EXP NO : No.9 DATE: 10/4/25	**Develop Neural Network-Based Time Series Forecasting Model.**

AIM:

To develop a Neural Network-based time series forecasting model using Long Short-Term Memory (LSTM) architecture for predicting future values of the Air Quality Index (AQI) in India based on historical synthetic data.

ALGORITHM:

1. Generate Synthetic AQI Data
2. Preprocess the Time Series Data
3. Create Time Series Sequences
4. Build the LSTM Neural Network Model
5. Train the Model
6. Forecast Future Values
7. Inverse Transform the Forecasted Values
8. Visualize the Forecast

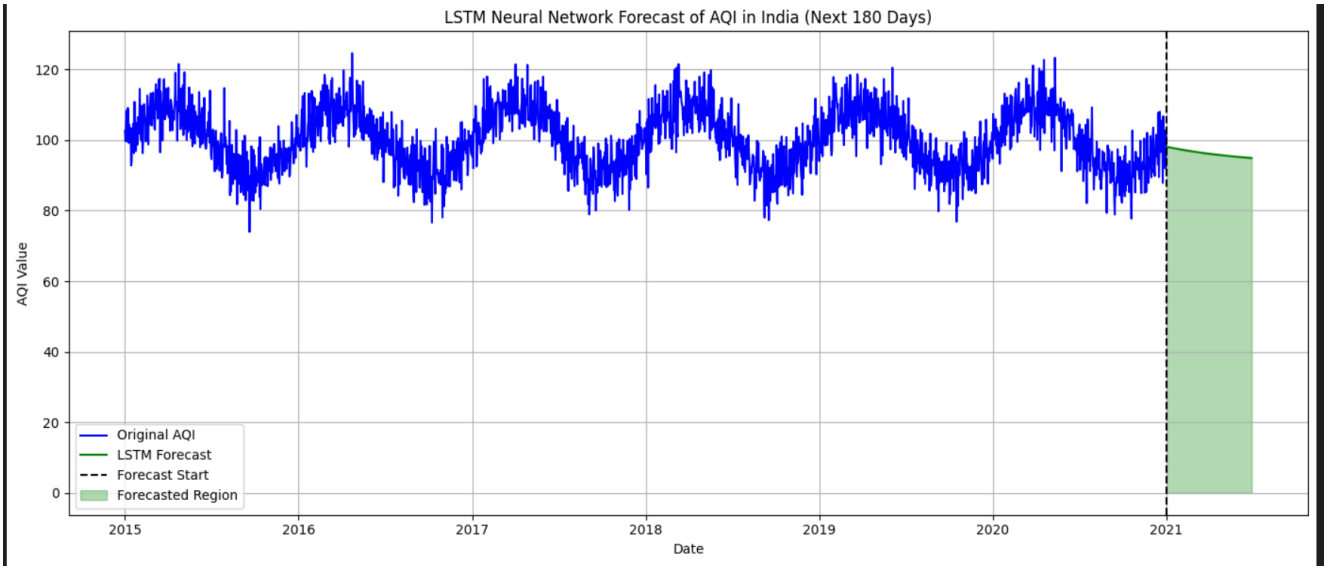
CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
np.random.seed(42)
date_range = pd.date_range(start='2015-01-01', end='2020-12-31', freq='D')
aqi_values = 100 + 10 * np.sin(2 * np.pi * date_range.dayofyear / 365.25) + np.random.normal(0, 5,
len(date_range))
df = pd.DataFrame({
    'Date': date_range,
    'Country': 'India',
```

```
    'AQI Value': aqi_values
})
df_country = df[df['Country'] == 'India'].groupby("Date")['AQI Value'].mean()
df_country = df_country.asfreq('D').interpolate()
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(df_country.values.reshape(-1, 1))
def create_sequences(data, time_steps=30):
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:i + time_steps])
        y.append(data[i + time_steps])
    return np.array(X), np.array(y)
time_steps = 30
X, y = create_sequences(scaled_data, time_steps)
X = X.reshape(X.shape[0], X.shape[1], 1)
model = Sequential()
model.add(LSTM(64, return_sequences=True, input_shape=(time_steps, 1)))
model.add(LSTM(32))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
model.fit(X, y, epochs=20, batch_size=32, verbose=1)
forecast_days = 180
input_seq = scaled_data[-time_steps:]
forecast = []
for _ in range(forecast_days):
    pred_input = input_seq[-time_steps:].reshape(1, time_steps, 1)
    pred = model.predict(pred_input, verbose=0)
    forecast.append(pred[0, 0])
    input_seq = np.append(input_seq, pred, axis=0)
forecast_scaled = np.array(forecast).reshape(-1, 1)
forecast_actual = scaler.inverse_transform(forecast_scaled)
forecast_dates = pd.date_range(start=df_country.index[-1] + pd.Timedelta(days=1),
periods=forecast_days)
```

```
forecast_series = pd.Series(forecast_actual.flatten(), index=forecast_dates)
plt.figure(figsize=(14, 6))
plt.plot(df_country, label='Original AQI', color='blue')
plt.plot(forecast_series, label='LSTM Forecast', color='green')
plt.axvline(x=df_country.index[-1], color='black', linestyle='--', label='Forecast Start')
plt.fill_between(forecast_series.index, forecast_series.values,
                 color='green', alpha=0.3, label='Forecasted Region')
plt.title('LSTM Neural Network Forecast of AQI in India (Next 180 Days)')
plt.xlabel('Date')
plt.ylabel('AQI Value')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

OUTPUT:



RESULT:

Thus the program has been completed and verified successfully.