

2023-2 캡스톤디자인 I



이공즈 팀

2020170815 김경호

2020170830 조혜운

2020170845 유혜원

2020170854 조혜린

2020170859 최유진

대형마트의 공조 제어 데이터를 활용한 실내 온도 예측 모델 개발



목차

● 01

주제 및 개요

● 02

데이터 탐색

● 03

예측모델 선택 과정

● 04

최종 모델 선정

● 05

모델 평가

● 06

결론

01

주제 및 개요

1.1 주제 선정 배경

1.2 프로젝트 목적



1.1 주제 선정 배경

“

대형마트에서는 **제품의 신선도와 고객의 편의**를
유지하기 위해 **적절한 온도**를 유지하는 것이 중요

”

대형 마트의 공조제어 시스템 데이터를 활용한 실내 온도 예측 모델 개발

1.2 프로젝트 목적

- 1 동남아의 계절성을 고려한 다변량 시계열 분석
- 2 높은 성능을 가지는 예측 모델 생성
- 3 다양한 방법론을 통한 lag 해결



02

데이터 탐색

2.1 데이터 EDA

2.2 변수 선택

2.3 데이터 전처리

2.4. 최종 데이터셋



2.1

데이터 EDA



- ①porest_area_ahu_automatic
- ②porest_area_chiller_automatic
- ③porest_device_plant_log
- ④porest_device_e_log
- ⑤porest_devices
- ⑥area_log

① porest_area_ahu_automatic



데이터셋 인사이트

- 'inv', 'occupancy', 'zero_min_temp'은 skewed 분포를 가짐.
- 'inv'에는 4915개(23.2%)의 0 값 존재
- 'power', 'inv'는 높은 피어슨 상관관계를 가짐(0.98).



2.1

데이터 EDA



①porest_area_ahu_automatic

②porest_area_chiller_automatic

③porest_device_plant_log

④porest_device_e_log

⑤porest_devices

⑥area_log

② porest_area_chiller_automatic



데이터셋 인사이트

- 'power'는 모두 상수 값 1을 가짐.
- 'power'와 'inv'는 매우 높은 스피어맨 상관관계를 가짐(0.99).
⇒ 강한 순위 상관관계
- ' power ' 와 ' inv ' 는 높은 피어슨 상관관계를 가짐(0.69).



2.1

데이터 EDA



①porest_area_ahu_automatic

②porest_area_chiller_automatic

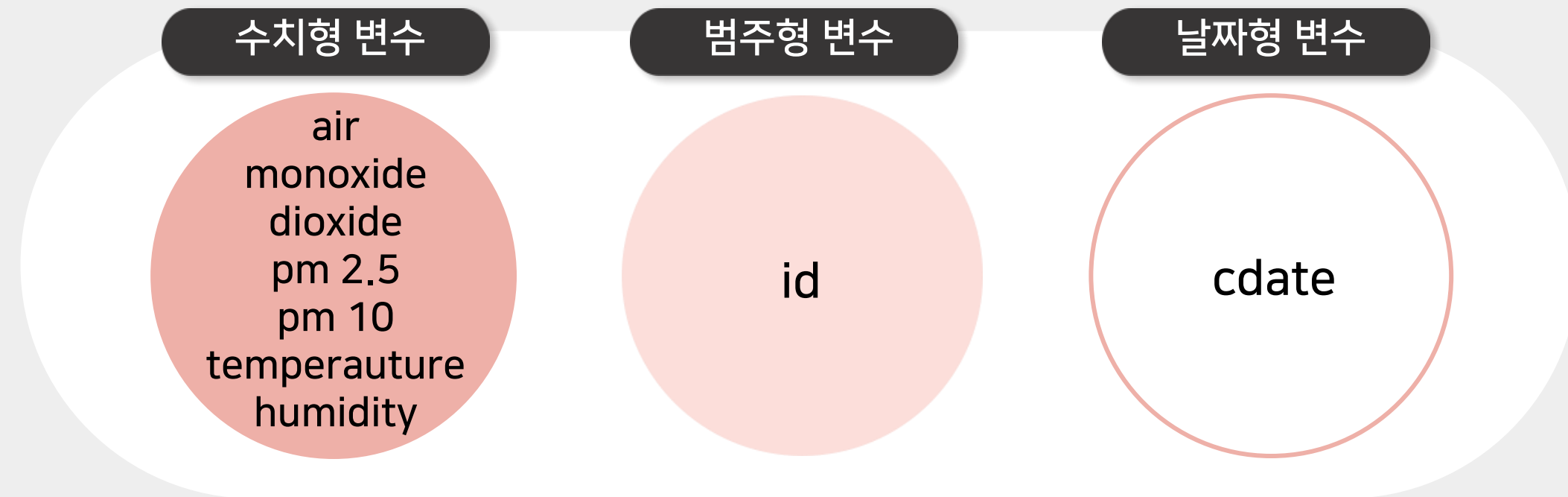
③porest_device_plant_log

④porest_device_e_log

⑤porest_devices

⑥area_log

③ porest_device_plant_log



데이터셋 인사이트

- 'air', 'monoxide', 'dioxide', 'pm2.5', 'pm10' 은 skewed 분포를 가짐
- 'air'에는 1219839개(27.25%)의 0 값 존재
- 'monoxide'에는 2438966개(54.48%)의 0 값 존재
- 'pm2.5'와 'pm10' 사이의 피어슨 상관계수와 스피어먼 상관계수는 모두 1

2.1
데이터 EDA



- ①porest_area_ahu_automatic
- ②porest_area_chiller_automatic
- ③porest_device_plant_log
- ④porest_device_e_log
- ⑤porest_devices
- ⑥area_log

④ porest_device_e_log
⑤ porest_devices

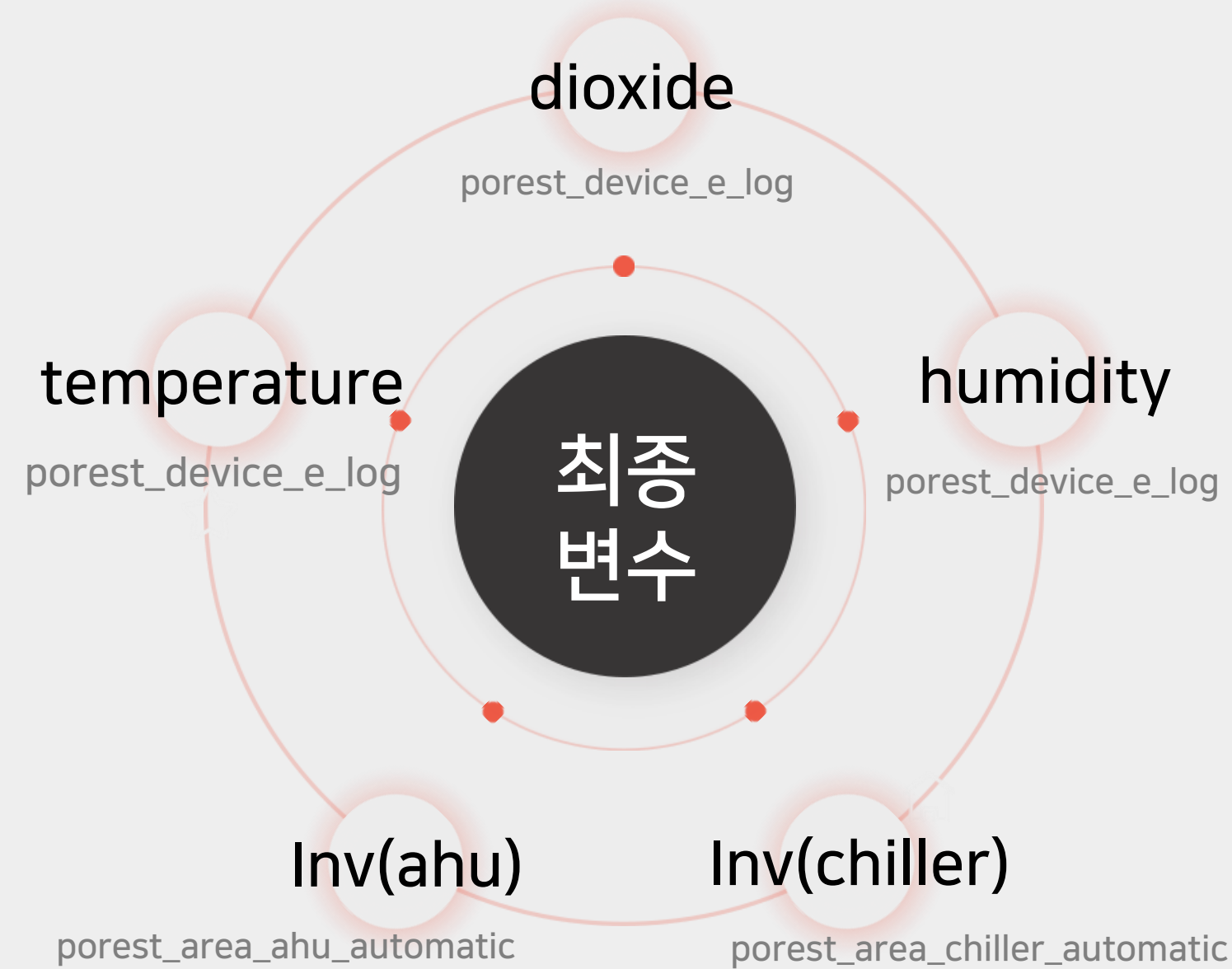
ahu	id_e , name, place, id_plant
chiller	name, id_e
kitchen	name, id_plant, id_e
eplus	id, panel_location, panel_board, load_name
plant	id, name, etc

⑥ area_log

TYPE	MODE
ahu	자동제어변경
chiller	스케줄 제어
kitchen	수동 제어



2.2 변수 선택



변수 선택 이유

- dioxide**..... CO2 농도는 단독으로 온도를 예측하는 데 사용되지는 않지만, 실내 환경 예측 조건 중 하나이므로 고려
- humidity**..... 공기 중의 수증기량은 온도와 상호작용하여 열을 보다 잘 전달하고 보존하는 데 영향을 줌
- inv(ahu)**..... 공기 처리 장치로, 실내 공기를 처리하고 분배하는 시스템이기에 온도에 영향을 미침
- inv(chiller)**..... 냉수를 생산하여 냉난방 시스템에 사용되는 장치로 AHU나 다른 냉난방 장치를 통해 실내 온도를 제어

2.3

데이터 전처리



- ①porest_area_ahu_automatic
- ②porest_area_chiller_automatic
- ③porest_device_plant_log

[제어변수]

AHU - INV

- 대상이 되는 AHU no: 1~14
- 14개의 AHU가 어느 정도 세기로 작동되는지에 따라 온도에 큰 영향을 줌
→ ahu1_inv , ahu2_inv, ...

Chiller - INV

- 대상이 되는 chiller no: 1, 2
- 2개의 chiller가 어느 정도 세기로 작동되는지에 따라 온도에 큰 영향을 줌
- 2개의 chiller는 동시에 사용되지 않고 매번 하나의 chiller만 작동함 → chiller_inv

- 01 각 제어방식 데이터 병합
- 02 1차 결측값 처리
- 03 10분 단위로 Resample 수행
- 04 2차 결측값 처리

[제어방식에 따른 기록 시점]

- 자동제어: 일정 시간 간격으로 기록
- 수동제어: 값에 변화가 생겼을 때 기록
- 스케줄제어: 값에 변화가 생겼을 때 기록

[병합 규칙]

cdate를 기준으로 병합 및 정렬

[결측값 처리 규칙]

- 1차와 2차 결측값 처리 규칙은 동일

1. 영업시간(7시,22시) 외 데이터에 대해서는 inv=0
2. 동일 날짜에 한해 직전 값으로 결측치에 값 부여
(값에 변화가 생긴 시점에 데이터가 기록됨을 고려)



2.3

데이터 전처리

[환경변수]

Plant

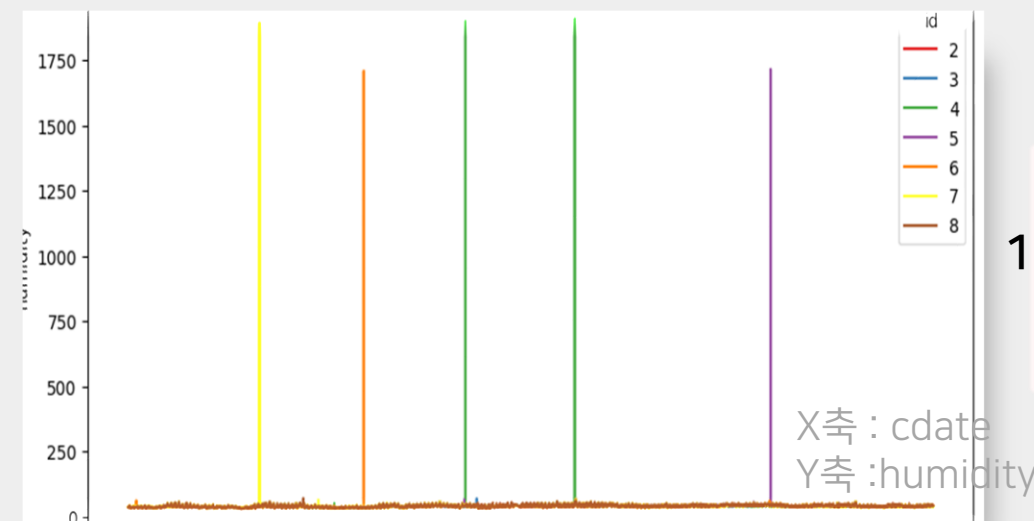
- 대상이 되는 plant no: 2~8
- 각 Plant는 환경변수 값들을 수집하는데 이중 temperature가 본 task의 예측값

Temperature
Humidity
Dioxide

plant2_temperature, plant3_humidity, plant4_dioxide, ...

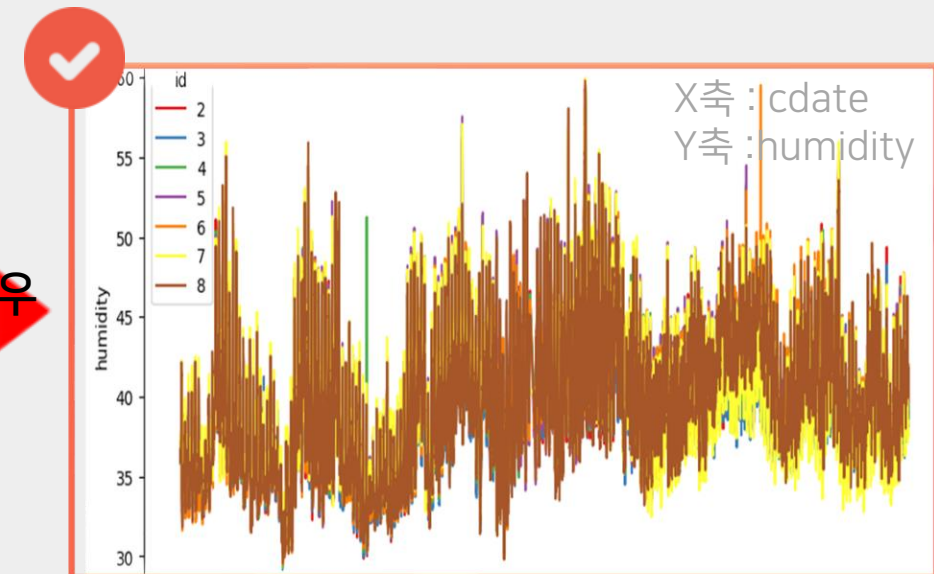
- ①porest_area_ahu_automatic
- ②porest_area_chiller_automatic
- ③porest_device_plant_log

01 이상치 제거

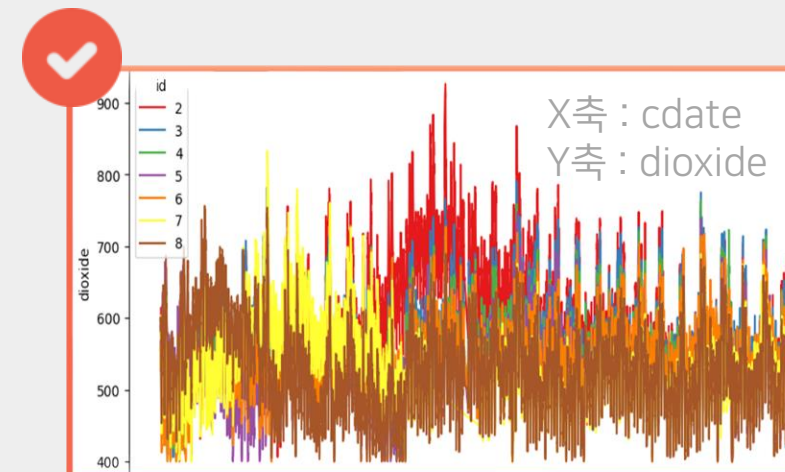


이상치 제거 전 humidity

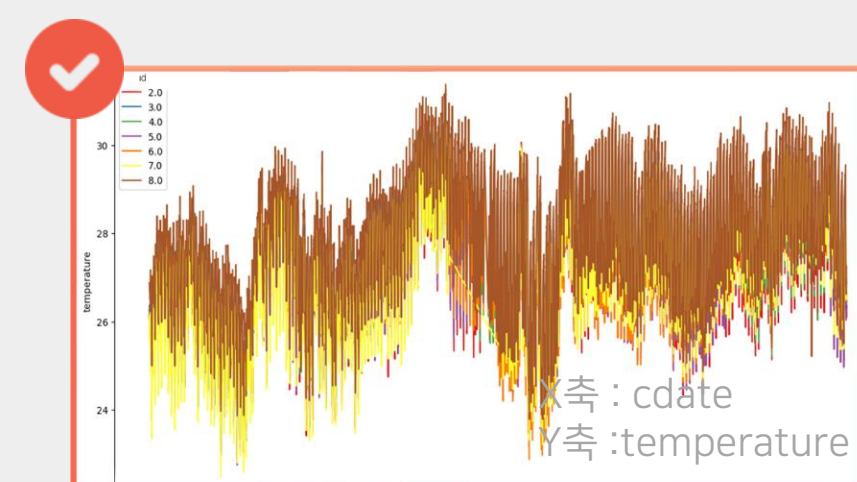
100을 초과하는 경우
노이즈로 판단



이상치 제거 후 humidity



dioxide



temperature

2.3

데이터 전처리



- ①porest_area_ahu_automatic
- ②porest_area_chiller_automatic
- ③porest_device_plant_log

[환경변수]

Plant

- 대상이 되는 plant no: 2~8
- 각 Plant는 환경변수 값들을 수집하는데 이중 temperature가 본 task의 예측값

Temperature
Humidity
Dioxide } plant2_temperature, plant3_humidity, plant4_dioxide, ...

out_temp Plant 9,10은 외부 환경변수를 나타냄. Plant 9, 10의 temperature 평균값 사용

- 01 노이즈 제거
- 02 10분 단위로 Resample 수행
- 03 결측값 처리

[기록 시점]

- 일정 시간 간격으로 기록

[Resample 규칙]

00:00<x<00:10 → 평균값 계산
→ 00:10 데이터로 넣기

[결측값 처리 규칙]

1. 옆 Plant에서 동일 시점 데이터 수집
 - 양 옆 Plant 모두 값 존재 시 평균값 사용
 - 한 Plant에만 값 존재 시 해당 값 사용
2. 해당 시점 모든 Plant에 결측값 존재 시
 - 24시간 단위로 해당 행 삭제
 - 연속된 시간이 짧은 결측값의 경우 Linear하게 보간



2.4. 최종 데이터 셋

Total 38953 entries, 38 columns

	cdate	ahu1_inv	ahu2_inv	ahu3_inv	ahu4_inv	ahu5_inv	ahu6_inv	ahu7_inv	ahu8_inv	ahu9_inv	...	plant6_temperature	plant6_humidity
0	2023-01-02 00:00:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25.931	35.506
1	2023-01-02 00:10:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25.932	35.550
2	2023-01-02 00:20:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25.917	35.566
3	2023-01-02 00:30:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25.890	35.619
4	2023-01-02 00:40:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	25.878	35.628
...
8779	2023-09-30 23:10:00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	27.740	46.280

03

예측모델 선택 과정

3.1 모델 탐색 과정

3.2 최종 모델 선정



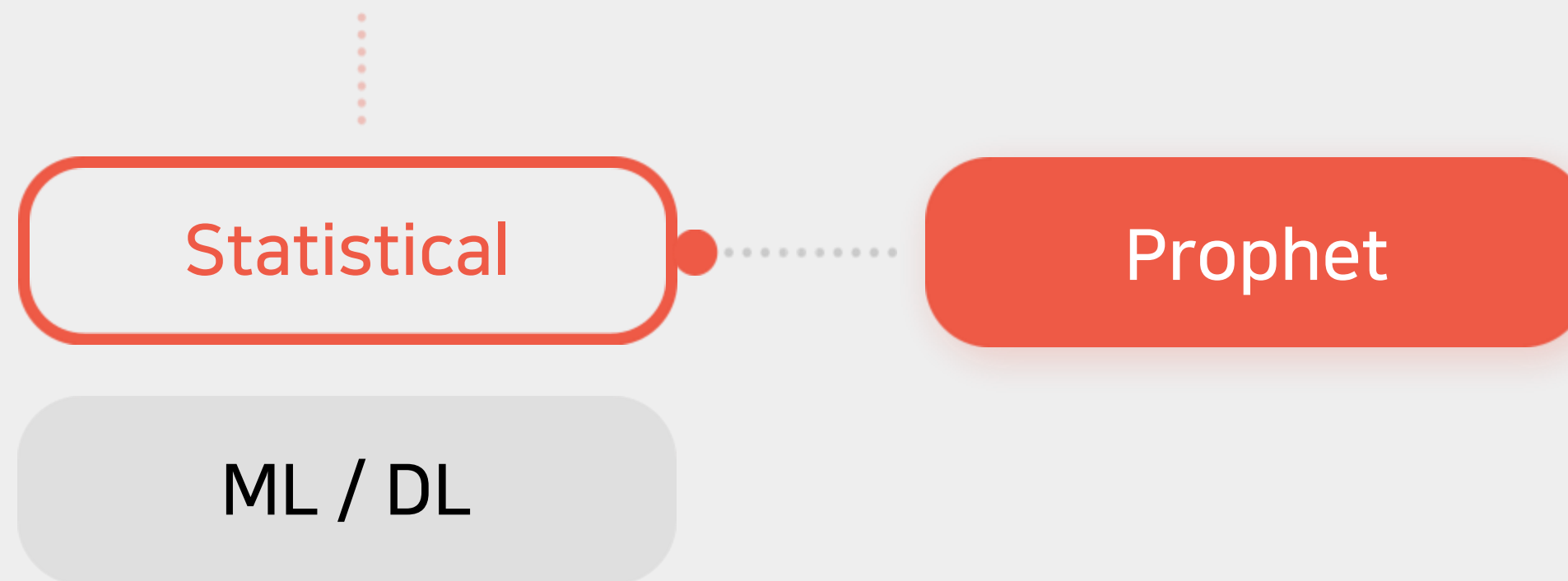
3.1

모델 선정 과정

시계열 예측 모델



Time Series Forecasting



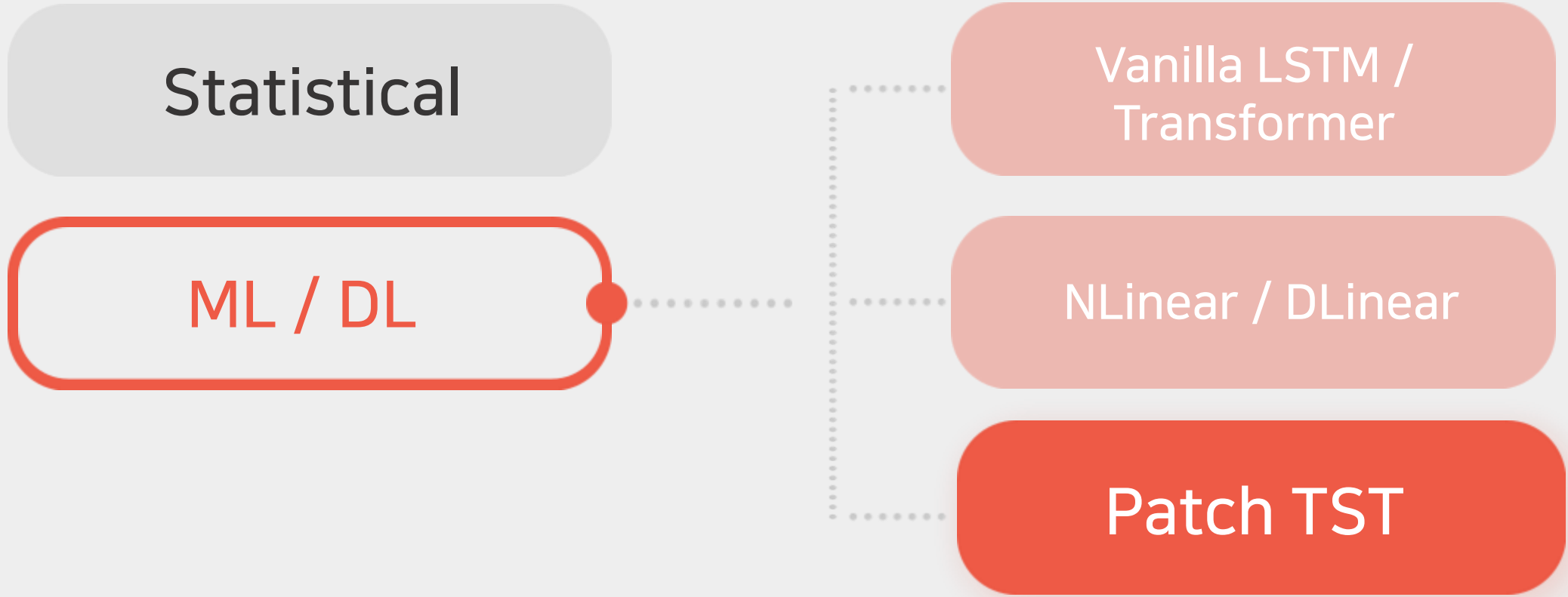
3.1

모델 선정 과정

시계열 예측 모델

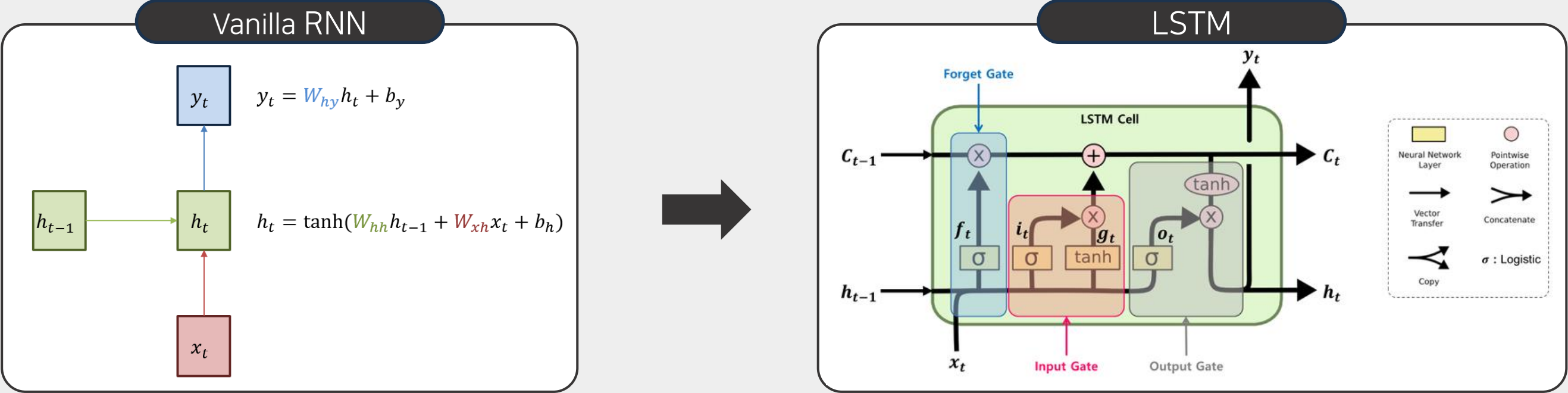


Time Series Forecasting



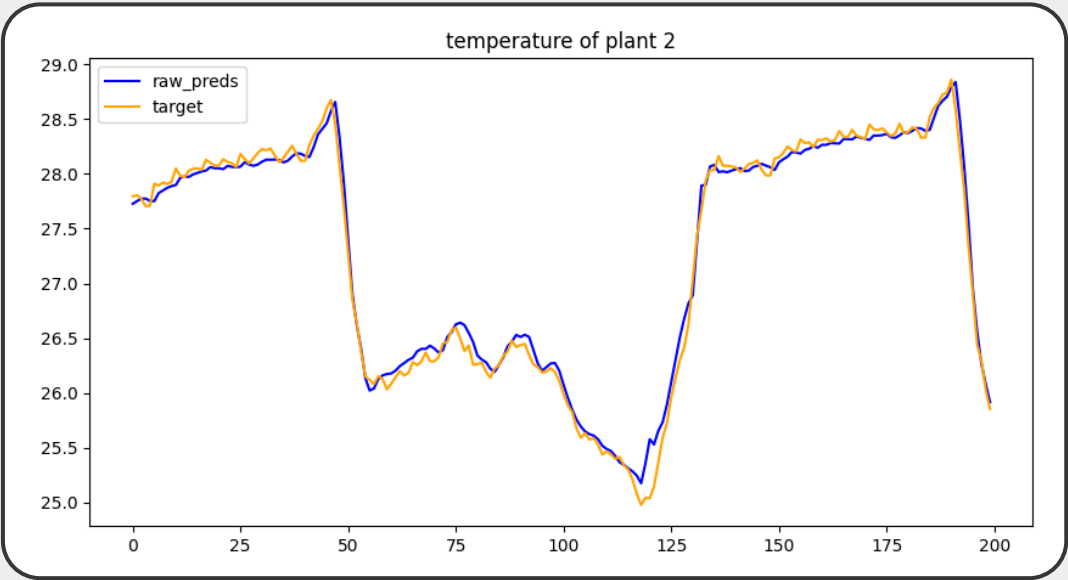
LSTM : Model Architecture & Evaluation

Long Short-Term Memory(1997)



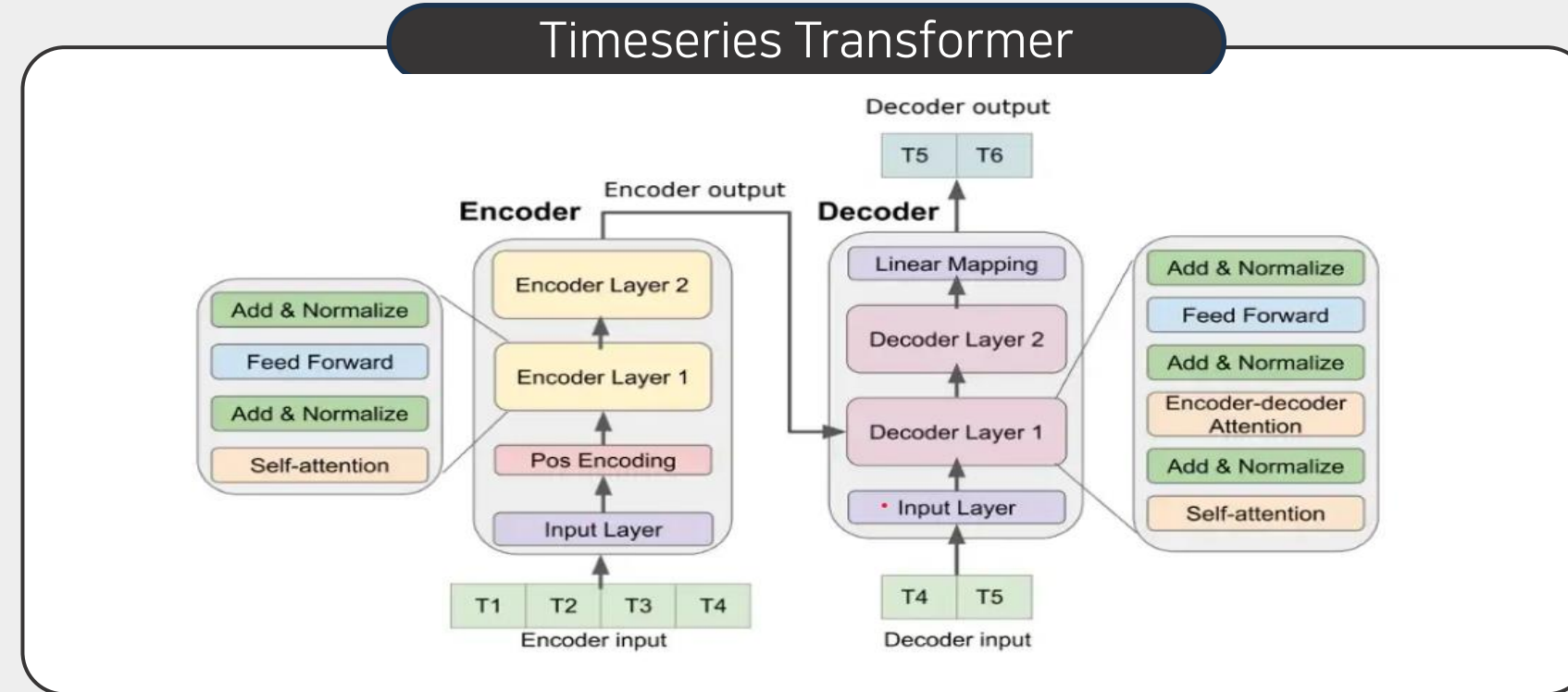
- 01 순환 신경망 RNN 방식을 기반으로 Long-range dependency(기울기 소실) 문제를 해결하고자 3가지의 Gate와 Cell-State를 추가한 모델
- 02 시계열 또는 시퀀스 데이터를 처리할 때 매우 유용하며, Gate를 통해 정보를 저장하고 어떤 정보를 버릴지 결정
- 03 Hyperparameter Tuning 결과, Window Size = 12, Learning Rate = 0.001 가 최적의 Parameter임을 확인
이때 성능지표와 Plot 결과는 아래와 같음

Metric	MSE	MAE	MAPE
LSTM	0.0191	0.096	0.0034



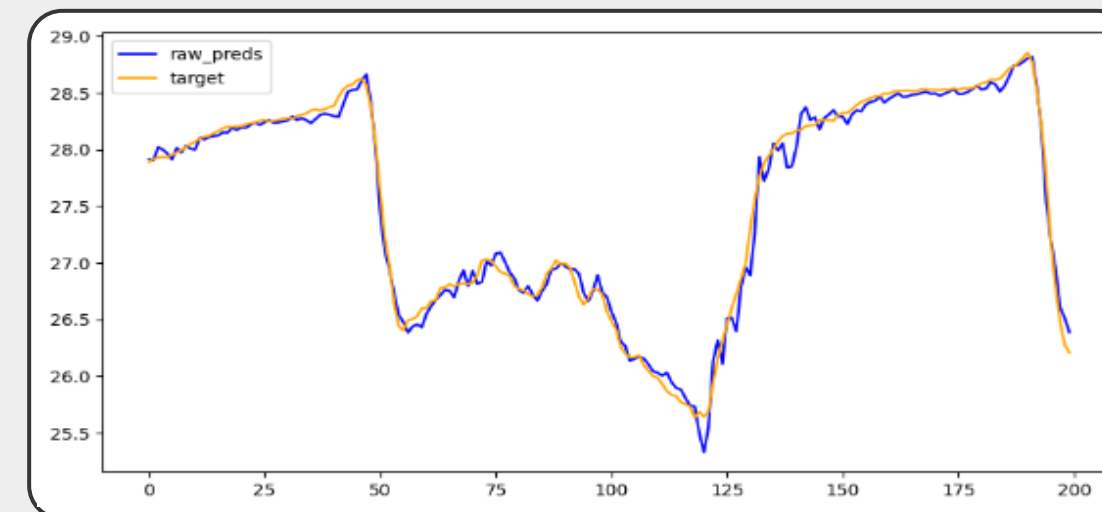
Time series Transformer : Model Architecture & Evaluation

Transformers in Time Series : A Survey(Wen et al. IJCAI 2023)



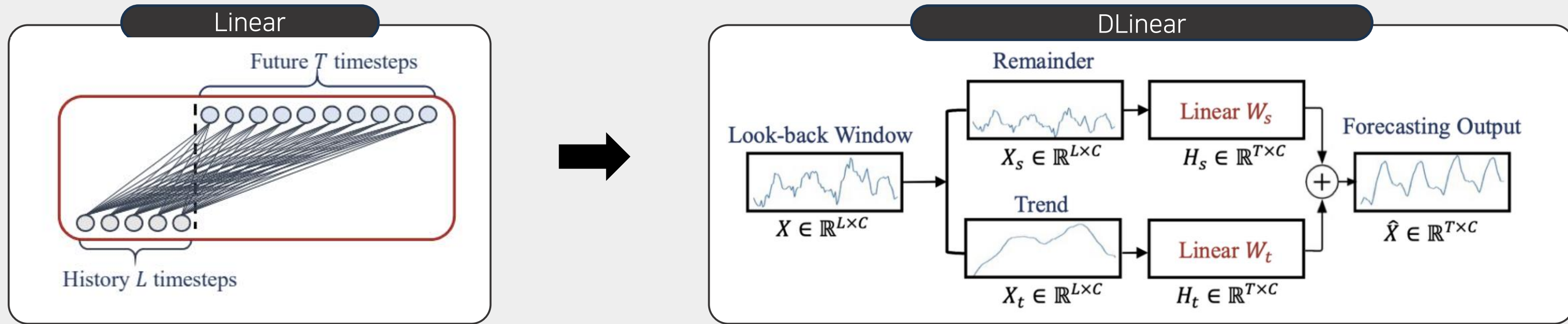
- 01 Transformer는 seq2seq의 encoder, decoder를 차용하되, RNN을 제거하고 attention을 사용한 모델
- 02 RNN은 시점에 따라 입력을 하나씩 받아 GPU를 사용한 병렬처리가 어렵지만, attention은 행렬 계산을 통해 문장을 처리하기 때문에 GPU를 사용한 병렬 처리가 가능
- 03 Self-Attention mechanism을 이용해 시계열 데이터를 처리하여 각 입력 시퀀스의 위치를 고려하여 주기성(Cycle), 추세(Trend) 등 시계열 데이터의 특징이라 보이는 패턴을 학습 가능
- 04 Hyperparameter Tuning 결과, Window Size =24, Learning Rate = 0.001, Batch size = 256 가 최적의 Parameter임을 확인 이때 성능지표와 Plot 결과는 아래와 같음

Metric	MSE	MAE	MAPE
TST	0.0234	0.1106	0.0040



DLinear : Model Architecture & Evaluation

Are Transformers Effective for Time Series Forecasting? (Zeng et al. AAAI 23)



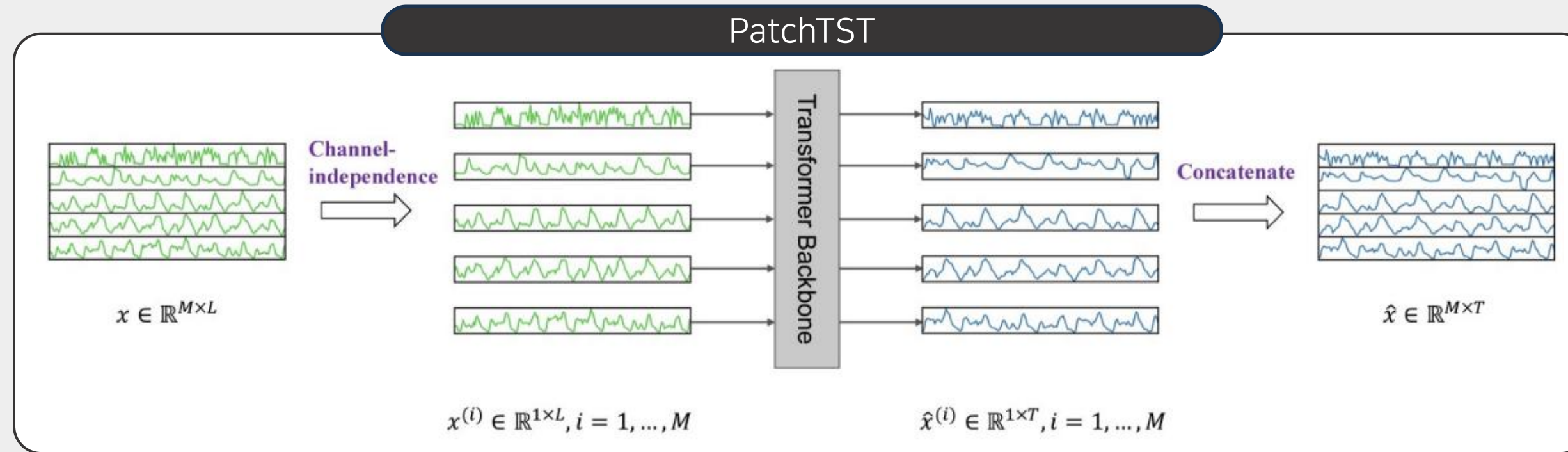
- 01 LTSF (Long Time Series Forecasting) 문제에서 Transformer 계열의 모델의 성능에 대한 의문점
- 02 간단한 선형 모델의 두가지 변형(NLinear, Dlinear)을 통해 장기 시계열 데이터에서 추세와 주기성에 대한 특징 추출에 뛰어난 성능을 보임
 - DLinear : Autoformer / FEDformer에서 사용된 Time Series Decomposition을 Linear모델에 적용
 - Nlinear : 마지막 시점의 값을 빼는 Normalization을 Linear 모델에 적용
- 03 Training Process : 모든 Plant에 대한 예측값의 성능지표

Metric	MSE	MAE	MAPE
DLinear	0.048	0.062	0.061

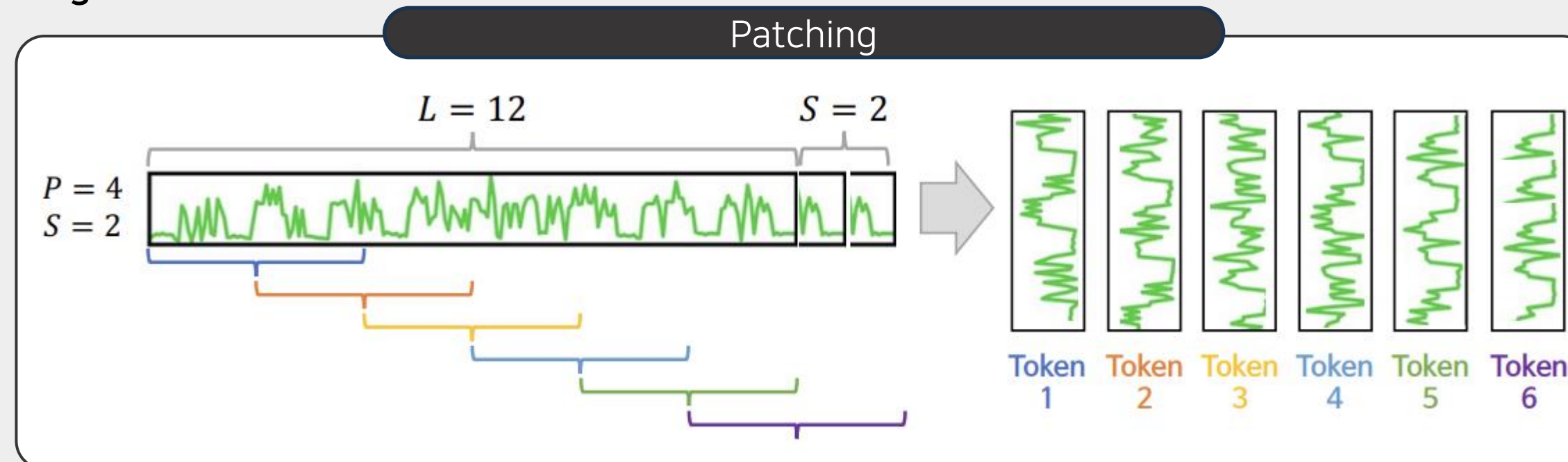


PatchTST : Model Architecture

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)



- 01 다변량 데이터의 Look – Back Window를 단변량 데이터로 분할한 뒤, 단변량 과거 데이터를 Transformer의 Input으로 넣어 Patching 및 Transformer를 통해 미래 데이터를 예측
- 02 모든 단변량 변수에 대한 미래 데이터를 통합하여 최종 다변량 미래 데이터를 도출함
- 03 Patching *



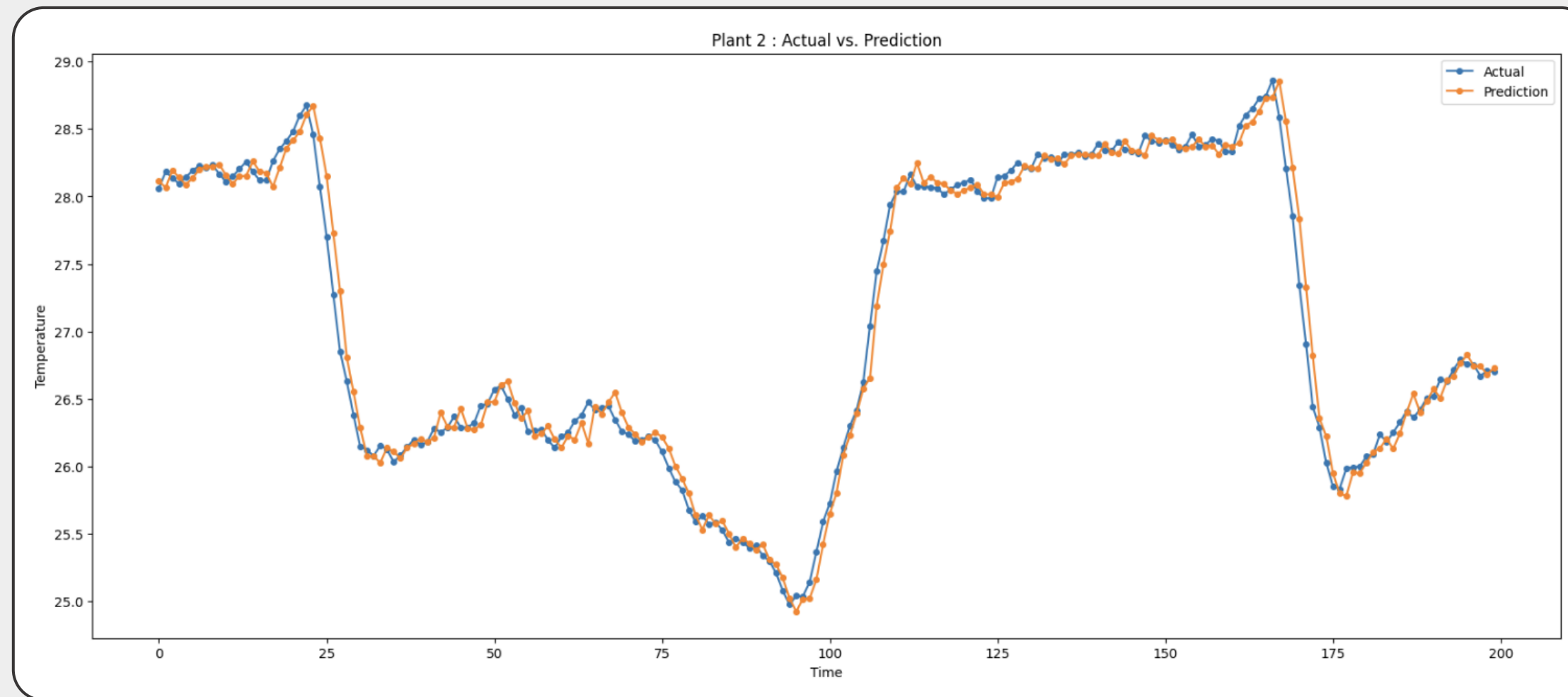
* 고려대학교 산업경영공학과 DSBA 연구실 최희정 박사과정 세미나 발표자료

PatchTST : Model Evaluation

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

01 Training Process : 모든 Plant의 예측값에 대한 성능 지표

Metric	MSE	MAE	MAPE
PatchTST	0.0105	0.0578	0.0021



02 Analysis

- Plot 결과, 아주 좋은 성능을 보이고 있음을 알 수 있음
- Prediction이 Actual을 따라가는 Lag이 생긴다는 점에서 Task 적합성이 조금 맞지 않음



Prophet: Model Architecture

Forecasting at Scale(Taylor et al. Facebook 2017)

Prophet 모형

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

- $g(t)$: 비주기적 변화를 반영하는 추세 함수
- $s(t)$: 주기적인 변화(ex, 주간/연간)
- $h(t)$: 휴일(불규칙 이벤트)의 영향력
- ϵ : 정규분포를 따르는 잔차라고 가정

01 Prophet: Facebook에서 개발한 시계열 예측 라이브러리

02 Prophet의 모형은 트렌드(growth), 계절성(seasonality), 휴일(holidays) 3가지의 main components로 이루어졌으며, 시간에 종속적인 구조를 가지지 않고 curve-fitting(곡선 맞춤)을 통해 예측 문제를 해결

- 유연성 : 계절적 요소와 트렌드에 대한 가정을 쉽게 반영
- ARIMA 모델과 달리 모델을 측정 간격을 regularly spaced 할 필요가 없고, 결측이 있어도 상관없음
- 빠른 fitting 속도 -> 여러 가지 시도 가능
- 직관적인 파라미터 조정을 통한 모델 확장 쉬움

03 조절 가능한 Hyperparameter의 종류가 많으며, 모델링 과정에서 아래와 같이 설정

growth = 'linear'
changepoints = None
n_changepoints = 25
changepoint_range = 0.8
changepoint_prior_scale = 0.05

seasonality_mode = 'additive'
seasonality_prior_scale = 10.0
yearly_seasonality = 'auto'
weekly_seasonality = 'auto'
daily_seasonality = 'auto'

holidays = None
holidays_prior_scale = 10.0
interval_width = 0.8
mcmc_samples = 0

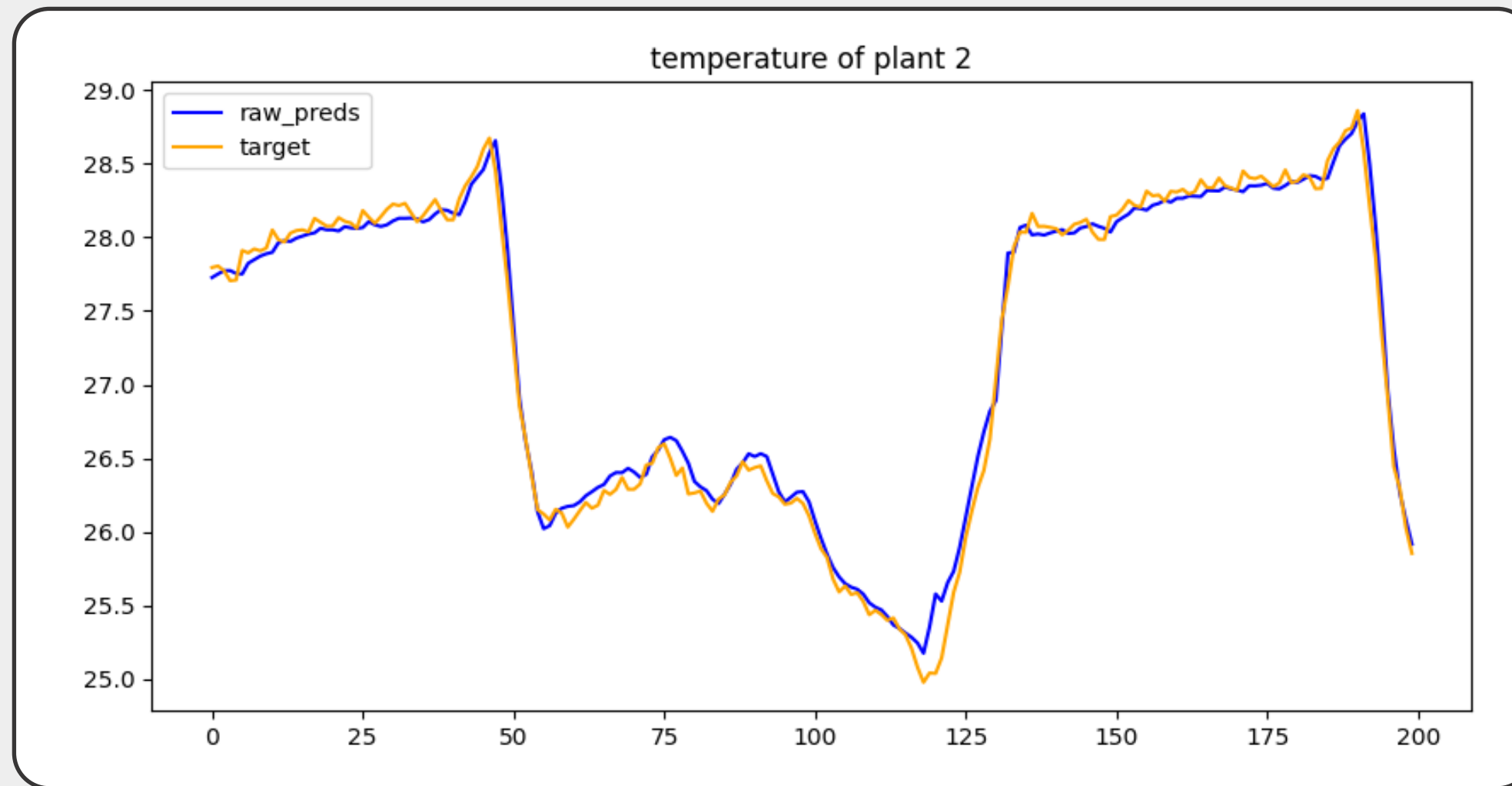


Prophet: Model Evaluation

Forecasting at Scale(Taylor et al. Facebook 2017)

01 Hyperparameter를 앞페이지와 같이 설정하였을 때 성능지표와 Plot 결과는 아래와 같음

Metric	MSE	MAE	MAPE
Prophet	0.0078	0.0704	0.0025



02 Analysis

- Plot 결과, 아주 좋은 성능을 보이고 있음을 알 수 있음
- Prediction이 Actual을 따라가는 Lag 또한 심하게 발생하지 않는다는 점에서 Task와 적합함



Model Evaluation : Validation Dataset

01 Evaluation Metric : MSE , MAE, MAPE

Metric	MSE	MAE	MAPE
LSTM	0.0191	0.0960	0.0034
TsT	0.0234	0.1106	0.0040
DLinear	0.0480	0.0620	0.0610
PatchTST	0.0105	0.0578	0.0021
Prophet	0.0078	0.0704	0.0025

02 Anlaysis

- 여러 Metric에서 PatchTST, Prophet이 좋은 성능을 보이는 것을 확인할 수 있음
- 최종 Model : PatchTST, Prophet



04

최종 모델 선정

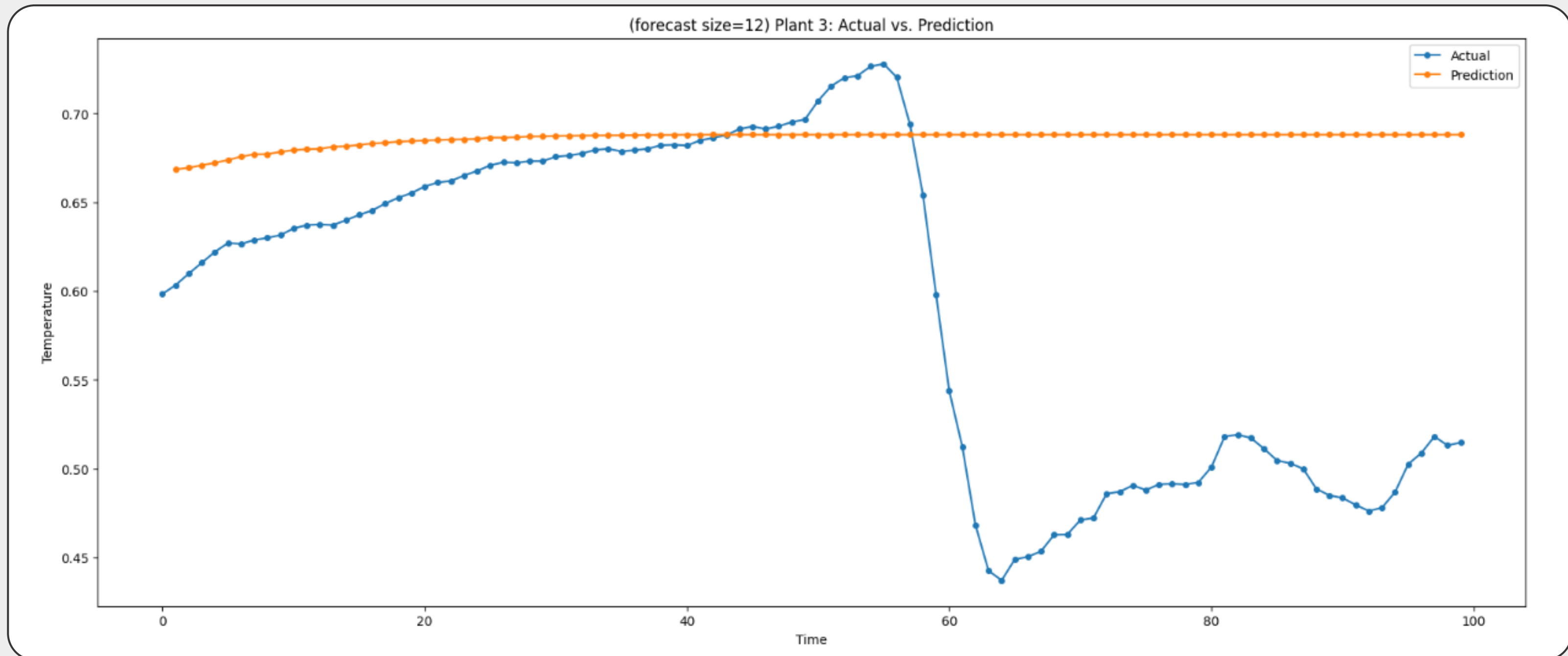


PatchTST: Inference

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

01 Inference Process : Inference with Validation Data

- ✓ Window Size = 24
- ✓ Forecast Size = 1



PatchTST: Inference

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

01 Inference Process : Inference with Validation Data

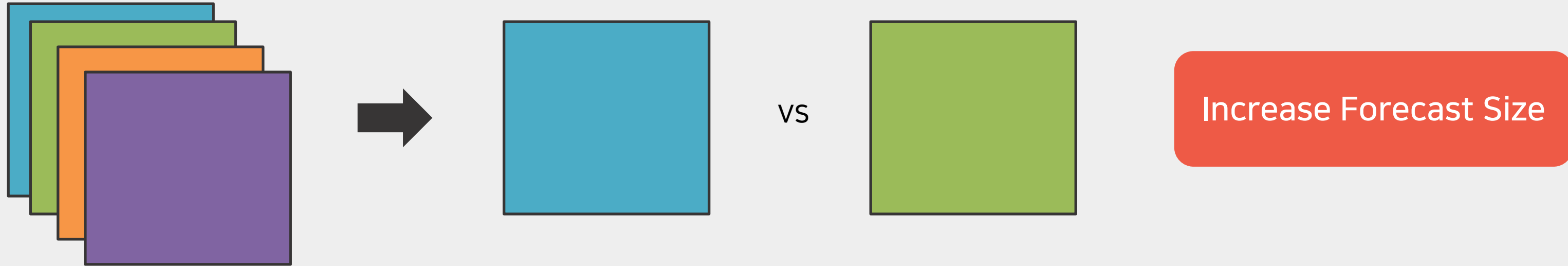
- ✓ Window Size = 24
- ✓ Forecast Size = 1



PatchTST: Inference

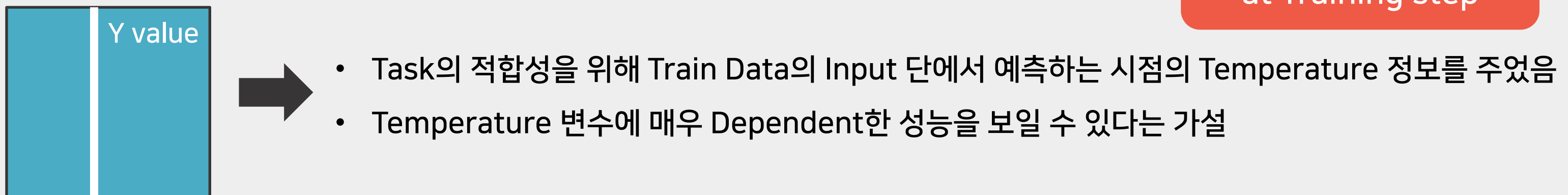
A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

02 Analysis on Result : Window Size (24) & Forecast Size (1)



- Blue Input과 Green Input은 24개의 Window Size (Input) 중에서 단 1개의 Row만 다른 값을 가지게 됨
- 이는 모델의 입력에 있어서 매우 작은 부분으로 작용하여 모델의 Output이 매우 Constant하다는 가설

03 Analysis on Result : Strong Dependency on Y Value

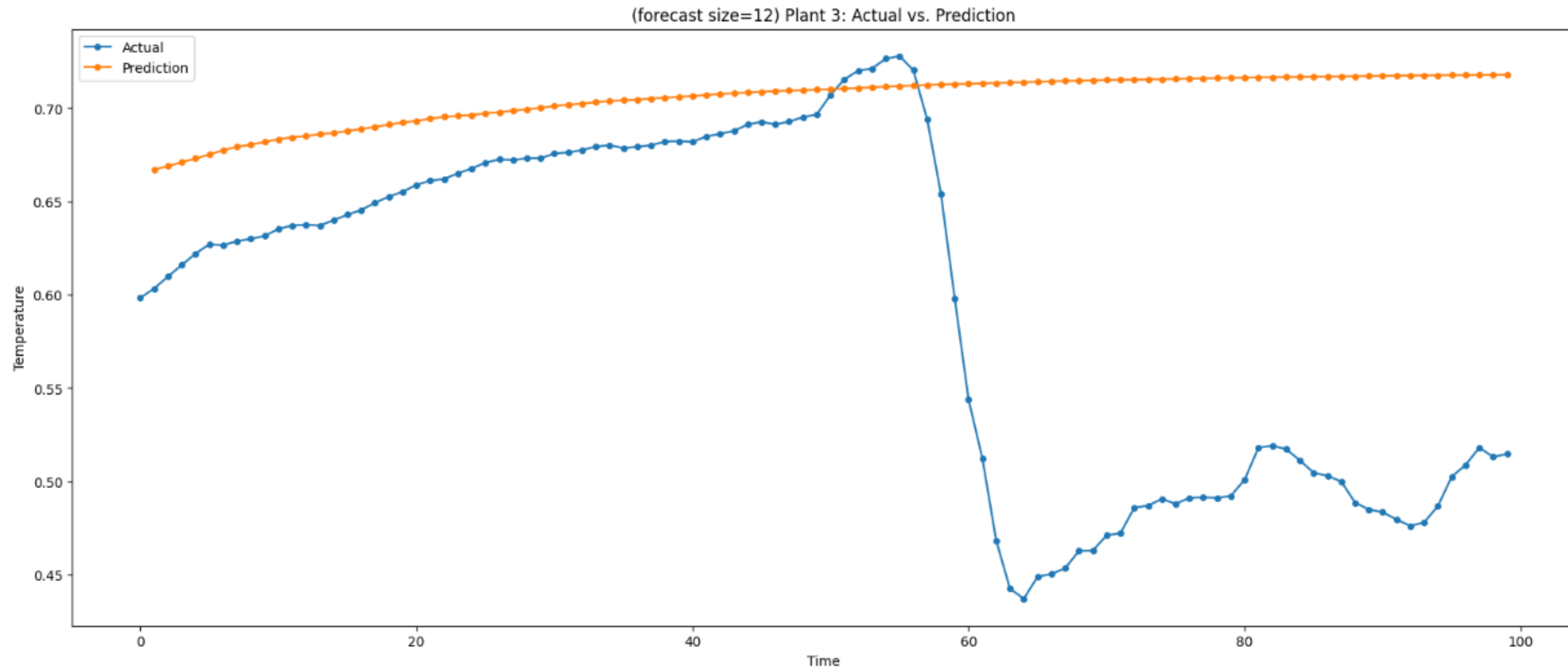


PatchTST: Inference

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

04 Inference Process : Inference with Validation Data

- ✓ Window Size = 24
- ✓ Forecast Size = 12
- ✓ Masking Ratio : 0.5



Prophet: Inference

Forecasting at Scale(2017)

01 모델 선정 과정에 사용하던 X변수들 중 실제 test data에 temperature값이 주어지지 않기에 temperature 변수를 삭제해 X변수 재정의

- X : ahu_inv, plant_dioxide, plant_humidity(plant 2~ 8), out_temp, chiller_inv
- y : ds, y * y: 예측하고자 하는 plant의 temperature 값

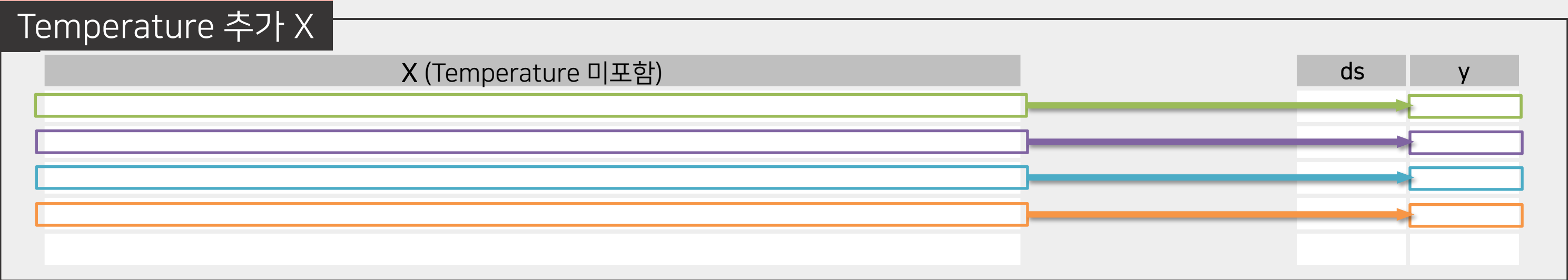
02 모델 선정 과정에서의 Hyperparameter와 동일하게 설정

growth = 'linear'
changepoints = None
n_changepoints = 25
changepoint_range = 0.8
changepoint_prior_scale = 0.05

seasonality_mode = 'additive'
seasonality_prior_scale = 10.0
yearly_seasonality = 'auto'
weekly_seasonality = 'auto'
daily_seasonality = 'auto'

holidays = None
holidays_prior_scale = 10.0
interval_width = 0.8
mcmc_samples = 0

03 X변수에 예측 temperature 변수 추가 유무를 기준으로 2가지 방식으로 Inference



Prophet: Inference

Forecasting at Scale(2017)

Temperature 추가 0

Transformed Dataset

	ahul_inv	...	plant7_humidity	plant8_dioxide	plant8_humidity	out_temp	chiller_inv
0	0.0	...	35.677000	602.200000	35.712000	25.621889	0.0
1	0.0	...	35.850000	605.400000	35.539000	25.563875	0.0
2	0.0	...	35.911000	602.818182	35.448182	25.528000	0.0
3	0.0	...	35.957778	611.333333	35.503333	25.444500	0.0
4	0.0	...	35.972000	607.900000	35.548000	25.425667	0.0
...
34627	0.0	...	40.190000	600.750000	39.410000	28.880000	42.0
34628	0.0	...	40.310000	599.500000	39.510000	28.820000	42.0
34629	0.0	...	40.650000	598.620000	39.690000	28.980000	42.0
34630	0.0	...	41.000000	599.380000	39.900000	29.000000	42.0
34631	0.0	...	41.320000	599.860000	39.730000	28.980000	42.0

▲ X_train

	ds	y
1	2023-01-02 00:10:00	26.102
2	2023-01-02 00:20:00	25.988
3	2023-01-02 00:30:00	25.957
4	2023-01-02 00:40:00	25.953
5	2023-01-02 00:50:00	26.044
...
34628	2023-08-31 23:10:00	28.320
34629	2023-08-31 23:20:00	28.280
34630	2023-08-31 23:30:00	28.270
34631	2023-08-31 23:40:00	28.320
34632	2023-08-31 23:50:00	28.320

▲ y_train

Prophet 모델은 t시점의 X가 t시점의 y를 예측하는 구조로 설계
 → Task에 맞추어 t시점의 X변수가 t+1시점의 y변수를 예측할 수 있도록 데이터 셋을 변형함

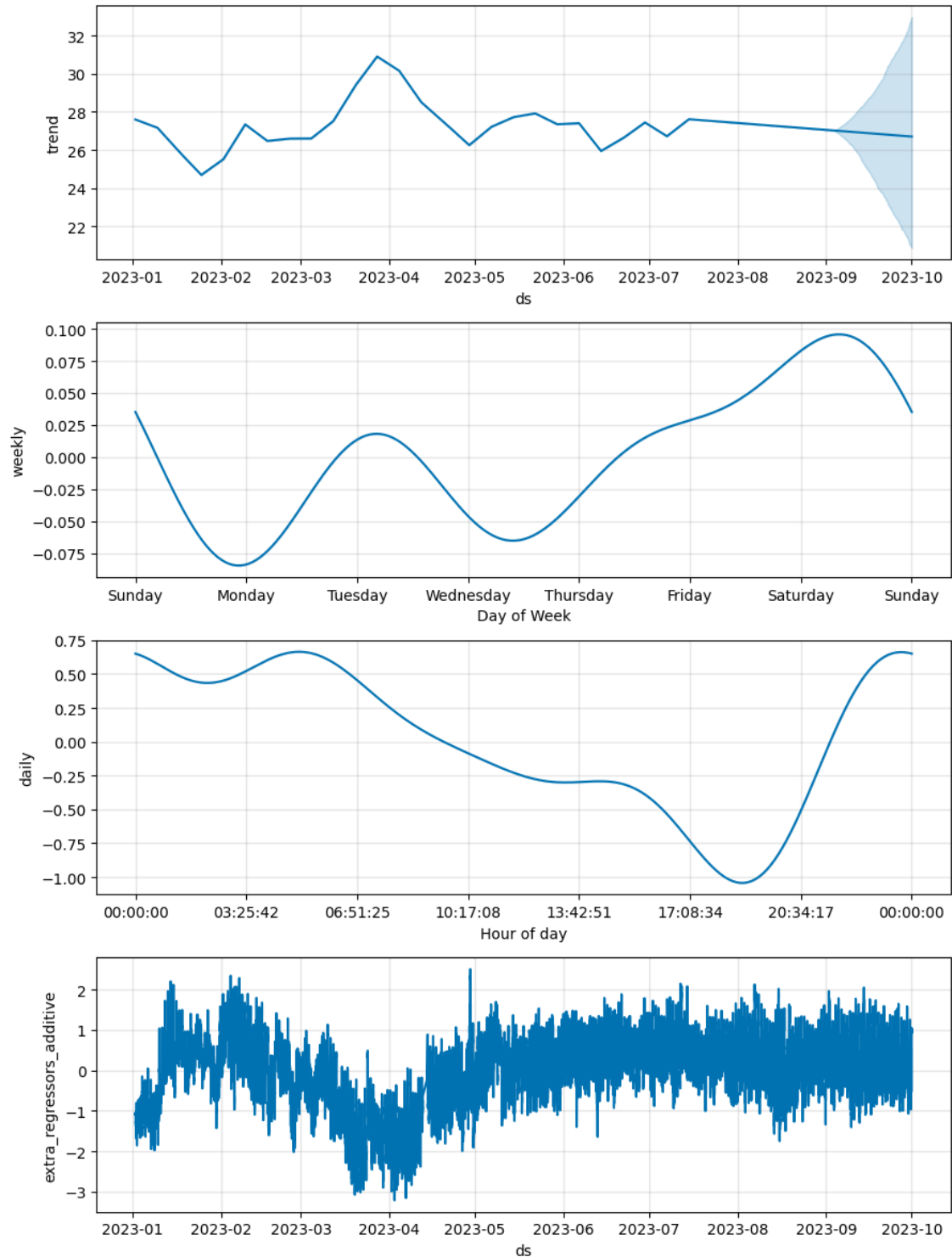


Prophet: Inference

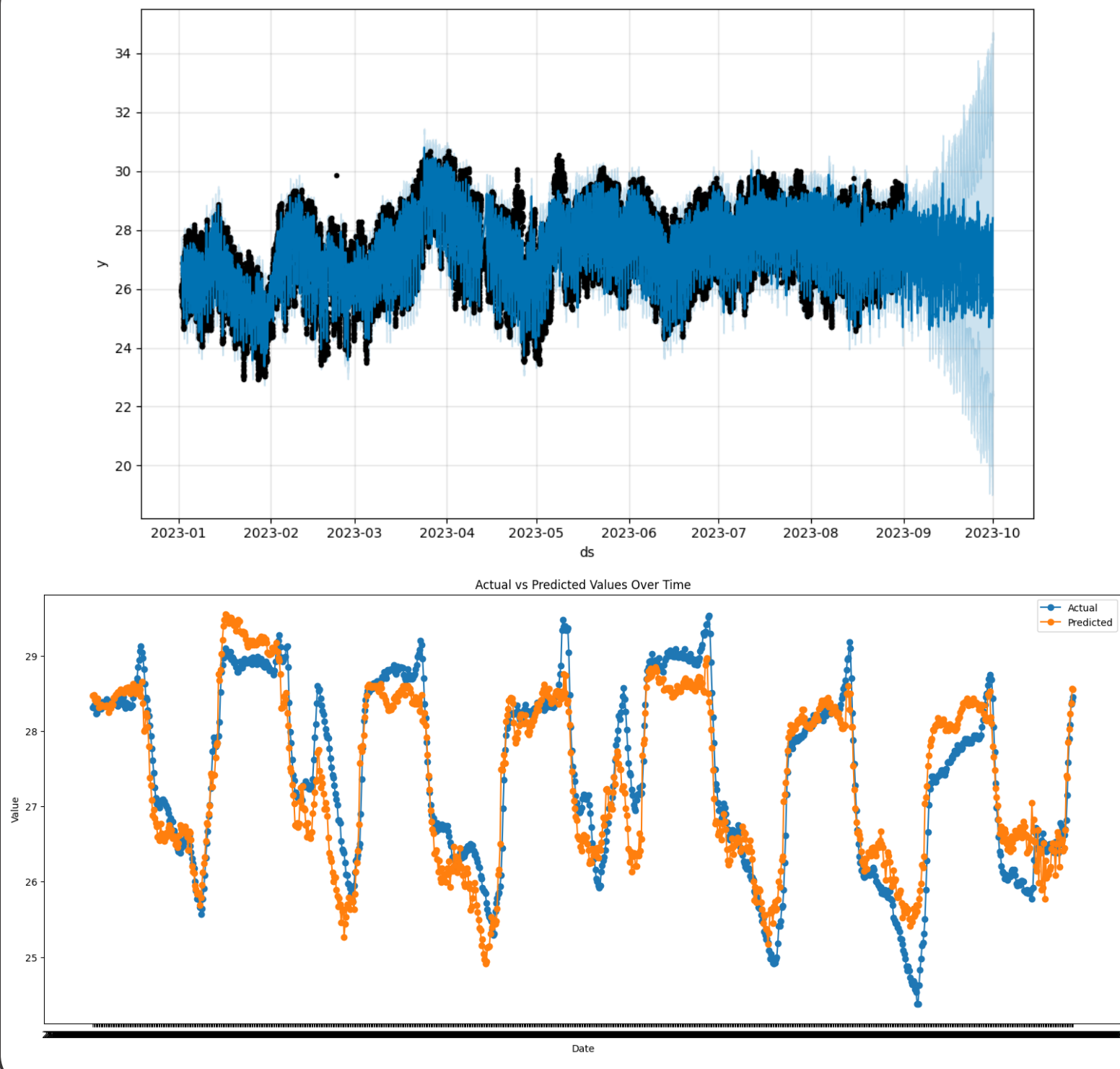
Forecasting at Scale(2017)

05 Temperature 추가 X

Components



Prediction

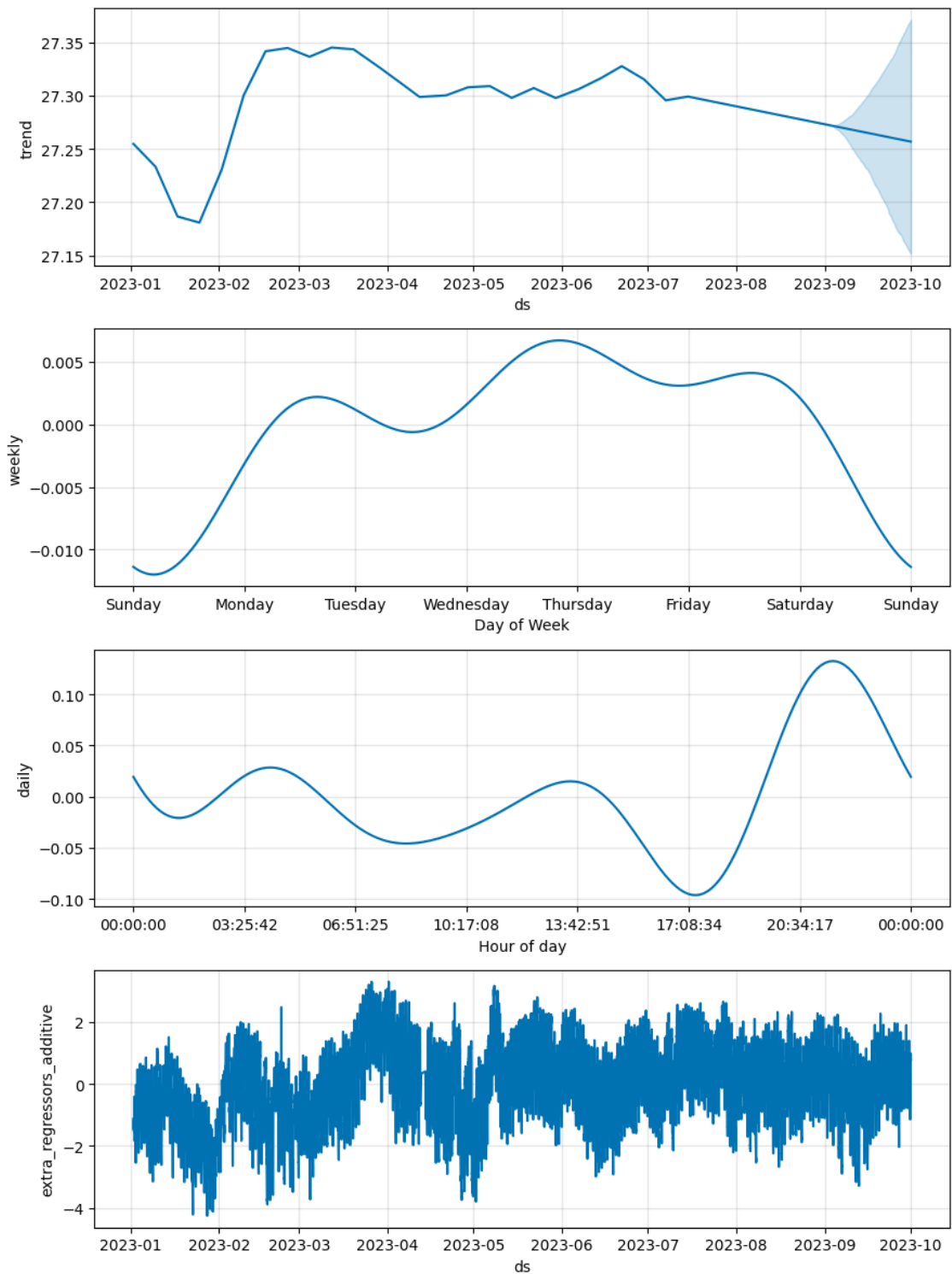


Prophet: Inference

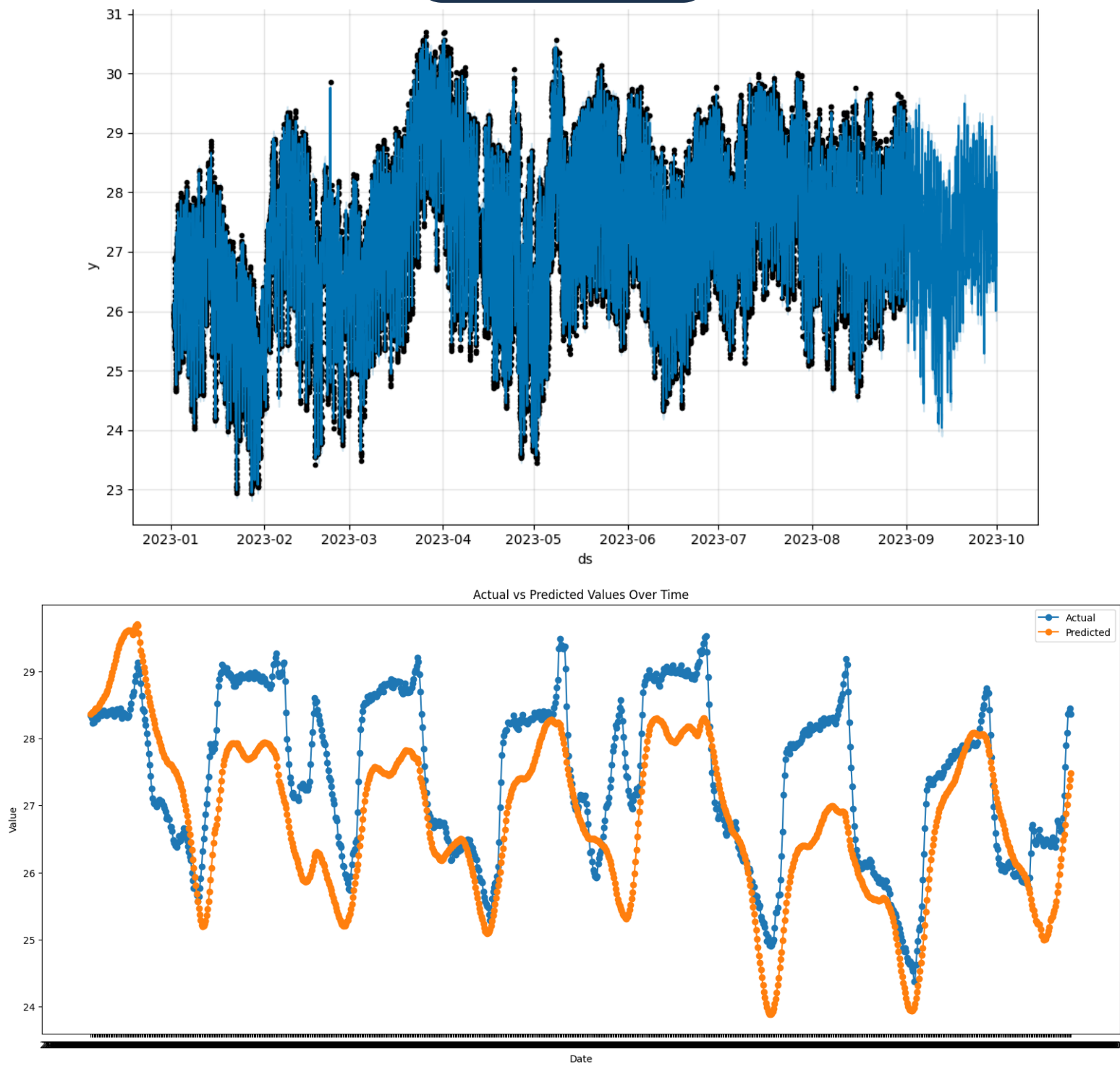
Forecasting at Scale(2017)

06 Temperature 추가 0

Components



Prediction



05

모델 평가 : Test Data

5.1. PatchTST Evaluation

5.2. Prophet Evaluation

5.3 Ensemble



5.1

PatchTST : Evaluation

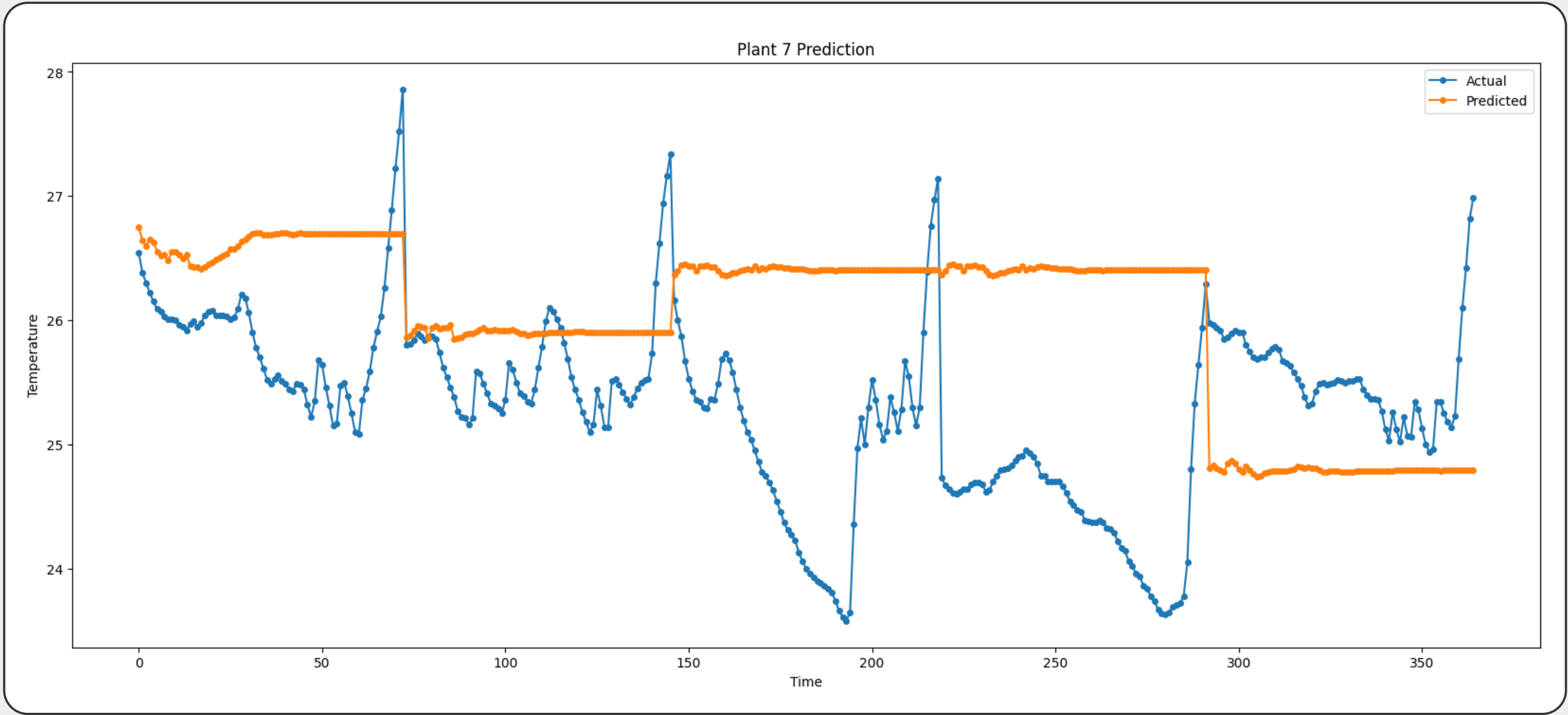


PatchTST : Model Evaluation

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

01 Test 데이터로 Inference 후 성능 평가 결과

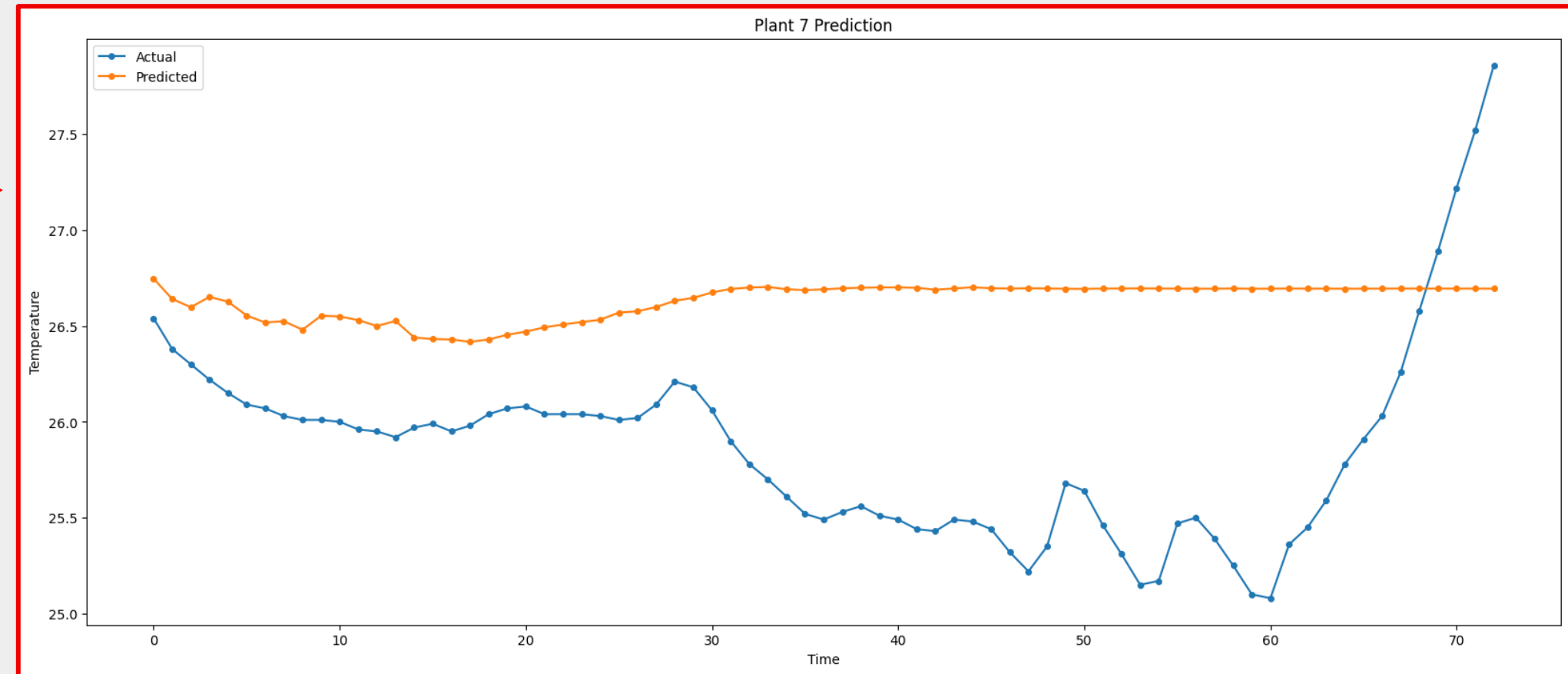
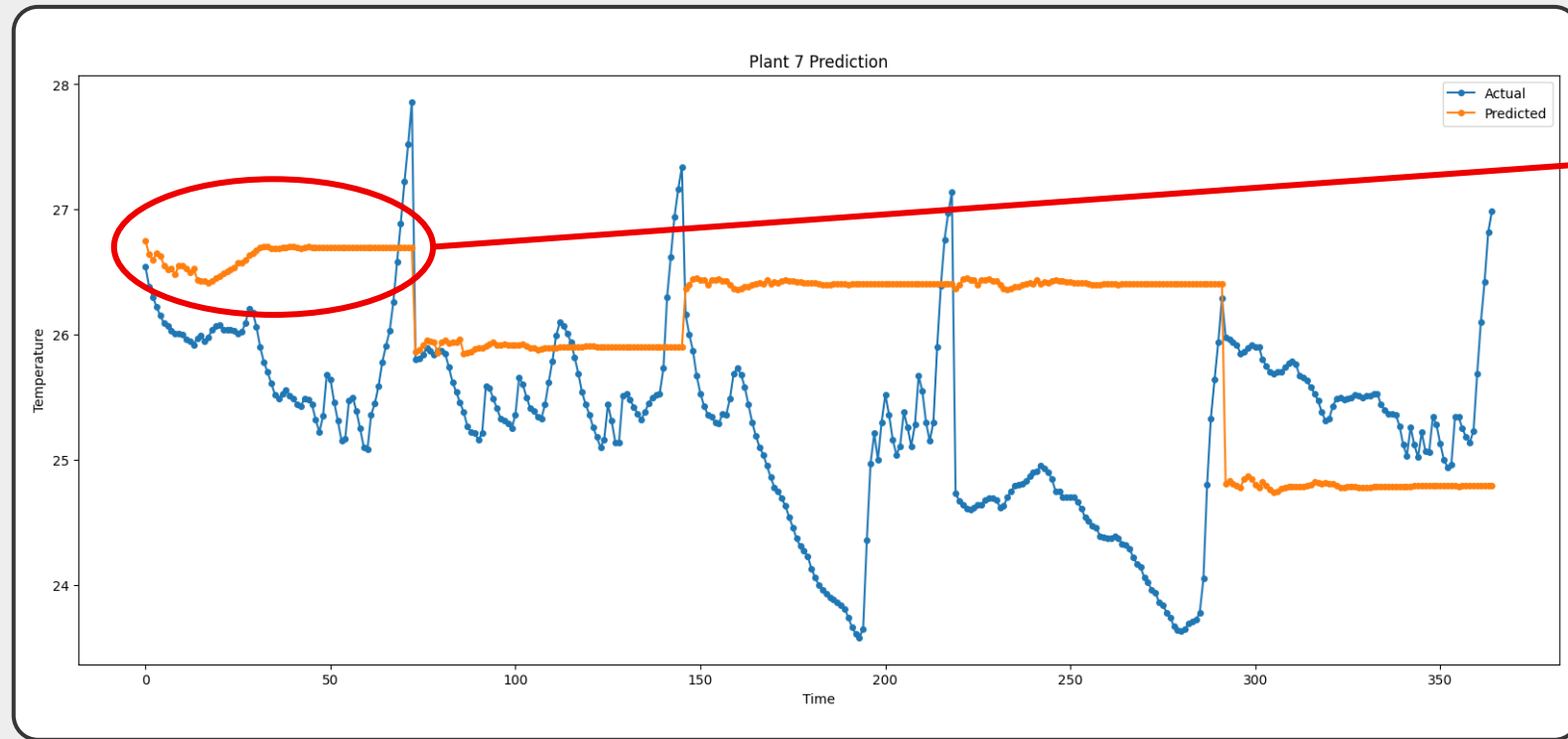
Metric	MSE	MAE	MAPE
PatchTST	1.2220	0.9250	0.0357



PatchTST : Model Evaluation

A Time Series is Worth 64 Words : Long – Time Forecasting with Transformers (Nie et al. ICLR 23)

02 Analysis



10-02 Plant 7 예측결과

초반 예측을 해나가다 어느 지점부터 예측 결과가 거의 일직선을 이룸

- 하루 내 예측값의 변화가 거의 없음
- t 시점 예측값 (\hat{y}_t) 예측 시 넣어준 이전 시점 예측값 (\hat{y}_{t-1})의 영향력이 매우 크다는 것을 알 수 있음
- 모델 학습 시 y 값들에 대해 masking을 수행했는데도 불구하고, inference 시 y 값에 의해 결과가 dominate됨을 알 수 있음
- 이는 PatchTST 모델의 한계로 볼 수도 있음



5.2

Prophet : Evaluation

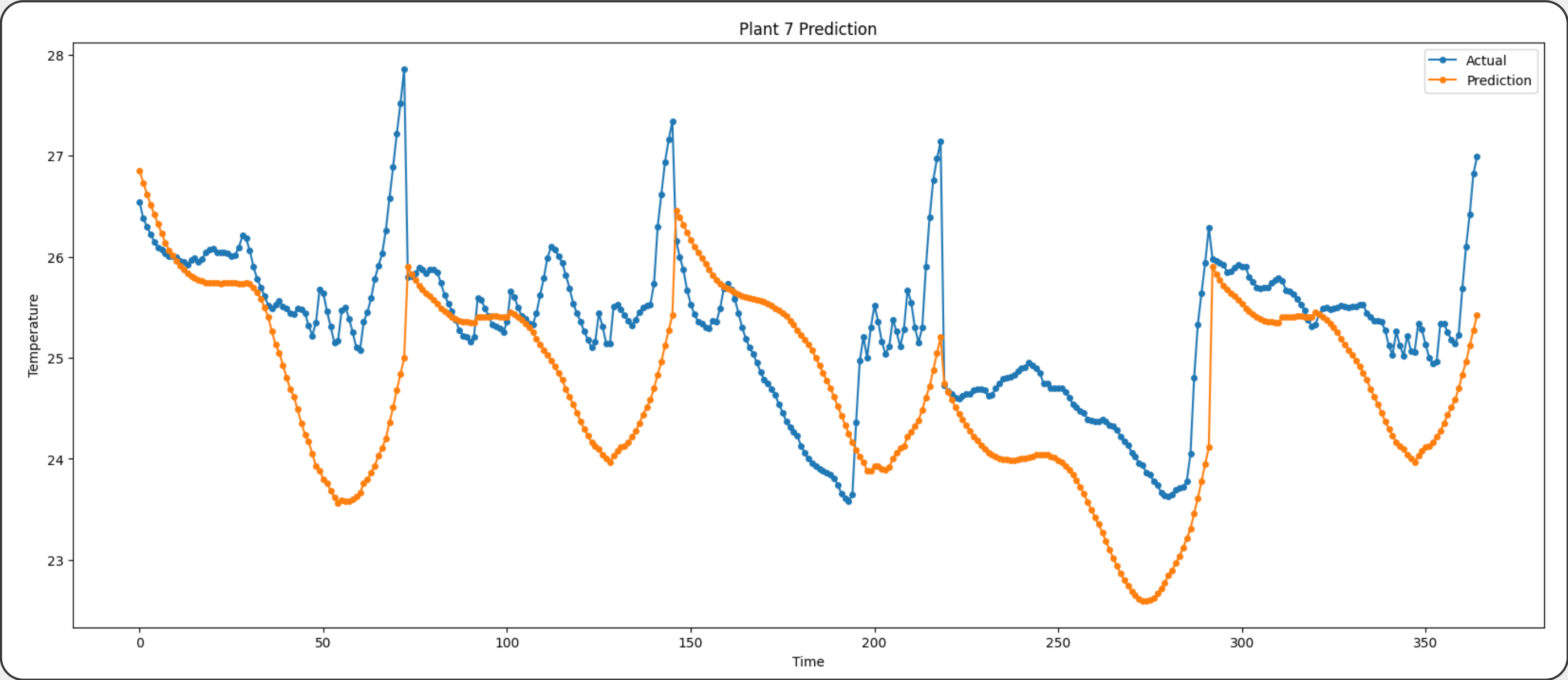


Prophet: Model Evaluation

Forecasting at Scale(2017)

01 Test 데이터로 Inference 후 성능 평가 결과

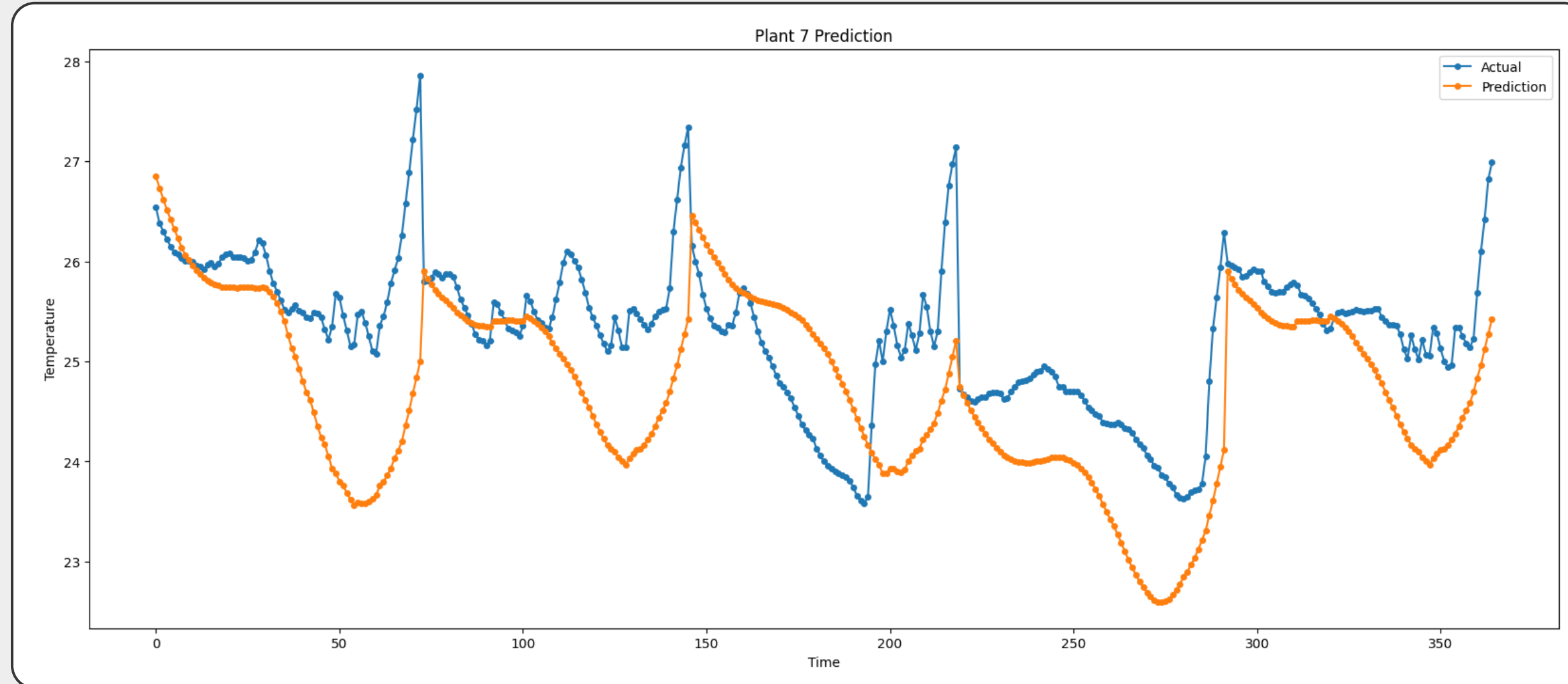
Metric	MSE	MAE	MAPE
Prophet	1.9601	1.1224	0.0427



Prophet: Model Evaluation

Forecasting at Scale(2017)

02 Analysis



예측 결과가 거의 일직선을 이루었던 PatchTST에 비해 추세를 반영하여 예측하고 있음

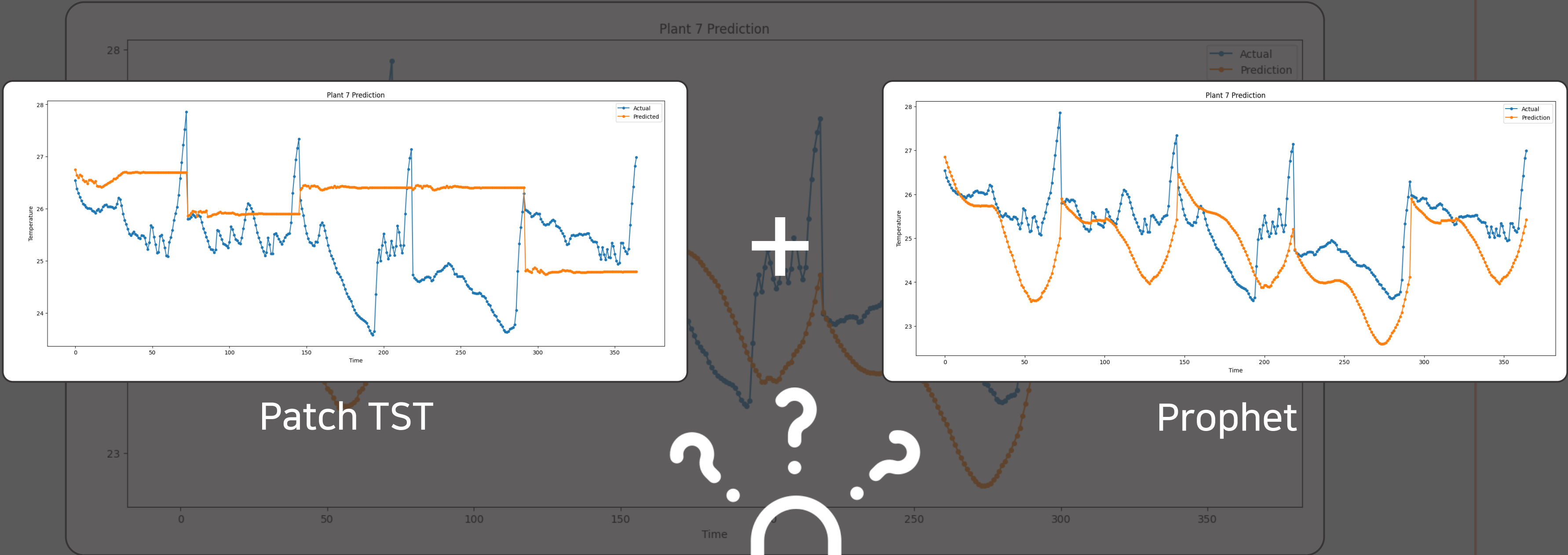
- 딥러닝이 아닌 선형 회귀를 기반으로 하기 때문에 y값에 덜 종속적
- 제어 데이터를 좀 더 잘 반영하여 예측할 수 있기 때문이라고 판단



Prophet: Model Evaluation

Forecasting at Scale(2017)

02 Analysis



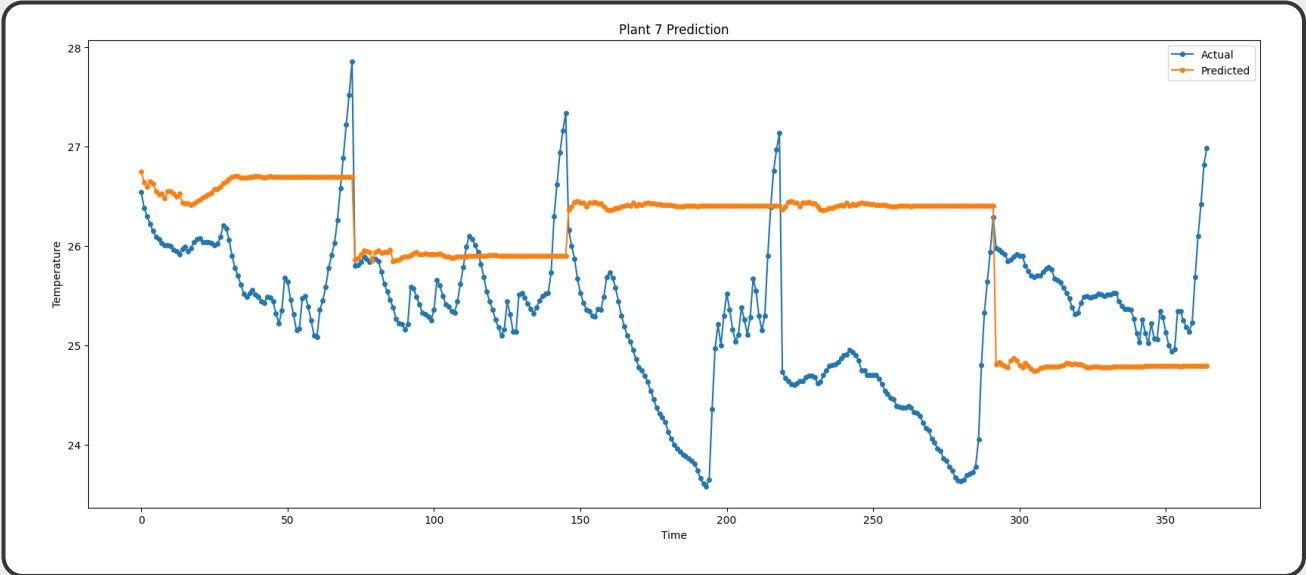
5.3

Ensemble : PatchTST + Prophet

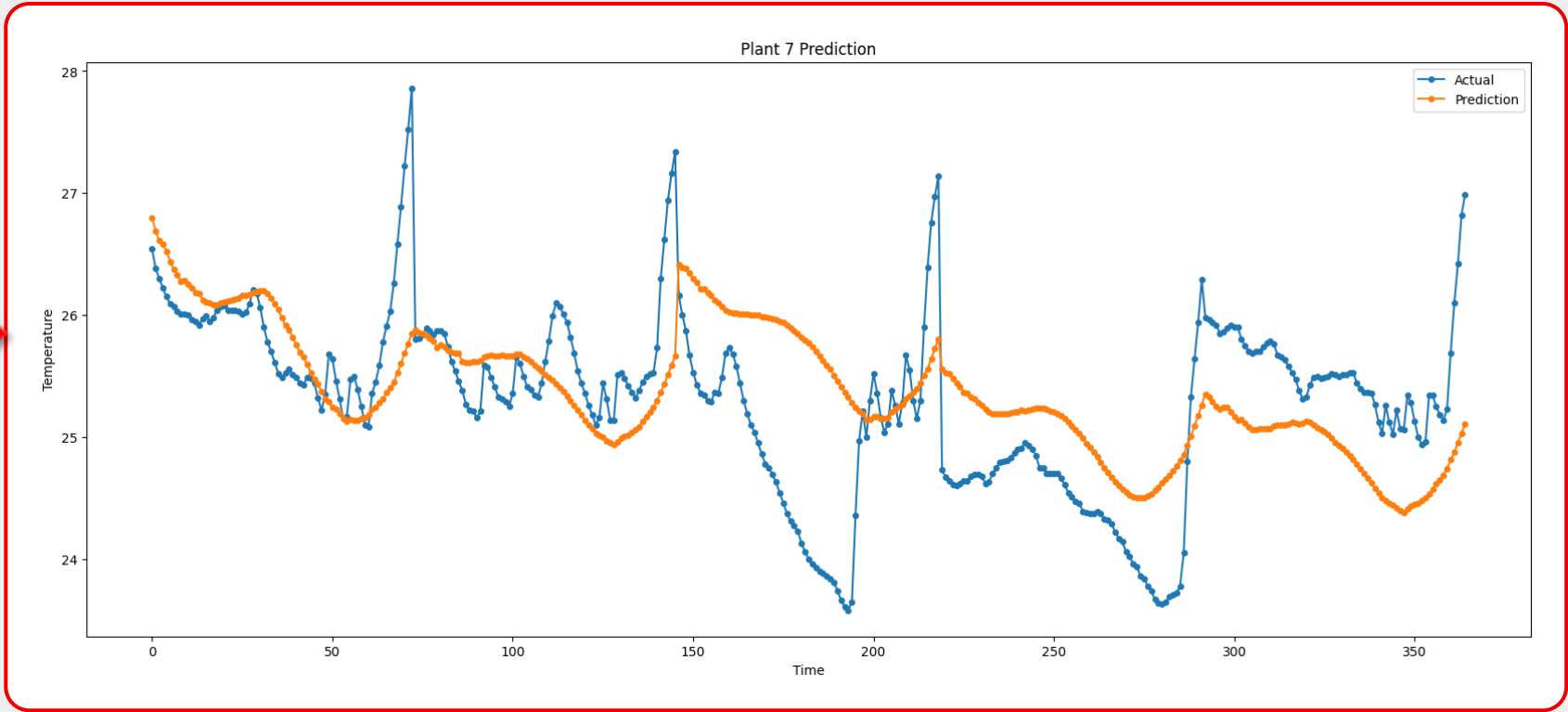
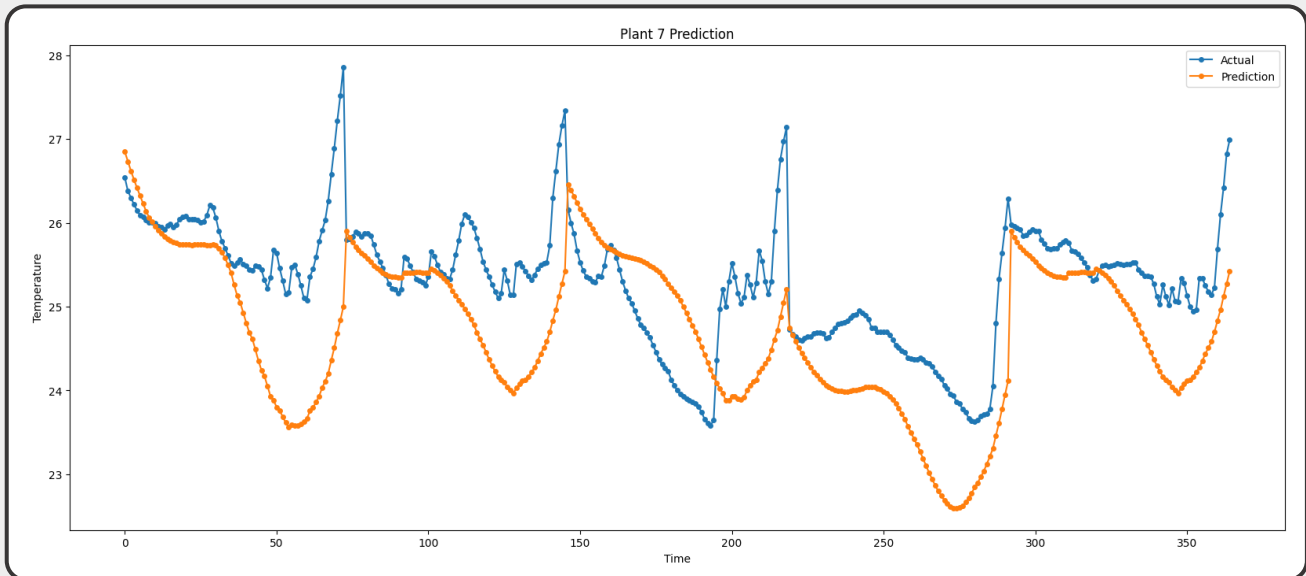


Ensemble: PatchTST + Prophet

01 Test 데이터로 Inference 후 성능 평가 결과



+



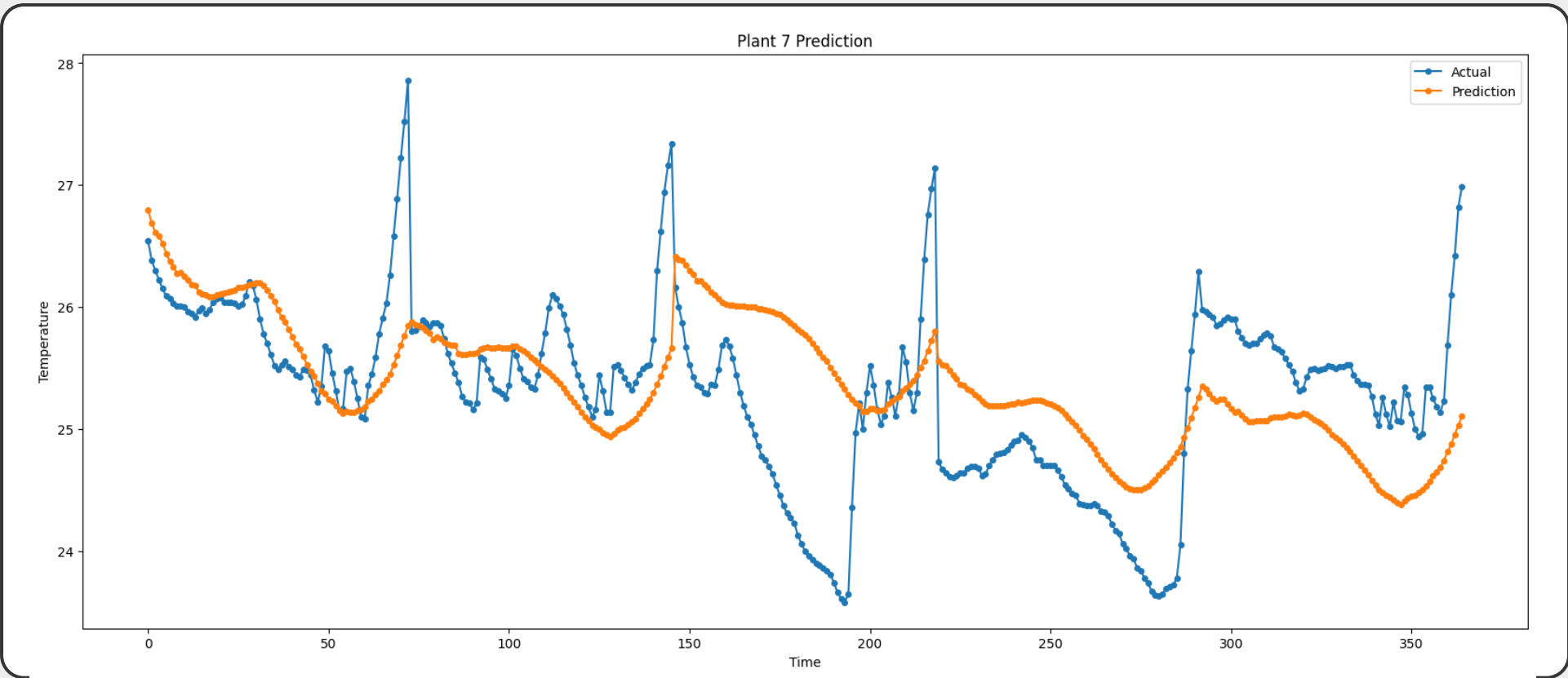
Ensemble: PatchTST + Prophet

01 Test 데이터로 Inference 후 성능 평가 결과

Metric	MSE	MAE	MAPE
PatchTST	1.2220	0.9250	0.0357

Metric	MSE	MAE	MAPE
Prophet	1.9601	1.1224	0.0427

Metric	MSE	MAE	MAPE
Ensemble	0.4043	0.4963	0.0189



두 모델의 예측값의 평균으로 평가한 결과, 모든 평가 지표값들이 개선되었음

- 하루 내 예측값의 변화가 거의 없음
- 학습 단계에서부터 두 모델의 앙상블 버전으로 학습시킬 경우 성능 향상 기대됨



06

Conclusion





발표를 들어주셔서

감사합니다

