Portfolio

윤 동 진

CONTENTS

About Me

1. Experiences

1) 금융 투자 알고리즘 개발

- **Enhanced Index Tracking: 유전 알고리즘 기반 지수 상향 추종 알고리즘**
  - Project Managing — Sequence Diagram(PlantUML), WBS
  - Scheme Definition — SQL Server, MySQL
  - EDA / Algorithm Modeling — Python(Dask, Numba, CUDA, scikit-learn)
  - Multi-node, multi-gpu distributed parallel computing — Dask, Numba, CUDA
- **WeightFormer: BERT 기반 포트폴리오 예측 알고리즘**
  - EDA / Algorithm Modeling — Python(PyTorch, Transformers, BERT)

2) 경진대회 참가

- **Wind Power Forecasting: 풍력 발전량 예측 알고리즘**
  - EDA / Algorithm Modeling — Python(PyTorch, Transformers, GRU)
- **Court Decision Prediction: 법원 판결 예측 알고리즘**
  - EDA / Algorithm Modeling — Python(PyTorch, spaCy, Transformers, T5, vicuna, LoRA)

3) 개발 환경 구축 및 배포

- **base-cuda: Prepared CUDA based Docker Image for Machine Learning Project**
  - Dockerhub — https://hub.docker.com/repository/docker/alchemine/base-cuda
- **analysis-tools: PyPI Analysis tools package for Machine Learning Project**
  - PyPI — https://pypi.org/project/analysis-tools

CONTENTS

Wind Power Forecasting

Wind Power Forecasting

1. 프로젝트 소개

풍력 발전 터빈의 위치, 온도와 풍속 등의 시공간 데이터를 학습하여 미래의 유효전력을 예측하는 프로젝트

2. 특징

- 시계열의 특성을 유지하며 **결측치와 에러값(전체 데이터의 23%)**을 처리
- **데이터를 가공하기 위한 다양한 기법들을 사용**
- 대학원생팀 1등으로 상금 200만원을 수상하였으며, **imputing 알고리즘**이 특히 좋은 평가를 받음

1) 문제 정의

Feature 설명

```
TurbID  - Wind turbine ID, 발전기 ID
Day     - Day of the record, 날짜
Tmstamp - Created time of the record, 시간 (10분 단위)
Wspd    - The wind speed recorded by the anemometer, 풍속(m/s)
Wdir    - wind direction, 터빈이 바라보는 각도와 실제 바람 방향 각도 차이(°)
Etmp    - Temperature of the surounding environment, 외부 온도(℃)
Itmp    - Temperature inside the turbine nacelle, 터빈 내부 온도(℃)
Ndir    - Nacelle direction, i.e., the yaw angle of the nacelle, 터빈이 바라보는 방향 각도(°)
Pab     - Pitch angle of blade, 터빈 당 3개의 날이 있으며 각각의 각도가 다름(°)
Prtv    - Reactive power, 무효전력 : 에너지원을 필요로 하지 않는 전력(kW)
Patv    - Active power(target variable), 유효전력 : 실제로 터빈을 돌리는 일을 하는 전력(kW)
```

Goal 134개의 터빈(TurbID)에 대한 유효전력(Patv) 예측

KPI 미래 2일 간 10분 단위로 sampling 된 유효전력에 대한 MSE와 MAE의 평균

1) 문제 정의 (continued)

Abnormal label 조건

1. Wspd > 2.5 and Patv ≤ 0

바람은 불지만 발전량이 없는 경우

2. |Pab| > 89

바람과 날개의 각도 차이가 커 바람의 영향을 받지 못하는 경우

3. |Wdir| > 180 or |Ndir| > 720

기기의 정상 범위를 넘어가는 경우

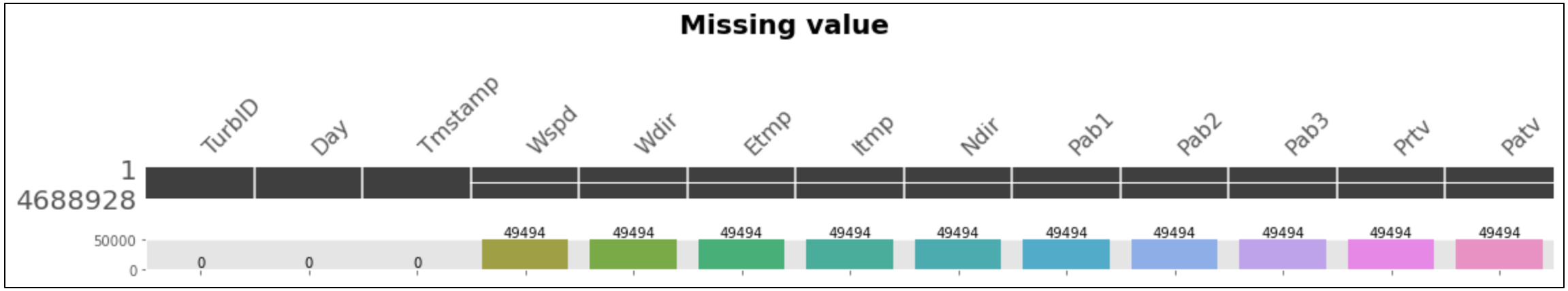
4. Patv is null

Target이 존재하지 않는 경우

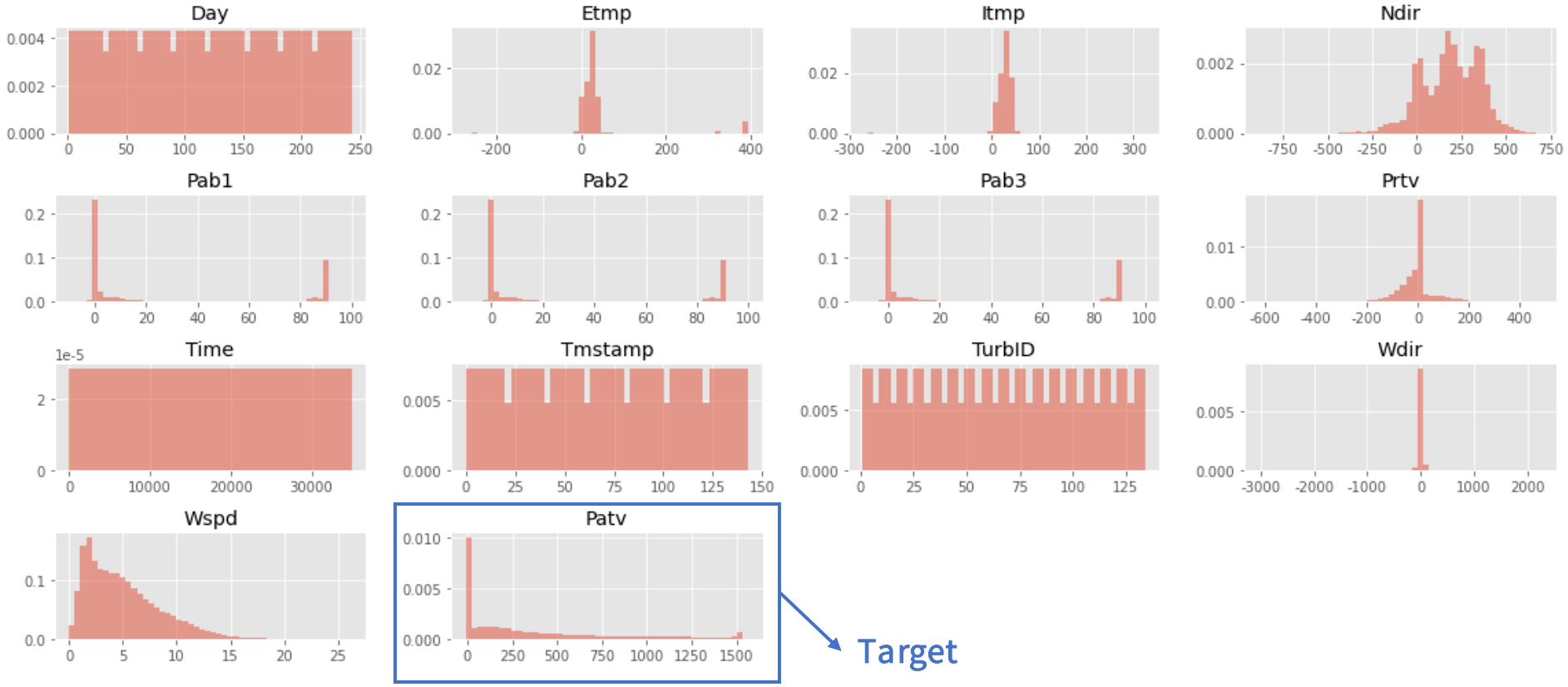
2) Explanatory Data Analysis (Training data)

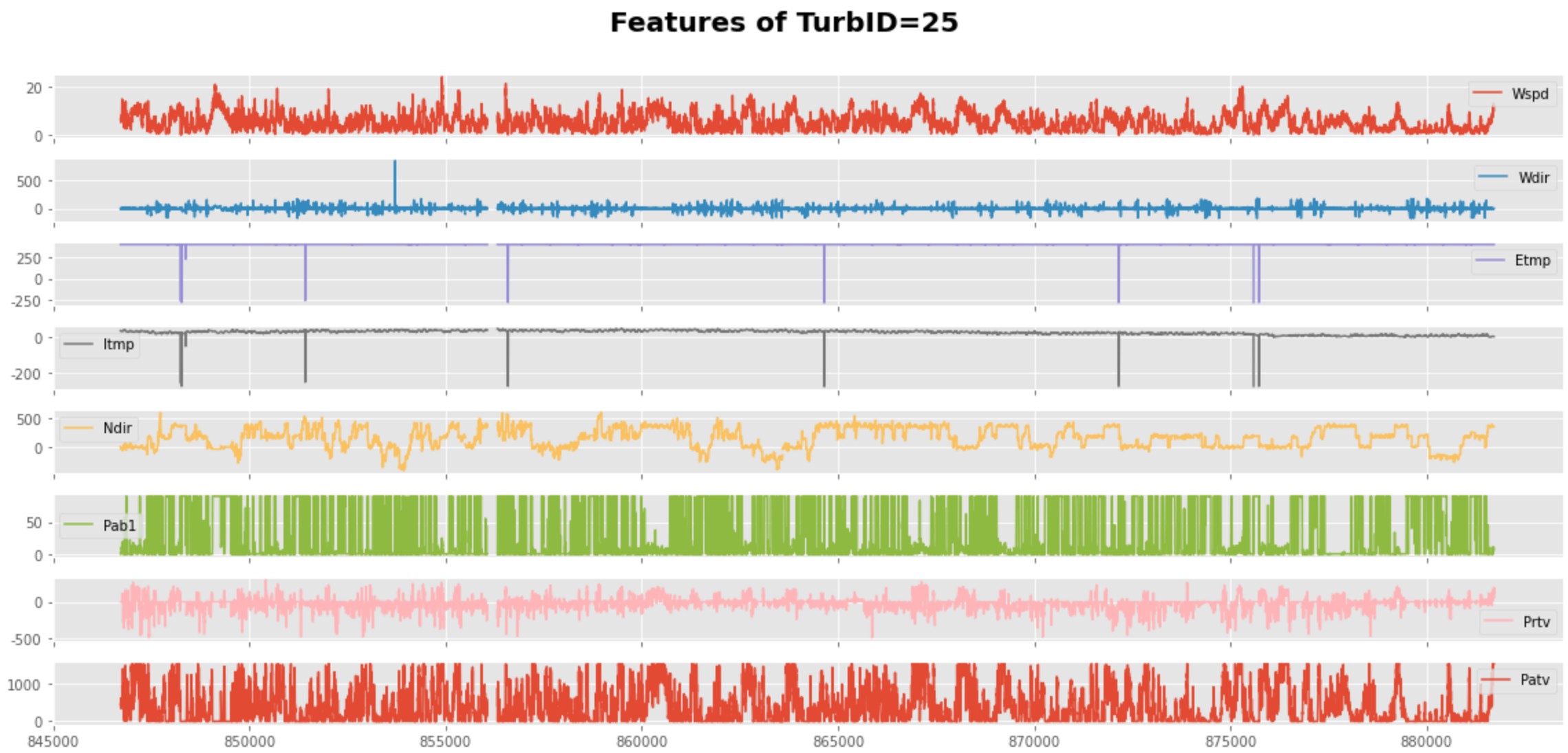
	TurbID	Day	Tmstamp	Wspd	Wdir	Etmp	Itmp	Ndir	Pab1	Pab2	Pab3	Prtv	Patv
0	1	1	00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1	1	00:10	6.17	-3.99	30.73	41.80	25.92	1.00	1.00	1.00	-0.25	494.66
2	1	1	00:20	6.27	-2.18	30.60	41.63	20.91	1.00	1.00	1.00	-0.24	509.76
3	1	1	00:30	6.42	-0.73	30.52	41.52	20.91	1.00	1.00	1.00	-0.26	542.53
4	1	1	00:40	6.25	0.89	30.49	41.38	20.91	1.00	1.00	1.00	-0.23	509.36
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4727227	134	243	23:10	10.98	-1.96	-5.11	-0.67	345.57	8.82	8.82	8.82	136.49	1152.60
4727228	134	243	23:20	11.82	-3.18	-5.46	-0.54	345.57	13.87	13.87	13.87	84.43	681.65
4727229	134	243	23:30	11.91	-1.42	-5.21	-0.42	345.57	10.69	10.69	10.69	145.72	1118.35
4727230	134	243	23:40	11.86	-0.95	-5.40	-0.38	345.57	13.94	13.94	13.94	89.56	683.49
4727231	134	243	23:50	11.72	0.04	-5.23	-0.37	345.57	10.90	10.90	10.90	120.18	1026.93

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4688928 entries, 0 to 4727231
Data columns (total 13 columns):
 #   Column   Dtype
---  ------   -----
 0   TurbID   int64
 1   Day      int64
 2   Tmstamp  object
 3   Wspd     float64
 4   Wdir     float64
 5   Etmp     float64
 6   Itmp     float64
 7   Ndir     float64
 8   Pab1     float64
 9   Pab2     float64
 10  Pab3     float64
 11  Prtv     float64
 12  Patv     float64
dtypes: float64(10), int64(2), object(1)
memory usage: 500.8+ MB
```



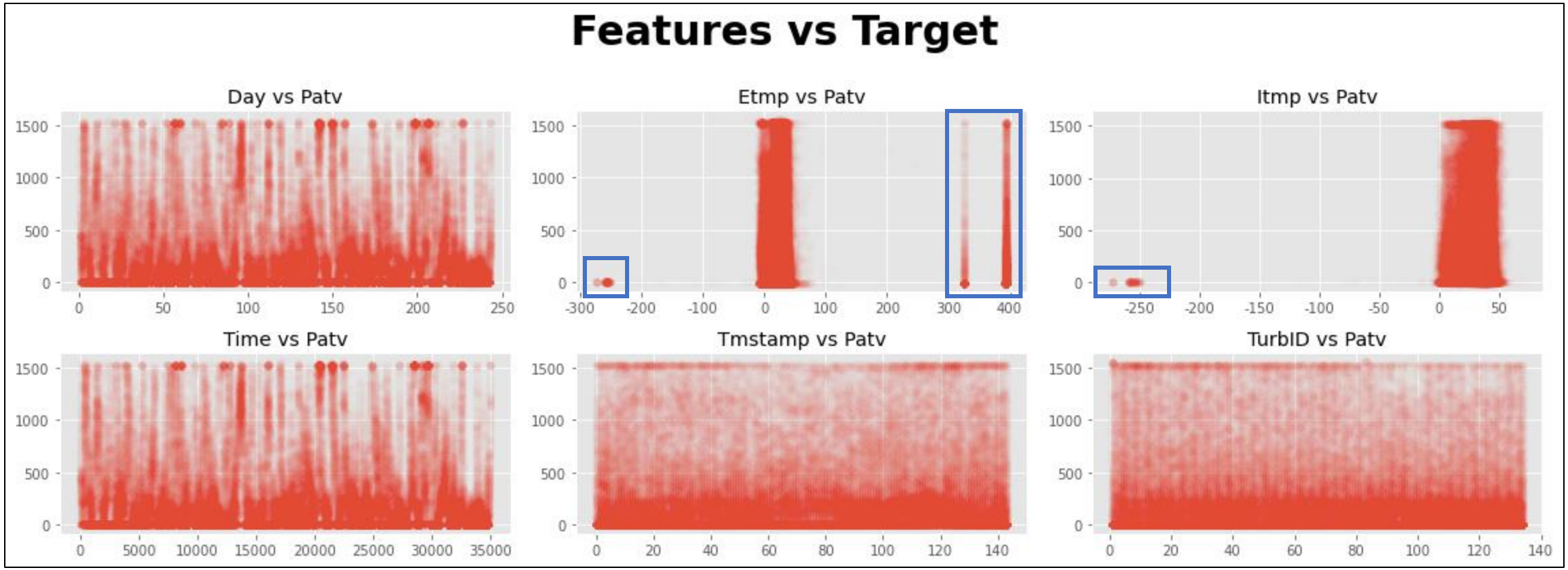
Features


Day
Etmp
Itmp
Ndir
Pab1
Pab2
Pab3
Prtv
Time
Tmstamp
TurbID
Wdir
Wspd
Patv
Target


Features of TurbID=25
Wspd
Wdir
Etmp
Itmp
Ndir
Pab1
Prtv
Patv
20
0
500
0
250
0
-250
0
-200
500
0
50
0
0
-500
1000
0
845000
850000
855000
860000
865000
870000
875000
880000

① Day, Etmp, Itmp, Time, Tmstamp, TurbID

- 뚜렷한 관계가 보이지 않음
- Etmp, Itmp: 이상치 처리가 필요

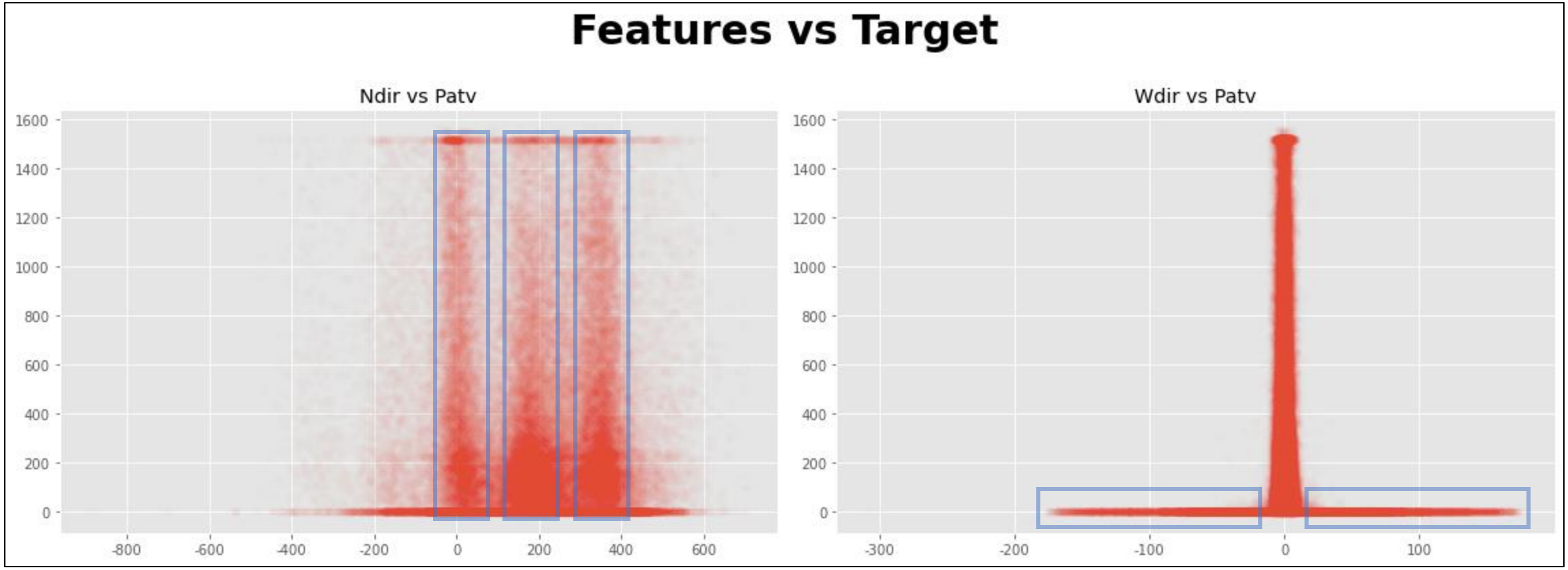


② Ndir

- 0도, 180도, 360도에서 기둥이 보임 (180도의 배수인 것처럼 보이나, -180도는 기둥이 없기 때문에 음수에 대한 구분이 필요)

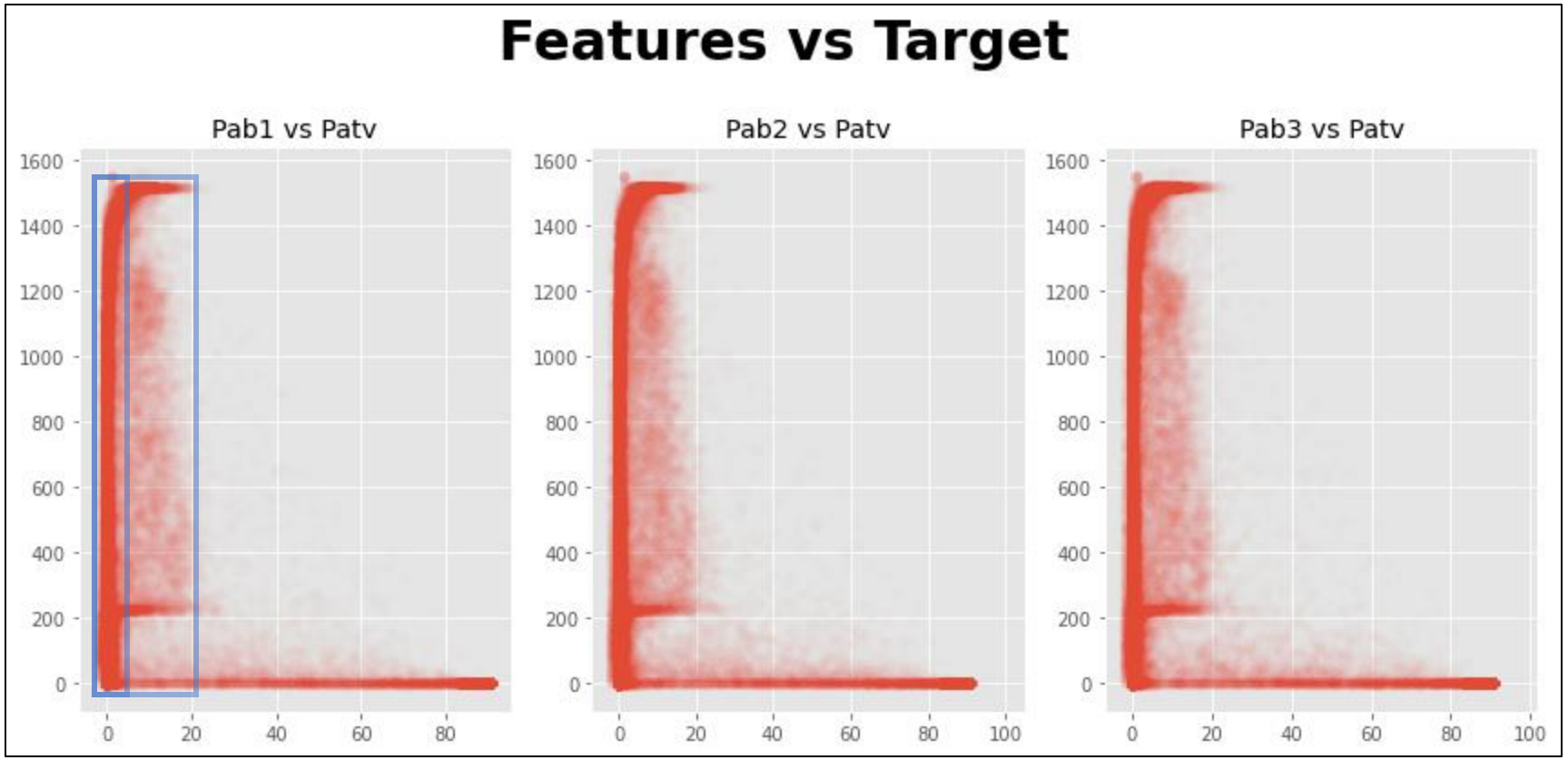
③ Wdir

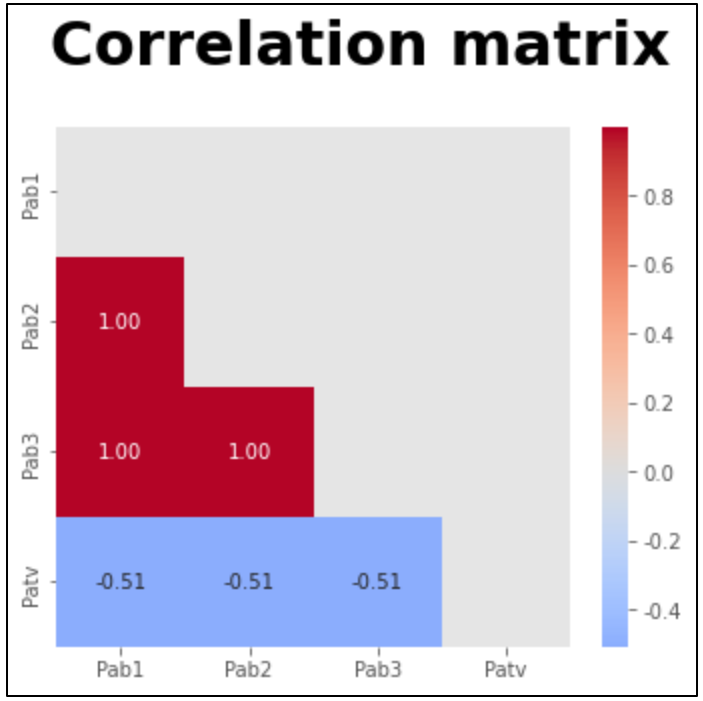
- 절댓값이 10도 이상이면, Patv ≈ 0



④ Pab1, Pab2, Pab3

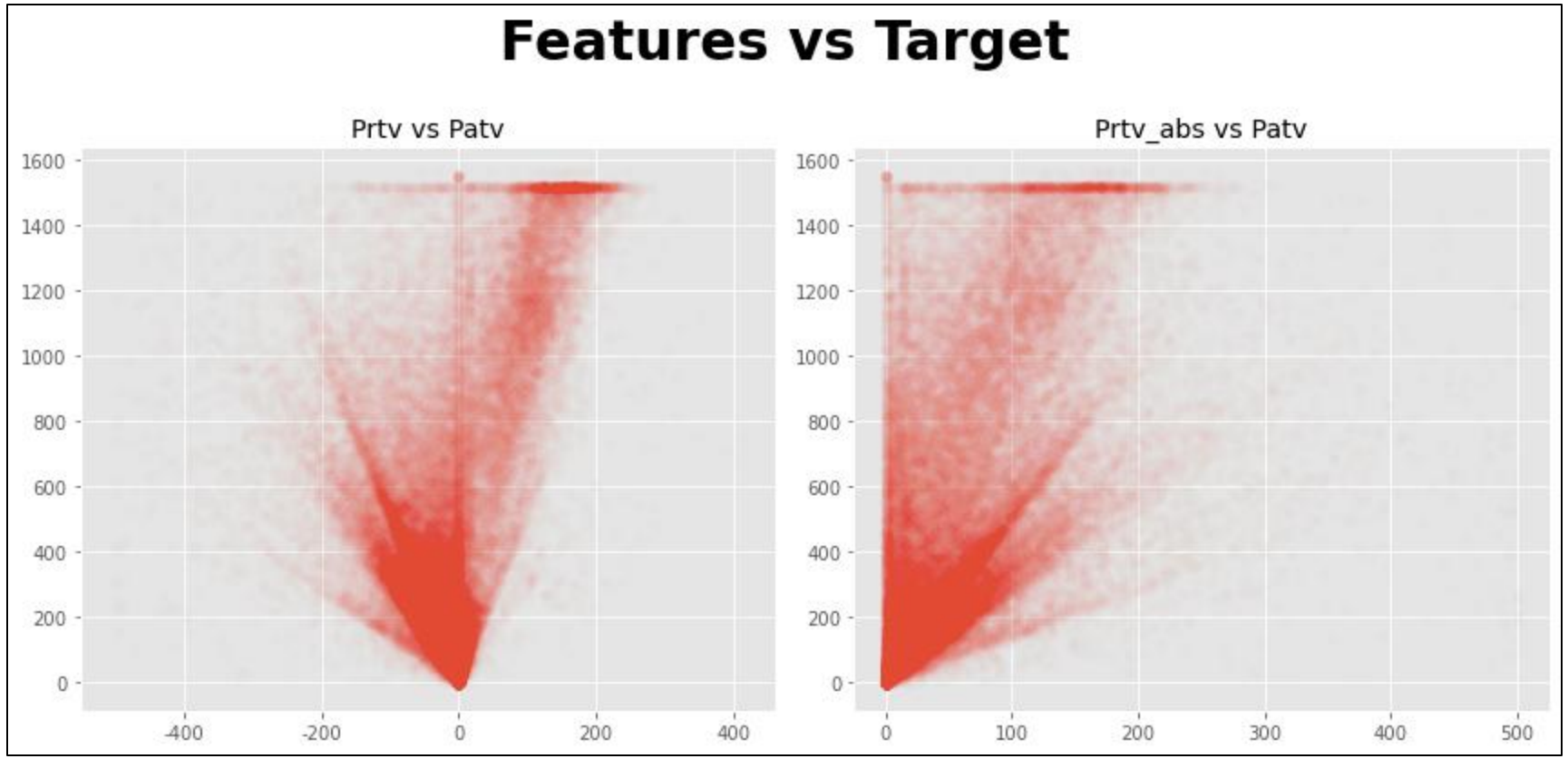
- 날개 3개의 각도로 서로 간의 correlation이 1이기 때문에 평균값 Pab를 사용
- 0.03도, 20도를 기준으로 층이 보임

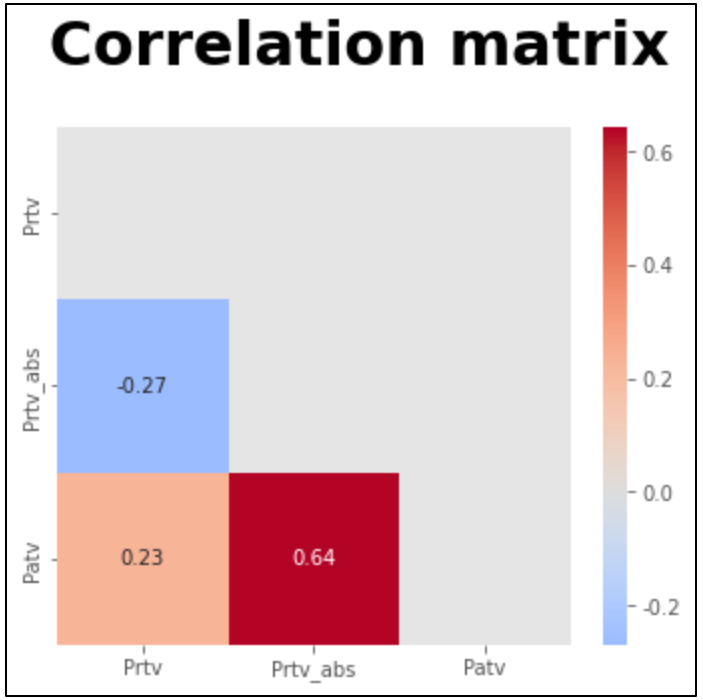




⑤ Prtv

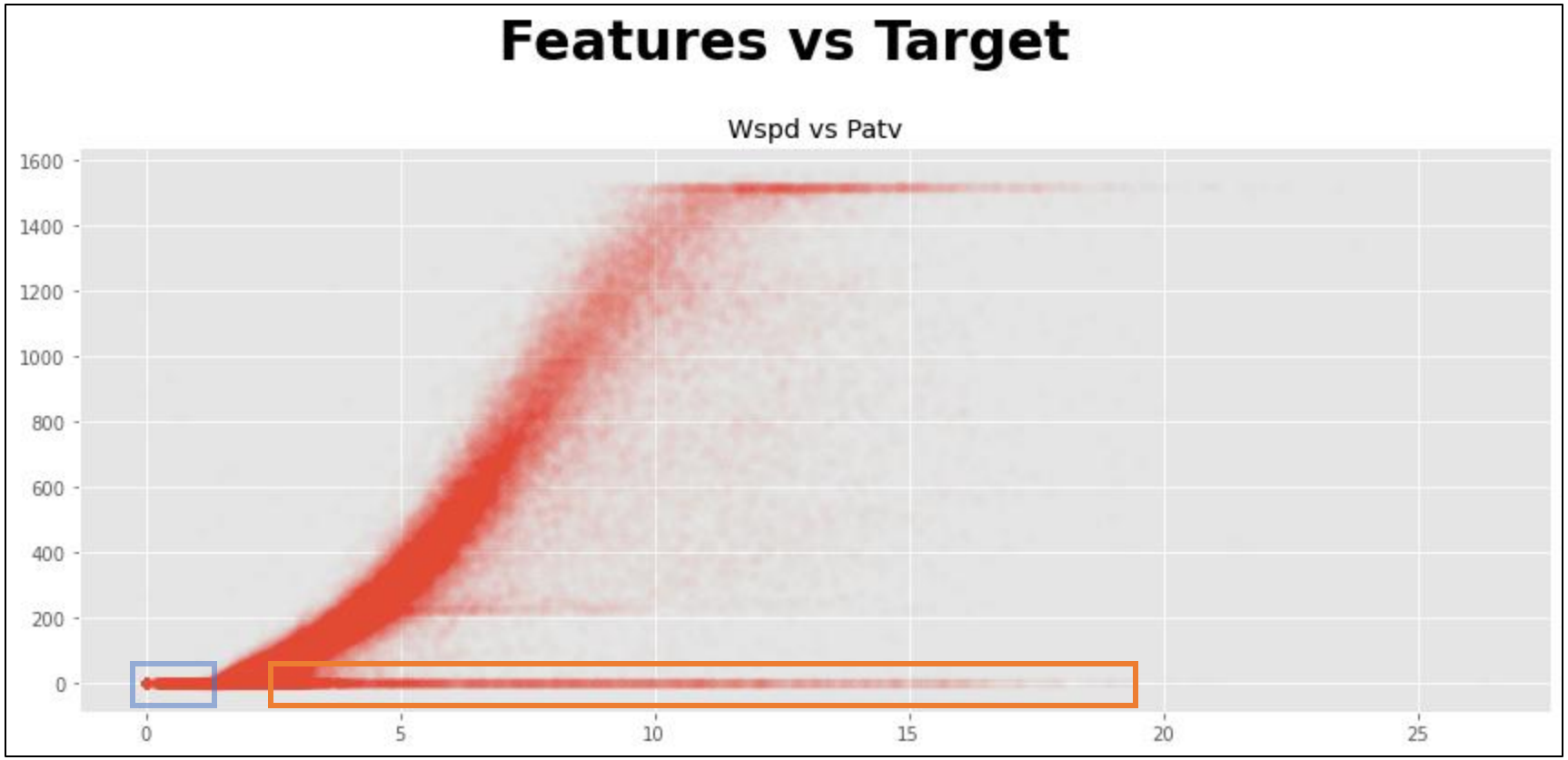
- 음수인 경우와 양수인 경우에 Patv와 다른 관계를 가지는 것 같음
- 절대값이 Patv와 강한 양의 상관관계를 가짐(0.64)

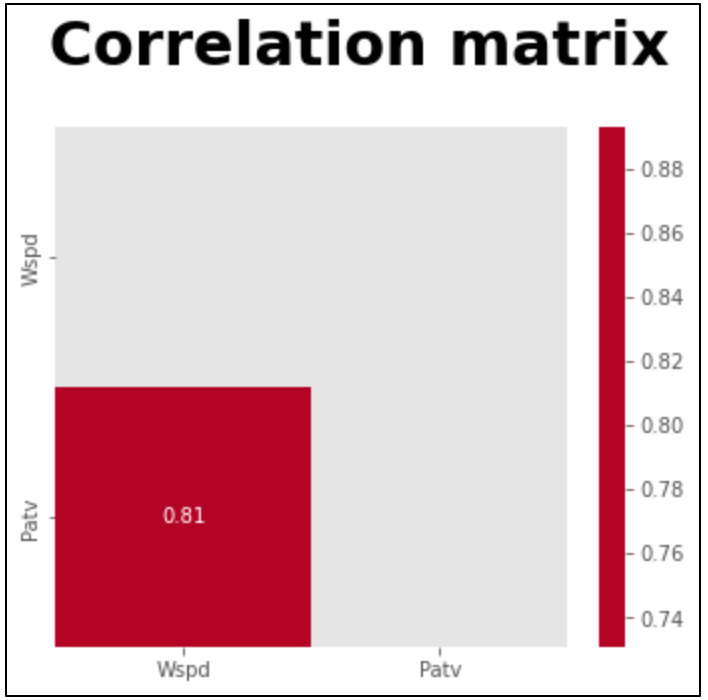




⑥ Wspd

- Patv와 강한 양의 상관관계를 가짐(0.81)
- 1m/s 이하: Patv = 0
- 1m/s 이상: 다른 요인으로 인해 Patv = 0 인 데이터를 구분할 수 있다면 Wspd의 correlation을 더 높일 수 있음





3) Preprocessing

① Mark abnormal Patv

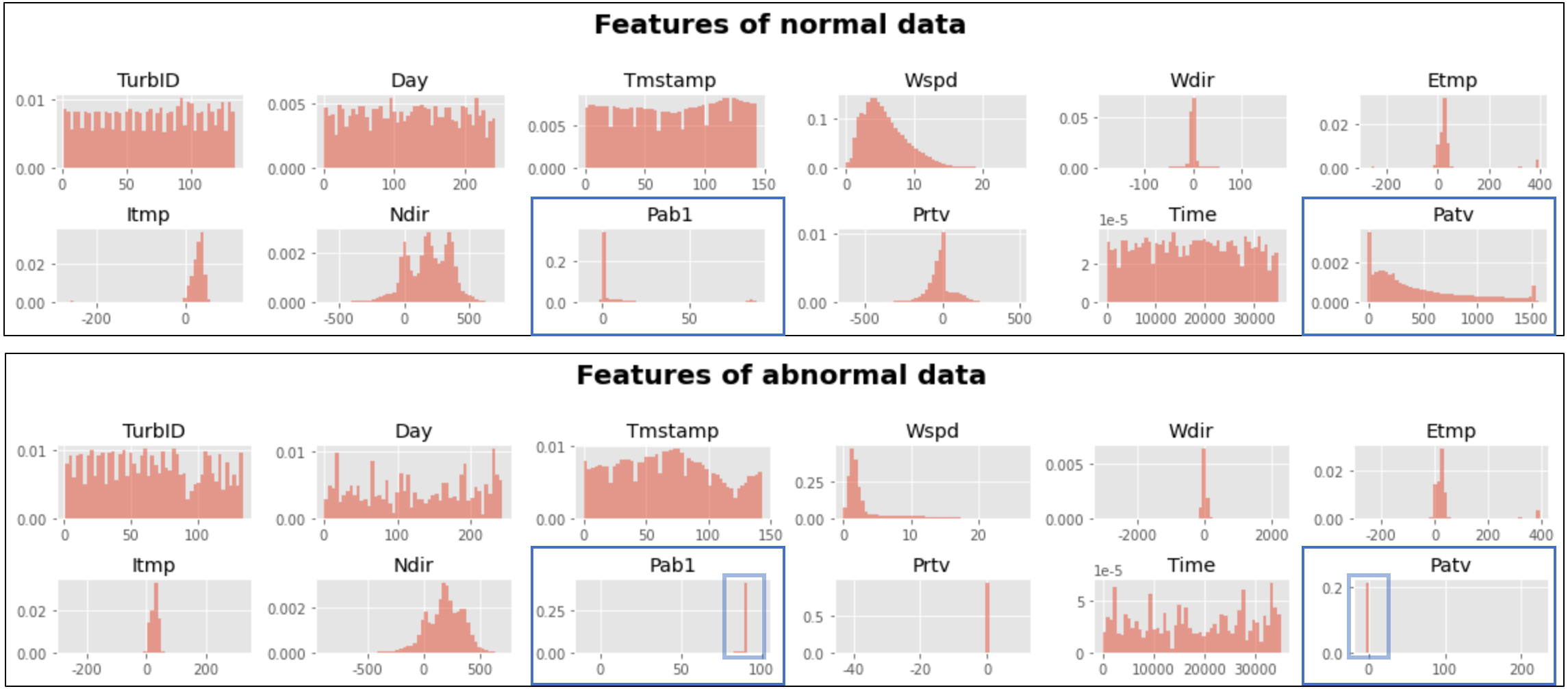
조건 1. Wspd > 2.5 and Patv <= 0
조건 2. Pab > 89
조건 3. |Wdir| > 180 or |Ndir| > 720
조건 4. Patv is null

하나라도 해당된다면 Abnormal = 1 o.w. 0

	TurbID	Day	Tmstamp	Wspd	Wdir	Etmp	Itmp	Ndir	Pab1	Pab2	Pab3	Prtv	Time	Patv	Abnormal
0	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	NaN	1
1	1	1	1	6.17	-3.99	30.73	41.80	25.92	1.00	1.00	1.00	-0.25	2	494.66	0
2	1	1	2	6.27	-2.18	30.60	41.63	20.91	1.00	1.00	1.00	-0.24	3	509.76	0
3	1	1	3	6.42	-0.73	30.52	41.52	20.91	1.00	1.00	1.00	-0.26	4	542.53	0
4	1	1	4	6.25	0.89	30.49	41.38	20.91	1.00	1.00	1.00	-0.23	5	509.36	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4688923	134	243	139	10.98	-1.96	-5.11	-0.67	345.57	8.82	8.82	8.82	136.49	34988	1152.60	0
4688924	134	243	140	11.82	-3.18	-5.46	-0.54	345.57	13.87	13.87	13.87	84.43	34989	681.65	0
4688925	134	243	141	11.91	-1.42	-5.21	-0.42	345.57	10.69	10.69	10.69	145.72	34990	1118.35	0
4688926	134	243	142	11.86	-0.95	-5.40	-0.38	345.57	13.94	13.94	13.94	89.56	34991	683.49	0
4688927	134	243	143	11.72	0.04	-5.23	-0.37	345.57	10.90	10.90	10.90	120.18	34992	1026.93	0

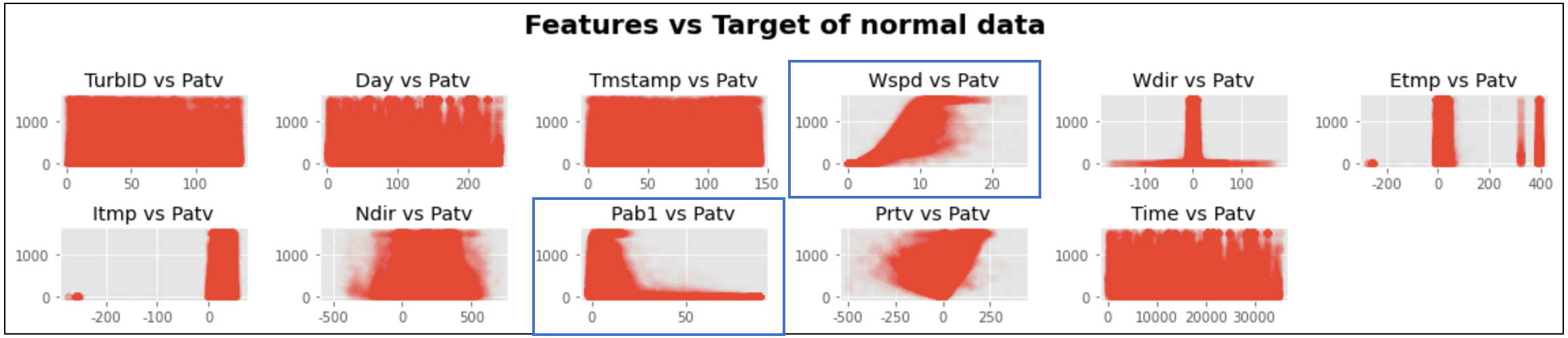
① Mark abnormal Patv (continued)

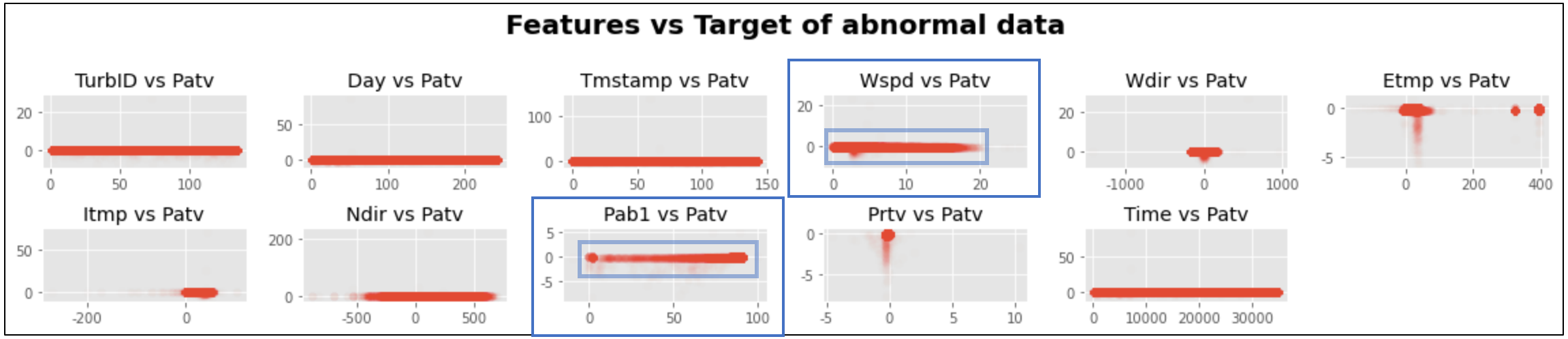
Pab가 90도가 넘어 Patv=0인 경우가 대부분



① Mark abnormal Patv (continued)

Pab로 인한 Patv = 0 분리 후, corr(Wspd, Patv) 상승: 0.81 → 0.88

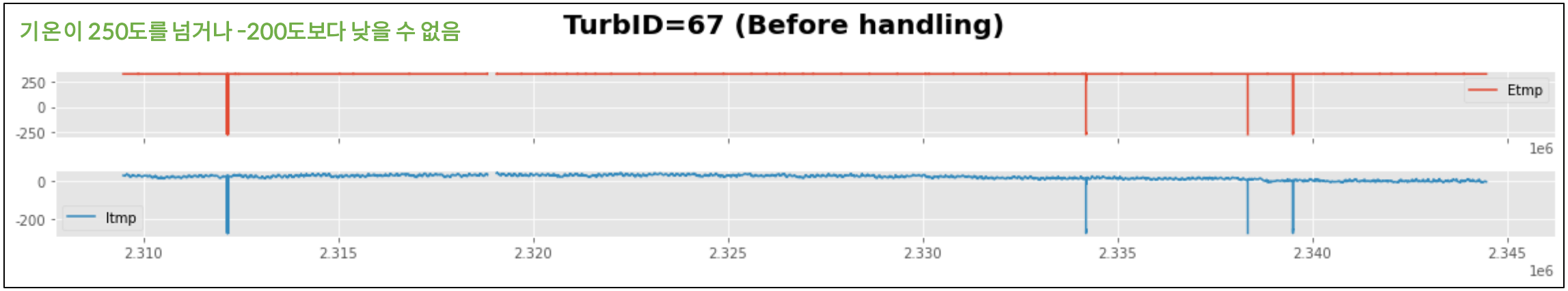


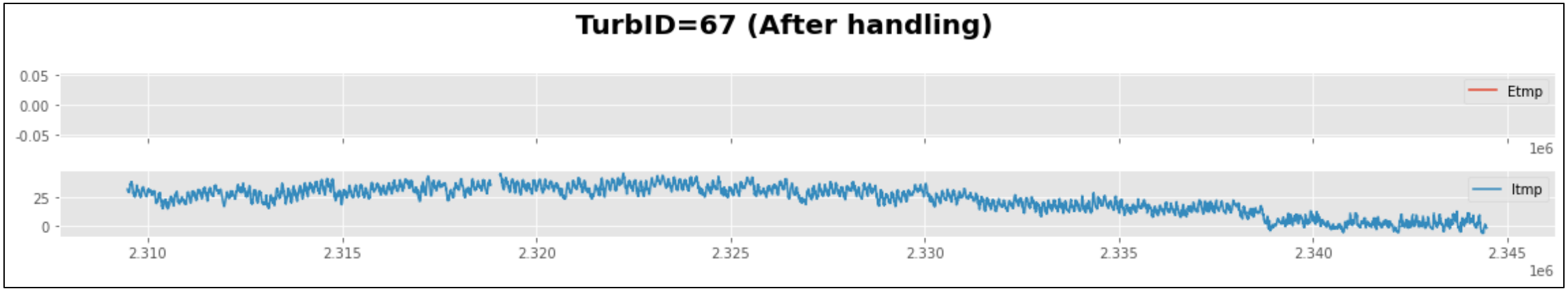


② Outlier handling

1. 다음과 같이 연속적으로 outlier가 이어지는 경우는 직접 null값으로 처리

기온이 250도를 넘거나 -200도보다 낮을 수 없음

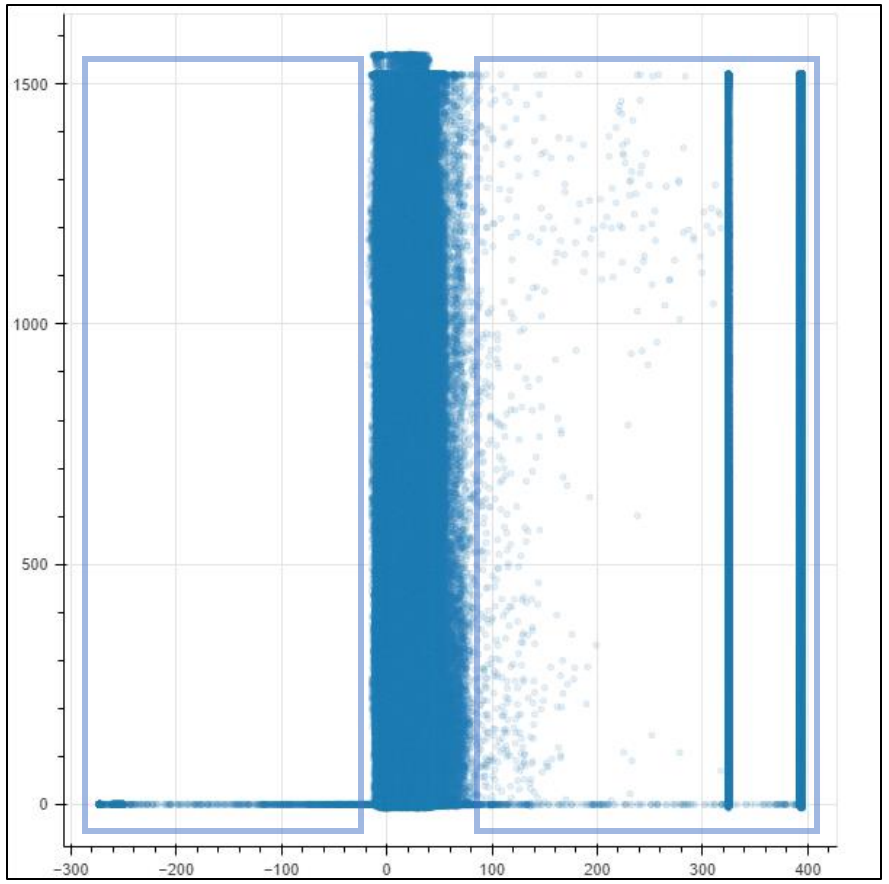




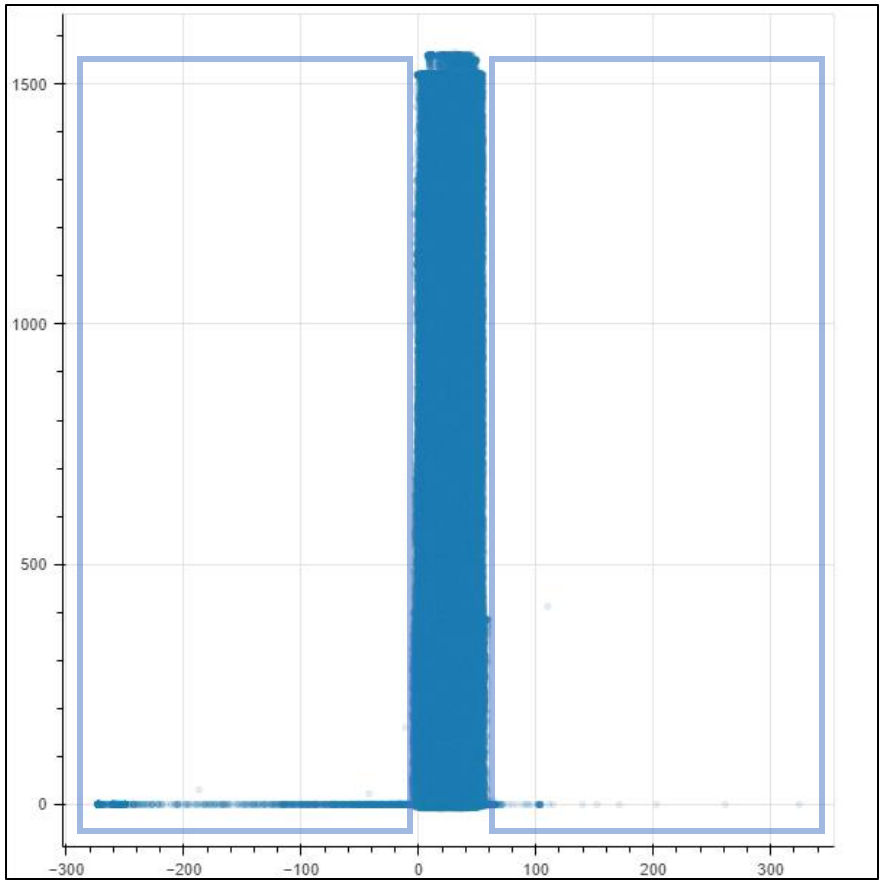
② Outlier handling

2. Ndir, Wdir, Pab: 주어진 정상 범위(not abnormal)로 clipping

Etmp: -20도~80도 안에 들도록 clipping

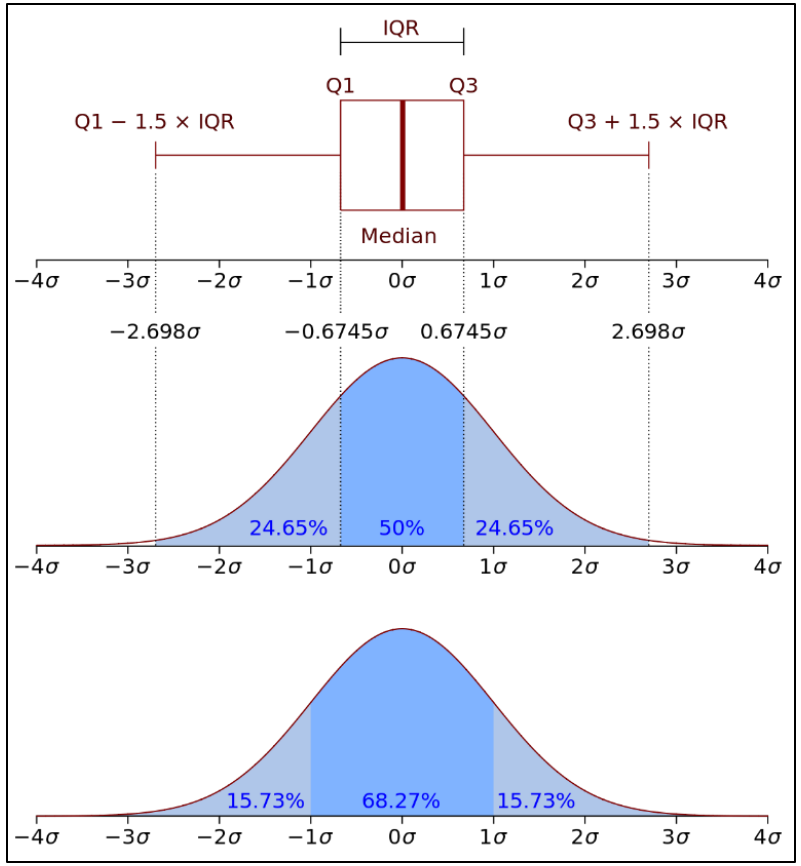


Itmp: -10도~65도 안에 들도록 clipping



② Outlier handling

3. 각 sample에 대하여 **해당값** 혹은 **1차 차분값**이 주변 2일 동안의 값들에 대하여 **이상치**라고 판단되는 경우 null로 채움
   (Etmp와 Itmp는 종모양의 분포를 가지고 있기 때문에 [Q1 – 1.5IQR, Q3 + 1.5IQR] 이외의 값을 이상치로 설정하는 것이 적절)



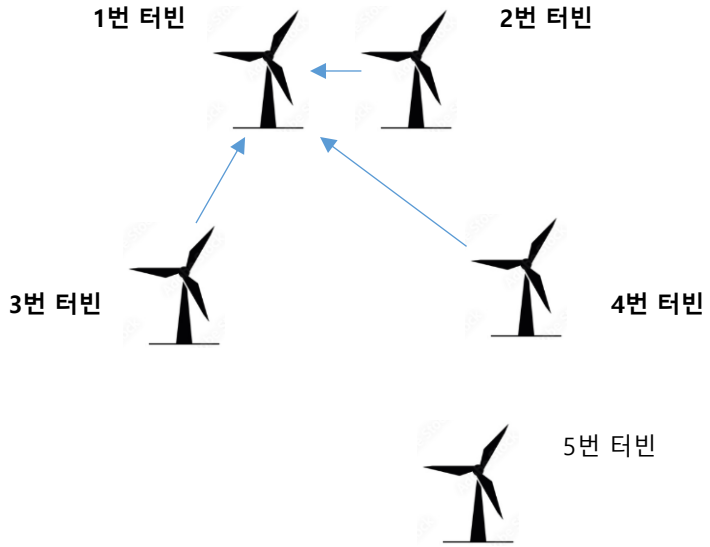
③ Imputing

1. Greedy imputing

이상치를 처리하고 난 후, 터빈의 데이터가 상당수 소실되는 경우 다음 알고리즘을 통해 값을 채움

Algorithm Greedy imputing

1. Imputing하고자 하는 터빈(대상 터빈)과 가장 가까운 k개의 터빈을 선택
2. 선택된 터빈들을 대상 터빈의 값과 비교하여 유사한 순으로 정렬
3. 대상 터빈의 값이 존재하지 않은 timestep의 값을 선택된 터빈들에서 순서대로 채워넣음



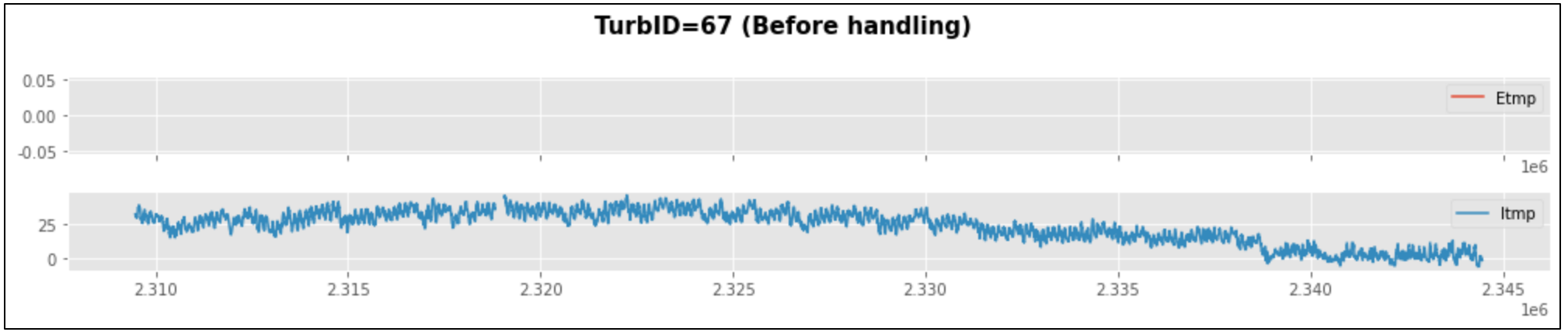
Ex) 1번 터빈에 대한 Greedy imputing 수행과정

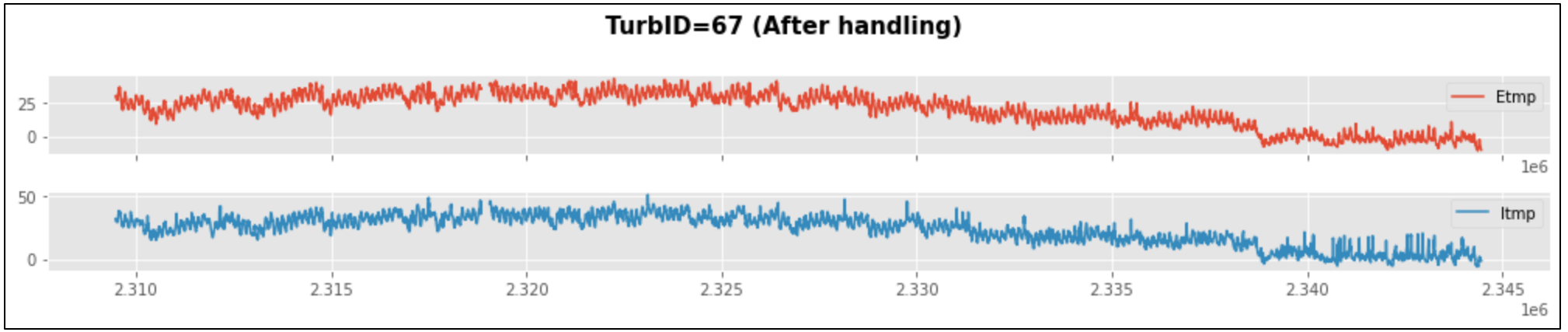
터빈 ID	거리	유사도 (MAE)	Time=1	Time=2	Time=3	Time=4	Time=5
1번 (기준)			10	20			
3번	2	0	10			40	
2번	1	1	11	21	31		
4번	3	2		22		44	55
5번	~~4~~	3	13	23	34	45	56

k = 3

③ Imputing

1. Greedy imputing(continued)





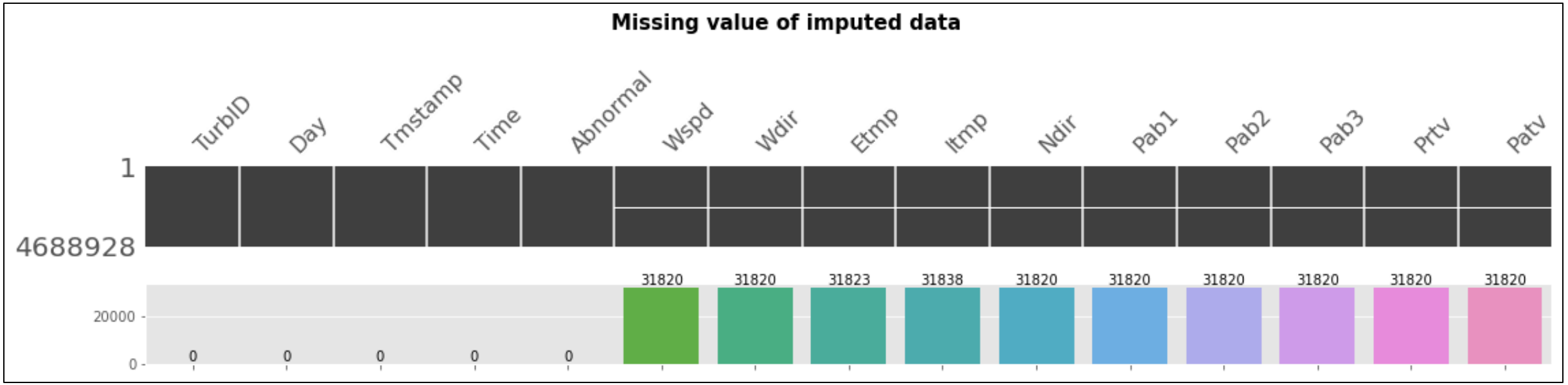
③ Imputing

2. Linear interpolation

각 turbine에 대하여 **threshold(e.g. 12시간) 이상 연속적이지 않은 결측치**를 linear interpolation으로 채움

(긴 sequence를 억지로 interpolation하면 시계열 특성이 망가지게 될 염려가 있음)

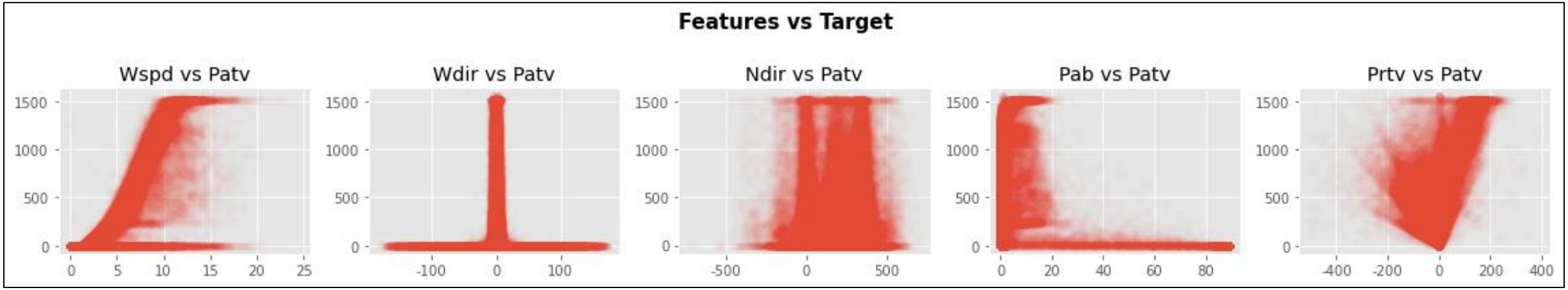
최종 imputing 결과



④ Feature engineering

1. Dummy variables (insight from EDA)

```
data['Wspd_extreme'] = data['Wspd'] < 1
data['Wdir_extreme'] = data['Wdir'].abs() > 10
data['Ndir_extreme'] = data['Ndir'] < -90
data['Pab_extreme1'] = data['Pab'] < 0.03
data['Pab_extreme2'] = data['Pab'] > 20
data['Prtv_pos']     = data['Prtv'] > 0
```



④ Feature engineering

2. Multiplicative variables(interactive effect)

```
data['Wspd_active']  = data['Wspd'] - 1
data['Wdir_active']  = data['Wdir'].abs() - 10
data['Ndir_cos_abs'] = np.abs(np.cos(Ndir_rad))  # 0도, 180도, 360도
data['Prtv_abs']     = data['Prtv'].abs()
```

```
data['Wspd_comb'] = data['Wspd_extreme'] * data['Wspd_active']
data['Wdir_comb'] = data['Wdir_extreme'] * data['Wdir_active']
data['Ndir_comb'] = data['Ndir_extreme'] * data['Ndir_cos_abs']
data['Prtv_comb'] = data['Prtv_pos'] * data['Prtv_abs']
```

④ Feature engineering

3. Feature extraction(domain knowledge)

```
ALPHA              = 40
data['Pab']        = (data['Pab1'] + data['Pab2'] + data['Pab3'])/3
Pab_rad            = np.radians(data['Pab']+ALPHA)
data['TSR']        = 1 / np.tan(Pab_rad)
data['RPM']        = data['Wspd_active'] * data['TSR']
data['Wspd_cube'] = data['Wspd_active']**3
data['Patv_pos']  = np.maximum(data['Patv'], min_val)
data['Patan_abs'] = np.arctan(data['Prtv_abs'] / data['Patv_pos'])

Wdir_rad          = np.radians(data['Wdir'])
data['Wdir_cos'] = np.cos(Wdir_rad)
data['Wdir_sin'] = np.sin(Wdir_rad)
data['Wspd_cos'] = data['Wspd_active'] * np.cos(Wdir_rad)
data['Wspd_sin'] = data['Wspd_active'] * np.sin(Wdir_rad)
data['Ndir_sin_abs'] = np.abs(np.sin(Ndir_rad))
```

④ Feature engineering

4. Time encoding

```
DAY             = 6*24  # 10minute * 6 * 24hour
Time_in_day     = data['Time'] * (2*np.pi) / DAY
data['Day_cos'] = np.cos(Time_in_day)
data['Day_sin'] = np.sin(Time_in_day)

YEAR             = 365*DAY
Time_in_year     = data['Time'] * (2*np.pi) / YEAR
data['Year_cos'] = np.cos(Time_in_year)
data['Year_sin'] = np.sin(Time_in_year)
```

4) Training

① Data split

Training set : Day ∈ {1일, ⋯, 217일}

Validation set : Day ∈ {218일, ⋯, 241일}

- Input sequence의 길이: 2일
- **Null값은 포함하는 샘플은 제외**하여 sliding window로 생성

② Model

GRU (# units=32, 1 hidden layer)

→ Seq2Seq 모델로 Transformer를 먼저 사용해보았으나 **오버피팅**이 너무 심해 가장 간단한 모델을 사용

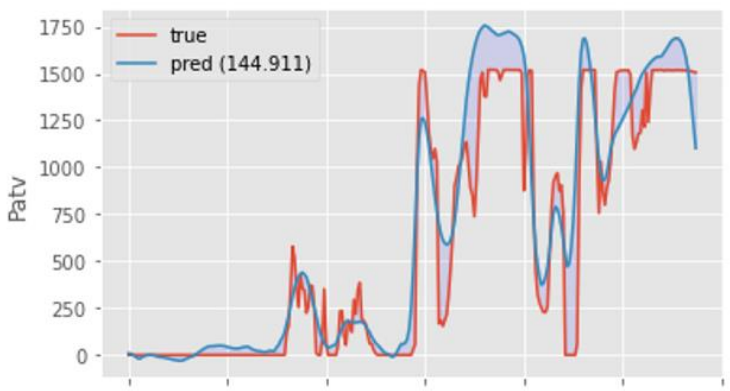
Outputs : [Wspd, Patv]

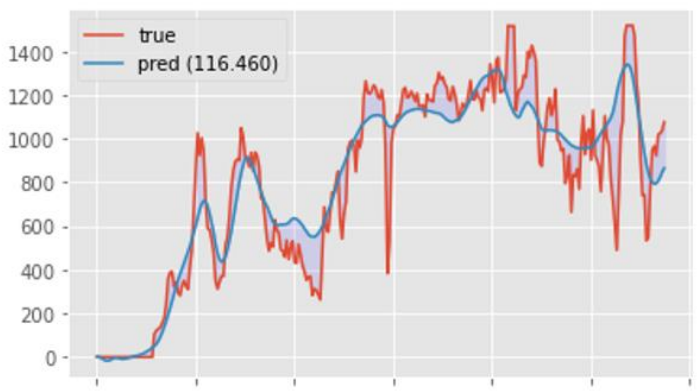
Loss : MSE

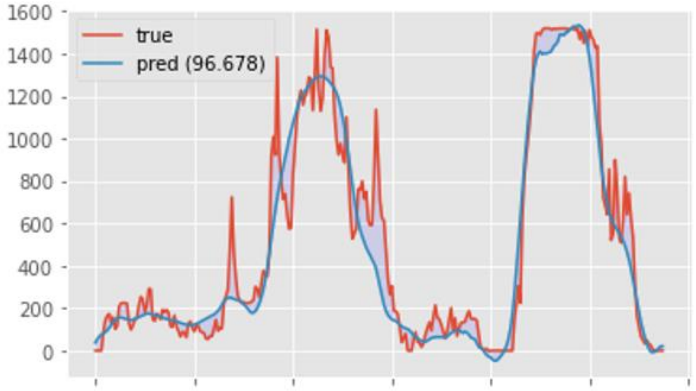
5) Result

① Training set

평가점수 = 129.80

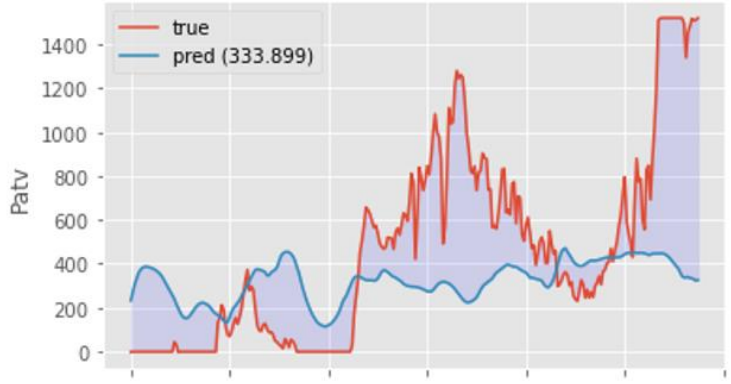


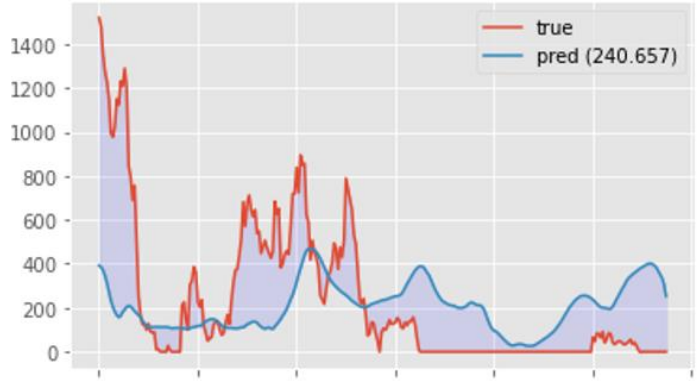


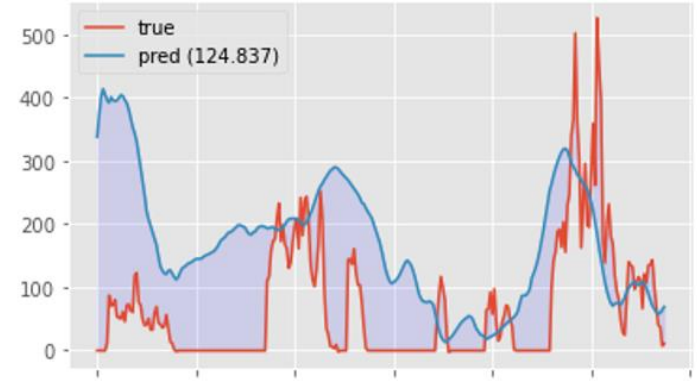


② Validation set

평가점수 = 250.07







5) Result

③ Test set (public)

평가점수 = 168.6642

#	팀	팀 멤버	점수
1	거북이알	한상 Lo	168.6642
2	찌개사랑	종버 ai ko	178.5241
3	시계열_사나이들	He 다오 DACON pr	179.41804
4	에이셉토치	팔협 킹보	192.2706
5	아기돼지삼형제	짱짱  간지	232.10154

Test set에서 2위 팀의 점수와 비교하여 **5.6%** 차이를 두며 1등 수상

3. 결론

- **데이터의 실측값과 이상치를 처리하는 전처리 작업**을 통해 좋은 성적을 낼 수 있었음
- 모델의 복잡도를 충분히 줄인 경우에도 overfitting이 발생한 것으로 보아,

1) **너무 많은 feature들을 사용하여 데이터를 구성**
   → 더욱 유의미한 feature들을 선택

2) **Training set의 데이터와 validation set의 분포, test set의 분포의 차이가 심함**
   → 학습 구간을 validation set 근처로 축소시키고 validation set 구간의 길이를 줄이는 것도 고려해볼만함

3) **모델이 데이터의 주요한 패턴을 제대로 학습하지 못함**
   → 유의미한 feature들을 선택한 후, 개선된 시계열 예측 모델을 사용

4) **다양한 크기의 sequence length를 고려**
   → 적절한 크기의 time dependency를 선택함으로써 오버피팅을 방지

5) **각 터빈의 위치 정보를 직접적으로 사용**
   → 위치 정보를 기반으로 GNN 모델을 사용

다양한 수행을 통해 개선된 결과를 얻을 수 있을 것으로 생각됨

THANK YOU

감사합니다