LSTM with Attention을 이용한 태양광 발전량 예측 - With Solar insolation Q calculation

폭풍대기 : 오승옥[1], 이선호[2], 신지은[1]

2023/11/28

[1]연세대학교 대기과학과

[2]서강대학교 컴퓨터공학과

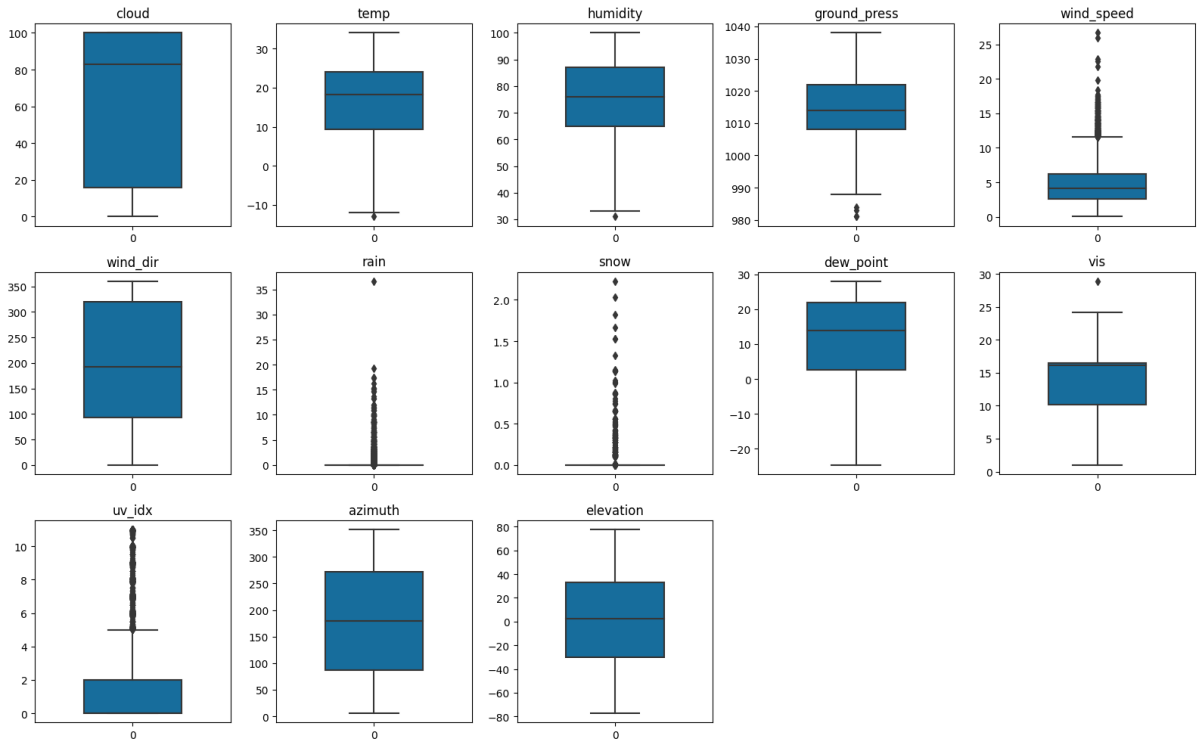
TABLE OF CONTENTS

1 Data preprocessing

● 데이터 시각화

- 이용 데이터 : 기상 관측 데이터, 태양광 예측 발전량 데이터, 태양광 발전량 데이터

1. 기상 관측 데이터



2. 태양광 예측 발전량 데이터

	round	time	model_id	amount
0	1	2022-06-19 01:00:00+09:00	0	0.0
1	1	2022-06-19 01:00:00+09:00	1	0.0
2	1	2022-06-19 01:00:00+09:00	2	0.0
3	1	2022-06-19 01:00:00+09:00	3	0.0
4	1	2022-06-19 01:00:00+09:00	4	0.0

3. 태양광 발전량 데이터

	time	amount
0	2022-06-19 01:00:00+09:00	0.0
1	2022-06-19 02:00:00+09:00	0.0
2	2022-06-19 03:00:00+09:00	0.0
3	2022-06-19 04:00:00+09:00	0.0
4	2022-06-19 05:00:00+09:00	0.0

1 Data preprocessing

● 데이터 결측 값 처리

1. 기상 관측 데이터

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11616 entries, 0 to 11615
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   time          11616 non-null  object
 1   cloud         11616 non-null  float64
 2   temp          11616 non-null  float64
 3   humidity      11616 non-null  float64
 4   ground_press  11616 non-null  float64
 5   wind_speed    11616 non-null  float64
 6   wind_dir      11616 non-null  float64
 7   rain          11616 non-null  float64
 8   snow          11616 non-null  float64
 9   dew_point     11616 non-null  float64
 10  vis           11616 non-null  float64
 11  uv_idx        11616 non-null  float64
 12  azimuth       11616 non-null  float64
 13  elevation     11616 non-null  float64
dtypes: float64(13), object(1)
memory usage: 1.2+ MB
```

11616개로, 결측 값이 없다.

2. 태양광 예측 발전량 데이터

model_id	0	1	2	3	4
round					
1	11616	11616	11616	11616	11616
2	11592	11592	11592	11592	11592

round 2에서 24개의 데이터가 부족하다.

- **2023-08-17 의 데이터 누락**
- **round 1의 값으로 대체**

3. 태양광 발전량 데이터

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11616 entries, 0 to 11615
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   time    11616 non-null  object
 1   amount  11616 non-null  float64
dtypes: float64(1), object(1)
memory usage: 181.6+ KB
```

11616개로, 결측 값이 없다.

```
df_elc_check['daily_sum'] = df_elc_check.amount.resample('D').sum()
zero_sum_indices = df_elc_check[df_elc_check['daily_sum'] == 0].index
print(zero_sum_indices)
```

```
DatetimeIndex([], dtype='datetime64[ns]', name='time', freq=None)
```

일간 총 태양광 발전량이 0인 날도 존재하지 않는다.

2 Feature engineering

● Solar insolation Q feature extraction

- time, azimuth, elevation을 한 개의 특성 Q_h 로 대체하였다.
- **Hourly extraterrestrial radiation on horizontal surface Q :**

$$Q_h(\infty) = S\left(\frac{r_0}{r}\right)^2\left(\sin\emptyset\sin\delta + \frac{24}{\pi}\cos\emptyset\cos\delta\sin\frac{\pi}{24}\cos h_i\right)$$

S : solar constant, r_0 : sun-earth mean distance, r : sun-earth distance,
$\emptyset$: latitude, δ : solar declination angle, h_i : hour angle

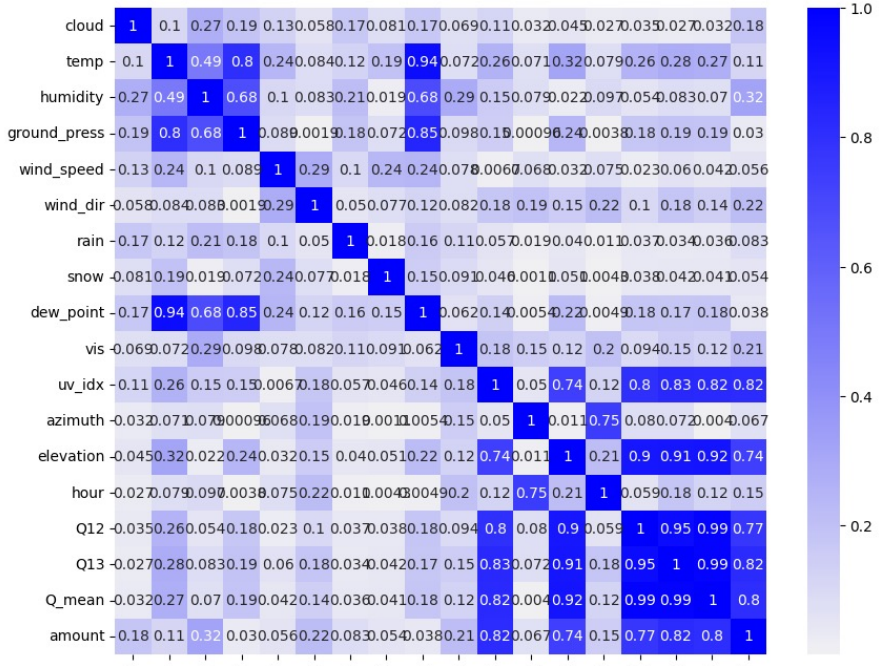
- r_0는 지구 장축 반지름으로 근사하여 계산하였다.
- $h_i = \frac{local\ noon - hour}{hour\ gap}$ 에서 local noon은 매일 다른 값을 가지지만, 12시와 13시, 그 둘의 평균값으로 대입해본 결과, 태양광 발전량과 가장 상관관계가 높은 12시로 근사하여 사용하였다.

2 Feature engineering

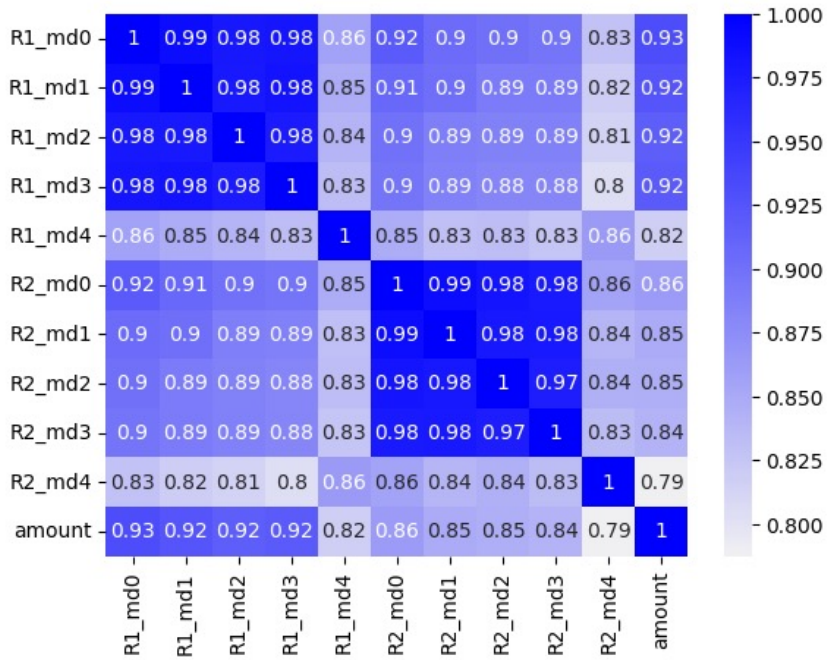
● 변수 선택

- 태양광 발전량과 상관관계가 낮은 변수들은 변수선택에서 제외하였다.
- 기상 관측 데이터 변수 선택 : cloud, humidity, wind_dir, vis, uv_idx, elevation, hour, Q13, (temperature)
- 태양광 예측 발전량 데이터 변수 선택 : model 0 - 3

1. 기상 관측 데이터



2. 태양광 예측 발전량 데이터



3 Model

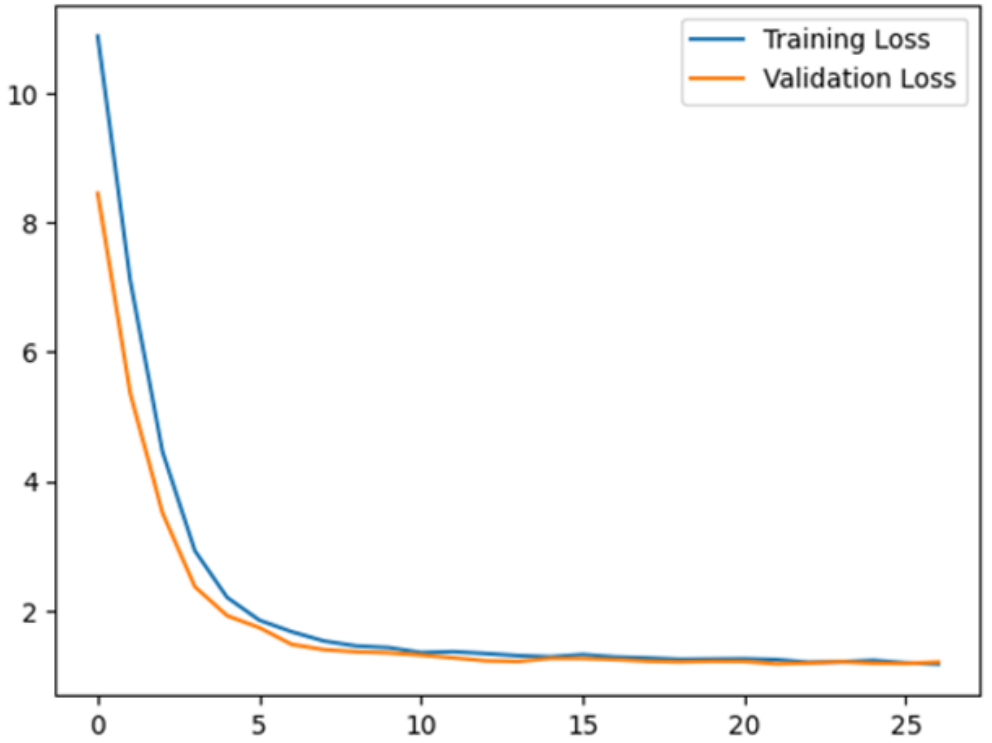
● 모델 설명

- 한정된 데이터, 단순한 앙상블 기법만으로 태양광 발전량 예측에서 좋은 결과를 얻기 어렵다고 판단하였다.
  - ➢ Random Forest 기반 앙상블 방법 적용 시 단순 평균을 취한 결과보다 나쁜 성능을 보였다.
- 따라서, 단순히 기존 모델의 결과를 ensembl하기보다는 **전처리한 데이터를 가지고 DNN (Deep Neural Network)를 학습하는 방법**을 택하였다.
- 태양광 발전량 데이터는 sequential data이므로, 이 특성을 활용할 수 있는 **Sequential model** 위주로 모델을 탐색하였다.
  - ➢ 여러 논문 자료에서 태양광 발전량 예측에 좋은 성능을 보인 LSTM을 이용하였다.
- 기존 모델의 예측 값만이 아니라 **기상 관측 데이터도 모두 feature로 사용하여 pretrained model에 fine-tuning 시키는 transfer learning**을 사용하였다.

3 Model

● 모델 설명

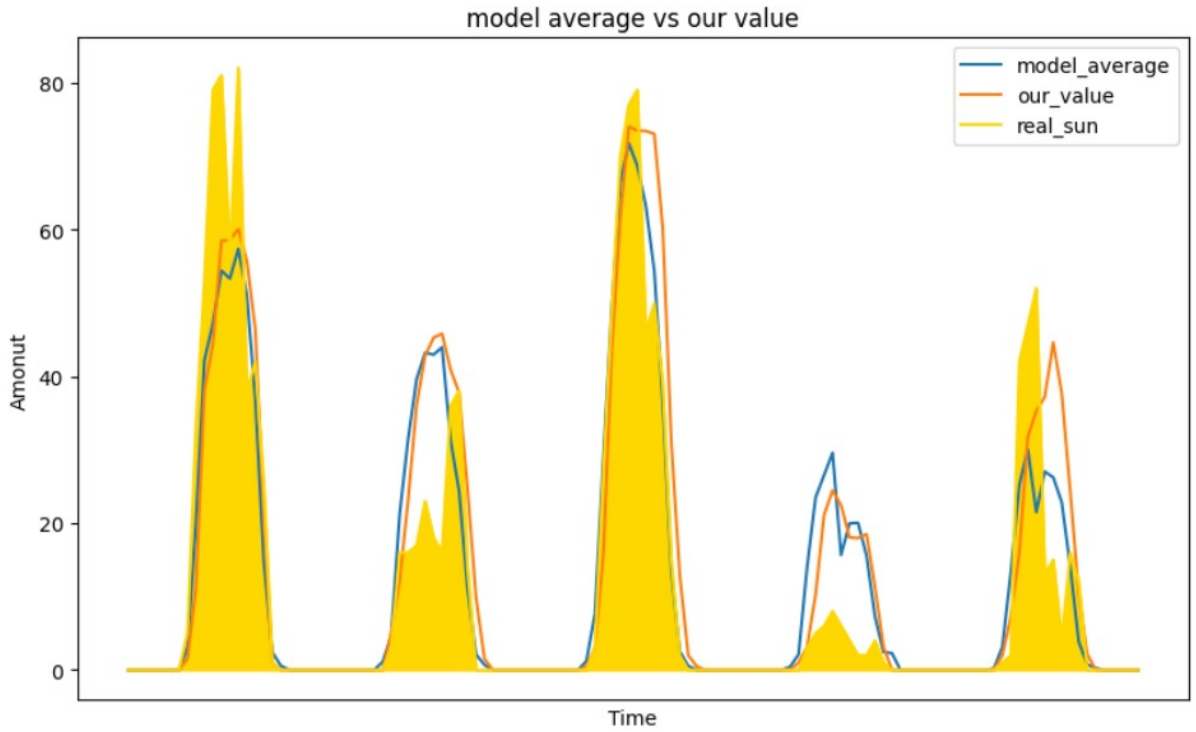
- **LSTM의 output state**를 해당 시간의 정보를 갖는 embedding feature로 고려하여, 이를 BERT의 input으로 넣었다.
  - ➢ 단, 시간에 관한 positional encoding은 고려하지 않았다.
- Pretrained bi-directional encoder 기반의 BERT를 사용했지만, 현재 시간의 output을 예측하는 데 있어서 미래의 input을 사용하지 못하도록 구현하였다.
- 즉, **transformer의 decoder와 유사한 masked multi-head self attention mechanism**을 사용하였다.
- 이는 attention score를 구할 때 현재 시간에서 미래의 정보를 반영하지 못하도록 하기 위한 용도이다.



4 Discussion

● 기존 모델과 성능 비교

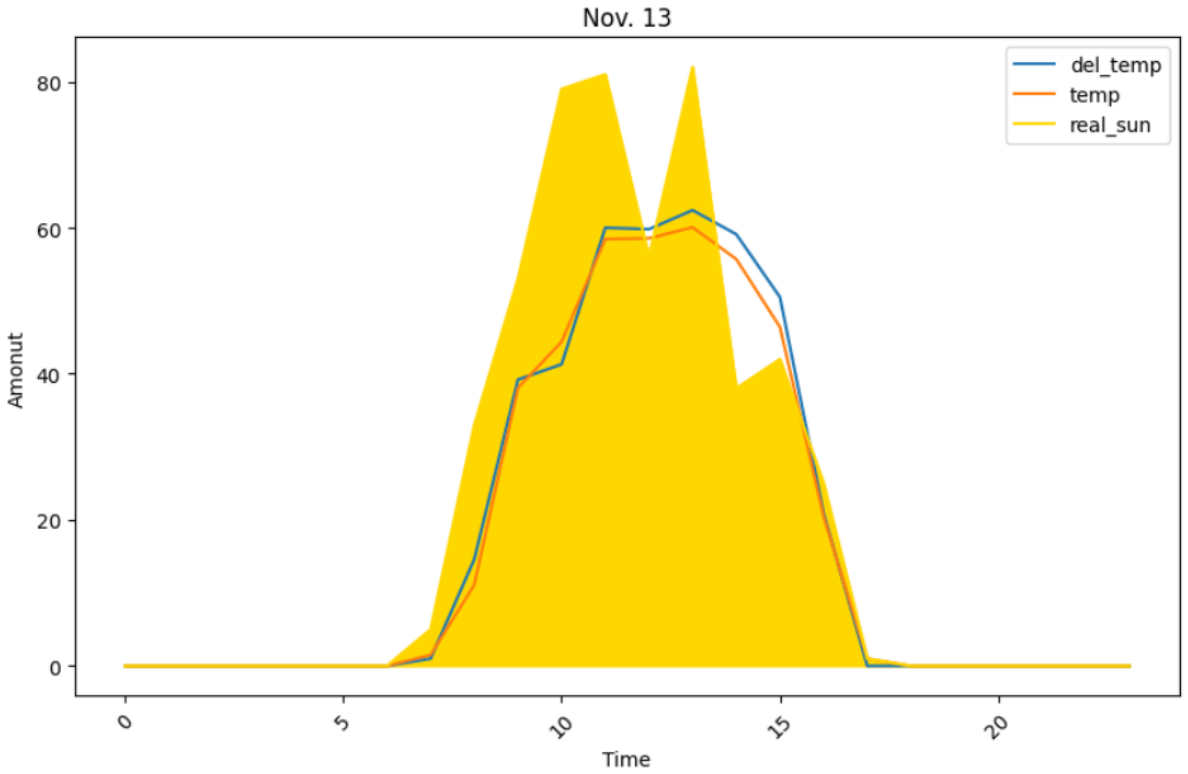
- 기존 모델을 단순 평균하여 예측한 것과 대회 기간 동안의 데이터를 가지고 성능을 비교하였다.
- 기존 모델에 비해 오차율이 낮으며, 특히 오후 3시 이후의 예측에서 특히 오차율이 낮아 높은 인센티브를 얻을 수 있었다.



4 Discussion

● 대회 기간 중 개선한 사항

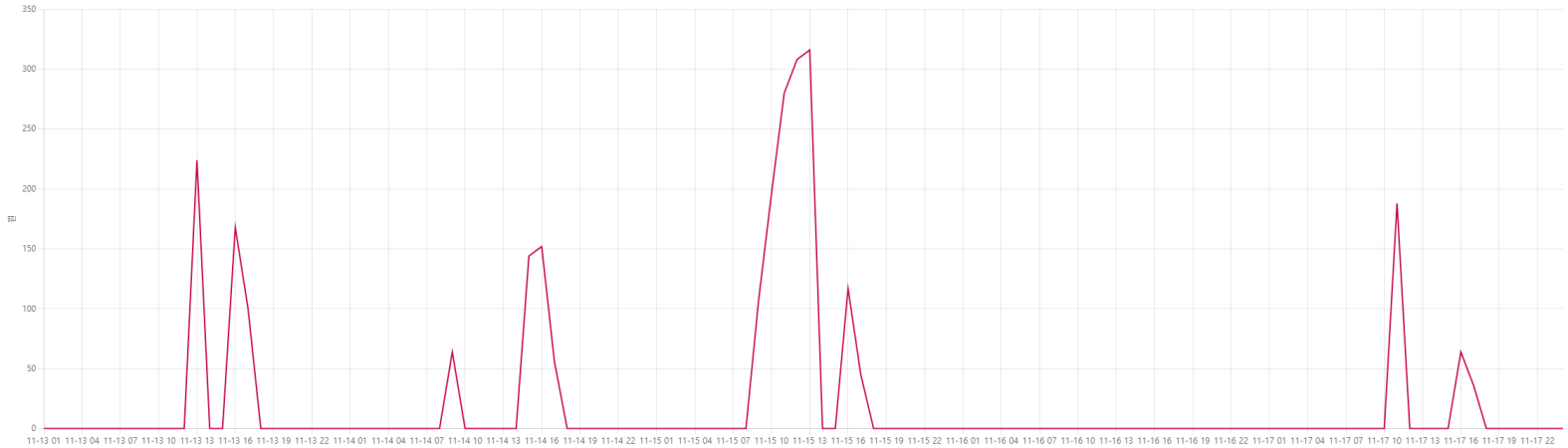
- 모델은 6월부터 10월의 데이터로 모델을 훈련하였지만, 대회기간은 입동을 지난 11월 13일부터 17일이었다.
- 훈련 데이터와 대회 기간의 데이터 간의 온도 차이로 인해, temperature를 변수에 포함한 13일에는 태양광 발전량을 과소평가하였다.
- 14일 부터는 temperature를 변수 선택에서 제거한 모델을 사용하였다.



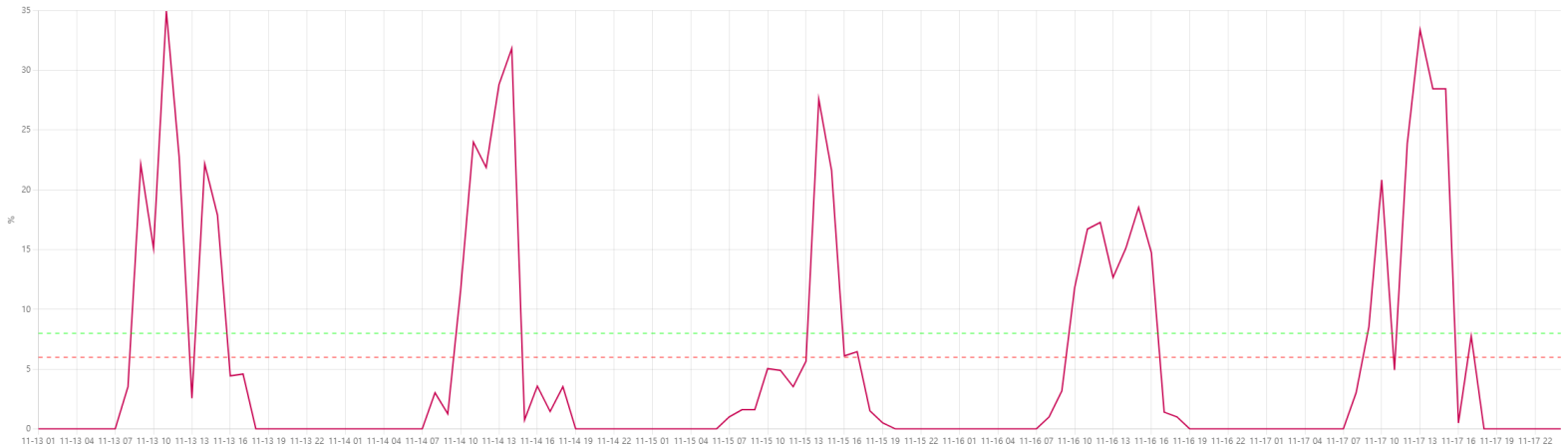
5 Conclusion

● 모델 평가

1. 인센티브



2. 입찰오차



5 Conclusion

● 개선 요소

- LSTM 을 이용한 선행 연구에서는 분 단위의 데이터로 모델링하였는데, 대회에서 제공된 날씨 측정 데이터의 시간 간격은 1시간이어서, window size를 상대적으로 짧게 설정할 수 밖에 없었다.

- 이로 인해 LSTM이 sequential data 예측에서 가지는 장점과 attention mechanism이 long-term dependency 유지에서 가지는 장점을 효과적으로 이용했다고 보기 어렵다.

- 모든 모델의 예측 값을 feature로 넣었으나, 모델 별 예측 평균 편차가 크다. 따라서, 평균 오차율이 작은 모델의 가중치를 더 높게 주는 방식으로 feature engineering 하는 것도 고려해볼 수 있으리라고 생각한다.

Q&A

Thanks for listening