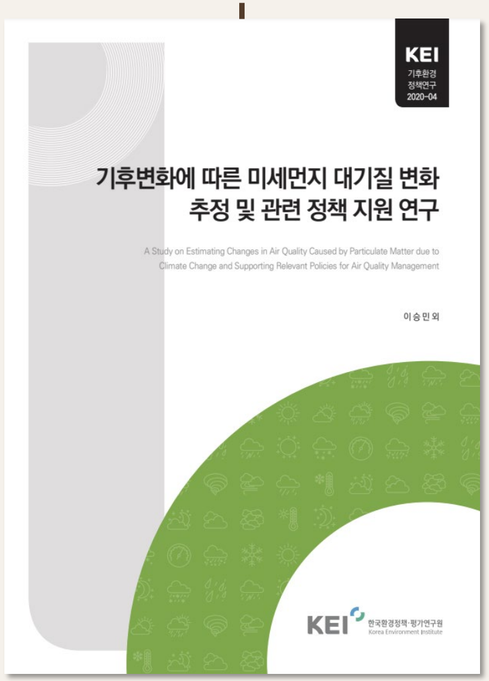
문제 해결 프로그래밍

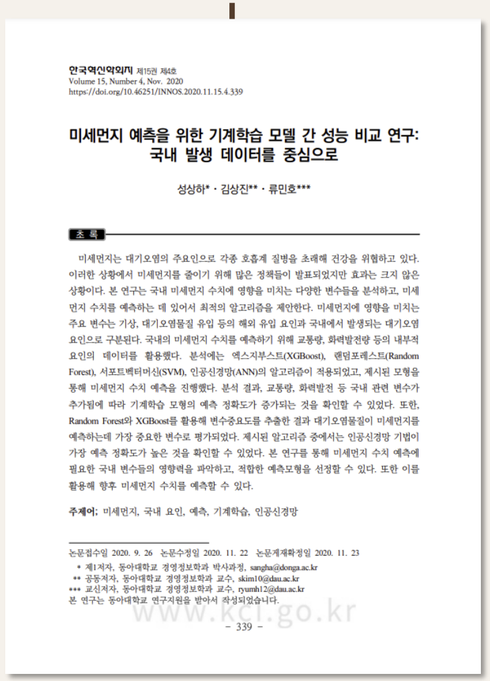
팀 프로젝트: 미세먼지 농도 예측

6조 | 김세현, 김나연, 천재민, 지호준

목차

변수 선정





한국액신학회지 제15권 제4호
Volume 15, Number 4, Nov. 2020
https://doi.org/10.46251/JNNOS.2020.11.15.4.339

미세먼지 예측을 위한 기계학습 모델 간 성능 비교 연구: 국내 발생 데이터를 중심으로

성상하* · 김상진** · 류민호***

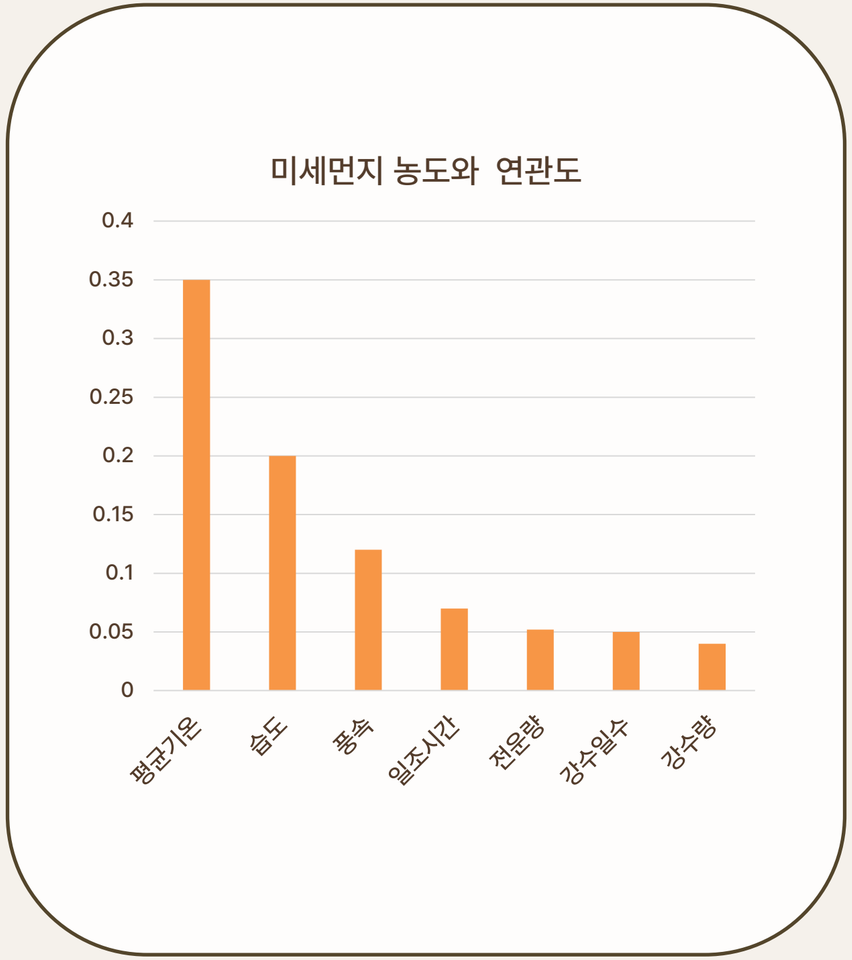
초 록

미세먼지는 대기오염의 주요인으로 각종 호흡계 질병을 초래해 건강을 위협하고 있다. 이러한 상황에서 미세먼지를 줄이기 위해 많은 정책들이 발표되었지만 효과는 크지 않은 상황이다. 본 연구는 국내 미세먼지 수치에 영향을 미치는 다양한 변수들을 분석하고, 미세먼지 수치를 예측하는 데 있어서 최적의 알고리즘을 제안한다. 미세먼지에 영향을 미치는 주요 변수는 기상, 대기오염물질 유입 등의 해외 유입 요인과 국내에서 발생되는 대기오염 요인으로 구분된다. 국내의 미세먼지 수치를 예측하기 위해 교통량, 화력발전량 등의 내부적 요인의 데이터를 활용했다. 분석에는 엑스지부스트(XGBoost), 랜덤포레스트(Random Forest), 서포트벡터머신(SVM), 인공신경망(ANN)의 알고리즘이 적용되었고, 제시된 모형을 통해 미세먼지 수치 예측을 진행했다. 분석 결과, 교통량, 화력발전 등 국내 관련 변수가 추가됨에 따라 기계학습 모형의 예측 정확도가 증가되는 것을 확인할 수 있었다. 또한, Random Forest와 XGBoost를 활용해 변수중요도를 추출한 결과 대기오염물질이 미세먼지를 예측하는데 가장 중요한 변수로 평가되었다. 제시된 알고리즘 중에서는 인공신경망 기법이 가장 예측 정확도가 높은 것을 확인할 수 있었다. 본 연구를 통해 미세먼지 수치 예측에 필요한 국내 변수들의 영향력을 파악하고, 적합한 예측모형을 선정할 수 있다. 또한 이를 활용해 향후 미세먼지 수치를 예측할 수 있다.

주제어: 미세먼지, 국내 요인, 예측, 기계학습, 인공신경망

논문접수일 2020. 9. 26 논문수정일 2020. 11. 22 논문게재확정일 2020. 11. 23

* 제1저자, 동아대학교 경영정보학과 박사과정, sangha@donga.ac.kr
** 공동저자, 동아대학교 경영정보학과 교수, skim10@dau.ac.kr
*** 교신저자, 동아대학교 경영정보학과 교수, ryumh12@dau.ac.kr
본 연구는 동아대학교 연구지원을 받아서 작성되었습니다.



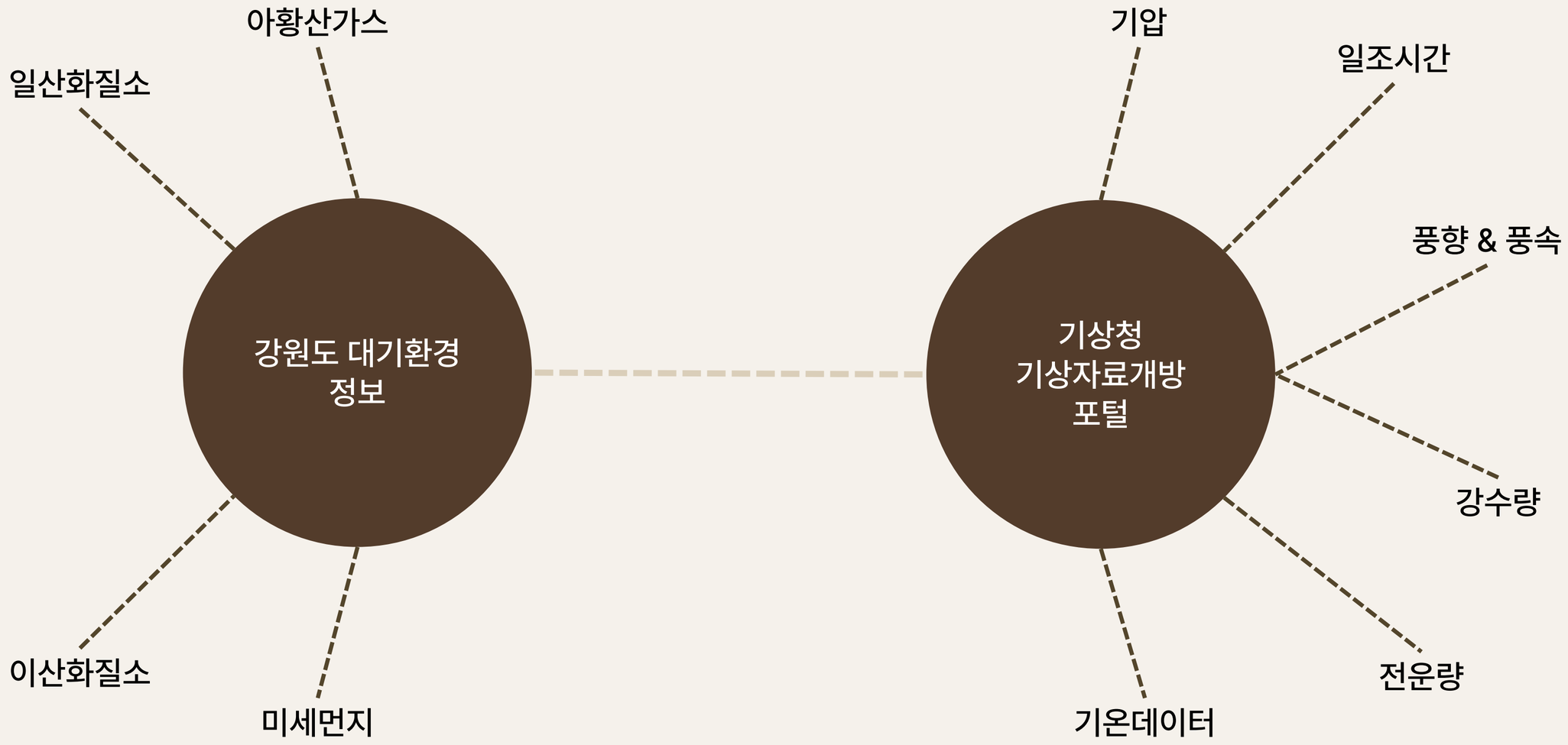
자료 기후변화에 따른 미세먼지 대기질 변화 추정 및 관련 정책 지원 연구, 이승민

자료 미세먼지 예측을 위한 기계학습 모델 간 성능 비교 연구: 국내 발생 데이터를 중심으로, 성상하

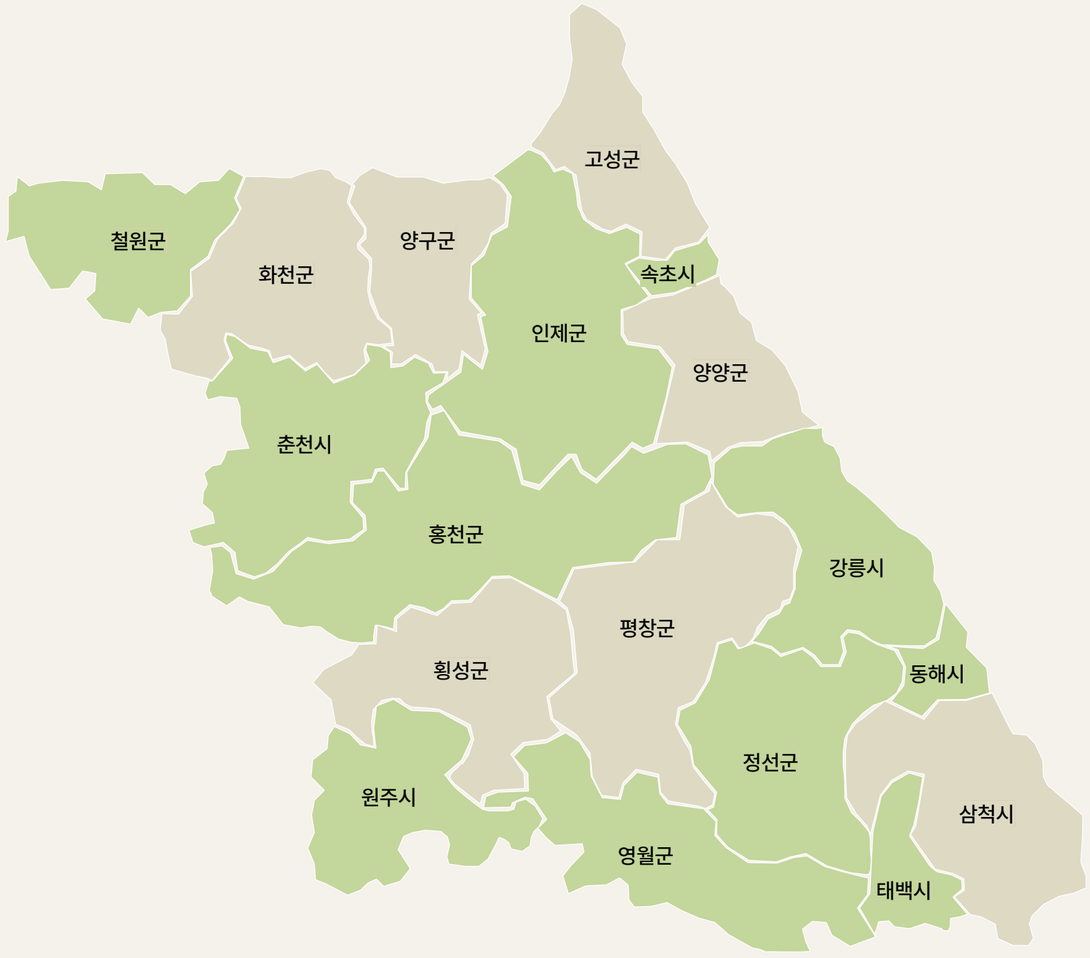
변수

강수량	평균전운량	기압	아황산가스
이산화질소	일산화질소	평균기온	최저기온
합계 일조시간	평균지면온도	평균풍속	최대풍속
□□□□□□	□□□□□□	□□□□□□□□	

변수 출처


아황산가스
일산화질소
강원도 대기환경
정보
이산화질소
미세먼지
기압
일조시간
풍향 & 풍속
기상청
기상자료개방
포털
강수량
전운량
기온데이터

지역 & 기간



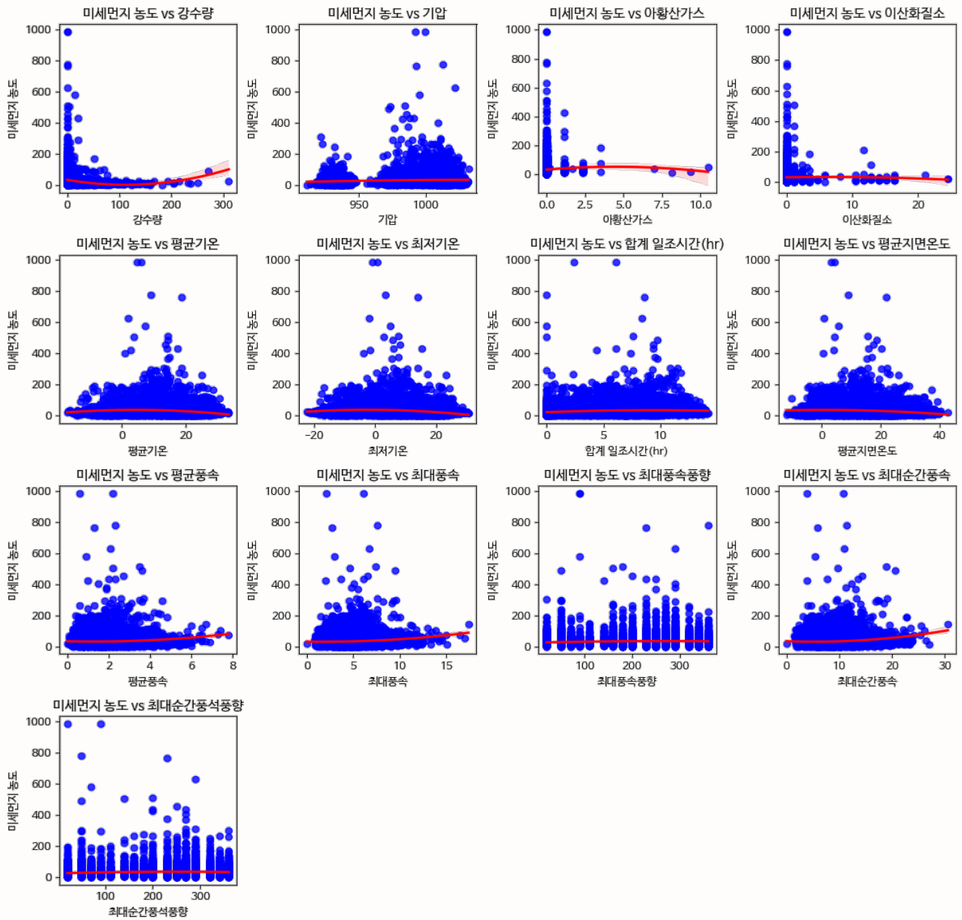
수집 지역

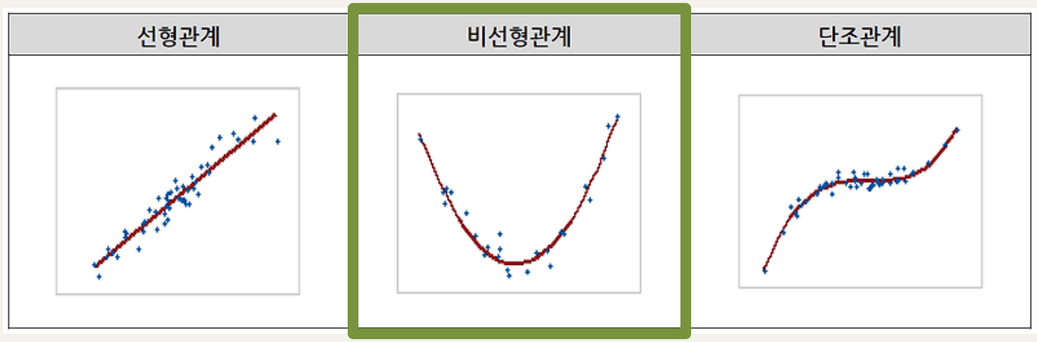
강릉시, 동해시, 속초시, 영월군, 원주시,
인제군, 정선군, 철원군, 춘천시, 태백시, 홍천군

수집 기간

2019년 1월 ~ 2023년 12월 (일 단위)

전처리 : 데이터 시각화





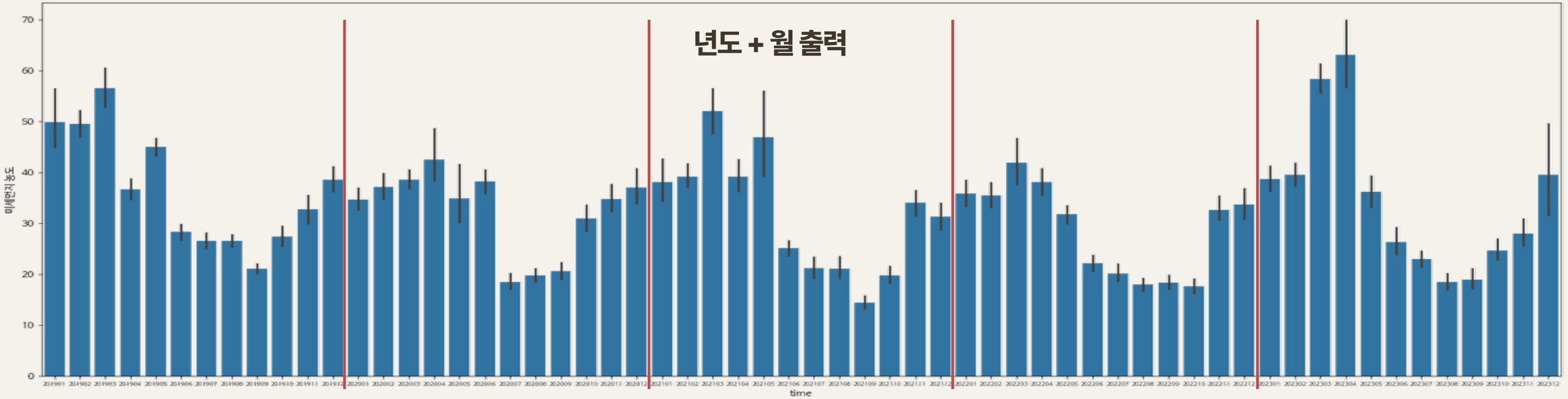
산점도와 다항 회귀선을 그려본 결과

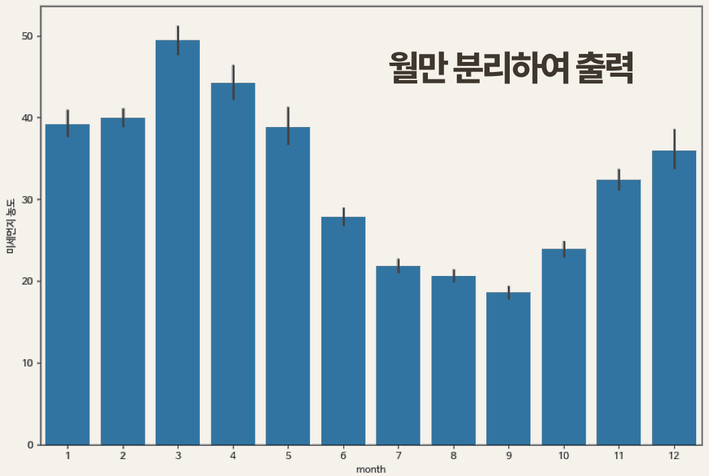
종속변수인 미세먼지 농도와 비교하여

강수량, 평균/최저 기온, 평균 지면온도,

평균/최대/순간 풍속에서 비선형 관계 형성

전처리 : 칼럼 추가





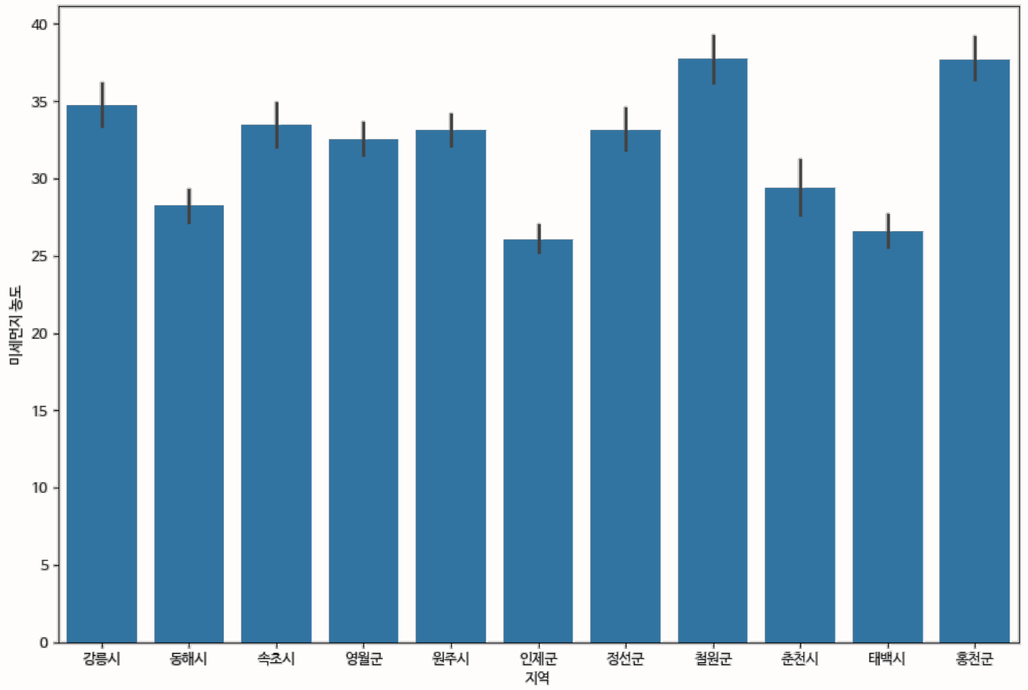
데이터 셋에서 연도+월 칼럼 생성 후 시간순으로 출력한 결과

19년~23년 전부 비슷한 형태

종속변수(y)에 월 값 또한 유효한 의미와 패턴 보유한 걸로 파악되어 추가

전처리 : 칼럼 추가

지역별 전체 평균 미세먼지 농도

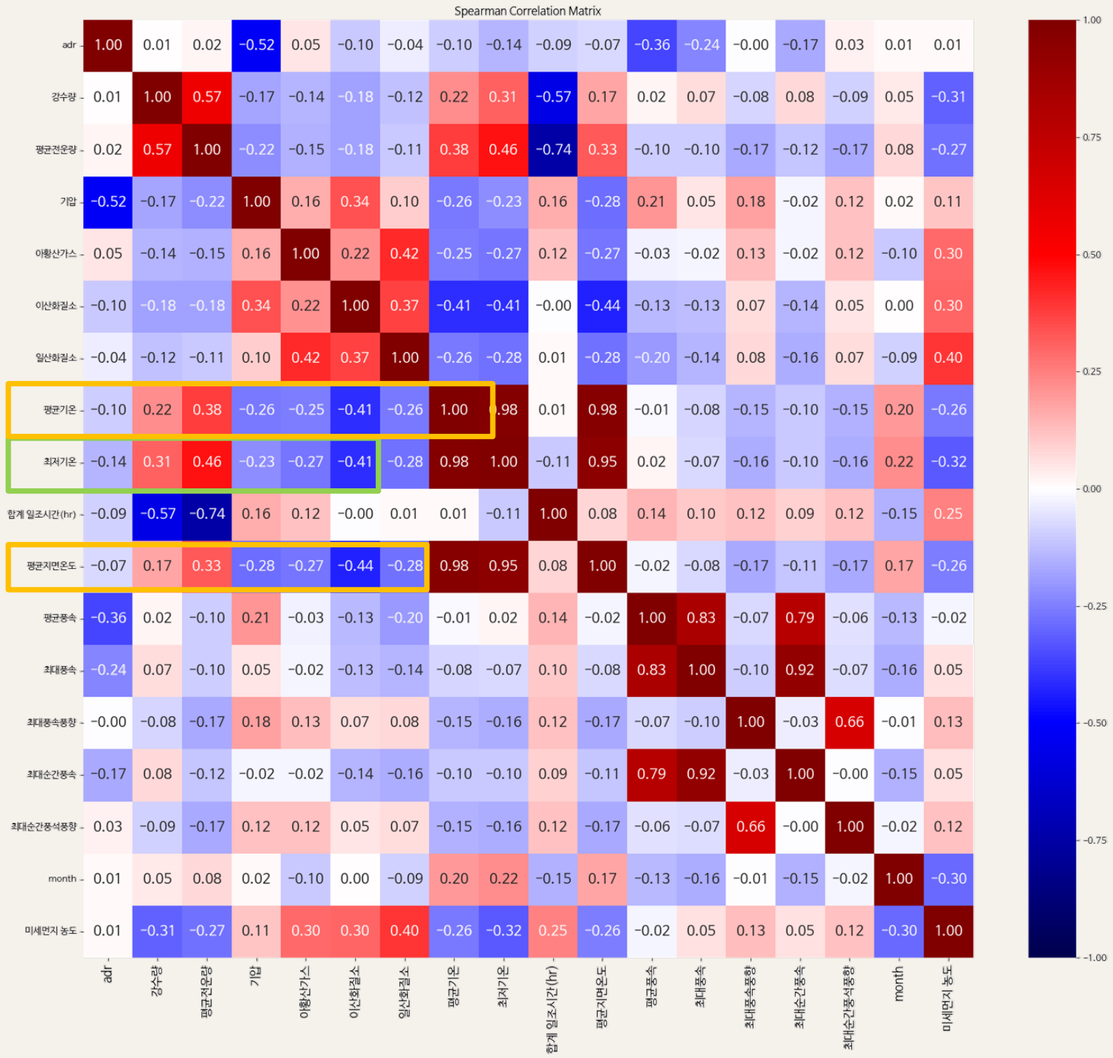


11개 지역의 데이터를 수집, 각 지역마다의 평균 미세먼지 농도 시각화

결과적으로 +/- 10 정도 차이가 발생
때문에 모델이 지역을 구분하여 학습 가능하도록
지역 숫자로 코드화 후 데이터에 추가

강릉시 => 1
동해시 => 2
속초시 => 3
영월군 => 4
원주시 => 5
인제군 => 6
정선군 => 7
철원군 => 8
춘천시 => 9
태백시 => 10
홍천군 => 11

전처리 : 상관관계



Spearman Rank Correlation Coefficient

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

데이터의 순위를 사용하므로 비선형이거나 비정규 분포를 가질 때에도 적용 가능한 **스피어만 상관계수**를 이용해서 상관관계 전처리 진행.

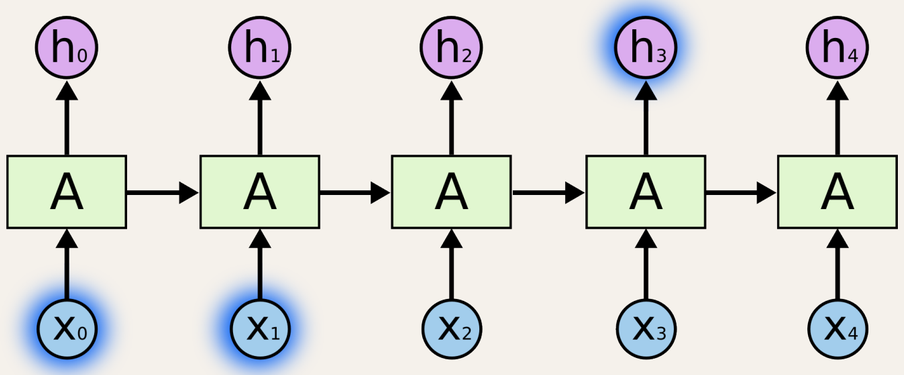
▼

평균기온, 평균지면온도 제거

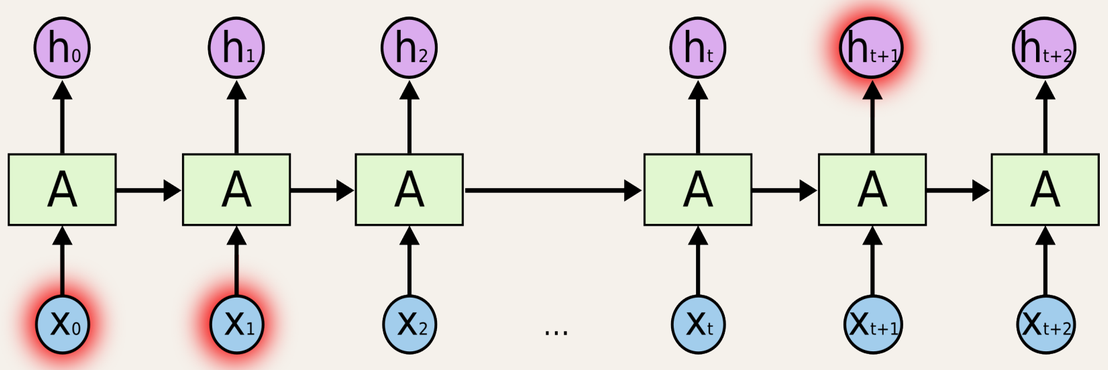
LSTM

Long Short-Term Memory

R N N



L S T M



분석 : LSTM

Long Short-Term Memory

```
data_cleaned = data_cleaned.sort_values(by=['일자'], ascending=True)
```

▼

	지역	일자	강수량	평균전운량	기압	아황산가스	이산화질소	일산화질소	평균기온	최저기온	...	평균지면온도	평균풍속	최대풍속	최대풍속풍향	최대순간풍속	최대순간풍속풍향	미세먼지	농도	year	month	adr
14447	동해시	2019-01-01	0.0	2.4	1023.5	0.002	0.016	0.3	0.0	-3.2	...	-3.3	2.0	4.9	320.0	8.8	340.0	19.0	2019	01	2
11258	태백시	2019-01-01	0.0	4.3	941.4	0.000	0.007	0.0	-7.1	-11.4	...	-7.1	1.6	3.9	360.0	9.6	340.0	0.0	2019	01	10
12372	철원군	2019-01-01	0.0	3.6	1013.8	0.004	0.009	0.6	-9.2	-16.5	...	-8.0	1.0	3.6	200.0	6.5	230.0	37.0	2019	01	8
12492	속초시	2019-01-01	0.0	3.5	1026.3	0.005	0.005	0.5	-1.5	-5.8	...	-3.1	1.7	5.9	290.0	10.9	290.0	18.0	2019	01	3
14187	홍천군	2019-01-01	0.0	2.6	1015.2	0.000	0.021	0.0	-7.8	-14.4	...	-6.3	1.0	4.0	270.0	6.1	290.0	0.0	2019	01	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
12848	철원군	2023-12-31	2.7	3.4	999.8	0.003	0.008	0.4	1.0	-2.8	...	0.1	0.9	4.3	340.0	8.4	340.0	17.0	2023	12	8
9883	태백시	2023-12-31	2.5	5.0	940.6	0.002	0.010	0.7	0.3	-1.7	...	-0.8	1.3	4.1	20.0	7.1	360.0	20.0	2023	12	10
855	강릉시	2023-12-31	14.7	9.5	1015.0	0.003	0.011	0.5	3.0	2.1	...	1.0	2.2	5.5	320.0	10.5	340.0	11.0	2023	12	1
8407	춘천시	2023-12-31	5.3	5.8	1018.1	0.003	0.029	0.7	1.7	-0.2	...	-0.1	0.4	1.9	250.0	2.8	110.0	27.0	2023	12	9
10268	원주시	2023-12-31	2.7	4.8	1009.1	0.004	0.023	0.6	2.3	1.5	...	-0.1	0.4	1.6	270.0	3.5	250.0	26.0	2023	12	5

18190 rows × 21 columns

분석 : LSTM 설정

Long Short-Term Memory

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

# 데이터 로드 (data_cleaned가 정의되어야 함)
# 예시: data_cleaned = pd.read_csv("path_to_your_data.csv")

# 독립 변수와 종속 변수 선택
df = data_cleaned[['adr', '강수량', '평균전운량', '기압', '아황산가스', '이산화질소', '일산화질소', '최저기온', '합계 일조시간(hr)', '평균풍속', '최대풍속', '최대풍속풍향', '최대순간풍속','최대순간풍석풍향','month']]
y = data_cleaned['미세먼지 농도']

# 독립 변수와 종속 변수 각각에 대해 StandardScaler 적용
X_scaler = StandardScaler()  # 독립 변수용 스케일러
y_scaler = StandardScaler()  # 종속 변수용 스케일러

scaled_X = X_scaler.fit_transform(df)  # 독립 변수 스케일링
scaled_y = y_scaler.fit_transform(y.values.reshape(-1, 1))  # 종속 변수 스케일링

# 시퀀스 데이터 생성 함수
def create_sequences(data, target, seq_length):
    X = []
    y = []
    for i in range(len(data) - seq_length):
        X.append(data[i:i + seq_length])
        y.append(target[i + seq_length])
    return np.array(X), np.array(y)

seq_length = 30  # 시퀀스 길이
X, y = create_sequences(scaled_X, scaled_y, seq_length)

# 학습 데이터와 테스트 데이터 분리
train_size = int(len(X) * 0.75)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
```

데이터 비율

Train : 75%

Test : 25%

시퀀스 길이 : 30

모델이 한 번에 처리하는 입력

데이터의 시간 단계

분석 : LSTM 설정

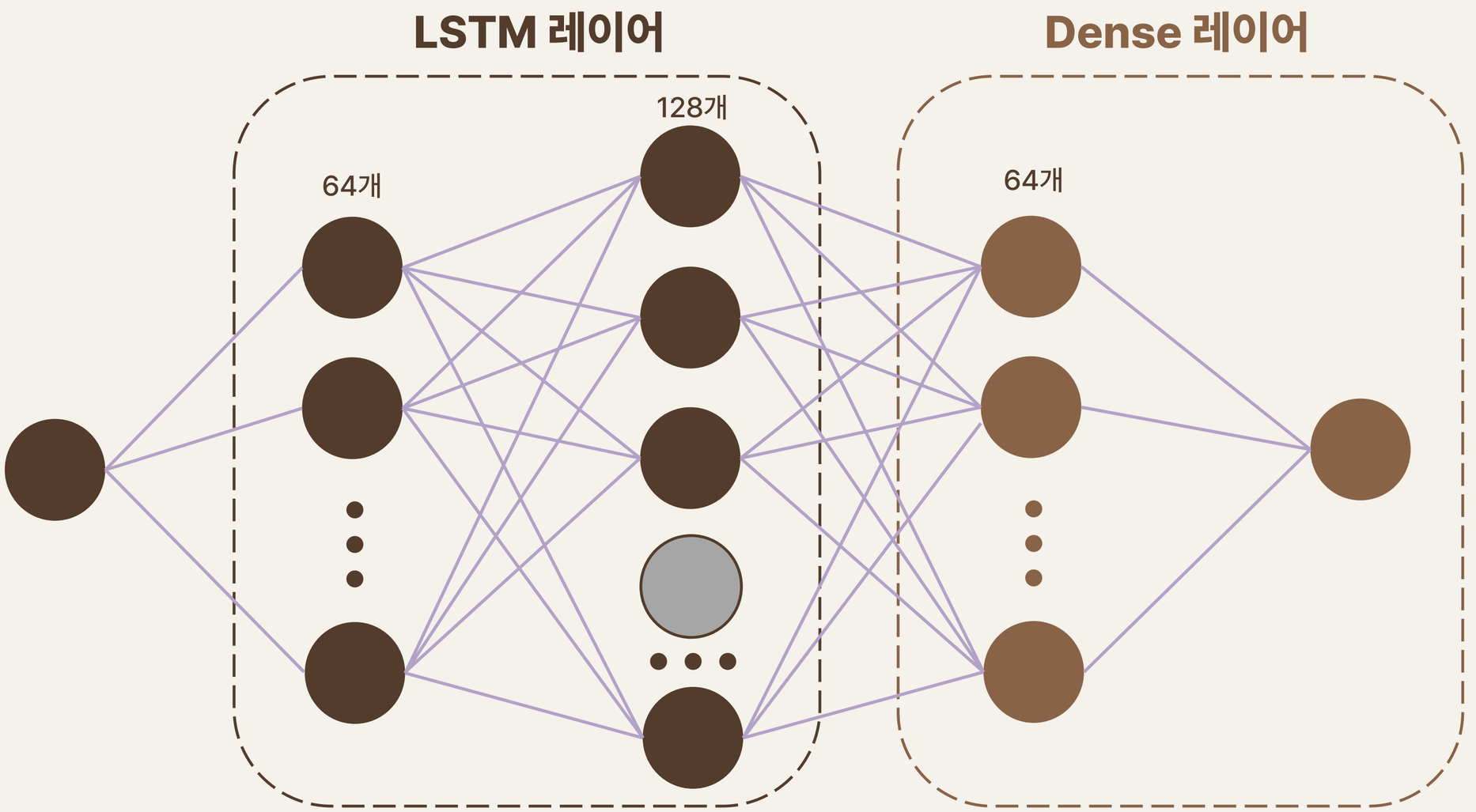
Long Short-Term Memory

```
# LSTM 모델 정의
model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(seq_length, X.shape[2]), dropout=0.2, recurrent_dropout=0.2),
    LSTM(128, dropout=0.3, recurrent_dropout=0.2),
    Dense(64, activation='relu'),
    Dense(1)
])

# 모델 컴파일
model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')

# 모델 학습
history = model.fit(X_train, y_train, epochs=60, batch_size=80, validation_split=0.2, shuffle=False)

# 모델 평가 및 예측
test_loss = model.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss}")

y_pred = model.predict(X_test)

# 정규화 복원
y_test_rescaled = y_scaler.inverse_transform(y_test.reshape(-1, 1))
y_pred_rescaled = y_scaler.inverse_transform(y_pred)


# 시각화
plt.figure(figsize=(12, 6))
plt.plot(y_test_rescaled, label='Actual PM10')
plt.plot(y_pred_rescaled, label='Predicted PM10')
plt.legend()
plt.title("Actual vs Predicted PM10 Concentration")
plt.xlabel("Time Steps")
plt.ylabel("PM10 Concentration")
plt.show()
```

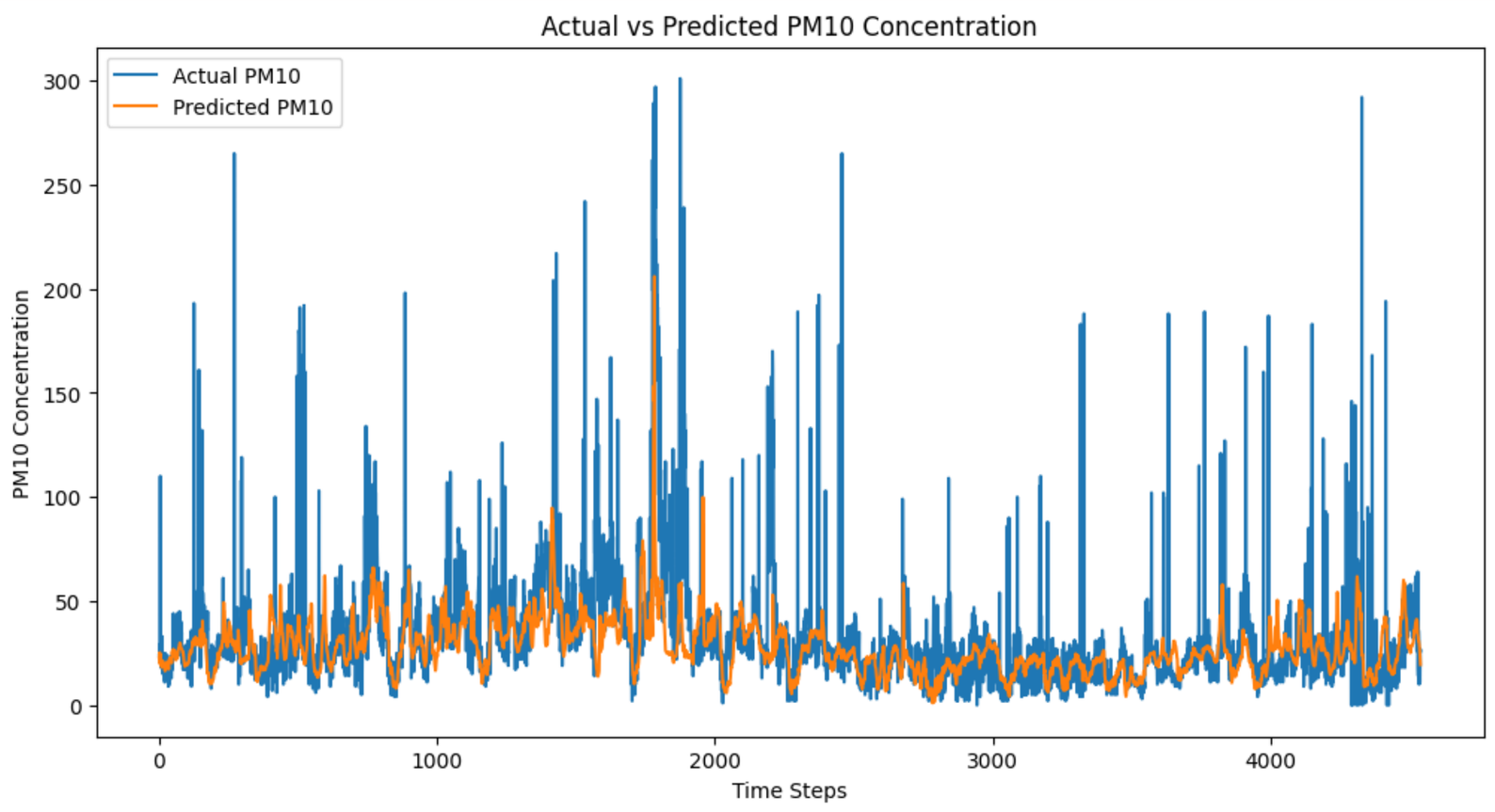
분석 : LSTM

Long Short-Term Memory

```
LSTM(64, return_sequences=True, input_shape=(30, 10)), # 첫 번째 LSTM: 64 유닛
LSTM(128, dropout=0.3), # 두 번째 LSTM: 128 유닛 , 30% 확률
Dense(64, activation='relu'), # Dense 레이어: 64 유닛
Dense(1) # 최종 출력
```



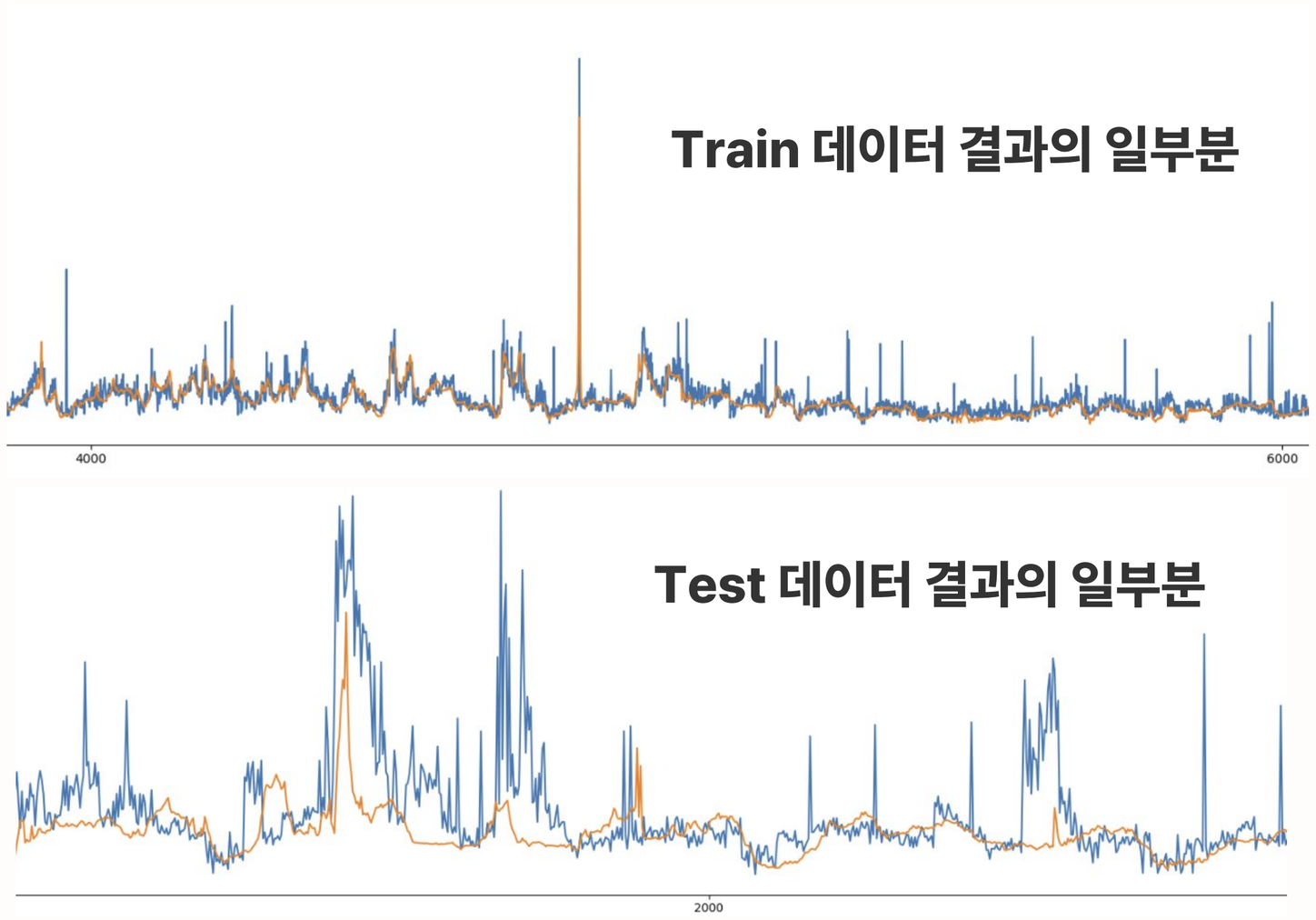
분석결과



학습된 모델에 Test 데이터 적용

극단적인 값은 예측하지 못하지만 일반적인 데이터들의 경우 잘 예측되고 있음.

분석결과



Train 데이터 결과와
Test 데이터 결과를 비교

분석결과

학습한 모델의 신뢰도

```
426/426 [==============================] - 5s 12ms/step
142/142 [==============================] - 2s 12ms/step
Train RMSE: 1.2615350865983082
Test RMSE: 1.2582707471386954
142/142 [==============================] - 2s 12ms/step
Mean Absolute Error: 0.5371960179178 18
R2 Score: 0.1695729075689698
```

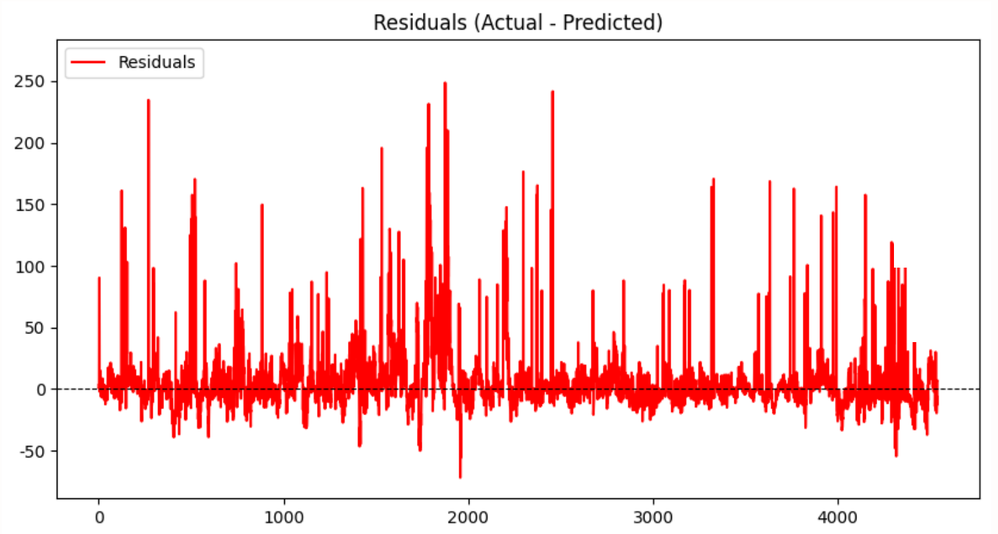
Train RMSE : 1.262

Test RMSE : 1.258

R^2 Score : 0.17

예측결과의 잔차

(실제 값 - 예측 값)



분석 개선방안

- 기간을 더 길게 설정하여 더 많은 데이터로 모델학습
- 유닛 수 최적화 후 모델학습
- LSTM 모델의 복잡도 증사설정

고성능의 컴퓨터 요구

감사합니다.