
AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting

Oleksandr Shchur, Caner Turkmen, Nick Erickson, Huibin Shen, Alexander Shirkov, Tony Hu, Yuyang Wang (Amazon Web Services, AWS AI Labs)

성균관대학교
통계학과

2024710133 이다인
2023711351 김민경

<https://arxiv.org/pdf/2308.05566>
2024년 05월 27일

목차

01. Intro

02. Review

03. Method

04. Experiment

05. Discussion

06. Data Analysis

01 Intro

중요성

- ### 시계열 예측(Time Series Forecasting)
- 시간의 흐름에 따라 변화하는 데이터를 분석하여 미래의 값을 예측하는 기법
 - 금융, 경제, 날씨, 주식 시장, 판매량 예측 등 다양한 분야에서 활용
 - 여러 분야에서 미래를 정확히 예측하는 것은 필수적

한계점

- 실무자들은 여전히 가장 적합한 방법을 선택하고 예측 파이프라인 구현, 평가가 어려움
- 시계열 데이터는 일반적인 테이블 형태의 데이터와는 다른 특성
- 시계열 예측에는 불확실성을 고려하는 것이 중요

01 Intro

AutoML과 시계열 예측의 발전

현재 상황

문제점

- 기존 AutoML 연구는 테이블 데이터 중심의 분류/회귀 문제
AutoML(Automated Machine Learning) : 기계학습 모델의 개발 과정을 자동화하는 기술
- 시계열 예측과 확률적 예측 지원 미흡

해결책

새로운 AutoML 프레임워크

- 사용하기 쉬운 인터페이스, 확률적 예측 지원
- 전문 지식이 부족한 실무자도 쉽게 사용 가능
- 더욱 정확한 예측을 통한 신뢰성 있는 의사결정
- 시계열 예측의 중요성 증대, AutoML 기술 수요 증가
- 실무자의 어려움 해소, 시계열 예측 분야 발전

01 Intro

시계열 예측을 위한 AutoML

필요성

- 시계열 데이터의 특성: 시간 의존성, 계절성
- 기존 연구 부족, 발전의 여지 큼

최근 연구

- 딥러닝과 AutoML의 접목
- 전처리, Feature 엔지니어링 자동화
- Probabilistic Forecasting 지원
- 대표 프레임워크: AutoTS, AutoPyTorch-Forecasting, AutoGluon-TimeSeries

01 Intro

AutoGluon-TimeSeries(AG-TS) 프레임워크 개발

AG-TS

- 사용 편의성과 견고성에 초점을 맞춘 오픈소스 AutoML 라이브러리
- 다양한 예측 모델의 앙상블을 활용하여 짧은 학습 시간 내에 높은 정확도를 제공

차별점

- 기존의 통계 모델, 머신러닝 기반 예측 기법, 그리고 앙상블 기술을 모두 통합한 프레임워크
- Point forecast뿐만 아니라 Probabilistic forecast까지 지원
- 사용 편의성을 극대화

02 Review

Probabilistic Time Series Forecasting 문제 정의

데이터 (D)

- $D = \{y_{i,1:T_i}\}_{i=1}^N$: N개의 univariate 시계열의 집합
- $y_{i,1:T_i} = (y_{i,1}, \dots, y_{i,T_i})$
- $y_{i,t}$: 시간 t에서의 i번째 시계열의 값 (예: t일에 판매된 i번 제품의 수량)
- T_i : i번째 시계열의 길이

목표

각 시계열의 미래 H 구간의 값을 예측

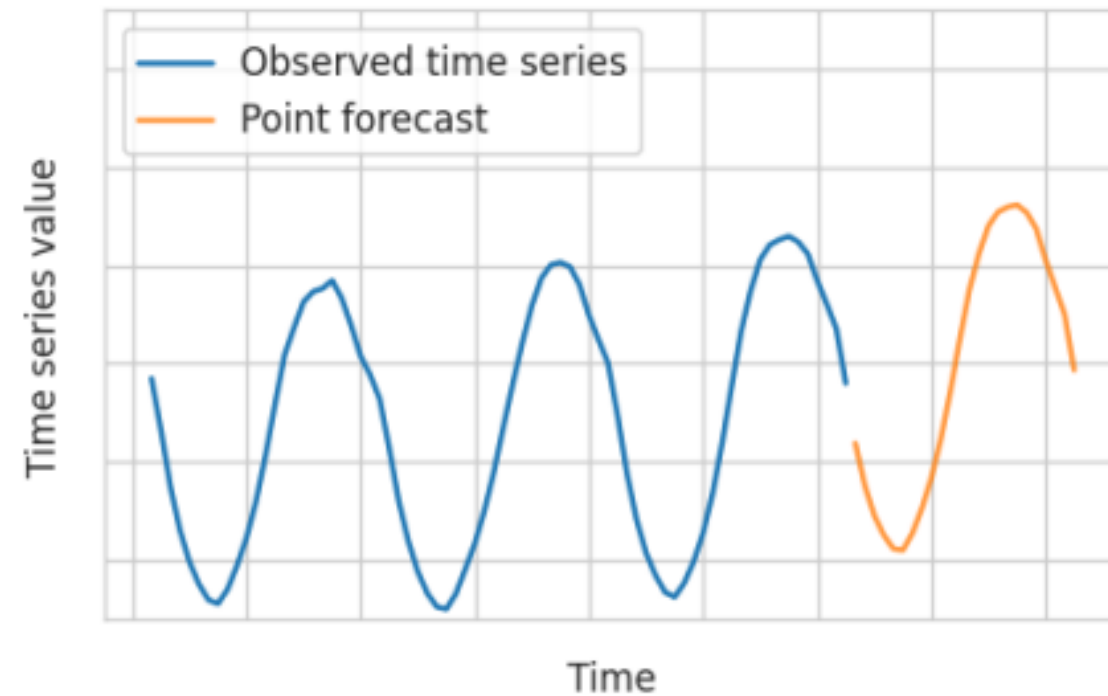
- H: 예측 길이 (prediction length) 또는 예측 구간 (forecast horizon)

Covariates

- Static Covariates: 상점 위치, 제품 ID
- Time-Varying Covariates:
 - 미래에 알려진 것: 요일, 프로모션
 - 과거에만 알려진 것: 날씨, 다른 제품의 판매량

02 Review

Point Forecast vs. Probabilistic Forecast

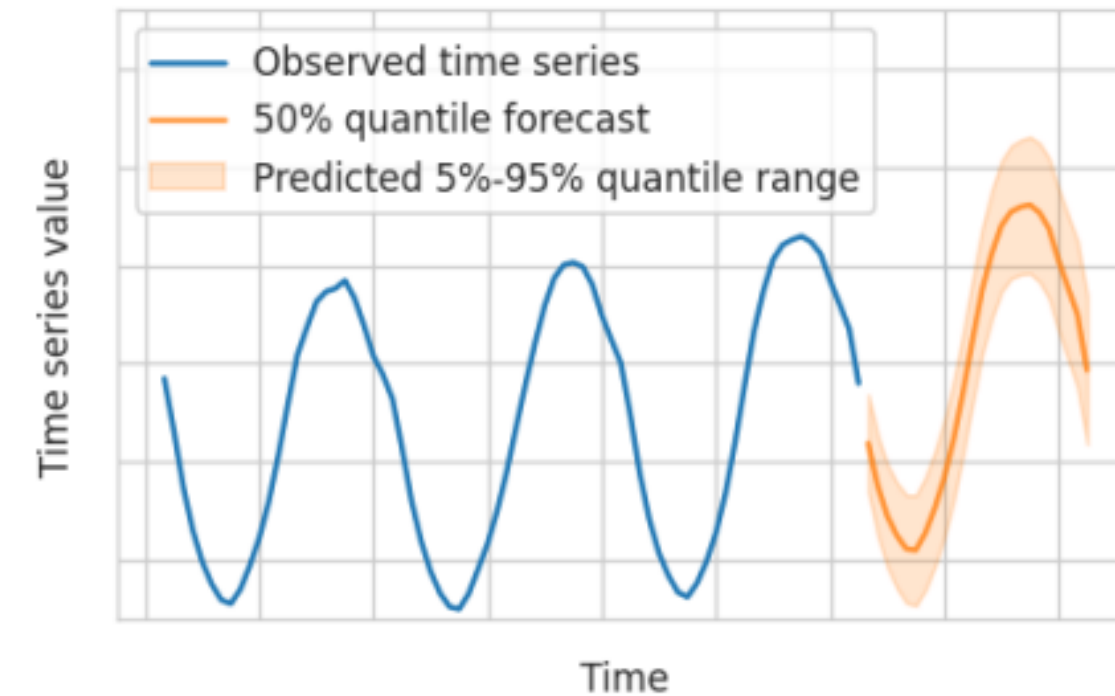


Point Forecast

- Conditional mean

$$\hat{y}_{i,T+1:T+H} = E_p[y_{i,1:T}, X_{i,1:T+H}]$$

- 미래 값에 대한 단일 예측을 제공



Probabilistic Forecast

- 조건부 분포 모델링

$$p(y_{i,T+1:T+H} \mid y_{i,1:T}, X_{i,1:T+H})$$

- Quantile forecast: $\hat{y}_{i,T+1:T+H}^q$

- 불확실성을 고려하여 가능한 미래 값의 범위를 제공

03 Method

AutoGluon-TimeSeries(AG-TS) 프레임워크

사용 예시코드

```
1 from autogluon.timeseries import TimeSeriesDataFrame, TimeSeriesPredictor
2
3 train_data = TimeSeriesDataFrame.from_path("train.csv")
4 predictor = TimeSeriesPredictor(prediction_length=30).fit(train_data)
5 predictions = predictor.predict(train_data) # forecast next 30 time steps
```

특징

- 사용 편의성과 견고성에 초점
- 전통적인 통계 모델과 머신러닝 기반 접근 방식을 모두 사용
- Forward Selection 알고리즘을 사용한 앙상블

04 Experiment

논문에서 실행된 실험 및 결과

Point
Forecast

Framework	Wins	Losses	Ties	Failures	Champion	Average rank	Average rescaled error	Win rate vs. baseline
AutoGluon (MASE)	-	-	-	0	19	2.08	0.073	100.0%
StatEnsemble	6	20	0	3	3	3.12	0.238	82.8 %
AutoPyTorch (MASE)	4	25	0	0	2	4.12	0.257	93.1%
AutoETS	4	25	0	0	1	4.64	0.374	75.9 %
AutoTheta	4	23	0	2	0	4.92	0.427	72.4 %
DeepAR	4	24	0	1	2	5.08	0.434	93.1 %
AutoARIMA	4	22	0	3	1	5.92	0.612	79.3 %
TFT	2	27	0	0	1	6.12	0.635	75.9 %

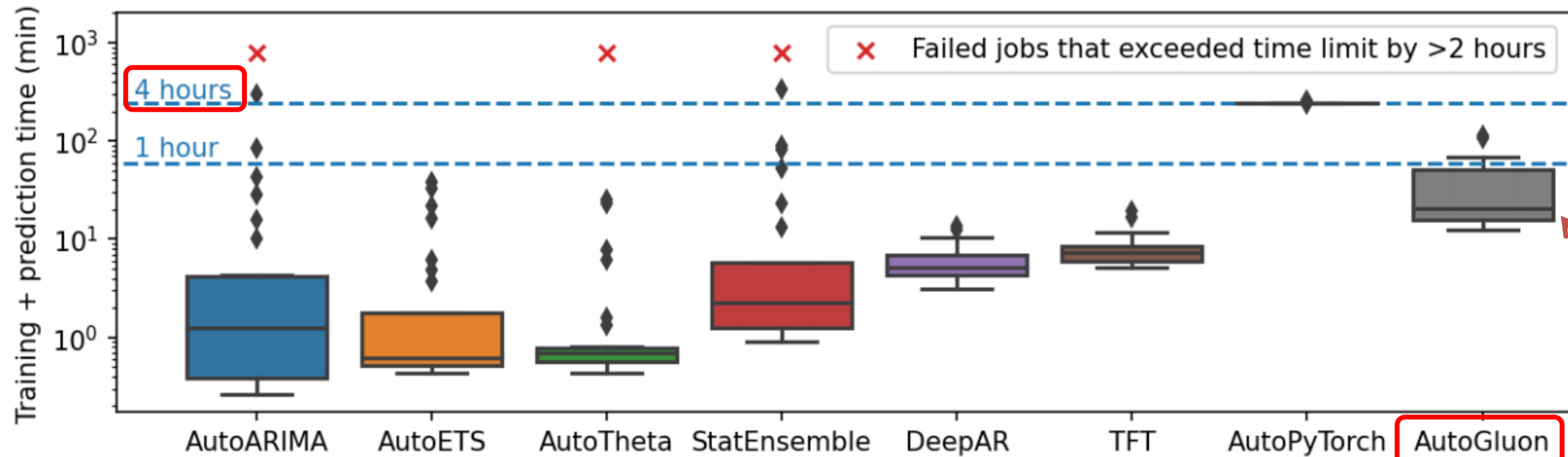
Quantile
Forecast

Framework	Wins	Losses	Ties	Failures	Champion	Average rank	Average rescaled error	Win rate vs. baseline
AutoGluon (wQL)	-	-	-	0	19	1.80	0.086	100.0%
StatEnsemble	3	23	0	3	0	3.36	0.330	86.2%
DeepAR	5	23	0	1	1	4.08	0.455	89.7%
TFT	5	24	0	0	5	4.24	0.487	89.7%
AutoETS	3	26	0	0	2	4.40	0.489	69.0%
AutoTheta	2	25	0	2	1	5.00	0.545	69.0%
AutoARIMA	4	22	0	3	1	5.12	0.641	82.8%

04 Experiment

논문에서 실행된 실험 및 결과

Total runtime



결과

- 앙상블을 사용하지 않을 경우 성능이 크게 저하
- AG-TS가 사용하는 세 가지 유형의 모델이 모두 전체 성능에 기여하고 있음을 확인

05 Discussion

AG-TS Framework 의 한계 및 제안사항

한계

- 대규모 데이터셋에서의 확장성 문제
- AG-TS를 적용할 때 사용자의 개입을 필요

제안사항

- Conformal Prediction 기법을 도입
- Cold-Start problem에서의 예측
- Large Language Model (LLM)과의 연계

06 Data Analysis



100개 건물들의 전력 사용량 데이터

2022-06-01 ~ 2022-08-24 (204,000행)

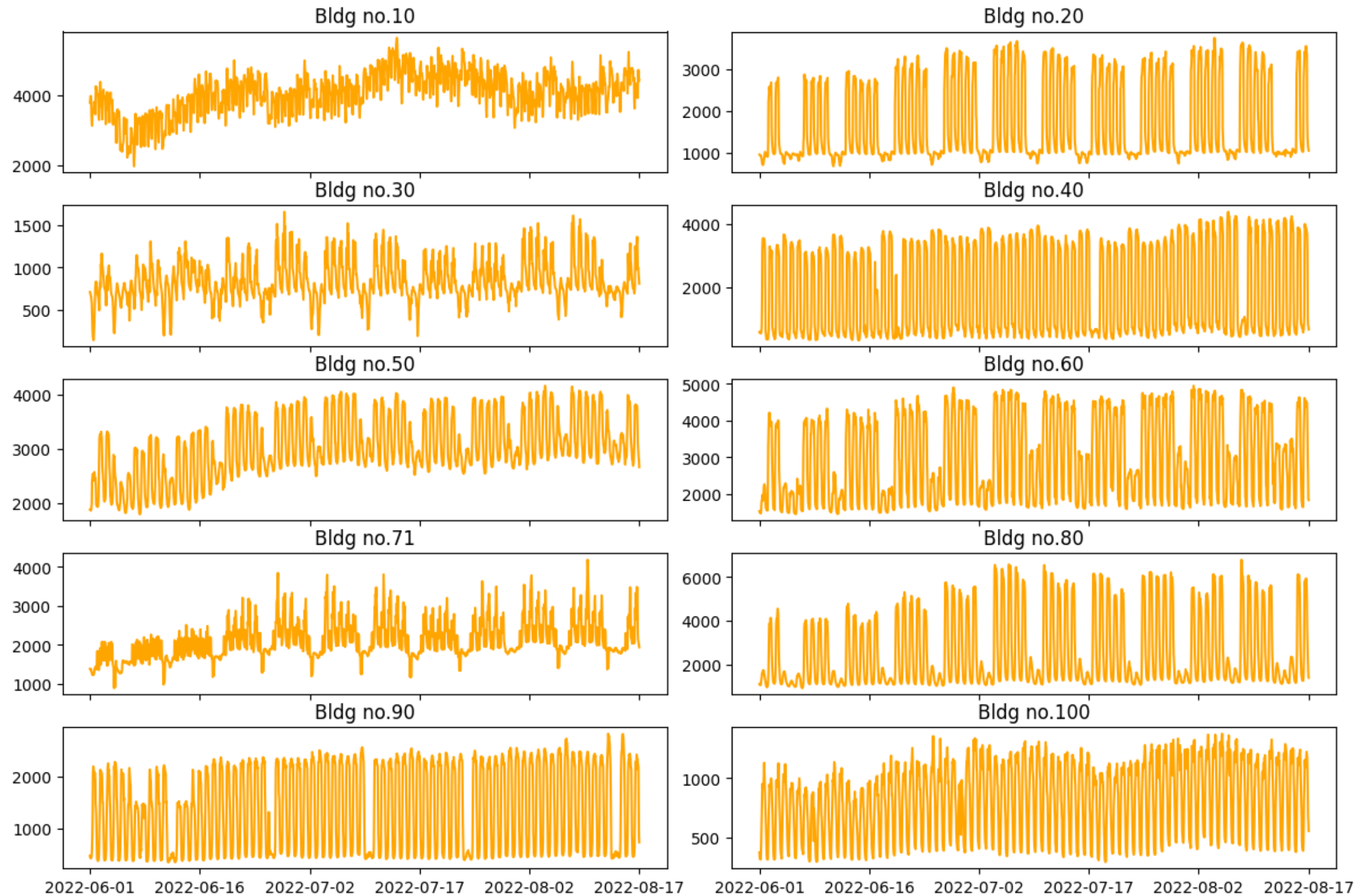
[데이터 구성]

- 건물번호 > item_id
- 일시 > timestamp
- 전력사용량(kWh) > target
- 기상자료, 건물정보 등

Train - 2022-06-01 00:00:00 ~ 2022-08-17 23:00:00

> 다음 일주일 2022-08-18 ~ 08-24의 전력 사용량을 예측

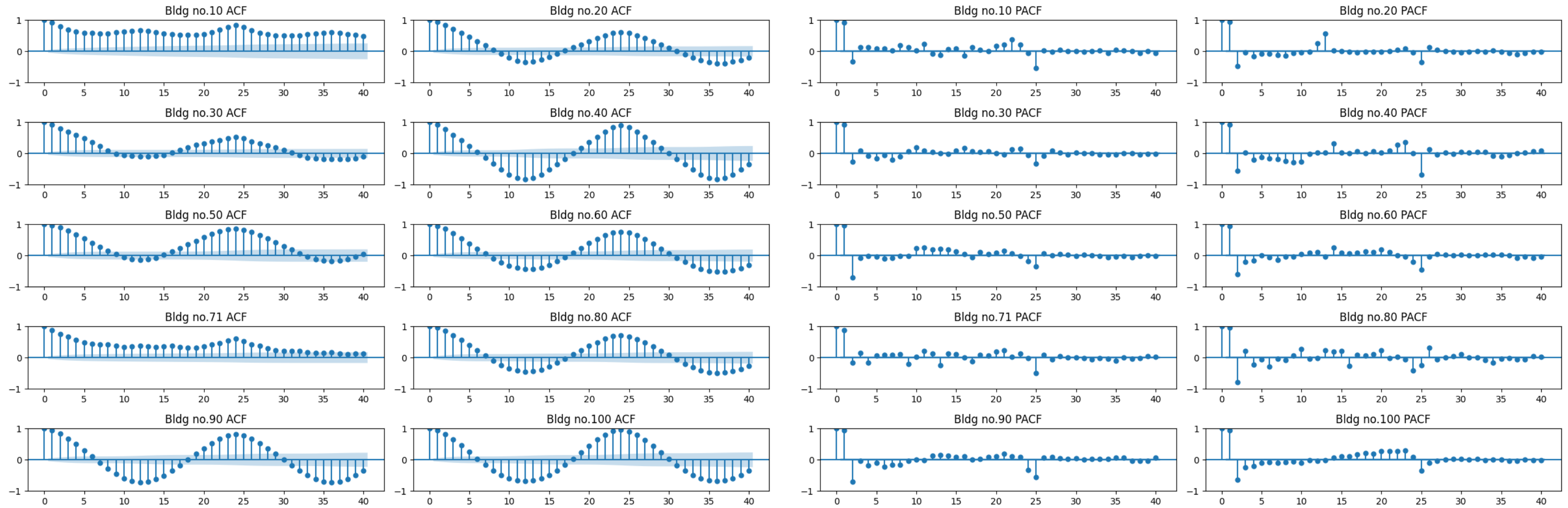
06 Data Analysis - AutoARIMA와 성능 비교



실험 1:

- 1) 10개 데이터를 동일하게 추출
- 2) 각각 AutoARIMA와 AutoGluon-TimeSeries 모델에 적합
- 3) 성능 비교 (MSPE)

06 Data Analysis - AutoARIMA와 성능 비교



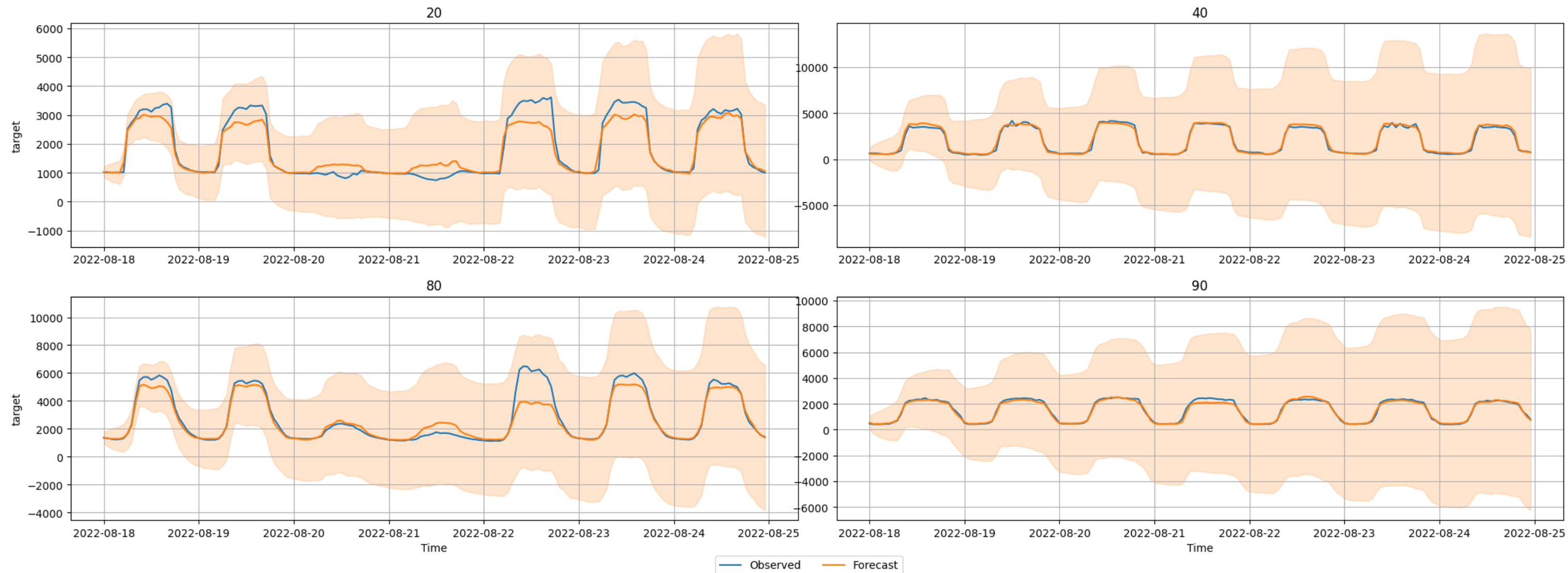
AG-TS

eval_metric = "MSE",
presets = "high-quality", time_limit = 3600

AutoARIMA

시간대별 데이터이므로 s=24, Seasonal=TRUE
AutoARIMA 로 각각 최적의 모델 탐색

06 Data Analysis - AutoARIMA와 성능 비교



item_id	mspe
10	238690
20	652184
30	59913.54
40	68820.8
50	52226.15
60	245507.1
71	220131.7
80	1431280
90	13394.32
100	5808.682
Total	298795.6

[random10 MSPE –
auto.arima]

[random10 forecast – AG-TS]

Models trained: ['SeasonalNaive', 'RecursiveTabular', 'DirectTabular', 'CrostonSBA', 'NPTS', 'DynamicOptimizedTheta', 'AutoETS', 'AutoARIMA', 'WeightedEnsemble']

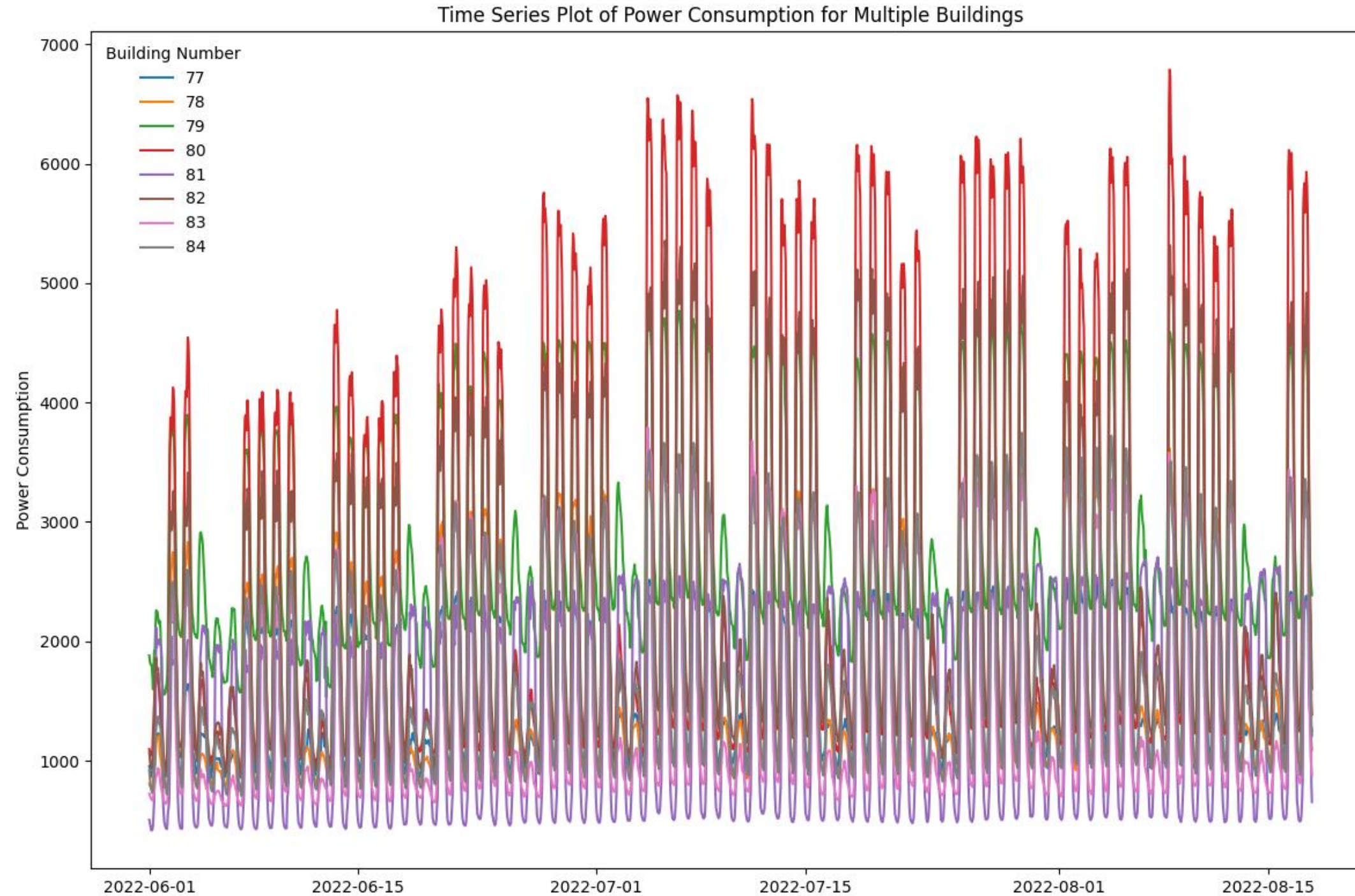
Total runtime: 437.56 s

Best model: WeightedEnsemble {'DirectTabular': 0.39, 'NPTS': 0.31, 'RecursiveTabular': 0.31}

AG-TS의 MSPE: 86753.62
AutoARIMA의 MSPE: 298795.6

AG-TS의 우수한 성능 확인

06 Data Analysis - Evaluating forecast accuracy



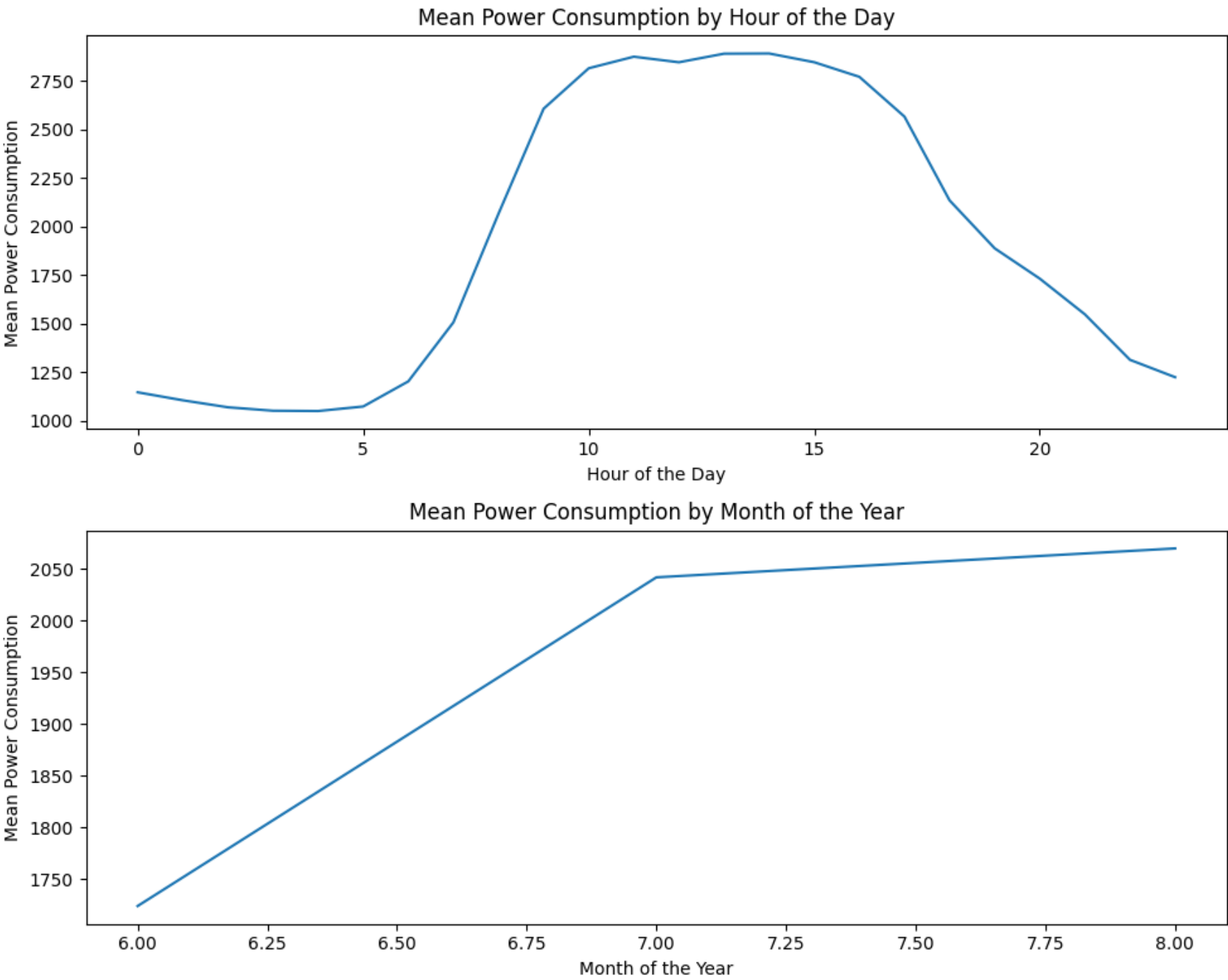
실험 2:

1) 지식산업센터 건물 8개에 대한
전력 사용량 예측

2) 기본 AG-TS 모델 (item_id,
timestamp, target으로 구성)과
추가적인 공변량을 사용한 모델 ($+ \alpha$)
각각에 적합

3) 성능 비교 (MSPE)

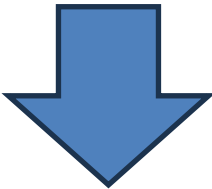
06 Data Analysis - Evaluating forecast accuracy



근무시간에 전력사용이 집중됨
6월보다 7, 8월에 전력사용량 늘어남 (에어컨 사용량 증가)

가용 데이터

building_number, date_time, temperature, rainfall, windspeed, humidity, sunshine, solar_radiation, power_consumption, building_type, total_area, cooling_area, solar_power_capacity, ess_capacity, pcs_capacity



사용 데이터

Item_id	building_number
timestamp	date_time
target	power_consumption
known_covariates	holidays, weekend
past_covariates	building_type, temperature, rainfall, windspeed, humidity, sunshine, solar_radiation
static_features	total_area, cooling_area

06 Data Analysis - Evaluating forecast accuracy

Models trained: ['SeasonalNaive',
'RecursiveTabular', 'DirectTabular', 'CrostonSBA',
'NPTS', 'DynamicOptimizedTheta', 'AutoETS',
'AutoARIMA', 'WeightedEnsemble']

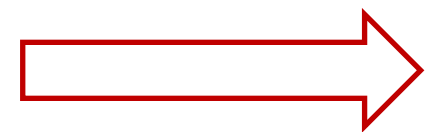
Total runtime: 408.18 s

Best model: WeightedEnsemble

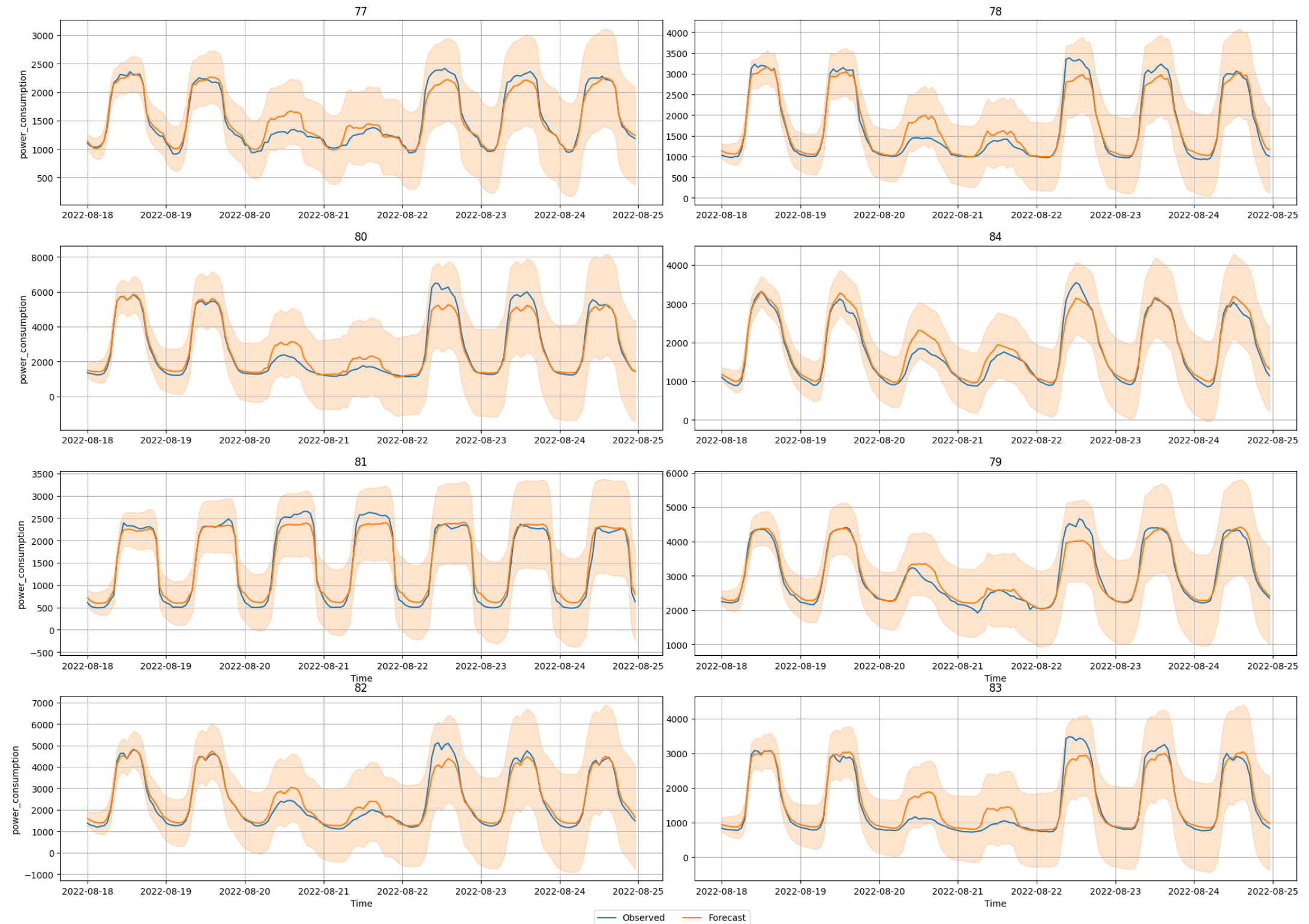
Ensemble weights: {'CrostonSBA': 0.09,
'DynamicOptimizedTheta': 0.03,
'RecursiveTabular': 0.88}
-167553.9510 = Validation score (-MSE)

기본 모델 MSPE: 123324.37

53% 개선



MSPE: 56824.39



Thank you