2022 POSTECH OIBC CHALLENGE

태양광 발전량 예측 경진대회

조성하 (수학과), 이은미 (환경공학부), 이동진 (인공지능대학원), POSTECH
Sun Capturer

Content

- 1. 대회 소개 및 문제 정의
- 2. 데이터 분석 및 전처리
- 3. 태양광 발전량 예측 모델
- 4. 클로징
- 5. 부록

대회 소개

- 타 지역 태양광 발전소들의 발전량 데이터를 활용하여 신규 설치된 광명시 태양광 발전소의 다음 날 24시간 동안의 매시간 발전량 구간 예측

광명시 발전소를 포함한
21개 태양광 발전소

2년치 기상 예보 및 실측 데이터 /
모듈 일사량 및 온도 데이터

사전 대회 포함 한 달 동안
광명시 발전소의 발전량을 매시간 예측

사진 출처: 광명시

태양광 발전량 예측의 중요성

정확한 태양광 발전량 예측의 어려움
일사량, 운량, 온도 등 기후 정보 변화로 인한
태양광 발전량 무작위적 변화 [2]

2015년 10월 기준, 태양광 발전은
전체 신재생 에너지의 80% 이상 차지 [1]

잘못된 예측으로 인해 기저부하 발전소들
의 재가동 비용 및 시간 소비 [3]

[1] 신동하, 박준호, 김창복., 일기 예보와 예측 일사 및 일조를 이용한 태양광 발전 예측, 한국항행학회논문지, 2017
[2] 이동훈, 김관호., 기후 및 계절정보를 이용한 딥러닝 기반의 장기간 태양광 발전량 예측 기법, 한국전자거래학회지, 2019
[3] 송재주, 정윤수, 이상호., 태양광 발전을 위한 발전량 예측 모델 분석, Journal of Digital Convergence, 2014

문제 해결을 위한 세 가지 요구 사항

Q1

타 발전소 데이터로부터 광명시 발전소의 발전량 패턴을 학습할 수 있을까?

Q2

어떤 기상 정보가 광명시 발전소의 발전량 예측에 유의미할까?

Q3

어떻게 기상 예보의 불확실성을 반영할 수 있을까?

제공 데이터

- 발전소별 발전량 데이터
- 기상 예보 데이터 1~3
- 기상 관측 데이터 1~3
- 모듈 경사일사량계, 모듈 표면 온도계

사진 출처: 광명시

사용한 데이터와 근거

- forecasts1.csv 및 gens.csv 사용
  - forecasts3: 짧은 예보 시간으로 인한 활용 제한 (예보 시점 이후 4~16시간)
  - forecasts2: 제공하는 피쳐가 forecasts1의 피쳐의 부분 집합
  - weathers 및 envs: 실측 데이터로서 다음날 데이터를 미리 알 수 없음 → 예보 불확실성 모델링시 활용 가능 (부록 참조)

사진 출처: 광명시

데이터 전처리 – forecasts1 (1)

- 예보 시점 반올림
- 16시에 예보된 다음날 데이터 선택
- 연도, 월, 일, 시간 정보 추가
- Min-Max 스케일링

예보 시점	예보 시간	온도	습도	…	강수량
5일 16:18	5일 17:00	10.55	58.0	…	0.0
5일 16:18	5일 18:00	9.44	67.0	…	0.0
⋮	⋮	⋮	⋮	⋱	⋮
5일 16:18	8일 16:00	14.44	45.0	…	0.0

예보 시점	예보 시간	온도	습도	…	강수량
5일 16:00	6일 01:00	6.11	80.0	…	0.0
5일 16:00	6일 02:00	5.56	84.0	…	0.0
⋮	⋮			⋱	⋮
5일 16:00	7일 00:00	6.67	86.0	…	0.0

예보 시점	예보 시간	온도	습도	…	강수량
5일 16:18	5일 17:00	0.495	0.540	…	0.0
5일 16:18	5일 18:00	0.485	0.602	…	0.0
⋮	⋮	⋮	⋮	⋱	⋮
5일 16:18	8일 16:00	0.514	0.510	…	0.0

(그림) forecasts1 데이터 전처리 예시

데이터 전처리 – forecasts1 (2)

- 기상 정보 선택
  - 민감도 분석을 통해 광명시 발전소의 발전량 예측에 유의미한 4개 기상 정보 확인 *민감도 분석 부록 참조
  - 하지만, 모든 발전소에 대해 일반화된 모델을 만들기 위해 모든 기상 정보 사용

(그림) 민감도 분석에서 높은 점수를 받은 4개 기상 정보

데이터 전처리 – gens (1)

- 발전소별 발전량 스케일 차이 존재
- 21개 발전소 데이터를 모두 사용하기 위하여 최대 용량 대비 발전량 비율을 예측
  - 발전량을 각 발전소의 최대 용량으로 나눠줌

(그림) 발전량 (왼쪽) 및 발전 비율 (오른쪽). 빨간 점선은 광명시 발전소, 회색 실선은 나머지 발전소를 나타낸다.

데이터 전처리 – gens (2)

- 광명시 발전소와 유사한 발전 패턴을 보이는 발전소 확인
  - 하지만 21개 발전소 모두 사용했을 때 더 좋은 성능을 보임
  - 21개 발전소 사전 훈련 → 7개 발전소 미세 조정 → 광명시 발전소 미세 조정 전략 시도했으나 성능 감소

(그림) 특정 기간 발전소 id 0, 1, 6, 7, 8, 9, 20를 갖는 발전소 (오른쪽)와 그 외 발전소 (왼쪽)의 발전 비율

데이터 전처리 – gens (3)

- 기타 이상 데이터 제거
  - 9번 발전소 2022년 8월 중순 ~ 9월 초 발전 중지

(그림) 특정 기간 9번 발전소의 발전 비율

WaveNet

- 구글 딥마인드의 1차원 CNN 기반 오디오 생성 모델
  - 오디오 생성을 위한 확률 모델
  - 입·출력 시퀀스의 길이 동일
  - 필터를 학습하는 CNN 특성상 이후 길이가 더 긴 시퀀스에 대해 예측 가능

(그림) WaveNet

사진 출처: van den Oord *et al.*, WaveNet: A Generative Model for Raw Audio, 2016

WaveNet 기반 태양광 발전량 확률 모델

- 발전량의 하한 및 상한 예측을 위한 확률 모델 고안 *모델 상세 부록 참조
- 24시간 예보 데이터 입력, 24시간 태양광 발전량의 평균과 표준편차 출력
  - 입력 $\boldsymbol{X} \in \mathbb{R}^{24\times17}$
  - 출력 $\hat{\mathbf{y}} = [\boldsymbol{\mu}_\theta\ \boldsymbol{\sigma}_\theta] \in \mathbb{R}^{24\times2}$

시간	피쳐1	피쳐2	⋯	피쳐17
01	0.495	0.938	⋯	0.043
02	0.485	0.948	⋯	0.086
⋮	⋮		⋱	⋮
24	0.514	0.887	⋯	0.000

→ WaveNet →

시간	평균	표준편차
01	0	0
02	0	0
⋮	⋮	⋮
24	0	0

(그림) WaveNet 입력 및 출력 예시

사용자 정의 손실 함수 (1)

- Negative log likelihood 손실함수 *코드 부록 참조
  - 출력된 평균과 표준편차를 파라미터로 갖는 정규분포를 가정

$$J(\theta) := \mathbb{E}_{(\boldsymbol{X},\boldsymbol{y})\in\mathcal{D}}[-\log p(y\,;\mu_\theta(x),\sigma_\theta(x)^2\,)] \approx -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i\,;\mu_\theta(x_i),\sigma_\theta(x_i)^2\,)$$

  - 정규 분포의 MLE는 maximum entropy를 갖기 때문에 큰 표준편차를 갖음

사용자 정의 손실 함수 (2)

- 경진대회 평가지표를 손실 함수로 구현 *코드 부록 참조
  - $l_1(\mu_\theta, y) \coloneqq |\mu_\theta - y|$
  - $l_2(\sigma_\theta) \coloneqq 2k\sigma_\theta$
  - $l_3(\mu_\theta, \sigma_\theta, y) \coloneqq y * \mathrm{ReLU}\big(y - (\mu_\theta + k\sigma_\theta)\big) + y * \mathrm{ReLU}\big((\mu_\theta - k\sigma_\theta) - y\big)$
  - $l(\mu_\theta, \sigma_\theta, y) \coloneqq \lambda_1 l_1(\mu_\theta, y) + \lambda_2 l_2(\sigma_\theta) + \lambda_3 l_3(\mu_\theta, \sigma_\theta, y)$

모델 훈련 상세

- 가중치 초기화: orthogonal initialization / zero bias
- 경사하강법: Adam(base_lr = 0.001, weight decay=0.01)
- 학습률 계획: Linear warmup (~10 에폭) 이후 cosine decay (~500 에폭) *그림 부록 참조
- 학습 로깅 툴: Weight and Bias (WanDB) *그림 부록 참조
- 에폭 / 미니 배치 사이즈: 500, 128

- 데이터 분할 전략
  - 훈련 데이터 발전소 및 기간: 21개 발전소, 2020년 10월 23일 ~ 2022년 10월 28일 → 약 15,000개
  - 검증 데이터 발전소 및 기간: 21개 발전소, 2022년 10월 29일 ~ 2022년 11월 12일 (2주) → 294개
  - 시험 데이터 발전소 및 기간: 광명시 발전소, 2022년 10월 29일 ~ 2022년 11월 12일 (2주) → 14개

대회기간 입찰을 위한 예측 (1)

- 전날 9시 및 16시 기상예보 데이터 수집 및 전처리
- 익일 24시간 발전량의 평균과 표준편차 예측
- 상한 및 하한 예측: $\boldsymbol{\mu}_\theta(\boldsymbol{X}) \pm k \cdot \boldsymbol{\sigma}_\theta(\boldsymbol{X})$
  - 사전 대회 기간 동안 $k = 0.5$ 사용, 실제 대회 기간 동안 $k = 0.01$ 사용

(그림) k값에 따른 평가지표 변화. 일반 스케일 (왼쪽) 및 로그 스케일 (오른쪽)

대회기간 입찰을 위한 예측 (2)

- 모델 앙상블
  - 회귀 문제에서 여러 모델의 예측 값의 평균을 취함으로써 편향 및 노이즈 완화
  - 휴리스틱에 기반하며 3개의 모델 선택
  - 각 모델이 예측한 평균과 표준편차의 평균값을 계산하여 최종 예측

모델	검증 평가지표 점수
모델1	0.320
모델2	0.346
모델 3	0.335
앙상블	0.320

(표) 선택한 3개 모델 및 앙상블 모델의 검증 평가지표

대회기간 입찰을 위한 예측 (3)

- 예측 결과 예시
  - 검증 데이터, $k = 0.01$, 기간 내 평균 평가 지표 0.215

(그림) 앙상블 모델의 검증 데이터 예측 (파랑) 및 실제 (빨강) 발전량. $k = 0.01$.

대회기간 입찰을 위한 예측 (4)

- 예측 결과 예시
  - 검증 데이터, $k = 0.5$, 기간 내 평균 평가 지표 0.308

(그림) 앙상블 모델의 검증 데이터 예측 (파랑) 및 실제 (빨강) 발전량. $k = 0.5$.

대회기간 입찰을 위한 예측 (5)

➢ 예측 결과 *17시 입찰 및 구간 오차 부록 참조

- 대회 기간 10시 입찰, $k = 0.01$

(그림) 대회 기간 동안의 예측 (빨강) 및 실제 (노랑) 발전량. $k = 0.01$.

사진 출처: : https://o.solarkim.com/cmpt2022

4. 클로징

요약

Q1

타발전소 데이터로부터 광명시 발전소의
발전량 패턴을 학습할 수 있을까?

21개 발전소를 모두 이용했을 때
성능 최대, 미세 조정 가능성

Q2

어떤 기상 정보가 광명시 발전소의
발전량 예측에 유의미할까?

광명시에 맞는 4개 기상 정보 확인,
하지만 일반적 모델을 위해 모두 사용 권장

Q3

어떻게 기상 예보의 불확실성을
반영할 수 있을까?

확률 모형으로 발전량 구간 예측

4. 클로징

느낀점 및 회고

- 시계열 데이터 예측에 가장 유의미한 정보인 "과거 발전량"을 사용하지 않았다.
  - 이틀 전 발전량, 일주일 전 발전량 정보를 사용했으면, 단기 예측의 오프셋으로 작용하여 예측 성능 향상 기대
  - 하지만, 과적합 가능성이 있다.
- 기상 예보 데이터로부터 기상 실측 데이터를 예측하는 모델을 학습하여, 기상 예보 대신 예측된 실측 데이터를 사용할 수 있었을 것이다.
- 미세 조정 디테일이 부족했다.
  - 21개 발전소로 사전학습 이후 광명시 발전소 데이터만 사용하여 가장 마지막 컨볼루션 층을 미세 조정하였는데, 먼 과거 데이터로 인해 미래 예측 성능은 더 떨어졌다.
  - 최근 데이터로만 미세 조정 수행시 성능 향상 기대
- 구간 예측 모형을 평가하는 좋은 지표는 무엇이 있을까?
  - ✓ 1:1:1 가중치 사용 또는 negative log likelihood 손실을 사용하면 구간 예측에 더 주안을 둘 수 있을 것이다.

부　록

부록

기상 예보 보정 모델 예시

- 기상 예보로부터 기상 실측을 예측하는 모델 $\mathcal{O}$
  - 입력 $\boldsymbol{f} \in$ forecasts1
  - 출력 $\boldsymbol{w} \in$ weathers1
  - 모델 $\mathcal{O}$ Fourier Neural Operator
- 이후 발전량 예측 모델에 기상 예보 대신 $\mathcal{O}(\boldsymbol{f})$ 입력하여 예보와 실측 사이의 편차 감소 효과 기대
  - 시간적 한계로 실제 구현 포기

(그림) 기상 예보 데이터로부터 실측 데이터를 예측하는 모형 예시

부록

민감도 분석 (1)

- **목적**: 민감도 분석을 통해 모델 예측도를 높이는 기상 정보 찾기
- **평가 모델**: Gated Recurrent Unit (GRU)
- **평가 지표**: Nash-Sutcliffe Efficiency (NSE)

$$\mathrm{NSE} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

부록

민감도 분석 (2)

➢ 1단계: 한가지 기상 정보만 사용하여 모델 훈련 및 평가

NSE는 상대적으로 낮지만 태양광 발전량에 영향이 있을 인자로 추정하여 사용

(그림) 민감도 분석에서 높은 점수를 받은 4개 기상 정보 (빨간색 박스) 및 추가 3개 기상 정보 (초록색 박스)

민감도 분석 (3)

➢ **2단계**: 상위 4개 기상 정보 고정 후 나머지 기상 정보를 하나씩 추가하며 사용하여 모델 훈련 및 평가

고정 Input

Applied Input	A		B		C
temperature	O	O	O	O	O
humidity	O	O	O	O	O
uv_idx	O	O	O	O	O
forecast	O	O	O	O	O
wind_dir	X	O	X	X	X
wind_spd	X	X	O	X	O
precip_1h	X	X	X	O	O
NSE	0.876	0.602	0.892	0.824	0.870

(그림) 민감도 분석에서 높은 점수를 받은 4개 기상 정보를 고정시키고,
추가 기상 정보를 하나씩 추가하며 모델 훈련 및 평가

부록

민감도 분석 (4)

➢ **3단계**: 상위 기상 정보 조합의 잔차 분석 수행

A

temperature, humidity, uv_idx, forecast

NSE: 0.876

B

temperature, humidity, uv_idx, forecast, wind_spd

NSE: 0.892

(BEST Input Combination)

C

temperature, humidity, uv_idx, forecast, wind_spd

NSE: 0.87

(그림) 기상 정보 조합에 따른 예측 (y축) 및 실제 (x축) 발전량

부록

WaveNet 구조 상세

- 발전량 평균과 표준편차의 양수를 보장하기 위하여 출력층 활성화 함수 ReLU 사용
- 기상 예보를 사용한다는 점에서 causal 패딩을 사용하지 않음

층수	입력 필터수	출력 필터수	필터 크기	패딩	Dilation 비율	활성화 함수
1 (입력)	21	10	2	same	1	ReLU
2	10	10			2	
3	10	10			4	
4	10	10			8	
5	10	10			1	
6	10	10			2	
7	10	10			4	
8	10	10			8	
9 (출력)	10	2	1	same	1	ReLU

(표) WaveNet 구현 상세

부록

Negative log likelihood 코드

```python
class BaseLoss(nn.Module):
    def __init__(self):
        super(BaseLoss, self).__init__()

    def _unroll_batch(self, input, target):
        input = input.reshape((-1, 2))
        target = target.reshape(-1)

        return input, target

    def forward(self, input, target):
        raise NotImplementedError


class NLGLLoss(BaseLoss):
    """Negative Log Gaussian Likelihood Loss"""
    def __init__(self, **kwargs):
        super(NLGLLoss, self).__init__()

    def forward(self, input, target):
        input, target = self._unroll_batch(input, target)
        mu = input[:, 0]
```

코드 하이라이팅 툴: https://carbon.now.sh/

부록

경진대회 평가지표 손실 함수 코드

```python
class MeanStdLoss(BaseLoss):
    def __init__(self, k=1.0, weights=(1.0, 0.2, 0.7), **kwargs):
        super(MeanStdLoss, self).__init__()
        self.k = k
        self.weights = weights

    def loss1(self, input, target):
        return torch.abs(input[:, 0] - target)

    def loss2(self, input):
        return 2 * self.k * input[:, 1]

    def loss3(self, input, target):
        return (
            F.relu(target - (input[:, 0] + self.k * input[:, 1])) +
            F.relu((input[:, 0] - self.k * input[:, 1]) - target)
        )

    def forward(self, input, target):
        input, target = self._unroll_batch(input, target)
        loss1 = self.loss1(input, target)
        loss2 = self.loss2(input)
        loss3 = self.loss3(input, target)

        return (self.weights[0] * loss1 + self.weights[1] * loss2 + self.weights[2] * loss3).sum()
```

코드 하이라이팅 툴: https://carbon.now.sh/

부록

학습률 계획 (learning rate scheduling)

(그림) 학습률 계획: 10 에폭까지 Linear warmup 후 500 에폭까지 cosine decaying

부록

WanDB 활용 예시 (1)

➢ 다양한 실험의 손실함수 값 및 평가지표의 각 항목을 한 눈에 비교 가능

(그림) WanDB 사용 예시: 손실 그래프 비교

부록

WanDB 활용 예시 (2)

- 각 실험의 하이퍼파라미터들을 저장하여 실험간 비교 용이

dongjinlee > Projects > solar_power_prediction > Table

Runs (27)

Filter | Group | Sort | Tag | Move | Create Sweep | Columns

Name (3 visualized)	State	Notes	User	Tags	Created	Runtime	dir_ckpt	epochs	hidden_dim	k	last_data	loss.k	loss.loss	loss.weight	optimizer.lr	optimizer.w	run_id	scheduler.T	scheduler.T	scheduler.T	scheduler.e	scheduler.g	seed
YoungCle	finished	Add notes	dongjir		1w ago	8m 43s	../ckpt/1668576298	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668576298	500	1	10	0.001	0.5	0
SE	finished	Add notes	dongjir	Selecte	1w ago	9m 21s	../ckpt/1668432245	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668432245	500	1	10	0.001	0.5	0
AdamW	finished	Add notes	dongjir		1w ago	7m 25s	../ckpt/1668400861	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668400861	500	1	10	0.001	0.5	0
GoodLuckAdamW	finished	Add notes	dongjir	Selecte	1w ago	7m 12s	../ckpt/1668400099	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668400099	500	1	10	0.001	0.5	0
GoodLuckk3	finished	Add notes	dongjir		2w ago	9m 38s	../ckpt/1668272078	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668272078	500	1	10	0.001	0.5	3
GoodLuckk2(X)	finished	Add notes	dongjir		2w ago	3m 27s	../ckpt/1668271850	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668271850	500	1	10	0.001	0.5	2
GoodLuckk1	finished	Add notes	dongjir		2w ago	9m 18s	../ckpt/1668271239	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668271239	500	1	10	0.001	0.5	1
GoodLuckk0	finished	Add notes	dongjir	Selecte	2w ago	9m 41s	../ckpt/1668270620	500	10	0.5	2022-11-1	1	MeanStdL	[1,0,0]	0	0.01	1668270620	500	1	10	0.001	0.5	0
GoodLuck4	finished	Add notes	dongjir		2w ago	8m 55s	../ckpt/1668269809	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668269809	500	1	10	0.001	0.5	4
GoodLuck3	finished	Add notes	dongjir		2w ago	9m 24s	../ckpt/1668269187	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668269187	500	1	10	0.001	0.5	3
GoodLuck2	finished	Add notes	dongjir		2w ago	9m 27s	../ckpt/1668268569	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668268569	500	1	10	0.001	0.5	2
GoodLuck1	finished	Add notes	dongjir		2w ago	8m 57s	../ckpt/1668267975	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668267975	500	1	10	0.001	0.5	1
GoodLuck0	finished	Add notes	dongjir		2w ago	9m 3s	../ckpt/1668267319	500	10	0.5	2022-11-1	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668267319	500	1	10	0.001	0.5	0
GoodLuck	finished	Add notes	dongjir	Selecte	2w ago	9m 5s	../ckpt/1668265483	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668265483	500	1	10	0.001	0.5	0
Baseline(-21)	finished	Add notes	dongjir		2w ago	9m 33s	../ckpt/1668259143	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668259143	500	1	10	0.001	0.5	4
Baseline(-21)	finished	Add notes	dongjir		2w ago	10m 1s	../ckpt/1668258344	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668258344	500	1	10	0.001	0.5	3
Baseline(-21)	finished	Add notes	dongjir		2w ago	9m 24s	../ckpt/1668257678	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668257678	500	1	10	0.001	0.5	2
Baseline(-21)	finished	Add notes	dongjir		2w ago	9m 6s	../ckpt/1668255283	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1668255283	500	1	10	0.001	0.5	1
CausalWaveNet	finished	Add notes	dongjir		2w ago	9m 1s	../ckpt/1667825844	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1667825844	500	1	10	0.001	0.5	0
Baseline(-21,FS2)	finished	Add notes	dongjir		2w ago	9m 58s	../ckpt/1667823071	500	10	0.5	2022-11-0	1	MeanStdL	[1,0.2,0.8]	0	0.01	1667823071	500	1	10	0.001	0.5	0

(그림) WanDB 사용 예시: 하이퍼파리미터 비교

부록

대회기간 예측 결과 (10시)

(그림) 사전 대회 및 대회 기간 10시 입찰 예측 (빨강) 및 실제 (노랑) 발전량

사진 출처: : https://o.solarkim.com/cmpt2022

부록

대회기간 예측 결과 (17시)

(그림) 사전 대회 및 대회 기간 17시 입찰 예측 (빨강) 및 실제 (노랑) 발전량

사진 출처: : https://o.solarkim.com/cmpt2022

부록

대회기간 예측 결과 (구간예측오차)

(그림) 사전 대회 및 대회 기간 구간 예측 오차

사진 출처: : https://o.solarkim.com/cmpt2022

부록

대회기간 평가지표 순위

			팀이름	구간예측오차	소속
1st	1	↑	별 x1	6.61	경북대
2nd	2	↑	IS_LAB	7.02	상명대
3rd	-	–	Sun Capturer	7.93	POSTECH
4	2	↑	naye0319	8.52	성균관대
5	-	–	그레이시리얼	8.83	인하대
6	5	↓	3축진동센서 x2	8.88	서울대
7	-	–	Duck Curve	9.06	한국공대
8	3	↑	돌고래 x1	9.39	울산대
9	3	↑	헬창1	9.78	단국대
10	1	↓	선사진관	10.21	건국대

(그림) 대회 기간 평가지표 순위

사진 출처: : https://o.solarkim.com/cmpt2022