```
import pandas as pd

# Load the CSV file
file_path = "/content/traffic_data_dyn (1).csv"
df = pd.read_csv(file_path)

# Display basic info
print(df.info())  # Check column names and data types
print(df.head())  # Preview first few rows
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86635 entries, 0 to 86634
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Time                         86635 non-null  float64
 1   Vehicle Count at Junction A  86635 non-null  int64
 2   Vehicle Count at Junction B  86635 non-null  int64
 3   Vehicle Count at Junction C  86635 non-null  int64
 4   Vehicle Count at Junction D  86635 non-null  int64
 5   Vehicle Count at Junction E  86635 non-null  int64
 6   Avg Speed on AB              86635 non-null  float64
 7   Avg Speed on BC              86635 non-null  float64
 8   Avg Speed on CA              86635 non-null  float64
 9   Avg Speed on CD              86635 non-null  float64
 10  Avg Speed on ED              86635 non-null  float64
 11  Avg Speed on DA              86635 non-null  float64
dtypes: float64(7), int64(5)
memory usage: 7.9 MB
None
   Time  Vehicle Count at Junction A  Vehicle Count at Junction B
0   1.0                            0                            0
1   2.0                            0                            0
2   3.0                            0                            0
3   4.0                            0                            0
4   5.0                            0                            0

   Vehicle Count at Junction C  Vehicle Count at Junction D  \
0                            0                            0
1                            0                            0
2                            0                            0
3                            0                            0
4                            0                            0

   Vehicle Count at Junction E  Avg Speed on AB  Avg Speed on BC
0                            0              0.0         0.000000
1                            0              0.0         0.000000
2                            0              0.0         2.181660
3                            0              0.0         4.622117
4                            0              0.0         6.796626

   Avg Speed on CA  Avg Speed on CD  Avg Speed on ED  Avg Speed on
0              0.0              0.0              0.0         0.000
1              0.0              0.0              0.0         1.317
2              0.0              0.0              0.0         3.199
3              0.0              0.0              0.0         5.222
4              0.0              0.0              0.0         6.901
```

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

# Load Data
file_path = "/content/traffic_data_dyn (1).csv"
df = pd.read_csv(file_path)
```

Resources

You are not subscribed. Learn more
You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units here.
At your current usage level, this runtime may last up to 1 hour 40 minutes.

Manage sessions

Want more memory and disk space?
Upgrade to Colab Pro

Not connected to runtime.

Change runtime type

```
# Select features and target
features = [
    "Vehicle Count at Junction A", "Vehicle Count at Junction B",
    "Vehicle Count at Junction D", "Vehicle Count at Junction E",
    "Avg Speed on AB", "Avg Speed on BC", "Avg Speed on CA",
    "Avg Speed on CD", "Avg Speed on ED", "Avg Speed on DA"
]
target = "Vehicle Count at Junction C"

# Normalize data
scaler = MinMaxScaler()
df_scaled = scaler.fit_transform(df[features + [target]])

# Convert to NumPy
df_scaled = np.array(df_scaled)

# Define sequence length
SEQ_LENGTH = 5  # Reduce to save memory

# Use NumPy array slicing for efficient sequence creation
X = np.array([df_scaled[i:i+SEQ_LENGTH, :-1] for i in range(len(df_sca
y = np.array([df_scaled[i+SEQ_LENGTH, -1] for i in range(len(df_scaled

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.

print("Optimized Shape - X_train:", X_train.shape, "y_train:", y_train
```

```
Optimized Shape - X_train: (69304, 5, 10) y_train: (69304,)
```

```
import tensorflow as tf
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Activation

# Define Pelliot activation function
def pelliot_activation(x, a=0.5):
    return x / (1 + a * K.abs(x))

# Create a custom activation layer
class PelliotActivation(Activation):
    def __init__(self, activation, **kwargs):
        super(PelliotActivation, self).__init__(activation, **kwargs)
        self.__name__ = 'pelliot_activation'

# Register the activation
tf.keras.utils.get_custom_objects().update({'pelliot_activation': Pell
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

# Build the LSTM Model
model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(SEQ_LENGTH, X_train.s
    LSTM(32, return_sequences=False),
    Dense(16, activation=PelliotActivation(pelliot_activation)),  # Pe
    Dense(1)  # Output layer (vehicle count)
])

# Compile the model
model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Summary
model.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.p
  super().__init__(**kwargs)
```

Model: "sequential_1"

Layer (type)	Output Shape
lstm_2 (LSTM)	(None, 5, 64)
lstm_3 (LSTM)	(None, 32)
dense_2 (Dense)	(None, 16)
dense_3 (Dense)	(None, 1)

Total params: 32,161 (125.63 KB)
Trainable params: 32,161 (125.63 KB)

```
# Train the model
history = model.fit(X_train, y_train, epochs=200, batch_size=32, valid

# Plot training loss
import matplotlib.pyplot as plt

plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.title("Loss Curve")
plt.show()
```

```
Epoch 1/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 18s 7ms/step - loss: 0.0244 - mae:
Epoch 2/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0185 - mae:
Epoch 3/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0180 - mae:
Epoch 4/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0176 - mae:
Epoch 5/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0173 - mae:
Epoch 6/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0171 - mae:
Epoch 7/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0164 - mae:
Epoch 8/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0158 - mae:
Epoch 9/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0159 - mae:
Epoch 10/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0155 - mae:
Epoch 11/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0152 - mae:
Epoch 12/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0151 - mae:
Epoch 13/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0150 - mae:
Epoch 14/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0147 - mae:
Epoch 15/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0147 - mae:
Epoch 16/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0144 - mae:
Epoch 17/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0143 - mae:
Epoch 18/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0141 - mae:
Epoch 19/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0139 - mae:
Epoch 20/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0135 - mae:
Epoch 21/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 8ms/step - loss: 0.0135 - mae:
Epoch 22/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 8ms/step - loss: 0.0132 - mae:
Epoch 23/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0132 - mae:
Epoch 24/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0130 - mae:
Epoch 25/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0128 - mae:
Epoch 26/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 18s 8ms/step - loss: 0.0124 - mae:
Epoch 27/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0123 - mae:
Epoch 28/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 18s 7ms/step - loss: 0.0122 - mae:
Epoch 29/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0120 - mae:
Epoch 30/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 8ms/step - loss: 0.0118 - mae:
Epoch 31/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0117 - mae:
Epoch 32/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0114 - mae:
Epoch 33/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0115 - mae:
Epoch 34/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0111 - mae:
Epoch 35/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0110 - mae:
Epoch 36/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0108 - mae:
Epoch 37/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0106 - mae:
```

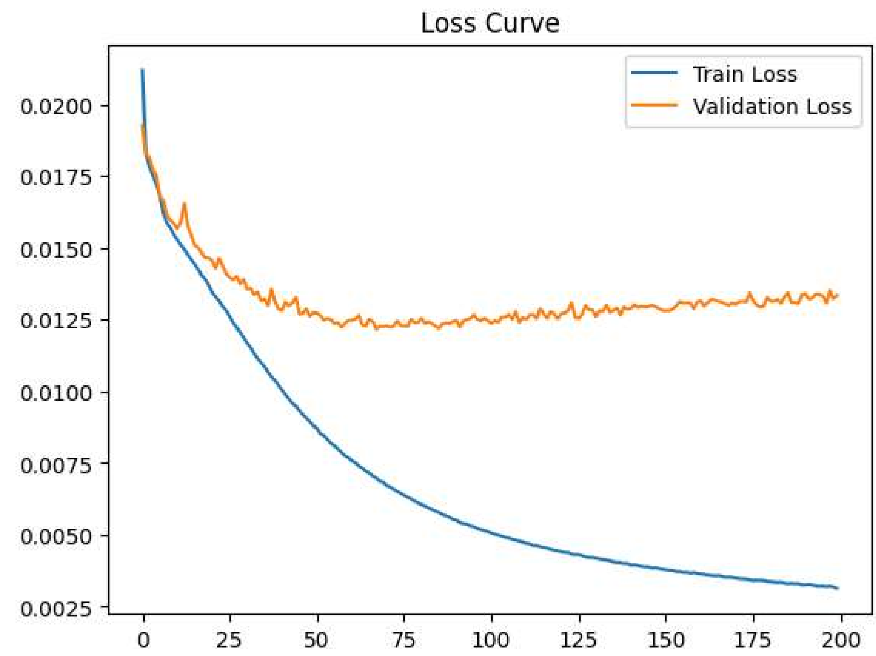
```
Epoch 38/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0104 - mae:
Epoch 39/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0104 - mae:
Epoch 40/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0101 - mae:
Epoch 41/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0099 - mae:
Epoch 42/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0098 - mae:
Epoch 43/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0096 - mae:
Epoch 44/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0095 - mae:
Epoch 45/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0094 - mae:
Epoch 46/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0092 - mae:
Epoch 47/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0090 - mae:
Epoch 48/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0089 - mae:
Epoch 49/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0088 - mae:
Epoch 50/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0087 - mae:
Epoch 51/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0086 - mae:
Epoch 52/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0085 - mae:
Epoch 53/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0083 - mae:
Epoch 54/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0081 - mae:
Epoch 55/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0081 - mae:
Epoch 56/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0079 - mae:
Epoch 57/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0079 - mae:
Epoch 58/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0078 - mae:
Epoch 59/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0076 - mae:
Epoch 60/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0075 - mae:
Epoch 61/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0075 - mae:
Epoch 62/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0073 - mae:
Epoch 63/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0072 - mae:
Epoch 64/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0072 - mae:
Epoch 65/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0071 - mae:
Epoch 66/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0071 - mae:
Epoch 67/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0070 - mae:
Epoch 68/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0069 - mae:
Epoch 69/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0068 - mae:
Epoch 70/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0068 - mae:
Epoch 71/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 7ms/step - loss: 0.0066 - mae:
Epoch 72/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0065 - mae:
Epoch 73/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0065 - mae:
Epoch 74/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0063 - mae:
```

```
Epoch 75/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0063 - mae:
Epoch 76/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0062 - mae:
Epoch 77/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0062 - mae:
Epoch 78/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0061 - mae:
Epoch 79/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0061 - mae:
Epoch 80/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0060 - mae:
Epoch 81/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0059 - mae:
Epoch 82/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0059 - mae:
Epoch 83/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0058 - mae:
Epoch 84/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0058 - mae:
Epoch 85/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0057 - mae:
Epoch 86/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0057 - mae:
Epoch 87/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0056 - mae:
Epoch 88/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0055 - mae:
Epoch 89/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0055 - mae:
Epoch 90/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0054 - mae:
Epoch 91/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0053 - mae:
Epoch 92/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0054 - mae:
Epoch 93/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0052 - mae:
Epoch 94/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0053 - mae:
Epoch 95/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0052 - mae:
Epoch 96/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0051 - mae:
Epoch 97/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0051 - mae:
Epoch 98/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0050 - mae:
Epoch 99/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0051 - mae:
Epoch 100/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0050 - mae:
Epoch 101/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0049 - mae:
Epoch 102/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0048 - mae:
Epoch 103/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0049 - mae:
Epoch 104/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0048 - mae:
Epoch 105/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0048 - mae:
Epoch 106/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0048 - mae:
Epoch 107/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0046 - mae:
Epoch 108/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0047 - mae:
Epoch 109/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0047 - mae:
Epoch 110/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0046 - mae:
Epoch 111/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0046 - mae:
Epoch 112/200
```

```
Epoch 112/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0045 - mae:
Epoch 113/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0046 - mae:
Epoch 114/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0045 - mae:
Epoch 115/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0044 - mae:
Epoch 116/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 8ms/step - loss: 0.0044 - mae:
Epoch 117/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0044 - mae:
Epoch 118/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0043 - mae:
Epoch 119/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0043 - mae:
Epoch 120/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0043 - mae:
Epoch 121/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0043 - mae:
Epoch 122/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0042 - mae:
Epoch 123/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0042 - mae:
Epoch 124/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 8ms/step - loss: 0.0042 - mae:
Epoch 125/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0042 - mae:
Epoch 126/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0041 - mae:
Epoch 127/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0041 - mae:
Epoch 128/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0041 - mae:
Epoch 129/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0041 - mae:
Epoch 130/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0041 - mae:
Epoch 131/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0040 - mae:
Epoch 132/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0040 - mae:
Epoch 133/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0040 - mae:
Epoch 134/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0040 - mae:
Epoch 135/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0040 - mae:
Epoch 136/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0039 - mae:
Epoch 137/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0039 - mae:
Epoch 138/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0039 - mae:
Epoch 139/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0039 - mae:
Epoch 140/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0039 - mae:
Epoch 141/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0037 - mae:
Epoch 142/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0038 - mae:
Epoch 143/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0038 - mae:
Epoch 144/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0038 - mae:
Epoch 145/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0037 - mae:
Epoch 146/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0037 - mae:
Epoch 147/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0037 - mae:
Epoch 148/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0037 - mae:
Epoch 149/200
```

```
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0037 - mae:
Epoch 150/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0036 - mae:
Epoch 151/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0037 - mae:
Epoch 152/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0037 - mae:
Epoch 153/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0036 - mae:
Epoch 154/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0036 - mae:
Epoch 155/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0036 - mae:
Epoch 156/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0036 - mae:
Epoch 157/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0036 - mae:
Epoch 158/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0035 - mae:
Epoch 159/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0035 - mae:
Epoch 160/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0035 - mae:
Epoch 161/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0035 - mae:
Epoch 162/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0035 - mae:
Epoch 163/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0035 - mae:
Epoch 164/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0034 - mae:
Epoch 165/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0034 - mae:
Epoch 166/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0035 - mae:
Epoch 167/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0034 - mae:
Epoch 168/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0034 - mae:
Epoch 169/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0034 - mae:
Epoch 170/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0034 - mae:
Epoch 171/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0034 - mae:
Epoch 172/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0034 - mae:
Epoch 173/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 22s 8ms/step - loss: 0.0034 - mae:
Epoch 174/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0034 - mae:
Epoch 175/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0033 - mae:
Epoch 176/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0033 - mae:
Epoch 177/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0033 - mae:
Epoch 178/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0033 - mae:
Epoch 179/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0033 - mae:
Epoch 180/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0032 - mae:
Epoch 181/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0032 - mae:
Epoch 182/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0032 - mae:
Epoch 183/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 7ms/step - loss: 0.0032 - mae:
Epoch 184/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0032 - mae:
Epoch 185/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0032 - mae:
Epoch 186/200
```

```
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0032 - mae:
Epoch 187/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0032 - mae:
Epoch 188/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0032 - mae:
Epoch 189/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0032 - mae:
Epoch 190/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 19s 7ms/step - loss: 0.0031 - mae:
Epoch 191/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0032 - mae:
Epoch 192/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0031 - mae:
Epoch 193/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0031 - mae:
Epoch 194/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 16s 8ms/step - loss: 0.0031 - mae:
Epoch 195/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 17s 8ms/step - loss: 0.0031 - mae:
Epoch 196/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 15s 7ms/step - loss: 0.0031 - mae:
Epoch 197/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - loss: 0.0031 - mae:
Epoch 198/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0030 - mae:
Epoch 199/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 20s 7ms/step - loss: 0.0031 - mae:
Epoch 200/200
2166/2166 ━━━━━━━━━━━━━━━━━━━━ 21s 8ms/step - loss: 0.0031 - mae:
```



```
# Make predictions
y_pred = model.predict(X_test)

# Convert back to original scale
y_test_actual = scaler.inverse_transform(np.column_stack([X_test[:, -1
y_pred_actual = scaler.inverse_transform(np.column_stack([X_test[:, -1

# Evaluate
from sklearn.metrics import mean_absolute_error, r2_score

mae = mean_absolute_error(y_test_actual, y_pred_actual)
r2 = r2_score(y_test_actual, y_pred_actual)

print(f"Mean Absolute Error: {mae:.4f}")
print(f"R² Score: {r2:.4f}")
```

```
542/542 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step
Mean Absolute Error: 1.2711
R² Score: 0.5928
```

```
import numpy as np
import matplotlib.pyplot as plt

# Function to aggregate data into 5-minute intervals
def aggregate_to_5min(data, interval=120, method="mean"):
    data = data[: len(data) // interval * interval]  # Trim to fit exact
    data = data.reshape(-1, interval)  # Reshape into (num_intervals, ir

    if method == "mean":
        return data.mean(axis=1)
    elif method == "sum":
        return data.sum(axis=1)

# Aggregate actual and predicted values
y_train_actual_5min = aggregate_to_5min(y_train_actual)
y_train_pred_actual_5min = aggregate_to_5min(y_train_pred_actual)
y_test_actual_5min = aggregate_to_5min(y_test_actual)
y_test_pred_actual_5min = aggregate_to_5min(y_test_pred_actual)

# Generate time axis (5-minute intervals)
time_train = np.arange(0, len(y_train_actual_5min) * 5, 5)  # Every 5 mi
time_test = np.arange(len(y_train_actual_5min) * 5, (len(y_train_actual_

# Plot
plt.figure(figsize=(12, 6))
plt.plot(time_train, y_train_actual_5min, label="Actual Data", color="bl
plt.plot(time_train, y_train_pred_actual_5min, label="Training Predictic
plt.plot(time_test, y_test_pred_actual_5min, label="Testing Predictions'

plt.xlabel("Time (minutes)")
```