프로젝트 활동 보고서

제조업 생산데이터 활용을 통한
에너지(전력) 사용량 예측 모델

광주 인공지능 사관학교
7반 ＡＰＩ

팀 명	API	
팀구성원	이름	역할
	백 현 선	총 괄
	김 한 수	데이터 전처리 및 모델 구현하기(머신러닝), WBS문서
	박 지 호	데이터 전처리 및 모델 구현하기(머신러닝), 회의록
	오 준 묵	데이터 탐색 및 수집 및 분석 빅데이터 분석 정의서, PPT
	이 치 원	모델 구현하기(딥러닝) 및 시각화

1. 문제 제기

1.1 프로그램 목표

제조공정 데이터 활용을 통해 최적의 에너지(전력) 사용량 예측 모델을 만들고, 이를 통해 제조업체의 생산원가를 낮추는데 일조한다. 정형 데이터 기반으로 머신러닝 방법을 사용하여 그 전력 사용량을 예측할 수 있도록 한다. 추가로 딥러닝의 순환신경망을 사용하여 전력 사용량의 추세를 예측하는 모델을 만든다.

1.2 데이터 셋

자원 최적화 AI 데이터 셋 - 출처(KAMP, 인공지능중소벤처제조플랫폼)

데이터 정의	총 18개의 제조업 생산데이터 : **날짜**(공장운영 날짜), **시간**(공장운영 시간), **15분** (0-15분까지의 피크전기 사용량), **30분** (15-30분까지의 피크전기 사용량), **45분** (30-45분까지의 피크전기 사용량), **60분** (45-60분까지의 피크전기 사용량), **평균** (15분, 30분, 45분, 60분 속성의 평균값), **생산량** (해당 시점에 생산해야할 생산량), **기온**, **풍속**, **습도**, **강수량**, **전기요금(계절),** **day**(해당 시점의 요일 값), **d(**해당 시점의 일 값), **m**(해당 시점의 달 값), **공장인원**, **인건비** 파일유형 : CSV

2. 데이터 탐색

2.1 데이터 내부 관찰

(1) 원천 데이터의 각 수치적 특성들의 기본 통계치를 확인해본다.

날짜, 시간, day, d, m, 인건비는 범주형 특성이므로 통계수치에서 제외한다.

	15분	30분	45분	60분	평균	생산량
count	6168.000000	6168.000000	6168.000000	6168.000000	6168.000000	6168.000000
mean	90.410182	92.695363	95.106355	95.037938	93.424125	467.344682
std	55.349403	57.942122	59.285709	59.347554	57.355938	857.571815
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	23.000000	23.000000	23.000000	23.000000	23.000000	0.000000
50%	101.000000	104.000000	105.000000	107.000000	104.000000	45.000000
75%	133.000000	143.000000	149.000000	149.000000	144.000000	637.250000
max	207.000000	222.000000	218.000000	214.000000	208.000000	9830.000000

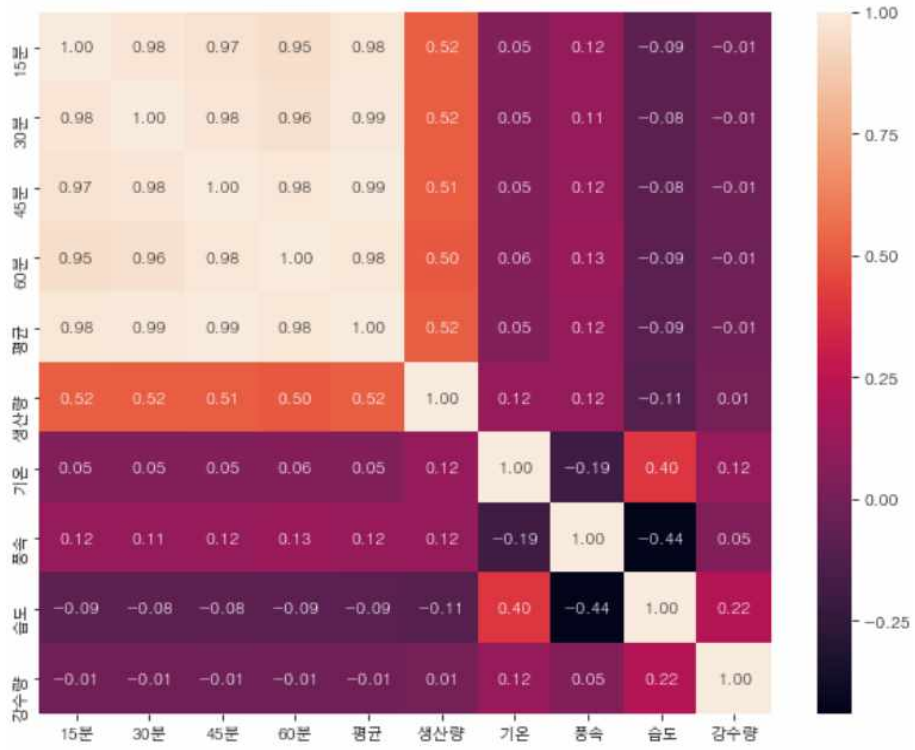
	기온	풍속	습도	강수량	전기요금(계절)	공장인원
count	6168.000000	6165.000000	6168.000000	6167.000000	6168.000000	6151.000000
mean	15.906064	2.063633	70.098735	2.244252	162.757198	0.901336
std	9.160356	1.164118	22.996164	9.613491	30.820855	1.985511
min	-12.000000	0.000000	8.000000	0.000000	109.800000	0.000000
25%	9.600000	1.200000	53.000000	0.000000	167.200000	0.000000
50%	17.400000	1.900000	74.000000	0.000000	167.200000	0.112971
75%	23.300000	2.800000	91.000000	0.100000	191.600000	1.162285
max	33.400000	7.600000	98.000000	122.400000	191.600000	48.386364

(2) 원천데이터에 결측치가 포함 되어 있는지 확인한다.

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   날짜        6168 non-null   int64
 1   시간        6168 non-null   int64
 2   15분       6168 non-null   int64
 3   30분       6168 non-null   int64
 4   45분       6168 non-null   int64
 5   60분       6168 non-null   int64
 6   평균        6168 non-null   int64
 7   생산량       6168 non-null   int64
 8   기온        6168 non-null   float64
 9   풍속        6165 non-null   float64
 10  습도        6168 non-null   int64
 11  강수량       6167 non-null   float64
 12  전기요금(계절)  6168 non-null   float64
 13  day       6168 non-null   int64
 14  d         6168 non-null   int64
 15  m         6168 non-null   int64
 16  공장인원      6151 non-null   float64
 17  인건비       6168 non-null   float64
dtypes: float64(6), int64(12)
```

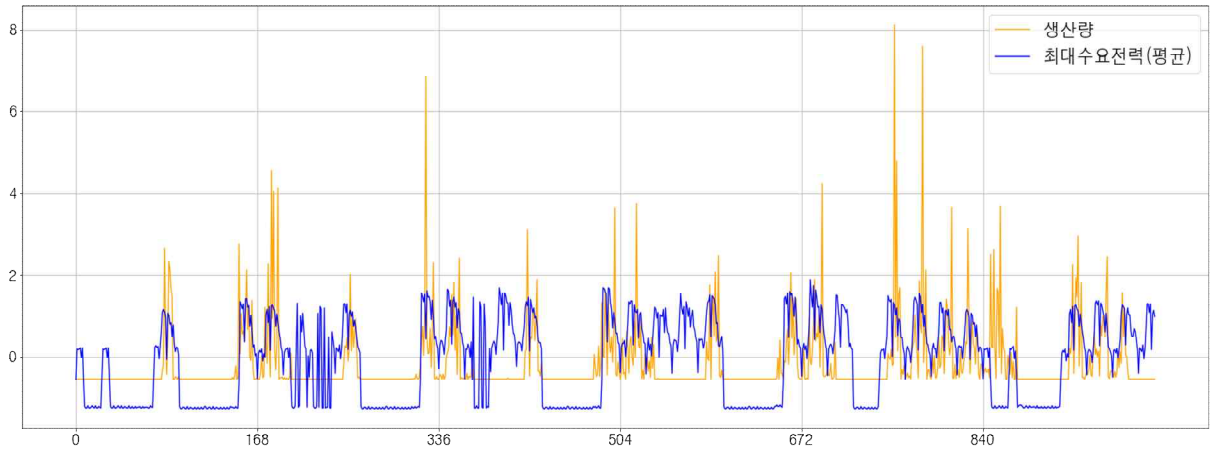
풍속 특성에서 3개, 강수량 특성에서 1개, 공장 인원 특성에서 17개의 결측치가 존재한다.

(3) 각 변수간의 상관관계를 히트맵으로 확인한다.

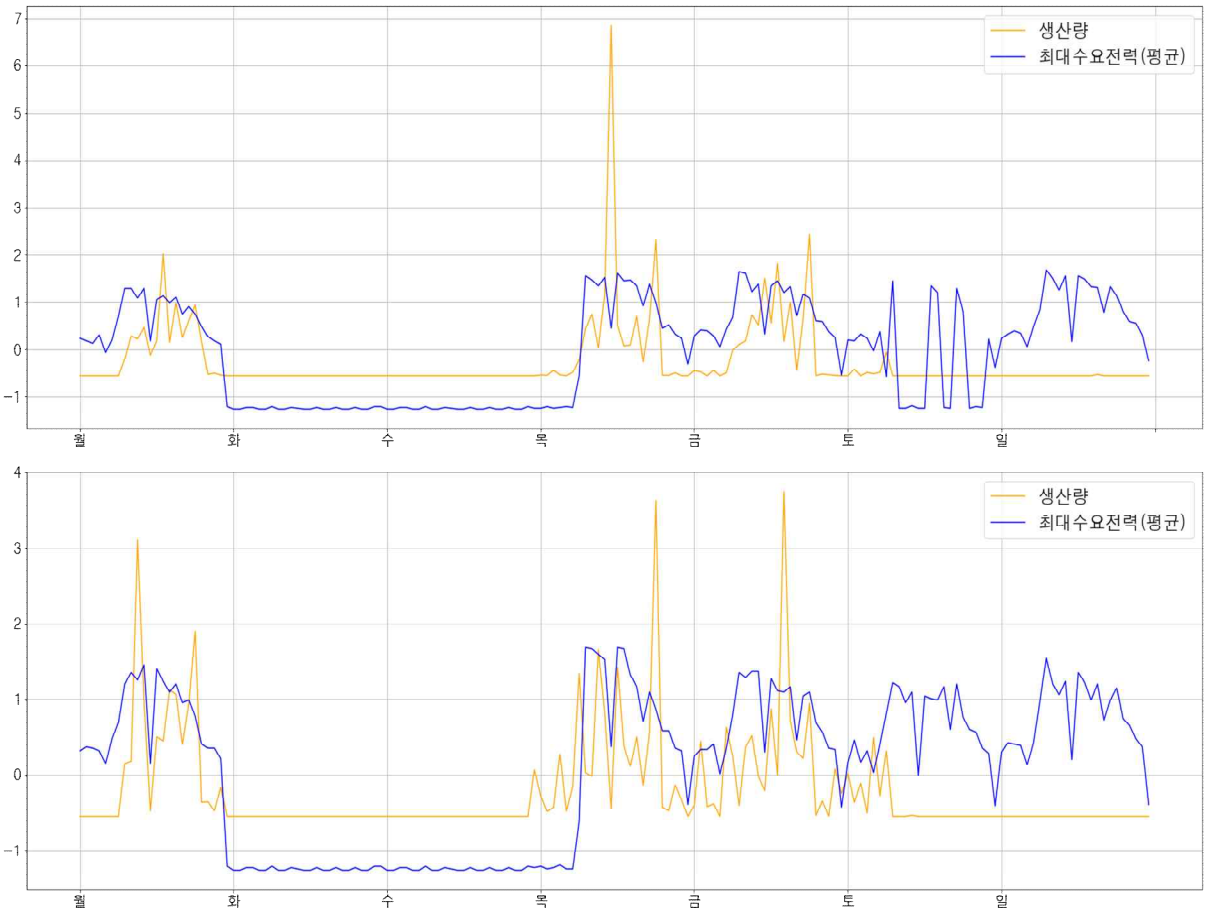


생산량과 전력소비량이 가장 큰 상관계수를 가지고 있으나 0.52 이므로 독립변수로 활용할 수 있다. 환경 조건의 상관계수는 낮지만 독립변수로 활용할 수 있다.

(4) 시간흐름에 따른 주기성을 확인해 본다.



전체데이터의 일부를 확인했을 때 일정 주기를 보이는 것으로 판단된다. 따라서 데이터의 특성을 반영하여 1주 단위로 그려본다.



서로 다른 1주간의 데이터를 보았을 때 1주 단위로 주기성을 보이는 것을 확인할 수 있다. 이를 통해 각 요일에 해당하는 특성이 학습에 도움을 줄 수 있을 것 추론할 수 있다. 또한 주기성을 가진 데이터는 시계열 데이터 분석에서 효과적일 것으로 예상된다.

2.2 제약사항

원천데이터의 사이즈는 총 6168개로 학습하는 데에 충분하지 않을 것으로 판단된다. 머신러닝기반의 모델에서는 어느 정도 훈련이 될 수 있지만, 딥러닝기반 모델에서는 문제가 될 것으로 예상된다.

이를 해결하기 위해 6168개의 데이터에 대해서 15분, 30분, 45분, 60분의 수치를 각각 하나의 데이터로 확장시켜 4배의 데이터를 확보하는 방안을 마련한다. 이를 더불어 부족한 데이터를 보완하는 방법을 생각해 본다.

3. 데이터 가공

3.1 결측 데이터 처리

풍속 특성에서 3개, 강수량 특성에서 1개, 공장 인원 특성 17개의 결측치가 존재한다. 모두 연속적 데이터이므로 평균을 이용해서 결측치를 채운다.

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   날짜        6168 non-null   int64
 1   시간        6168 non-null   int64
 2   15분       6168 non-null   int64
 3   30분       6168 non-null   int64
 4   45분       6168 non-null   int64
 5   60분       6168 non-null   int64
 6   평균        6168 non-null   int64
 7   생산량       6168 non-null   int64
 8   기온        6168 non-null   float64
 9   풍속        6165 non-null   float64
 10  습도        6168 non-null   int64
 11  강수량       6167 non-null   float64
 12  전기요금(계절)  6168 non-null   float64
 13  day       6168 non-null   int64
 14  d         6168 non-null   int64
 15  m         6168 non-null   int64
 16  공장인원      6151 non-null   float64
 17  인건비       6168 non-null   float64
dtypes: float64(6), int64(12)
```

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   날짜        6168 non-null   int64
 1   시간        6168 non-null   int64
 2   15분       6168 non-null   int64
 3   30분       6168 non-null   int64
 4   45분       6168 non-null   int64
 5   60분       6168 non-null   int64
 6   평균        6168 non-null   int64
 7   생산량       6168 non-null   int64
 8   기온        6168 non-null   float64
 9   풍속        6168 non-null   float64
 10  습도        6168 non-null   int64
 11  강수량       6168 non-null   float64
 12  전기요금(계절)  6168 non-null   float64
 13  day       6168 non-null   int64
 14  d         6168 non-null   int64
 15  m         6168 non-null   int64
 16  공장인원      6168 non-null   float64
 17  인건비       6168 non-null   float64
dtypes: float64(6), int64(12)
```

결측치를 해결한 데이터를 이용하여 데이터 가공을 진행한다.

3.2 데이터 확장

원천데이터에서 15분 단위로 측정된 피크전력 사용량을 이용, 데이터 크기를 늘린다. 15분 단위로 측정된 데이터(15분, 30분, 45분, 60분) 4개와 전력량을 제외한 나머지 데이터를 조합하여 기존보다 4배 늘어난 데이터로 만든다.

15분 단위의 전력량을 가진 총 24672개의 데이터를 만들 수 있다..

```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   날짜        24672 non-null  float64
 1   시간        24672 non-null  float64
 2   15분       24672 non-null  float64
 3   30분       24672 non-null  float64
 4   45분       24672 non-null  float64
 5   60분       24672 non-null  float64
 6   평균        24672 non-null  float64
 7   생산량       24672 non-null  float64
 8   기온        24672 non-null  float64
 9   풍속        24672 non-null  float64
 10  습도        24672 non-null  float64
 11  강수량       24672 non-null  float64
 12  전기요금(계절)  24672 non-null  float64
 13  day       24672 non-null  float64
 14  d         24672 non-null  float64
 15  m         24672 non-null  float64
 16  공장인원      24672 non-null  float64
 17  인건비       24672 non-null  float64
 18  target    24672 non-null  float64
dtypes: float64(19)
```

3.3 데이터 정규화

연속변수에 대해서 정규화를 시행한다. target 값은 모델의 예측 결과를 다시 원래의 데이터로 변환해야하기 때문에 inverse_transform을 할 수 있도록 별도의 스케일러를 저장해 둔다.

	생산량	기온	풍속	습도	강수량	target	전기요금(계절)	공장인원
count	2.467200e+04	2.467200e+04	2.467200e+04	2.467200e+04	2.467200e+04	2.467200e+04	2.467200e+04	2.467200e+04
mean	-4.602209e-15	-2.395817e-15	-3.129666e-15	-2.269739e-14	-1.588422e-14	-9.227373e-16	9.816446e-14	1.432063e-15
std	1.000020e+00	1.000020e+00	1.000020e+00	1.000020e+00	1.000020e+00	1.000020e+00	1.000020e+00	1.000020e+00
min	-5.450069e-01	-3.046642e+00	-1.771061e+00	-2.700614e+00	-2.334471e-01	-1.607964e+00	-1.718365e+00	-4.532387e-01
25%	-5.450069e-01	-6.884639e-01	-7.406904e-01	-7.436076e-01	-2.334471e-01	-1.211627e+00	1.441609e-01	-4.532387e-01
50%	-4.925289e-01	1.631004e-01	-1.396408e-01	1.696622e-01	-2.334471e-01	1.841681e-01	1.441609e-01	-3.972118e-01
75%	1.981398e-01	8.072324e-01	6.331374e-01	9.089759e-01	-2.230434e-01	8.562176e-01	9.358968e-01	1.318384e-01
max	1.091852e+01	1.909899e+00	4.754621e+00	1.213399e+00	1.250067e+01	2.217549e+00	9.358968e-01	2.394520e+01

3.4 머신러닝에 사용할 데이터

머신러닝의 단변량 회귀모델에 사용할 데이터를 설정한다. 독립변수로는 생산량, 기온, 풍속, 습도, 강수량의 총 5개의 특성을 사용한다. 종속변수로는 target(15분 단위의 피크전력량)을 사용한다. 해당 특성을 갖는 데이터들을 '훈련을 위한 데이터셋'과 '평가를 위한 데이터셋'으로 구분하여 준비한다.

데이터 셋을 구분할 때 '9월의 데이터'는 <테스트셋>으로 이외의 데이터는 <훈련셋>으로 나누어서 사용할 수 있도록 준비한다.

3.5 딥러닝(순환신경망)에 사용할 데이터

순환신경망에 사용할 데이터를 설정한다. 생산량, 기온, 풍속, 습도, 강수량, target(15분 단위의 피크전력량)의 총 6개 특성을 사용한다. 1주일의 데이터(672개의 데이터 포인트)를 입력값으로 다음 하루의 피크전력량(96개의 데이터 포인트)을 타겟값으로 설정한다. 이외에도 다양한 타임스텝을 구성하여 목표한 프로그램에 적합한 수치를 찾아볼 수 있도록 한다.

해당 데이터 셋도 마찬가지로 '훈련용 데이터 셋'과 '테스트용 데이터 셋'으로 구분하여 준비한다.

4. 모델 탐색 및 설정

4.1 머신러닝 모델 탐색

	model	score
0	HistGradientBoostingRegressor	0.8350686163499208
1	GradientBoostingRegressor	0.8258984561220296
2	ExtraTreesRegressor	0.8202433804014658
3	RandomForestRegressor	0.8073834812616958
4	MLPRegressor	0.8066455544573319
5	BaggingRegressor	0.7946679383974198
6	SVR	0.7033515400777091
7	NuSVR	0.7027724006697037
8	AdaBoostRegressor	0.6803639186975778
9	ExtraTreeRegressor	0.6642862801077527
10	DecisionTreeRegressor	0.6030229684929503
11	KNeighborsRegressor	0.4652774046732411
12	LinearSVR	0.3327701653736943
13	OrthogonalMatchingPursuit	0.31552624377802896
14	OrthogonalMatchingPursuitCV	0.31552624377802896
15	HuberRegressor	0.3127064528536697
16	ARDRegression	0.30650676750378447
17	LassoCV	0.30504927637483603
18	KernelRidge	0.30455654854065695
19	LarsCV	0.3044284506316701
20	LassoLarsCV	0.3044284506316701

사이킷런의 주요 회귀 모델들을 사용하여 R2 점수를 측정하여 이중 상위 점수를 가진 모델을 우선적으로 사용한다. 최초 점수 측정에는 데이터셋을 8:2로 분할한 데이터를 이용한다.

히스토그램 기반 그래디언트 부스팅 모델은 LightGBM 모델로 대체하고, 그래디언트 부스팅 모델은 XGBoost 모델로 대체하여 평가한다.

그리고 ExtraTreesRegressor, RandomForestRegressor를 포함한 총 4개의 모델을 이용하여 이후의 모델 정교화에 사용한다.

	model	score
0	LGBMRegressor	0.840912
1	XGBRegressor	0.819080

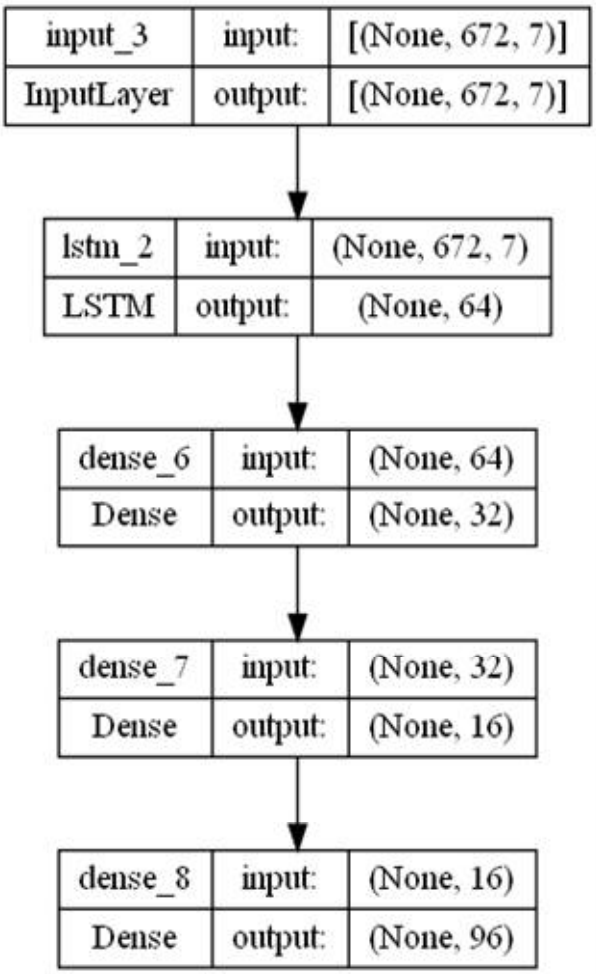
LightGBM과 XGBoost를 사용하여 평가지표를 구해본다.
위의 결과지표와 더불어 LGBMRegressor가 가장 좋은 성능을 보여주고 있다.

4.2 딥러닝(순환신경망) 모델 탐색

순환신경망 구조를 사용하기 위해 LSTM(Long-Short Term Memory), GRU(Gated Recurrent Unit)을 이용한 신경망 구조를 설계한다. 또한 양방향 순환신경망을 사용하기 위해 Bidirectional 층을 사용하여 신경망 구조를 설계한다.

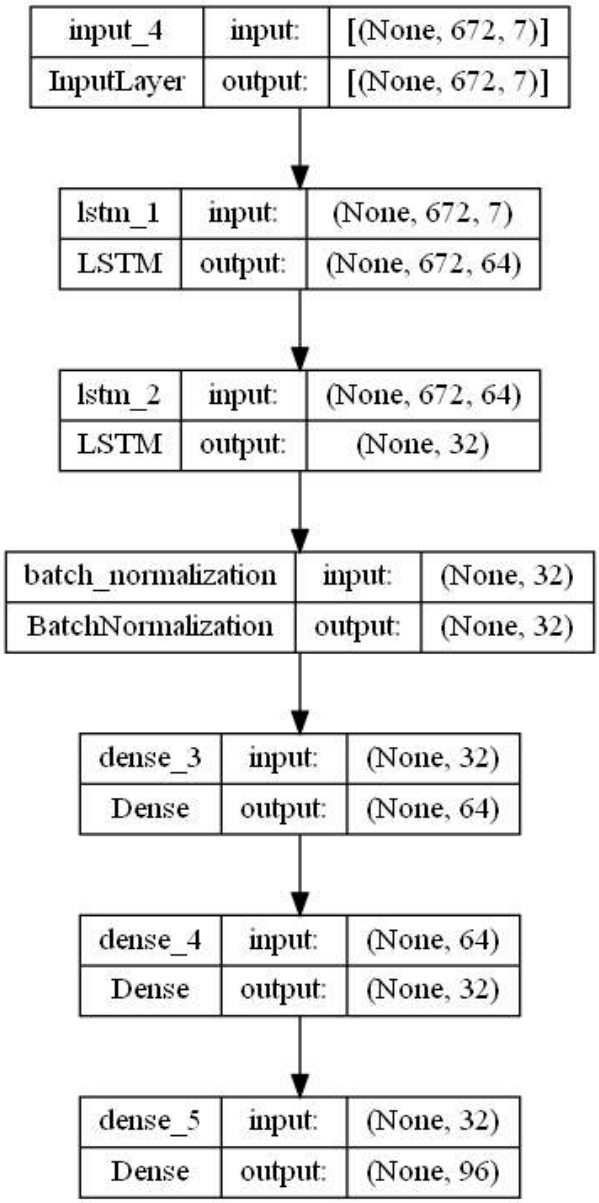
총 5~7개의 은닉층을 지닌 신경망 구조를 설계하고, 시계열 학습데이터를 이용하여 모델을 학습시킨다. 이를 이용하여 R2점수와 MAE(Mean Absolute Error)를 가지고 모델의 성능을 확인한다.

모든 모델의 훈련에 사용된 파라미터는 배치사이즈 32, 에폭 100, 조기종료 허용치를 10으로 설정하고 진행한다.



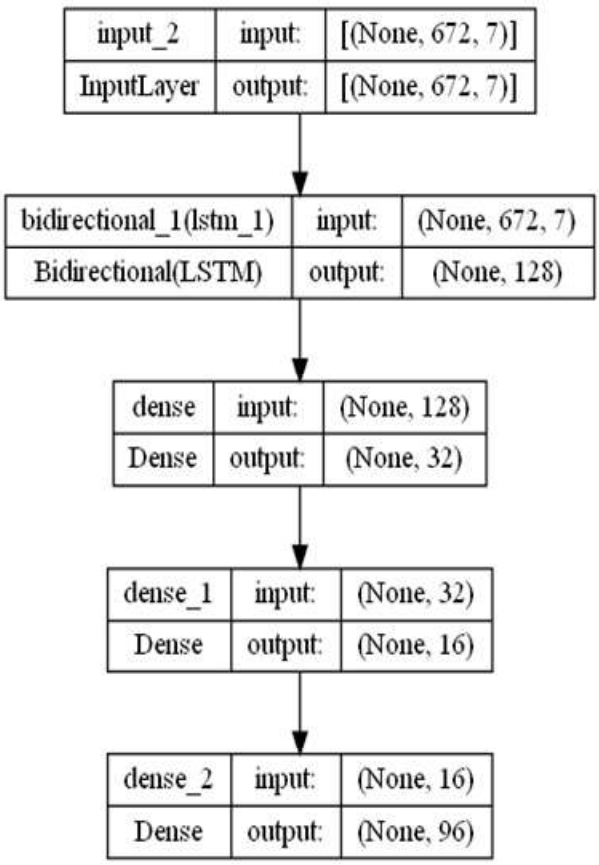
[모델 1]

R2_score : 0.84689597771



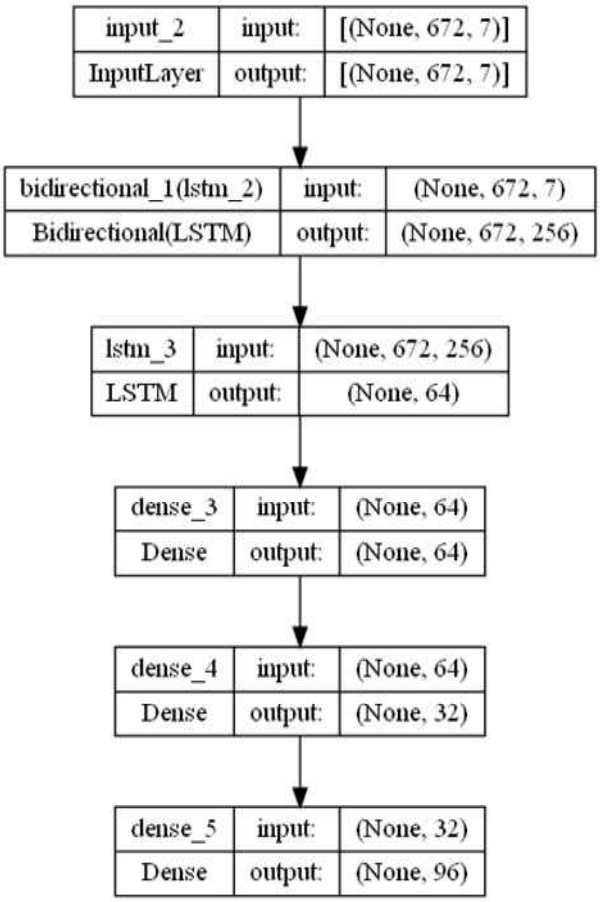
[모델 2]

R2_score : 0.70190340726



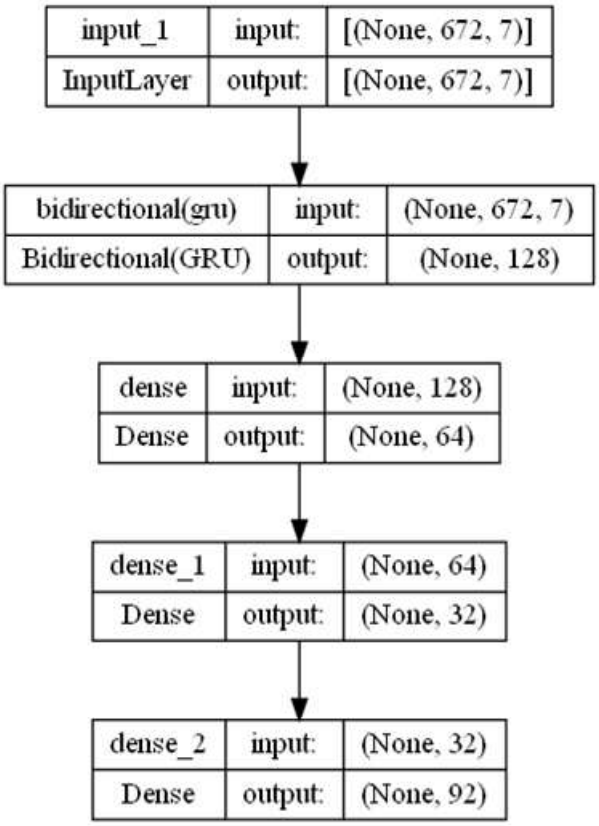
[모델 3]

R2_score : 0.81067882350



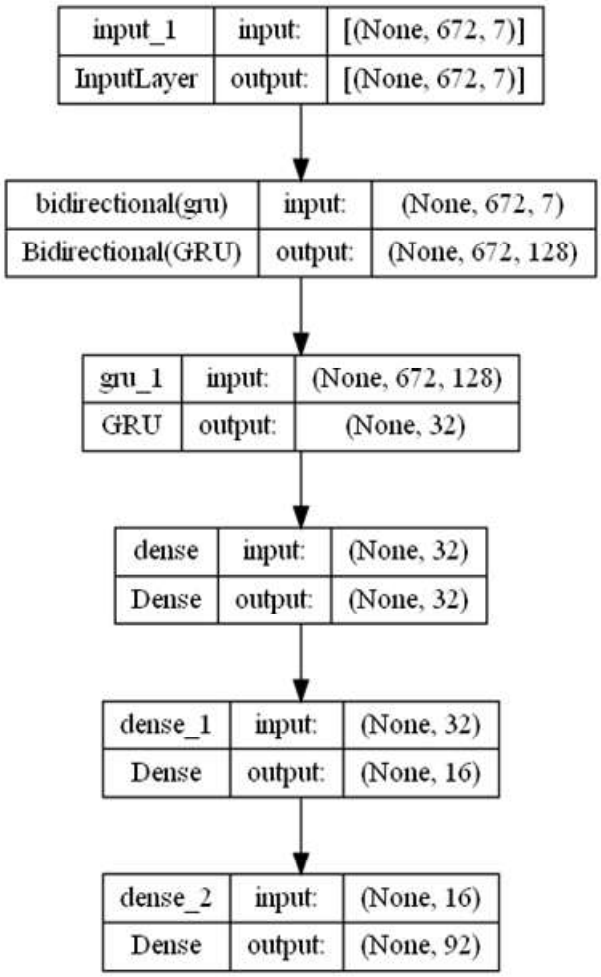
[모델 4]

R2_score : 0.87870106111



[모델 5]

R2_score : 0.76268254451



[모델 6]

R2_score : 0.86148043040

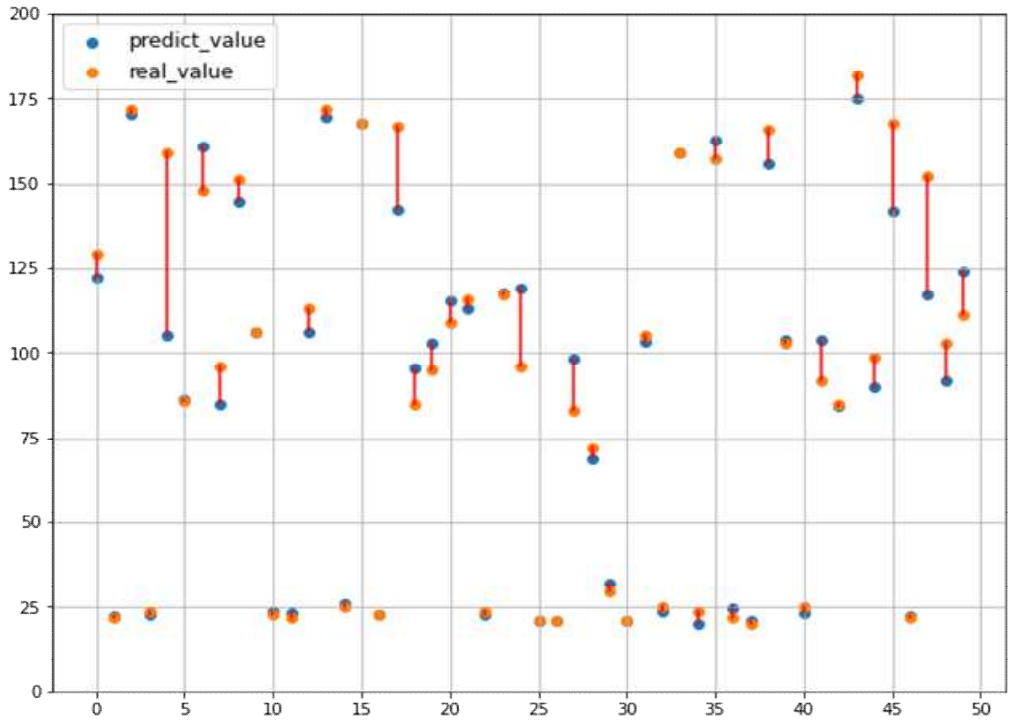
결과 : 양방향 순환신경망 구성이 보다 좋을 성능을 보여주었고, 순환신경망을 2개층으로 쌓았을 때 더 나은 성능을 보여준다.

LSTM과 GRU는 크게 성능차이가 나지 않았으나 LSTM이 조금 더 좋은 성능을 보여주었기 때문에 양방향 LSTM층과 LSTM층을 쌓은 '모델 4번'을 이용하여 튜닝해 본다.

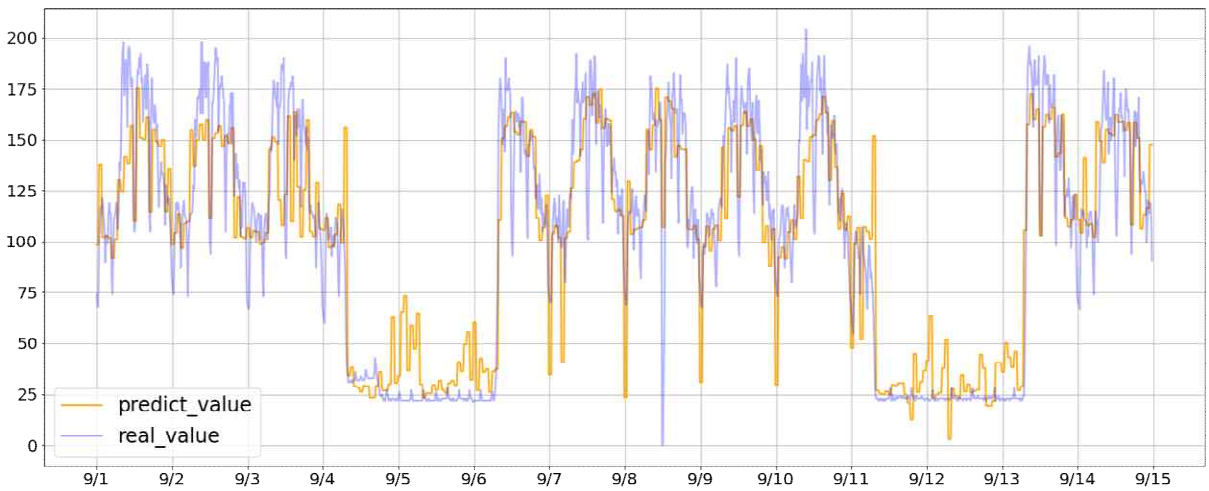
5. 모델 평가

5.1 머신러닝 모델

먼저 하나의 데이터 포인트에 대해 예측한 머신러닝기반 모델의 R2 점수와 테스트셋을 예측한 결과를 확인한다. 또한 9월 데이터를 테스트셋으로 분리하여 평가한 모델의 R2점수, 그리고 테스트셋(9월의 데이터)을 예측한 결과를 확인한다.



[무작위 데이터포인트를 예측한 결과]
R2_score : 0.93874121864

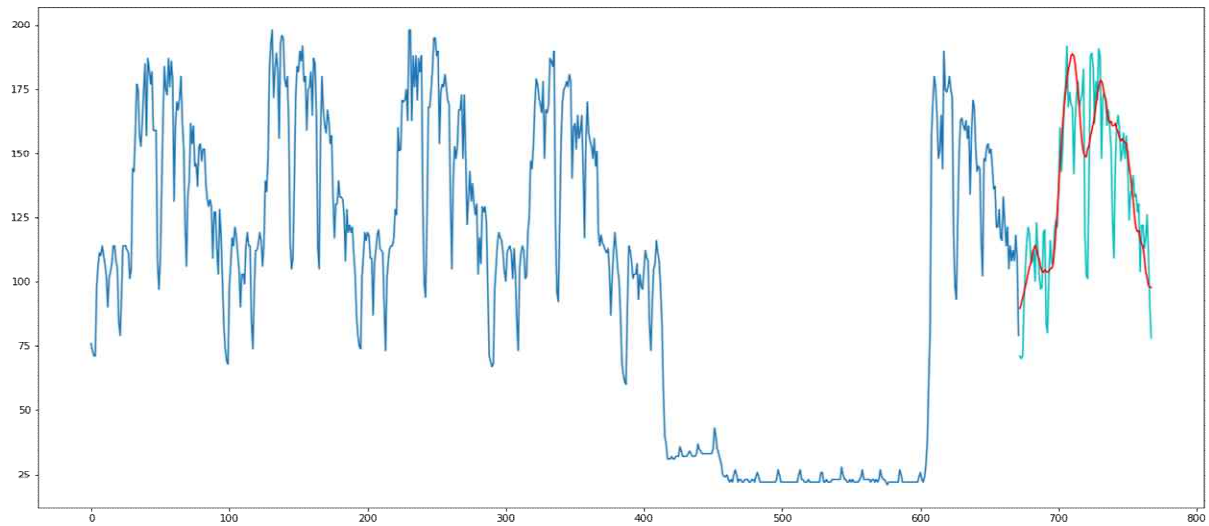


[9월의 데이터포인트를 예측한 결과]
R2_score : 0.82177097191

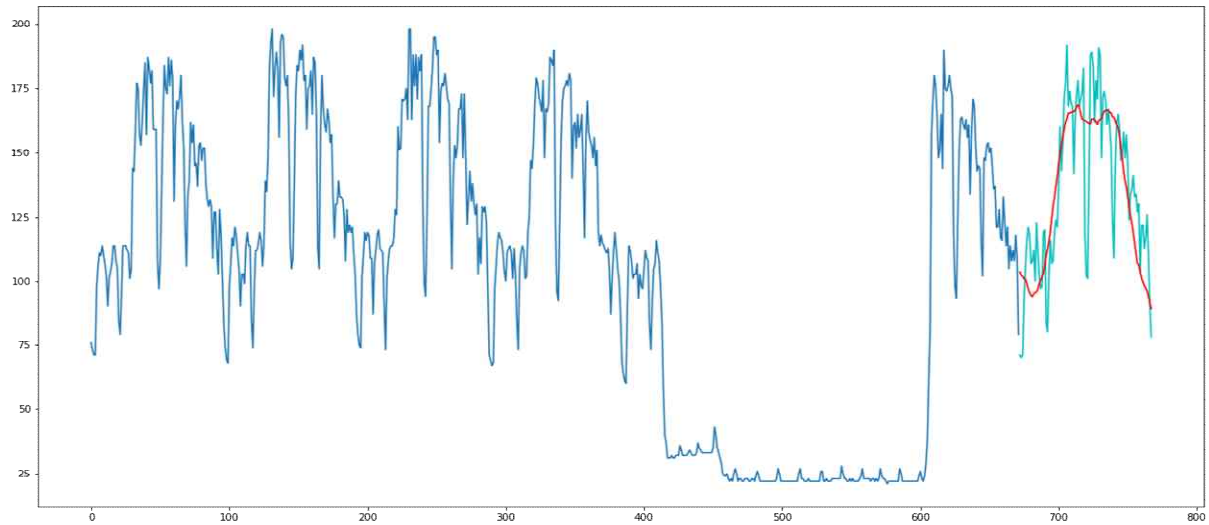
각 데이터를 포인트별로 예측한 결과에서 의미 있는 평가지표를 확인할 수 있었으나, 최초 목표였던 9월 데이터를 모아둔 테스트셋에서는 평가지표가 하락하였다. 또한 전력량의 등락이 심한 구간과 최대 피크가 등장하는 구간에서 예측을 잘 못하는 것으로 파악된다. 프로젝트의 목표가 전력 최대 피크를 잘 예측하는 것이기에 해당 머신러닝 모델로 목표를 달성하기에는 부족하다 할 수 있다.

5.2 딥러닝(순환신경망) 모델

구성한 6개의 모델 가운데 높은 지표를 보인 2개의 모델에 대해서 9월의 특정한 하루 데이터를 예측한 결과를 확인해 본다. 각 모델은 양방향 LSTM과 LSTM층을 가진 신경망 구조(모델4)와 양방향 GRU와 GRU층을 가진 신경망 구조(모델6)이다. 두 모델을 이용하여 변화가 가장 두드러지게 나타난 9월 6일의 데이터를 통해 확인한다.

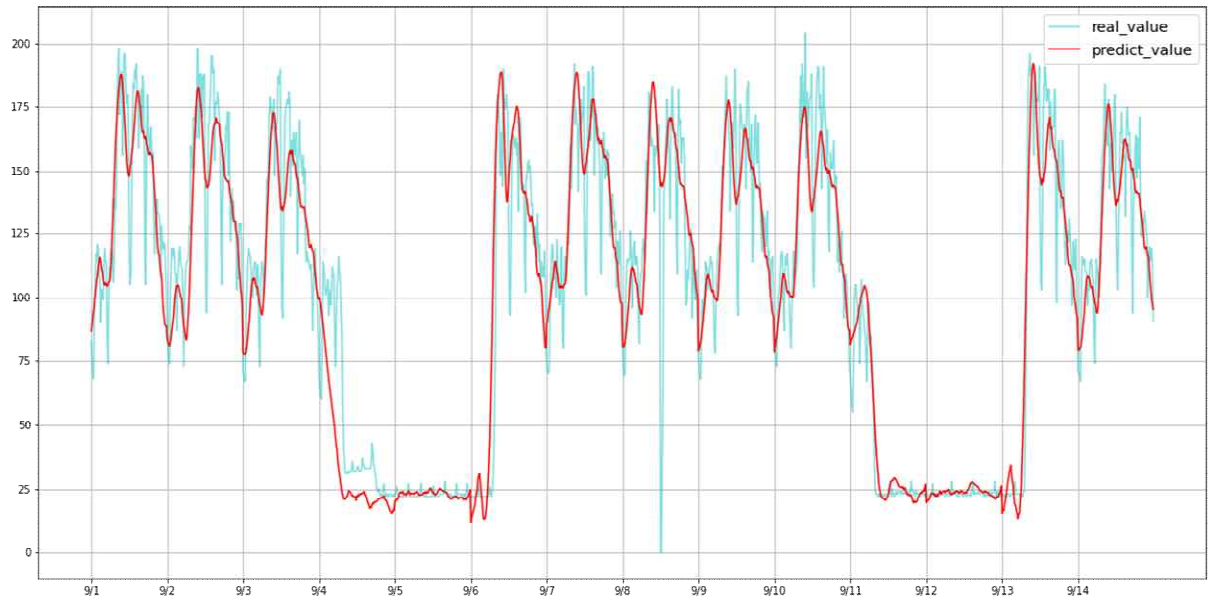


[모델 4]



[모델 6]

모델4가 피크전력 변화추세를 더 잘 나타내고 있다.
모델4를 이용한 9월 전체 데이터 예측 자료를 시각화한다.



해당 모델은 변화추세를 잘 예측하는 것으로 보이지만 프로젝트 목표에 필요한 최대 피크전력이 나타나는 포인트에서 실제 값보다 더 낮게 예측하는 경향이 보인다.
해당 모델 이외에도 다른 모델에서도 전반적으로 이러한 경향을 보였으므로 이를 기존 R2 점수로 평가하는 것보다 더 적합한 평가지표를 찾아 분석할 필요가 있다.

5.3 평가지표 설정

새로운 모델 평가지표를 고려해본다. 각 1일 단위 예측 결과에서 실제 피크 지점과 예측한 피크 지점의 시간차이를 이용하여 지표로 나타낸다. 해당 지표는 피크지점이 어느 시점에 나타나는지에 대한 예측을 정교하게 하는 데에 이용할 수 있다.
또 다른 지표는 피크 지점의 실제값과 그 지점에서 예측한 값의 차이, 혹은 구간내의 실제 피크전력과 예측한 피크전력의 차이를 이용해 측정할 수 있다. 이는 최대피크전력을 예측하여 실제 하루단위의 전력소요를 예측하는 것에 이용할 수 있다.
위의 2가지 평가지표를 이용해서 만들어둔 6개의 모델에 대한 점수를 산출한다.

	time_gap	value_gap
model1	16.615385	18.925923
model2	19.923077	24.487587
model3	17.538462	22.070190
model4	10.000000	8.914862
model5	20.145752	23.175115
model6	12.769231	10.568480

단위
time_gap (15분)
value_gap (kW)

새롭게 정의한 평가지표에서도 모델4의 지표가 다른 모델에 비해 더 우수하다. 피크전력이 나타나는 지점을 예측하는 데에는 약 2시간 30분의 오차가 나타나고, 최대 피크전력은 8.9 kW의 차이를 보인다.

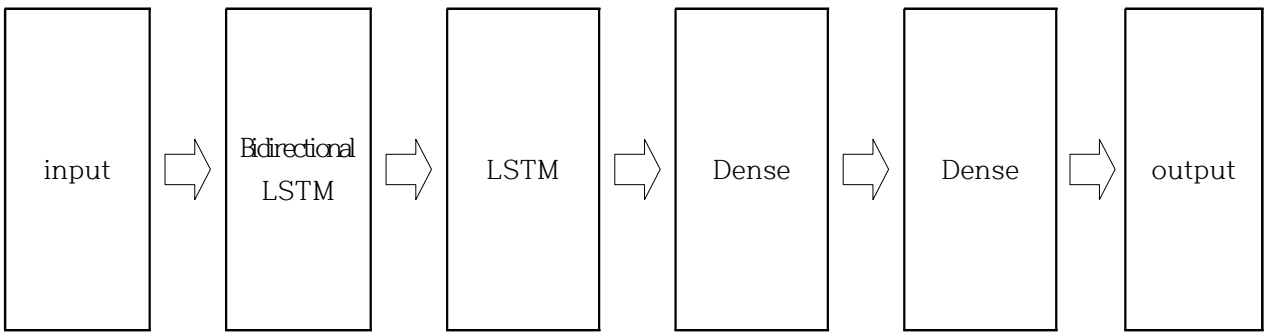
6. 결론

6.1 최종모델 선정기준

제조공정 데이터를 통해 전력 사용량을 예측하는 모델 구현을 시도하였다. 최종모델을 선정하는 기준은 '전력 사용량 예측'과 '피크전력 등장 시간 예측', 총 2가지 기준이며 이를 통해 가장 정확한 모델을 최종모델로 선정하였다.

6.2 최종모델

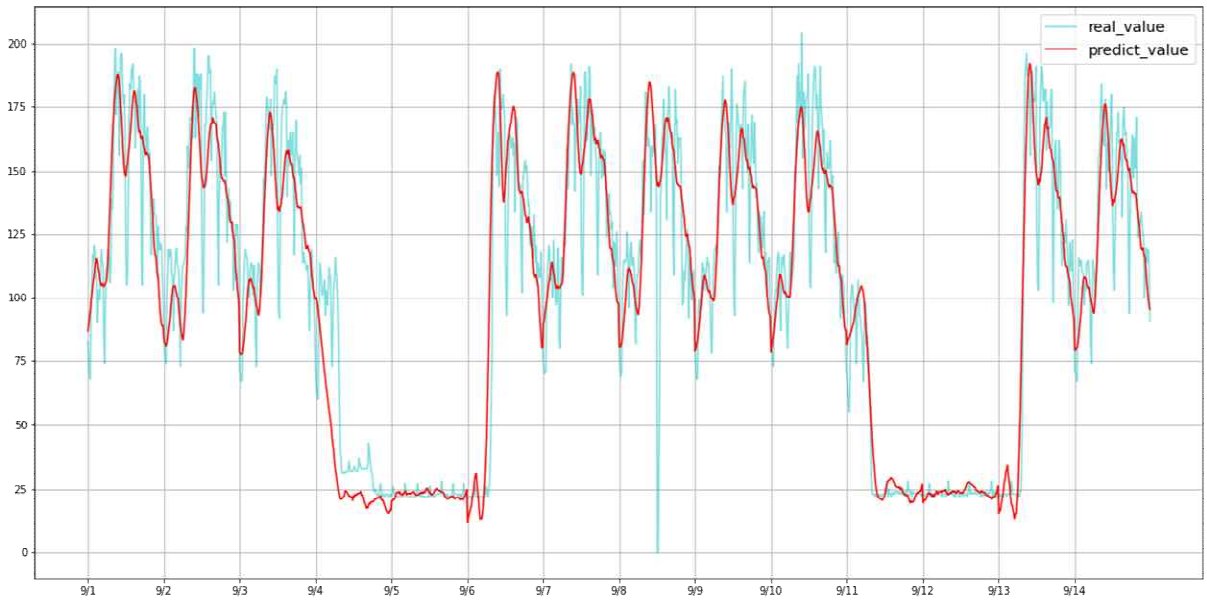
최종모델은 양방향 LSTM층과 LSTM층을 쌓은 신경망 구조(모델4)를 채택하였다.
해당 신경망 구조는 아래와 같다.



평가지표	
R2	0.878701
time_gap	10.000000
value_gap	8.914862

왼쪽의 지표를 통해 해당 모델의 R2 점수, time_gap 점수, value_gap 점수를 확인할 수 있다.

아래 그래프는 실제값과 예측값의 추세를 나타낸다.



6.3 최종모델의 기대효과

최종모델을 통해 다음날의 전력 피크가 나타나는 지점을 예측할 수 있다. 최대 피크 전력을 초과하는 지점이 예측된다면, 그 구간의 작업량을 분산하여 최대 피크 전력이 상향되지 않도록 조정할 수 있는 근거로 사용할 수 있다.

또한 하루 전체의 전력 소요량을 예측하여 전력 공급수준에 대비할 수 있다. 이를 향후 ESS 시스템과 연동한다면, 전력 사용이 적은 지점에서는 에너지를 저장하고 늘어나는 지점에서는 저장된 전력을 끌어와서 사용하는 등의 효율적 전력 환경을 조성을 기대할 수 있다.