 더 좋은 FEDformer-f(푸리에 변환을 통한 버전)와 비교

5-2. Transformer와의 비교

- 양적 결과

		SOTA																			
Methods	IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer		Pyraformer*		LogTrans		Repeat*			
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE			
Electricity	96	27.40%	0.140	0.237	0.141	0.237	0.140	0.237	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	0.258	0.357	1.588	0.946	
	192	23.88%	0.153	0.250	0.154	0.248	0.153	0.249	0.201	0.315	0.222	0.334	0.296	0.386	0.386	0.443	0.266	0.368	1.595	0.950	
	336	21.02%	0.169	0.268	0.171	0.265	0.169	0.267	0.214	0.329	0.231	0.338	0.300	0.394	0.378	0.443	0.280	0.380	1.617	0.961	
	720	17.47%	0.203	0.301	0.210	0.297	0.203	0.301	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	0.283	0.376	1.647	0.975	
	96	45.27%	0.082	0.207	0.089	0.208	0.081	0.203	0.148	0.278	0.197	0.323	0.847	0.752	0.376	1.105	0.968	0.812	0.081	0.196	
	192	42.06%	0.167	0.304	0.180	0.300	0.157	0.293	0.271	0.380	0.300	0.369	1.204	0.895	1.748	1.151	1.040	0.851	0.167	0.289	
Exchange	336	33.69%	0.328	0.432	0.331	0.415	0.305	0.414	0.460	0.500	0.509	0.524	1.672	1.036	1.874	1.172	1.659	1.081	0.305	0.396	
	720	46.19%	0.964	0.750	1.033	0.780	0.643	0.601	1.195	0.841	1.447	0.941	2.478	1.310	1.943	1.206	1.941	1.127	0.823	0.681	
	96	30.15%	0.410	0.282	0.410	0.279	0.410	0.282	0.587	0.366	0.613	0.388	0.719	0.391	2.085	0.468	0.684	0.384	2.723	1.079	
	192	29.96%	0.423	0.287	0.423	0.284	0.423	0.287	0.604	0.373	0.616	0.382	0.696	0.379	0.867	0.467	0.685	0.390	2.756	1.087	
	336	29.95%	0.436	0.295	0.435	0.290	0.436	0.296	0.621	0.383	0.622	0.337	0.777	0.420	0.869	0.469	0.734	0.408	2.791	1.095	
	720	25.87%	0.466	0.315	0.464	0.307	0.466	0.315	0.626	0.382	0.660	0.408	0.864	0.472	0.881	0.473	0.717	0.396	2.811	1.097	
Traffic	96	18.89%	0.176	0.236	0.182	0.232	0.176	0.237	0.217	0.296	0.266	0.336	0.300	0.384	0.896	0.556	0.458	0.490	0.259	0.254	
	192	21.01%	0.218	0.276	0.225	0.269	0.220	0.282	0.276	0.336	0.307	0.367	0.598	0.544	0.622	0.624	0.658	0.589	0.309	0.292	
	336	22.71%	0.262	0.312	0.271	0.301	0.265	0.319	0.339	0.380	0.359	0.395	0.578	0.523	0.739	0.753	0.797	0.652	0.377	0.338	
	720	19.85%	0.326	0.363	0.338	0.348	0.323	0.362	0.403	0.428	0.419	0.428	1.059	0.741	1.004	0.934	0.869	0.675	0.465	0.394	
	96	47.86%	1.947	0.985	1.683	0.858	2.215	1.081	3.228	1.260	3.483	1.287	5.764	1.677	1.420	2.012	4.480	1.444	6.587	1.701	
	192	36.43%	2.182	1.036	1.703	0.859	1.963	0.963	2.679	1.080	3.103	1.148	4.755	1.467	7.394	2.031	4.799	1.467	7.130	1.884	
Weather	48	34.43%	2.256	1.060	1.719	0.884	2.130	1.024	2.622	1.078	2.669	1.085	4.763	1.469	7.551	2.057	4.800	1.468	6.575	1.798	
	60	34.33%	2.390	1.104	1.819	0.917	2.368	1.096	2.857	1.157	2.770	1.125	5.264	1.564	7.662	2.100	5.278	1.560	5.893	1.677	
	96	0.80%	0.375	0.397	0.374	0.394	0.375	0.399	0.376	0.419	0.449	0.459	0.865	0.713	0.664	0.612	0.878	0.740	1.295	0.713	
	192	3.57%	0.418	0.429	0.408	0.415	0.405	0.416	0.420	0.448	0.500	0.482	1.008	0.792	0.790	0.681	1.037	0.824	1.325	0.733	
	336	6.54%	0.479	0.476	0.476	0.429	0.427	0.439	0.443	0.459	0.465	0.521	0.496	1.107	0.809	0.891	0.738	1.238	0.932	1.323	0.744
	720	13.04%	0.624	0.592	0.440	0.453	0.472	0.490	0.506	0.507	0.514	0.512	1.181	0.865	0.963	0.782	1.135	0.852	1.339	0.756	
ETTh1	96	19.94%	0.288	0.352	0.277	0.338	0.289	0.353	0.346	0.388	0.358	0.397	3.755	1.525	0.645	0.597	2.116	1.197	0.432	0.422	
	192	19.81%	0.377	0.413	0.344	0.381	0.383	0.418	0.429	0.439	0.456	0.452	5.602	1.931	0.788	0.683	4.315	1.635	0.534	0.473	
	336	25.93%	0.452	0.461	0.357	0.400	0.448	0.465	0.496	0.487	0.482	0.486	4.721	1.835	0.907	0.747	1.124	1.604	0.591	0.508	
	720	14.25%	0.698	0.595	0.394	0.436	0.605	0.551	0.463	0.474	0.515	0.511	3.647	1.625	0.963	0.783	3.188	1.540	0.588	0.517	
	96	21.10%	0.308	0.352	0.306	0.348	0.299	0.343	0.379	0.419	0.505	0.475	0.672	0.571	0.543	0.510	0.600	0.546	1.214	0.665	
	192	21.36%	0.340	0.369	0.349	0.375	0.335	0.365	0.426	0.441	0.553	0.496	0.795	0.669	0.557	0.537	0.837	0.700	1.261	0.690	
ETTh2	336	17.07%	0.376	0.393	0.375	0.388	0.369	0.386	0.445	0.459	0.621	0.537	1.212	0.871	0.754	0.655	1.124	0.832	1.283	0.707	
	720	21.73%	0.440	0.435	0.433	0.422	0.425	0.421	0.543	0.490	0.671	0.561	1.166	0.823	0.908	0.724	1.153	0.820	1.319	0.729	
	96	17.73%	0.168	0.262	0.167	0.255	0.167	0.260	0.203	0.287	0.255	0.339	0.363	0.453	0.435	0.507	0.768	0.642	0.266	0.328	
	192	17.84%	0.232	0.303	0.221	0.293	0.224	0.303	0.269	0.328	0.281	0.340	0.533	0.563	0.730	0.673	0.989	0.757	0.340	0.371	
	336	15.69%	0.320	0.373	0.274	0.327	0.281	0.342	0.325	0.366	0.339	0.372	1.363	0.887	1.201	0.845	1.334	0.872	0.412	0.410	
	720	12.58%	0.413	0.435	0.368	0.384	0.397	0.421	0.421	0.415	0.433	0.432	3.379	1.338	3.625	1.451	3.048	1.328	0.521	0.465	

* Methods* are implemented by us; Other results from FEDformer [31].

Table 2. Multivariate long-term forecasting errors in terms of MSE and MAE, the lower the better. Among them, ILI dataset is with forecasting horizon $T \in \{24, 36, 48, 60\}$. For the others, $T \in \{96, 192, 336, 720\}$. Repeat repeats the last value in the look-back window.

The best results are highlighted in bold and the best results of Transformers are highlighted with a underline. Accordingly, IMP. is the best result of linear models compared to the results of Transformer-based solutions.

- 아홉 가지 다양한 시계열 데이터셋에서 Transformer 모델들과 LTSF-Linear을 평가한 결과를 제시
 - LTSF-Linear가 변수 간 상관 관계를 모델링하지 않으면서도 나타남
- 다른 시계열 벤치마크에서 NLinear과 DLinear은 분포 변화와 추세-계절성 특성을 처리하는 데 뛰어남
 - 단변량 예측 결과도 마찬가지로 LTSF-Linear이 Transformer 기반 LTSF 솔루션을 크게 앞서는 것을 확인할 수 있음
- 단순한 Repeat 방법은 장기적 계절 데이터에서는 안 좋은 결과를 보이지만, Exchange-Rate와 같은 데이터에서는 모든 Transformer 기반 방법을 놀라운 정도로 앞선다
 - 이는 Transformer 기반 솔루션에서 추세를 잘못 예측하는 문제
 - 훈련 데이터의 급격한 변화 노이즈에 과적합될 수 있어서 중요한 정확도 하락을 초래
- 질적 결과

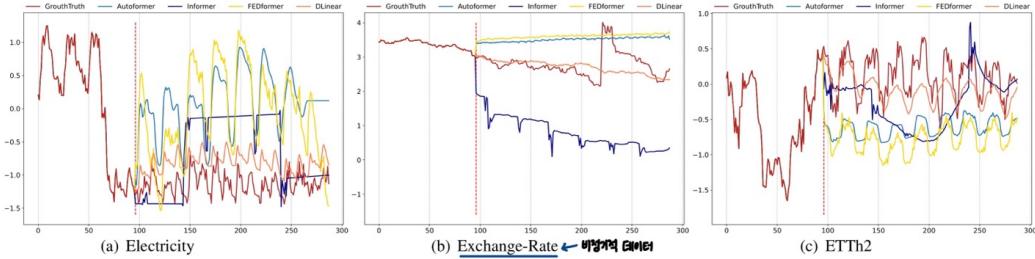


Figure 3. Illustration of the long-term forecasting output (Y-axis) of five models with an input length $L=96$ and output length $T=192$ (X-axis) on Electricity, Exchange-Rate, and ETTh2, respectively.

- 서로 다른 시계열 패턴을 가진 데이터셋에서 Transformer 기반 솔루션과 LTSF-Linear의 예측 결과를 그래프로 표시
- Transformers는 미래 데이터의 규모와 편향을 캡처하는 데 실패하며, Exchange-Rate와 같은 비정기적 데이터에서는 적절한 추세를 예측하기 어렵다는 것을 나타냄

⇒ 기존의 Transformer 기반 LTSF 솔루션이 LTSF 작업에 부적합함을 시사

5-3. LTSF-Transformers에 대한 추가 분석

기존 LTSF-Transformer들은 긴 입력 시퀀스에서 시간 관계를 잘 추출할 수 있을까?

- 기존 LTSF-Transformer 모델들은 긴 입력 시퀀스에서 시간적 관계를 추출하는 능력이 미흡한 것으로 나타남
 - look-back 창 크기가 예측 정확도에 큰 영향을 미침
 - 일반적으로 강력한 시계열 예측 모델은 더 큰 창 크기에서 더 나은 성능을 보여야 함
- 실험 결과, 기존 Transformer 기반 모델들은 창 크기가 증가할수록 성능이 저하되거나 안정적인 양상을 보인 반면, 모든 LTSF-Linear 모델은 창 크기가 증가함에 따라 성능이 크게 향상됨
 - ⇒ 기존 솔루션들은 긴 시퀀스에서 시간적 노이즈에 과적합되는 경향이 있으며, 입력 크기 96이 대부분의 Transformer 모델에 적합한 것으로 나타남
 - ⇒ 추가로 보다 정량적인 결과도 확인되었으며, 우리의 결론이 거의 모든 경우에 적용 가능함을 확인

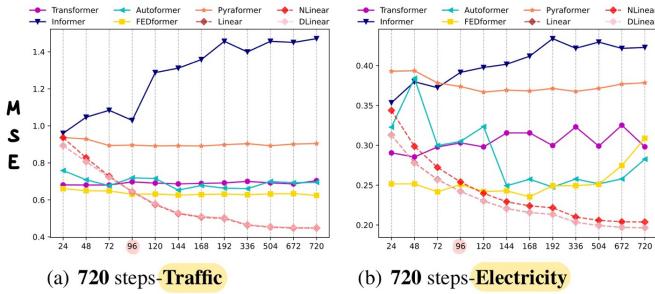


Figure 4. The MSE results (Y-axis) of models with different look-back window sizes (X-axis) of long-term forecasting ($T=720$) on the Traffic and Electricity datasets.

장기간 예측으로 무엇을 배울 수 있을까?

- 장기 시계열 예측은 주로 추세와 주기성을 잘 파악할 수 있는가에 의존하는 반면, 단기 시계열 예측은 look-back 창의 크기에 영향을 크게 받는다는 가설을 확인하려 함
- 실험 설계
 - 동일한 미래 720개의 시간 단계에 대한 예측 정확도를 두 가지 다른 look-back 창 크기의 데이터로 비교하여 검증
 - 결과적으로, 최신 **Transformer** 모델의 성능이 약간 감소
 - 주로 인접한 시계열 데이터에서 유사한 시간 정보만을 포착하는 것으로 나타남
 - 데이터의 본질적인 특성을 포착하기 위해 많은 수의 매개변수가 필요하지 않음
 - ⇒ 많은 매개변수를 사용하면 과적합이 발생할 수 있음
 - ⇒ LTSF-Linear이 Transformer 기반 방법보다 더 나은 성능을 내는 이유로 짐작됨

LTSF에서 self-attention 스키마가 효과적인가?

- 기존의 Transformer (예: Informer)에 있는 복잡한 디자인이 필수적인지 검증
- Informer를 점차적으로 Linear 모델로 변환하는 실험을 진행
 1. 각 셀프 어텐션 레이어를 가중치가 동적으로 변경되는 완전 연결 레이어로 대체 ⇒ Att.-Linear
 2. Informer의 다른 보조 디자인(예: FFN)을 제외하고 임베딩 레이어와 선형 레이어를 남겨둠 ⇒ Embed + Linear
 3. 모델을 하나의 선형 레이어로 간소화

Methods	Informer	Att.-Linear	Embed + Linear	Linear
Exchange	96	0.847	1.003	0.173
	192	1.204	0.979	0.443
	336	1.672	1.498	1.288
	720	2.478	2.102	2.026
ETTh1	96	0.865	0.613	0.454
	192	1.008	0.759	0.686
	336	1.107	0.921	0.821
	720	1.181	0.902	1.051

Table 4. The MSE comparisons of gradually transforming Informer to a Linear from the left to right columns. Att.-Linear is a structure that replaces each attention layer with a linear layer. Embed + Linear is to drop other designs and only keeps embedding layers and a linear layer. The look-back window size is 96.

⇒ 기존의 LTSF 벤치마크에는 적어도 셀프 어텐션 스키마와 다른 복잡한 모듈이 필요하지 않다는 것을 확인할 수 있음

기존의 LTSF-Transformer 모델들이 시간 순서를 잘 보존할 수 있는지 어떻게 판단할까?

Methods		Linear			FEDformer			Autoformer			Informer		
Predict Length	Ori.	Shuf.	Half-Ex.	Ori.	Shuf.	Half-Ex.	Ori.	Shuf.	Half-Ex.	Ori.	Shuf.	Half-Ex.	
Exchange	96	0.080	0.133	0.169	0.161	0.160	0.162	0.152	0.158	0.160	0.952	1.004	0.959
	192	0.162	0.208	0.243	0.274	0.275	0.275	0.278	0.271	0.277	1.012	1.023	1.014
	336	0.286	0.320	0.345	0.439	0.439	0.439	0.435	0.430	0.435	1.177	1.181	1.177
	720	0.806	0.819	0.836	1.122	1.122	1.122	1.113	1.113	1.113	1.198	1.210	1.196
Average Drop		N/A → 27.26%	46.81%	N/A	-0.09%	0.20%	N/A	0.09%	1.12%	N/A	-0.12%	-0.18%	
ETTh1	96	0.395	0.824	0.431	0.376	0.753	0.405	0.455	0.838	0.458	0.974	0.971	0.971
	192	0.447	0.824	0.471	0.419	0.730	0.436	0.486	0.774	0.491	1.233	1.232	1.231
	336	0.490	0.825	0.505	0.447	0.736	0.453	0.496	0.752	0.497	1.693	1.693	1.691
	720	0.520	0.846	0.528	0.468	0.720	0.470	0.525	0.696	0.524	2.720	2.716	2.715
Average Drop		N/A	81.06%	4.78%	N/A	73.28%	3.44%	N/A	56.91%	0.46%	N/A	1.98%	0.18%

Table 5. The MSE comparisons of models when shuffling the raw input sequence. Shuf. randomly shuffles the input sequence. Half-Ex. randomly exchanges the first half of the input sequences with the second half. Average Drop is the average performance drop under all forecasting lengths after shuffling. All results are the average test MSE of five runs.

▲ Linear 모델의 경우 데이터의 순서를 섞으면 성능이 저하됨
→ 데이터의 순서가 중요한 영향을 미치는 것을 확인할 수 있음

- 기존의 LTSF-Transformer 모델들은 시간 순서를 잘 보존하지 못하는 경향이 있음
 - 셀프 어텐션 메커니즘은 순열 무관성을 가짐
⇒ 시계열 예측에서는 시간적 순서가 중요한데, 이를 잘 캡처하지 못함
- 위치 및 시간 임베딩을 사용하더라도 기존 Transformer 기반 모델은 시간 정보를 제한적으로 보존하고 소음이 있는 금융 데이터에서 오버피팅하기 쉬움
⇒ 시간 순서를 잘 유지하지 못함
- 이에 반해 LTSF-Linear는 더 적은 매개변수로 순서를 자연스럽게 모델링하고 오버피팅을 피할 수 있음

⇒ 기존 Transformer는 현재의 LTSF 벤치마크에 대해 시간 순서를 잘 보존하지 못한다는 것을 시사

다른 임베딩 전략들은 얼마나 효과적인가?

Methods	Embedding	Traffic			
		96	192	336	720
FEDformer	All	0.597	0.606	0.627	0.649
	wo/Pos.	0.587	0.604	0.621	0.626
	wo/Temp.	0.613	0.623	0.650	0.677
	wo/Pos.-Temp.	0.613	0.622	0.648	0.663
Autoformer	All	0.629	0.647	0.676	0.638
	wo/Pos.	0.613	0.616	0.622	0.660
	wo/Temp.	0.681	0.665	0.908	0.769
	wo/Pos.-Temp.	0.672	0.811	1.133	1.300
Informer	All	0.719	0.696	0.777	0.864
	wo/Pos.	1.035	1.186	1.307	1.472
	wo/Temp.	0.754	0.780	0.903	1.259
	wo/Pos.-Temp.	1.038	1.351	1.491	1.512

Table 6. The MSE comparisons of different embedding strategies on Transformer-based methods with look-back window size 96 and forecasting lengths {96, 192, 336, 720}.

- 다양한 임베딩 전략의 효과를 연구한 결과, Transformer 기반 방법에서 위치와 타임스탬프 임베딩의 중요성이 드러남
- **Informer** 모델
 - 위치 임베딩이 없으면(wo/Pos.) 예측 오류가 크게 증가하고, 타임스탬프 임베딩이 없으면(wo(Temp.) 예측 길이가 길어짐에 따라 성능이 점차 악화됨
- 그러나 **FEDformer**와 **Autoformer**는 위치 임베딩 없이도 성능이 유지되거나 향상됨
 - 타임스탬프 시퀀스를 입력으로 사용하여 시간 정보를 잘 포착할 수 있었음
 - 타임스탬프 임베딩을 제거하면 Autoformer의 성능이 급격히 저하되었으나, FEDformer는 주파수 강화 모듈 덕분에 위치 및 타임스탬프 임베딩을 제거해도 상대적으로 성능 하락이 적었음

훈련 데이터 크기가 기존 LTSF Transformer의 제한 요소입니까?

Methods	FEDformer		Autoformer	
	Ori.	Short	Ori.	Short
Dataset				
96	0.587	0.568	0.613	0.594
192	0.604	0.584	0.616	0.621
336	0.621	0.601	0.622	0.621
720	0.626	0.608	0.660	0.650

Table 7. The MSE comparison of two training data sizes.

- 기존의 LTSF(장기 시계열 예측) Transformer 모델의 성능이 부족한 이유가 작은 훈련 데이터 크기 때문인지 확인하기 위해 Traffic 데이터셋을 사용한 실험을 수행

- 결과적으로, 훈련 데이터 크기를 축소한 경우에도 대부분의 경우 예측 오류가 더 낮았음
⇒ 모델 성능의 한계는 훈련 데이터 크기에 의한 것이 아니라는 것을 확인

효율성이 정말 최우선 순위입니까?

Method	MACs	Parameter	Time	Memory
DLinear	0.04G	139.7K	0.4ms	687MiB
Transformer×	4.03G	13.61M	26.8ms	6091MiB
Informer	3.93G	14.39M	49.3ms	3869MiB
Autoformer	4.41G	14.91M	164.1ms	7607MiB
Pyraformer	0.80G	241.4M*	3.4ms	7017MiB
FEDformer	4.41G	20.68M	40.5ms	4143MiB

- × is modified into the same one-step decoder, which is implemented in the source code from Autoformer.

- * 236.7M parameters of Pyraformer come from its linear decoder.

Table 8. Comparison of practical efficiency of LTSF-Transformers under L=96 and T=720 on the Electricity. MACs are the number of multiply-accumulate operations. We use Dlinear for comparison since it has the double cost in *LTSF-Linear*. The inference time averages 5 runs.

- 현재의 LTSF-Transformer 모델은 원래의 Transformer 모델의 복잡성($O(L^2)$)이 LTSF(장기 시계열 예측) 문제에 부적합하다고 주장
 - 이를 해결하기 위해 $O(L)$ 의 복잡성을 갖는 다양한 변형 모델을 개발
- 그러나 이러한 변형 모델들이 실제로는 원래의 Transformer 모델보다 효율적이지 않을 수 있으며, 메모리 문제가 심각한 정도는 아니라는 것을 실험적으로 입증함

⇒ 현재의 GPU 환경에서는 메모리 효율적인 Transformer 모델을 개발하는 것이 현재의 벤치마크에서는 최우선 과제가 아닐 수 있음을 시사

6. 결론 & 후속 연구 방향

- Transformer 기반의 장기 시계열 예측 문제에 대한 새로운 솔루션의 효과성에 의문을 제기하고 있음
 - 간단한 선형 모델인 LTSF-Linear을 DMS 예측 기준선으로 사용하여 이러한 주장 을 검증
 - 다양한 관점에서 왜 현재의 LTSF-Transformer가 이전 연구에서 주장된 것만큼 효과적이지 않은지를 설명함
 - 미래 연구 방향
 - LTSF-Linear 모델은 용량이 제한적이며, 변화점에 의한 시간적 동적을 캡처하기 어렵다는 한계점이 있음
- ⇒ 새로운 모델 디자인, 데이터 처리 및 벤치마크 등의 연구가 필요