Import Libaries and Load Data

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense
import matplotlib.pyplot as plt

# Read the CSV file into a pandas DataFrame
file_path = "london_weather[1].csv"
df = pd.read_csv(file_path)

# Prepare the data
df['date'] = pd.to_datetime(df['date'], format='%Y%m%d')
df.set_index('date', inplace=True)
mean_temp_data = df['mean_temp'].dropna().values
```

Normalise Data

In [2]:

```
# Normalize the data
scaler = MinMaxScaler()
mean_temp_data_normalized = scaler.fit_transform(mean_temp_data.reshape(-1, 1))
```

Train and Testing

In [3]:

```
# Create sequences and targets for training
sequence_length = 30
sequences = []
targets = []
for i in range(len(mean_temp_data_normalized) - sequence_length):
    seq = mean_temp_data_normalized[i:i+sequence_length]
    target = mean_temp_data_normalized[i+sequence_length]
    sequences.append(seq)
    targets.append(target)
sequences = np.array(sequences)
targets = np.array(targets)

# Split data into training and testing sets
train_size = int(0.8 * len(sequences))
train_sequences, test_sequences = sequences[:train_size], sequences[train_size:]
train_targets, test_targets = targets[:train_size], targets[train_size:]
```

Model Building

In [4]:

```
# Build and train the RNN model
model = Sequential()
model.add(SimpleRNN(50, input_shape=(sequence_length, 1)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(train_sequences, train_targets, epochs=50, batch_size=32, verbose=1)

# Make predictions on test data
test_predictions_normalized = model.predict(test_sequences)
```

```
test_predictions = scaler.inverse_transform(test_predictions_normalized)

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(mean_temp_data[train_size+sequence_length:], tes

# Print RMSE
print("RMSE:", rmse)
```

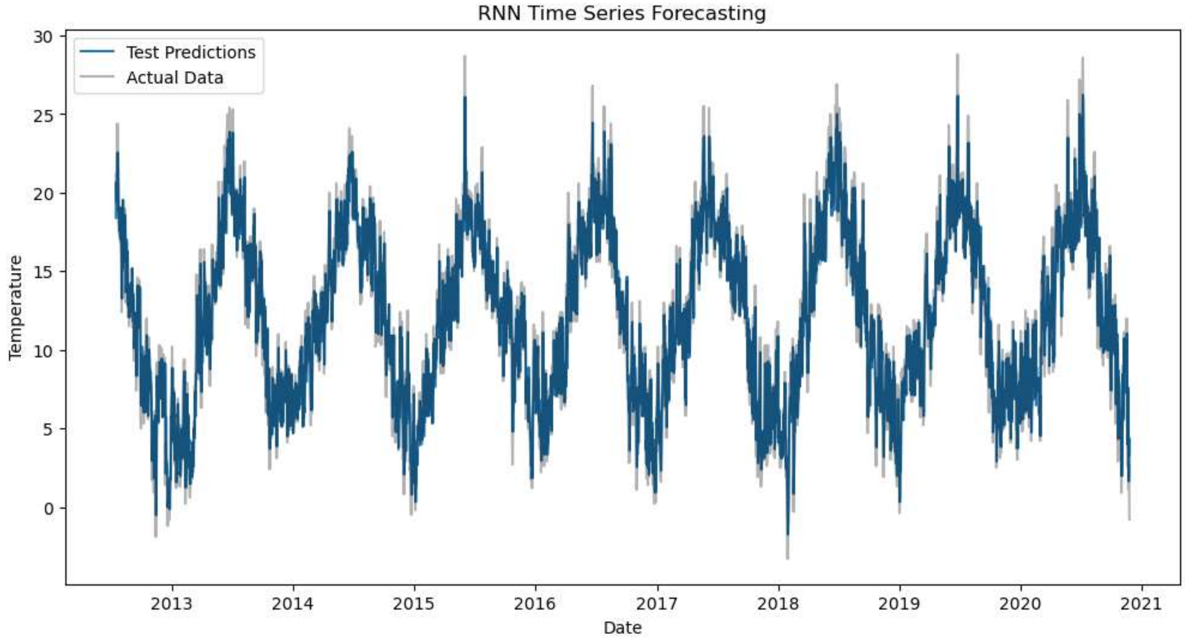
```
Epoch 1/50
382/382 [==============================] - 7s 11ms/step - loss: 0.0064
Epoch 2/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0035
Epoch 3/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0029
Epoch 4/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0028
Epoch 5/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0028
Epoch 6/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0028
Epoch 7/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0028
Epoch 8/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 9/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0028
Epoch 10/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 11/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 12/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 13/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 14/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 15/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 16/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 17/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 18/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 19/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 20/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 21/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 22/50
382/382 [==============================] - 4s 12ms/step - loss: 0.0027
Epoch 23/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 24/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 25/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 26/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 27/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 28/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 29/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 30/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 31/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 32/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
```

```
Epoch 33/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 34/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 35/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 36/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 37/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 38/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 39/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 40/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 41/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 42/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 43/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 44/50
382/382 [==============================] - 4s 11ms/step - loss: 0.0027
Epoch 45/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 46/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 47/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 48/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 49/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
Epoch 50/50
382/382 [==============================] - 4s 10ms/step - loss: 0.0027
96/96 [==============================] - 1s 5ms/step
RMSE: 1.9468221143417137
```

Plot Result

In [6]:

```python
# Plot the results
plt.figure(figsize=(12, 6))
plt.plot(df.index[train_size+sequence_length:train_size+sequence_length+len(test_pr
plt.plot(df.index[train_size+sequence_length:train_size+sequence_length+len(test_pr
plt.legend()
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.title('RNN Time Series Forecasting')
plt.show()
```


RNN Time Series Forecasting
Test Predictions
Actual Data
Temperature
30
25
20
15
10
5
0
2013
2014
2015
2016
2017
2018
2019
2020
2021
Date