CurrentLSTM_SmallerWindow

November 23, 2022

[ ]:
```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import *
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.losses import MeanSquaredError
from tensorflow.keras.metrics import RootMeanSquaredError
from tensorflow.keras.optimizers import Adam
```

[ ]:
```
df = pd.read_csv('meter1_combined.csv')
df.DateTime = pd.to_datetime(df.DateTime,dayfirst=True)
df.index = pd.to_datetime(df['DateTime'], format='%d/%m/%Y %H:%M:%S')
df["Active Total Energy in MWh"]= df['Active Total Energy (Wh) [Wh]'].
 ↪astype(float)/ 1000000
df["Combined Active Power in MW"] = df['Combined Active Power Line 1+2 +3␣
 ↪(watt)'].astype(float) / 1000000
print(df)
df['Combined Active Power in MW'].plot()
# plt.show()
target= df['Combined Active Power in MW']
```

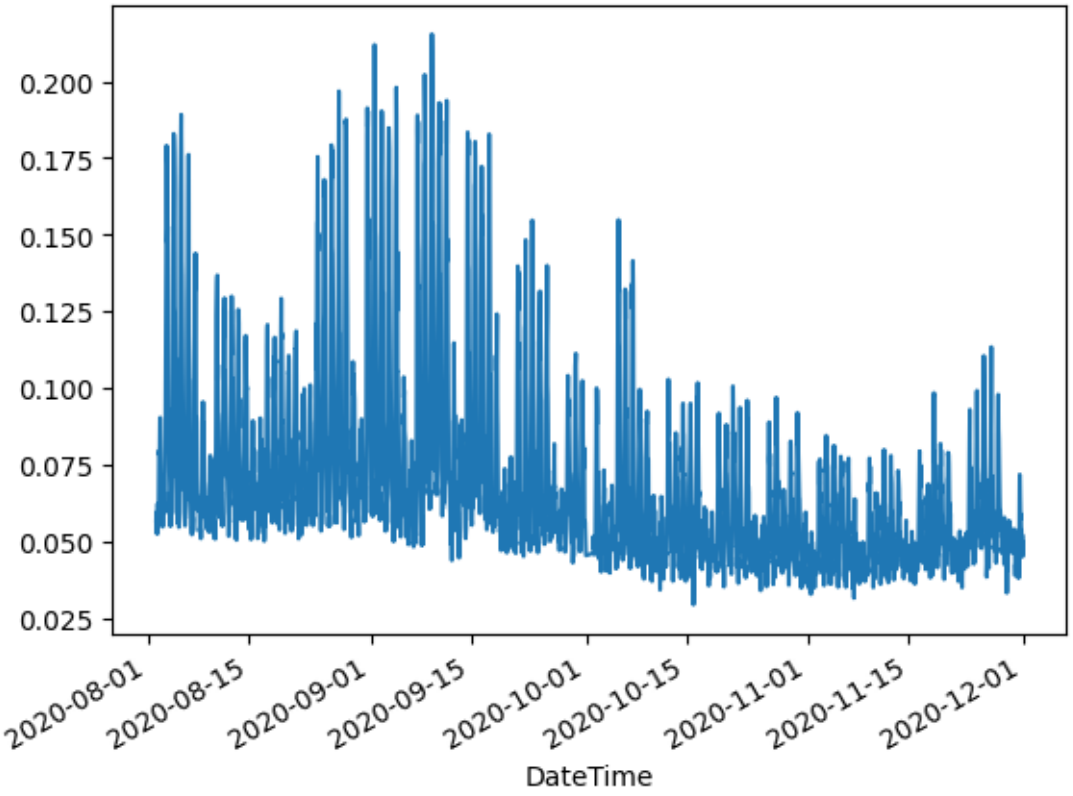
```
                             DateTime  Voltage Line 1 [V]  \
DateTime
2020-08-02 00:00:00 2020-08-02 00:00:00              232.39
2020-08-02 01:00:00 2020-08-02 01:00:00              234.30
2020-08-02 02:00:00 2020-08-02 02:00:00              234.14
2020-08-02 03:00:00 2020-08-02 03:00:00              234.99
2020-08-02 04:00:00 2020-08-02 04:00:00              233.60
…                                   …                   …
2020-11-30 19:00:00 2020-11-30 19:00:00              233.59
2020-11-30 20:00:00 2020-11-30 20:00:00              233.99
2020-11-30 21:00:00 2020-11-30 21:00:00              233.38
2020-11-30 22:00:00 2020-11-30 22:00:00              233.50
2020-11-30 23:00:00 2020-11-30 23:00:00              232.89

                     Voltage Line 2 [V]  Voltage Line 3 [V]  \
```

```
DateTime
2020-08-02 00:00:00              232.27              230.53
2020-08-02 01:00:00              234.51              232.35
2020-08-02 02:00:00              233.59              231.74
2020-08-02 03:00:00              234.59              232.55
2020-08-02 04:00:00              233.55              231.91
…                                 …                   …
2020-11-30 19:00:00              233.42              231.93
2020-11-30 20:00:00              234.11              232.65
2020-11-30 21:00:00              233.19              231.41
2020-11-30 22:00:00              232.91              231.56
2020-11-30 23:00:00              232.31              230.59

                    Current in Line 1 [Amp]  Current in Line 2 [Amp]  \
DateTime
2020-08-02 00:00:00                    80.68                    78.04
2020-08-02 01:00:00                    92.21                    78.21
2020-08-02 02:00:00                    80.72                    69.16
2020-08-02 03:00:00                    85.04                    72.15
2020-08-02 04:00:00                    81.88                    65.06
…                                      …                      …
2020-11-30 19:00:00                    75.79                    82.21
2020-11-30 20:00:00                    72.91                    62.53
2020-11-30 21:00:00                    70.29                    72.40
2020-11-30 22:00:00                    80.09                    76.39
2020-11-30 23:00:00                    69.22                    72.51

                    Current in Line 3 [Amp]  \
DateTime
2020-08-02 00:00:00                    91.81
2020-08-02 01:00:00                    91.04
2020-08-02 02:00:00                    80.74
2020-08-02 03:00:00                    88.16
2020-08-02 04:00:00                    81.17
…                                      …
2020-11-30 19:00:00                    78.28
2020-11-30 20:00:00                    67.83
2020-11-30 21:00:00                    81.45
2020-11-30 22:00:00                    71.99
2020-11-30 23:00:00                    74.78

                    Combined Active Power Line 1+2 +3 (watt)  \
DateTime
2020-08-02 00:00:00                                  56683.98
2020-08-02 01:00:00                                  59487.19
2020-08-02 02:00:00                                  52798.79
2020-08-02 03:00:00                                  55698.04
2020-08-02 04:00:00                                  52330.48
```

```
…                                                  …
2020-11-30 19:00:00                            52057.36
2020-11-30 20:00:00                            44937.44
2020-11-30 21:00:00                            48802.58
2020-11-30 22:00:00                            49585.46
2020-11-30 23:00:00                            46653.30

                    Combined Apparent Power Line 1+2+3  \
DateTime
2020-08-02 00:00:00                            57983.59
2020-08-02 01:00:00                            61091.12
2020-08-02 02:00:00                            53770.52
2020-08-02 03:00:00                            57424.54
2020-08-02 04:00:00                            53163.79
…                                                 …
2020-11-30 19:00:00                            55031.18
2020-11-30 20:00:00                            47462.15
2020-11-30 21:00:00                            52115.82
2020-11-30 22:00:00                            53158.36
2020-11-30 23:00:00                            50223.58

                    Combined Reactive Power Line 1+2+3  …  \
DateTime                                                   …
2020-08-02 00:00:00                            11684.22  …
2020-08-02 01:00:00                            13601.32  …
2020-08-02 02:00:00                             8885.98  …
2020-08-02 03:00:00                            13030.89  …
2020-08-02 04:00:00                             8584.88  …
…                                                  …  …
2020-11-30 19:00:00                            17593.00  …
2020-11-30 20:00:00                            15053.53  …
2020-11-30 21:00:00                            18048.29  …
2020-11-30 22:00:00                            18749.11  …
2020-11-30 23:00:00                            18488.18  …

                    Current Neutral Line  Total Current in Line 1+2+3  \
DateTime
2020-08-02 00:00:00                 18.17                       250.42
2020-08-02 01:00:00                 23.96                       261.31
2020-08-02 02:00:00                 25.79                       230.61
2020-08-02 03:00:00                 24.14                       245.26
2020-08-02 04:00:00                 25.93                       228.12
…                                       …                            …
2020-11-30 19:00:00                  6.32                       236.15
2020-11-30 20:00:00                 17.59                       203.26
2020-11-30 21:00:00                 11.32                       224.24
2020-11-30 22:00:00                 11.08                       228.40
2020-11-30 23:00:00                  6.16                       216.55
```

DateTime	Active Total Energy (Wh) [Wh]	Reactive Total Energy (VARh) [VARh]	Apparent Total Energy (Vah) [Vah]	Temp	Hi	Low	Active Total Energy in MWh	Combined Active Power in MW
2020-08-02 00:00:00	4282350848	1891807744	4681609216	28.7	29.6	27.6	4282.350848	0.056684
2020-08-02 01:00:00	4282409472	1891820160	4681668096	26.0	27.4	25.2	4282.409472	0.059487
2020-08-02 02:00:00	4282467584	1891832064	4681725952	25.9	26.9	25.1	4282.467584	0.052799
2020-08-02 03:00:00	4282525952	1891843968	4681783808	25.3	26.4	24.4		
2020-08-02 04:00:00	4282584320	1891856896	4681842688	24.7	26.3	24.1		
…	…	…	…	…	…	…		
2020-11-30 19:00:00	4480172544	1972983680	4895366656	14.4	15.1	14.1		
2020-11-30 20:00:00	4480225792	1973002752	4895422464	13.7	14.2	13.3		
2020-11-30 21:00:00	4480275968	1973021568	4895476224	13.4	13.7	13.2		
2020-11-30 22:00:00	4480325120	1973040000	4895528960	13.2	13.7	12.7		
2020-11-30 23:00:00	4480373248	1973057152	4895579648	12.8	13.4	12.6		

```
2020-08-02 03:00:00          4282.525952          0.055698
2020-08-02 04:00:00          4282.584320          0.052330
…                                    …                 …
2020-11-30 19:00:00          4480.172544          0.052057
2020-11-30 20:00:00          4480.225792          0.044937
2020-11-30 21:00:00          4480.275968          0.048803
2020-11-30 22:00:00          4480.325120          0.049585
2020-11-30 23:00:00          4480.373248          0.046653

[2878 rows x 23 columns]
```



[ ]:
```python
from sklearn.metrics import mean_squared_error as mse

def plot_predictions1(model, X, y, start=0, end=100):
  predictions = model.predict(X).flatten()
  df = pd.DataFrame(data={'Predictions':predictions, 'Actuals':y})
  plt.plot(df['Predictions'][start:end])
  plt.plot(df['Actuals'][start:end])
  plt.legend(['Actuals','Test Predictions'])
  return df, mse(y, predictions)
```

[ ]:
```
target.head()
```

[ ]:
```
DateTime
2020-08-02 00:00:00    0.056684
2020-08-02 01:00:00    0.059487
2020-08-02 02:00:00    0.052799
2020-08-02 03:00:00    0.055698
2020-08-02 04:00:00    0.052330
Name: Combined Active Power in MW, dtype: float64
```

[ ]:
```
temp_df = pd.DataFrame({'Power':target,"Avg Temp":df["Temp"],"High Temp":
 ↪df['Hi'],"Low Temp":df["Low"],"Combined Current":df['Total Current in Line␣
 ↪1+2+3']})
temp_df['Seconds'] = temp_df.index.map(pd.Timestamp.timestamp)
temp_df
```

[ ]:
```
                        Power  Avg Temp  High Temp  Low Temp  \
DateTime
2020-08-02 00:00:00  0.056684      28.7       29.6      27.6
2020-08-02 01:00:00  0.059487      26.0       27.4      25.2
2020-08-02 02:00:00  0.052799      25.9       26.9      25.1
2020-08-02 03:00:00  0.055698      25.3       26.4      24.4
2020-08-02 04:00:00  0.052330      24.7       26.3      24.1
…                         …         …          …         …
2020-11-30 19:00:00  0.052057      14.4       15.1      14.1
2020-11-30 20:00:00  0.044937      13.7       14.2      13.3
2020-11-30 21:00:00  0.048803      13.4       13.7      13.2
2020-11-30 22:00:00  0.049585      13.2       13.7      12.7
2020-11-30 23:00:00  0.046653      12.8       13.4      12.6

                     Combined Current       Seconds
DateTime
2020-08-02 00:00:00            250.42  1.596326e+09
2020-08-02 01:00:00            261.31  1.596330e+09
2020-08-02 02:00:00            230.61  1.596334e+09
2020-08-02 03:00:00            245.26  1.596337e+09
2020-08-02 04:00:00            228.12  1.596341e+09
…                                 …           …
2020-11-30 19:00:00            236.15  1.606763e+09
2020-11-30 20:00:00            203.26  1.606766e+09
2020-11-30 21:00:00            224.24  1.606770e+09
2020-11-30 22:00:00            228.40  1.606774e+09
2020-11-30 23:00:00            216.55  1.606777e+09

[2878 rows x 6 columns]
```

[ ]:
```
#Adding into consideration the time as a periodic function

day = 60*60*24
month= 30*day
temp_df['Day sin'] = np.sin(temp_df['Seconds'] * (2* np.pi / day))
temp_df['Day cos'] = np.cos(temp_df['Seconds'] * (2 * np.pi / day))
temp_df['Month sin'] = np.sin(temp_df['Seconds'] * (2 * np.pi / month))
temp_df['Month cos'] = np.cos(temp_df['Seconds'] * (2 * np.pi / month))
temp_df
```

[ ]:

DateTime	Power	Avg Temp	High Temp	Low Temp	\
2020-08-02 00:00:00	0.056684	28.7	29.6	27.6	
2020-08-02 01:00:00	0.059487	26.0	27.4	25.2	
2020-08-02 02:00:00	0.052799	25.9	26.9	25.1	
2020-08-02 03:00:00	0.055698	25.3	26.4	24.4	
2020-08-02 04:00:00	0.052330	24.7	26.3	24.1	
…	…	…	…	…	
2020-11-30 19:00:00	0.052057	14.4	15.1	14.1	
2020-11-30 20:00:00	0.044937	13.7	14.2	13.3	
2020-11-30 21:00:00	0.048803	13.4	13.7	13.2	
2020-11-30 22:00:00	0.049585	13.2	13.7	12.7	
2020-11-30 23:00:00	0.046653	12.8	13.4	12.6	

DateTime	Combined Current	Seconds	Day sin	Day cos	\
2020-08-02 00:00:00	250.42	1.596326e+09	-4.553737e-12	1.000000	
2020-08-02 01:00:00	261.31	1.596330e+09	2.588190e-01	0.965926	
2020-08-02 02:00:00	230.61	1.596334e+09	5.000000e-01	0.866025	
2020-08-02 03:00:00	245.26	1.596337e+09	7.071068e-01	0.707107	
2020-08-02 04:00:00	228.12	1.596341e+09	8.660254e-01	0.500000	
…	…	…	…	…	
2020-11-30 19:00:00	236.15	1.606763e+09	-9.659258e-01	0.258819	
2020-11-30 20:00:00	203.26	1.606766e+09	-8.660254e-01	0.500000	
2020-11-30 21:00:00	224.24	1.606770e+09	-7.071068e-01	0.707107	
2020-11-30 22:00:00	228.40	1.606774e+09	-5.000000e-01	0.866025	
2020-11-30 23:00:00	216.55	1.606777e+09	-2.588190e-01	0.965926	

DateTime	Month sin	Month cos
2020-08-02 00:00:00	-0.743145	0.669131
2020-08-02 01:00:00	-0.737277	0.675590
2020-08-02 02:00:00	-0.731354	0.681998
2020-08-02 03:00:00	-0.725374	0.688355
2020-08-02 04:00:00	-0.719340	0.694658
…	…	…
2020-11-30 19:00:00	-0.622515	0.782608

```
2020-11-30 20:00:00  -0.615661   0.788011
2020-11-30 21:00:00  -0.608761   0.793353
2020-11-30 22:00:00  -0.601815   0.798636
2020-11-30 23:00:00  -0.594823   0.803857

[2878 rows x 10 columns]
```

[ ]:
```
temp_df.drop("Seconds",axis=1,inplace=True)
temp_df.head()
```

[ ]:
```
                        Power  Avg Temp  High Temp  Low Temp  \
DateTime
2020-08-02 00:00:00  0.056684      28.7       29.6      27.6
2020-08-02 01:00:00  0.059487      26.0       27.4      25.2
2020-08-02 02:00:00  0.052799      25.9       26.9      25.1
2020-08-02 03:00:00  0.055698      25.3       26.4      24.4
2020-08-02 04:00:00  0.052330      24.7       26.3      24.1

                     Combined Current       Day sin   Day cos  Month sin  \
DateTime
2020-08-02 00:00:00            250.42 -4.553737e-12  1.000000  -0.743145
2020-08-02 01:00:00            261.31  2.588190e-01  0.965926  -0.737277
2020-08-02 02:00:00            230.61  5.000000e-01  0.866025  -0.731354
2020-08-02 03:00:00            245.26  7.071068e-01  0.707107  -0.725374
2020-08-02 04:00:00            228.12  8.660254e-01  0.500000  -0.719340

                     Month cos
DateTime
2020-08-02 00:00:00   0.669131
2020-08-02 01:00:00   0.675590
2020-08-02 02:00:00   0.681998
2020-08-02 03:00:00   0.688355
2020-08-02 04:00:00   0.694658
```

[ ]:
```
#Function to create timeseries for multiple parameters
def df_to_X_y2(df, window_size=2):
  df_as_np = df.to_numpy()
  X = []
  y = []
  for i in range(len(df_as_np)-window_size):
    row = [r for r in df_as_np[i:i+window_size]]
    X.append(row)
    label = df_as_np[i+window_size][0]
    y.append(label)
  return np.array(X), np.array(y)
```

[ ]:
```
X2, y2 = df_to_X_y2(temp_df)
X2.shape, y2.shape
```

[ ]:
```
((2876, 2, 9), (2876,))
```

[ ]:
```
X2_train, y2_train = X2[:2000], y2[:2000]
X2_val, y2_val = X2[2000:2435], y2[2000:2435]
X2_test, y2_test = X2[2435:], y2[2435:]
X2_train.shape, y2_train.shape, X2_val.shape, y2_val.shape, X2_test.shape,␣
 ↪y2_test.shape
```

[ ]:
```
((2000, 2, 9), (2000,), (435, 2, 9), (435,), (441, 2, 9), (441,))
```

[ ]:
```
#Standardization for the temperatures. Dividing mean/std
avg_temp_training_mean = np.mean(X2_train[:, :, 1])
avg_temp_training_std = np.std(X2_train[:, :, 1])
hi_temp_training_mean=np.mean(X2_train[:, :, 2])
hi_temp_training_std=np.std(X2_train[:, :, 2])
low_temp_training_mean = np.mean(X2_train[:, :, 3])
low_temp_training_std=np.std(X2_train[:, :, 3])
curr_training_mean = np.mean(X2_train[:, :, 4])
curr_training_std = np.std(X2_train[:, :, 4])

def preprocess(X):
  X[:, :, 1] = (X[:, :, 1] - avg_temp_training_mean) / avg_temp_training_std
  X[:, :, 2] = (X[:, :, 2] - hi_temp_training_mean) / hi_temp_training_std
  X[:, :, 3] = (X[:, :, 3] - low_temp_training_mean) / low_temp_training_std
  X[:, :, 4] = (X[:, :, 4] - curr_training_mean) / curr_training_std
  return X

preprocess(X2_train)
preprocess(X2_val)
preprocess(X2_test)
```

[ ]:
```
array([[[ 0.07168051, -1.02664378, -1.04978731, …, -0.96592583,
          0.97629601, -0.21643961],
        [ 0.07084134, -0.94738271, -1.03006985, …, -0.8660254 ,
          0.97437006, -0.22495105]],

       [[ 0.07084134, -0.94738271, -1.03006985, …, -0.8660254 ,
          0.97437006, -0.22495105],
        [ 0.06864545, -0.92756745, -0.99063492, …, -0.70710678,
          0.97236992, -0.23344536]],

       [[ 0.06864545, -0.92756745, -0.99063492, …, -0.70710678,
          0.97236992, -0.23344536],
        [ 0.05908783, -0.90775218, -0.97091746, …, -0.5       ,
```

```
        0.97029573, -0.2419219 ]],

       …,

       [[ 0.05205736, -2.21555971, -2.17368271, …,  0.25881905,
         -0.62251464,  0.78260816],
        [ 0.04493744, -2.35426657, -2.35113988, …,  0.5       ,
         -0.61566148,  0.78801075]],

       [[ 0.04493744, -2.35426657, -2.35113988, …,  0.5       ,
         -0.61566148,  0.78801075],
        [ 0.04880258, -2.41371237, -2.44972719, …,  0.70710678,
         -0.60876143,  0.79335334]],

       [[ 0.04880258, -2.41371237, -2.44972719, …,  0.70710678,
         -0.60876143,  0.79335334],
        [ 0.04958546, -2.4533429 , -2.44972719, …,  0.8660254 ,
         -0.60181502,  0.79863551]]])
```

[ ]:
```
model4 = Sequential()
model4.add(InputLayer((2, 9)))
model4.add(LSTM(100))
model4.add(Dense(8, 'relu'))
model4.add(Dense(8, 'relu'))
model4.add(Dense(1, 'linear'))

model4.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 100)               44000
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 808
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 72
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 9
=================================================================
Total params: 44,889
Trainable params: 44,889
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
cp4 = ModelCheckpoint('model4/', save_best_only=True)
model4.compile(loss=MeanSquaredError(), optimizer=Adam(learning_rate=0.0001),␣
 ↪metrics=[RootMeanSquaredError()])
```

```
[ ]: model4.fit(X2_train, y2_train, validation_data=(X2_val, y2_val), epochs=50,␣
      ↪callbacks=[cp4])
```

```
Epoch 1/50
63/63 [==============================] - 3s 11ms/step - loss: 0.0050 -
root_mean_squared_error: 0.0703 - val_loss: 8.4294e-04 -
val_root_mean_squared_error: 0.0290

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 2/50
63/63 [==============================] - 0s 3ms/step - loss: 0.0017 -
root_mean_squared_error: 0.0414 - val_loss: 2.1183e-04 -
val_root_mean_squared_error: 0.0146

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 3/50
63/63 [==============================] - 0s 3ms/step - loss: 8.5952e-04 -
root_mean_squared_error: 0.0293 - val_loss: 3.3844e-04 -
val_root_mean_squared_error: 0.0184
Epoch 4/50
63/63 [==============================] - 0s 3ms/step - loss: 5.1058e-04 -
root_mean_squared_error: 0.0226 - val_loss: 2.8822e-04 -
val_root_mean_squared_error: 0.0170
Epoch 5/50
63/63 [==============================] - 0s 3ms/step - loss: 2.8549e-04 -
root_mean_squared_error: 0.0169 - val_loss: 1.8366e-04 -
val_root_mean_squared_error: 0.0136

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets
```

```
INFO:tensorflow:Assets written to: model4\assets

Epoch 6/50
63/63 [==============================] - 0s 3ms/step - loss: 1.8130e-04 -
root_mean_squared_error: 0.0135 - val_loss: 1.3691e-04 -
val_root_mean_squared_error: 0.0117

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 7/50
63/63 [==============================] - 0s 3ms/step - loss: 1.6727e-04 -
root_mean_squared_error: 0.0129 - val_loss: 1.1042e-04 -
val_root_mean_squared_error: 0.0105

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 8/50
63/63 [==============================] - 0s 3ms/step - loss: 1.3879e-04 -
root_mean_squared_error: 0.0118 - val_loss: 1.0350e-04 -
val_root_mean_squared_error: 0.0102

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 9/50
63/63 [==============================] - 0s 3ms/step - loss: 1.3344e-04 -
root_mean_squared_error: 0.0115 - val_loss: 1.0139e-04 -
val_root_mean_squared_error: 0.0101

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
```

```
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 10/50
63/63 [==============================] - 0s 3ms/step - loss: 1.2766e-04 -
root_mean_squared_error: 0.0113 - val_loss: 9.4874e-05 -
val_root_mean_squared_error: 0.0097

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 11/50
63/63 [==============================] - 0s 3ms/step - loss: 1.1801e-04 -
root_mean_squared_error: 0.0108 - val_loss: 1.0041e-04 -
val_root_mean_squared_error: 0.0100
Epoch 12/50
63/63 [==============================] - 0s 3ms/step - loss: 1.2130e-04 -
root_mean_squared_error: 0.0110 - val_loss: 9.4626e-05 -
val_root_mean_squared_error: 0.0097

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 13/50
63/63 [==============================] - 0s 3ms/step - loss: 1.1123e-04 -
root_mean_squared_error: 0.0105 - val_loss: 9.4228e-05 -
val_root_mean_squared_error: 0.0097

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets
```

```
Epoch 14/50
63/63 [==============================] - 0s 3ms/step - loss: 1.1792e-04 -
root_mean_squared_error: 0.0109 - val_loss: 9.4649e-05 -
val_root_mean_squared_error: 0.0097
Epoch 15/50
63/63 [==============================] - 0s 3ms/step - loss: 1.0462e-04 -
root_mean_squared_error: 0.0102 - val_loss: 8.3833e-05 -
val_root_mean_squared_error: 0.0092

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 16/50
63/63 [==============================] - 0s 3ms/step - loss: 1.1090e-04 -
root_mean_squared_error: 0.0105 - val_loss: 8.5402e-05 -
val_root_mean_squared_error: 0.0092
Epoch 17/50
63/63 [==============================] - 0s 3ms/step - loss: 9.8536e-05 -
root_mean_squared_error: 0.0099 - val_loss: 8.0924e-05 -
val_root_mean_squared_error: 0.0090

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 18/50
63/63 [==============================] - 0s 3ms/step - loss: 1.0823e-04 -
root_mean_squared_error: 0.0104 - val_loss: 8.0413e-05 -
val_root_mean_squared_error: 0.0090

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets
```

```
Epoch 19/50
63/63 [==============================] - 0s 3ms/step - loss: 8.7269e-05 -
root_mean_squared_error: 0.0093 - val_loss: 8.3166e-05 -
val_root_mean_squared_error: 0.0091
Epoch 20/50
63/63 [==============================] - 0s 3ms/step - loss: 9.9617e-05 -
root_mean_squared_error: 0.0100 - val_loss: 7.8328e-05 -
val_root_mean_squared_error: 0.0089

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 21/50
63/63 [==============================] - 0s 3ms/step - loss: 1.0122e-04 -
root_mean_squared_error: 0.0100 - val_loss: 8.0279e-05 -
val_root_mean_squared_error: 0.0090
Epoch 22/50
63/63 [==============================] - 0s 3ms/step - loss: 9.4319e-05 -
root_mean_squared_error: 0.0097 - val_loss: 8.3540e-05 -
val_root_mean_squared_error: 0.0091
Epoch 23/50
63/63 [==============================] - 0s 3ms/step - loss: 9.1818e-05 -
root_mean_squared_error: 0.0096 - val_loss: 8.1809e-05 -
val_root_mean_squared_error: 0.0090
Epoch 24/50
63/63 [==============================] - 0s 3ms/step - loss: 9.1492e-05 -
root_mean_squared_error: 0.0096 - val_loss: 8.2637e-05 -
val_root_mean_squared_error: 0.0091
Epoch 25/50
63/63 [==============================] - 0s 3ms/step - loss: 8.3657e-05 -
root_mean_squared_error: 0.0091 - val_loss: 7.7989e-05 -
val_root_mean_squared_error: 0.0088

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 26/50
63/63 [==============================] - 0s 3ms/step - loss: 8.7476e-05 -
```

```
root_mean_squared_error: 0.0093 - val_loss: 7.8773e-05 -
val_root_mean_squared_error: 0.0089
Epoch 27/50
63/63 [==============================] - 0s 3ms/step - loss: 9.4620e-05 -
root_mean_squared_error: 0.0097 - val_loss: 8.8397e-05 -
val_root_mean_squared_error: 0.0094
Epoch 28/50
63/63 [==============================] - 0s 3ms/step - loss: 8.6920e-05 -
root_mean_squared_error: 0.0093 - val_loss: 8.8954e-05 -
val_root_mean_squared_error: 0.0094
Epoch 29/50
63/63 [==============================] - 0s 3ms/step - loss: 9.1303e-05 -
root_mean_squared_error: 0.0095 - val_loss: 7.4737e-05 -
val_root_mean_squared_error: 0.0086

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 30/50
63/63 [==============================] - 0s 3ms/step - loss: 8.3424e-05 -
root_mean_squared_error: 0.0091 - val_loss: 7.0160e-05 -
val_root_mean_squared_error: 0.0084

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 31/50
63/63 [==============================] - 0s 3ms/step - loss: 8.3333e-05 -
root_mean_squared_error: 0.0091 - val_loss: 6.8365e-05 -
val_root_mean_squared_error: 0.0083

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets
```

```
INFO:tensorflow:Assets written to: model4\assets

Epoch 32/50
63/63 [==============================] - 0s 3ms/step - loss: 8.7535e-05 -
root_mean_squared_error: 0.0094 - val_loss: 7.9424e-05 -
val_root_mean_squared_error: 0.0089
Epoch 33/50
63/63 [==============================] - 0s 3ms/step - loss: 7.9905e-05 -
root_mean_squared_error: 0.0089 - val_loss: 7.6372e-05 -
val_root_mean_squared_error: 0.0087
Epoch 34/50
63/63 [==============================] - 0s 3ms/step - loss: 9.3388e-05 -
root_mean_squared_error: 0.0097 - val_loss: 6.4635e-05 -
val_root_mean_squared_error: 0.0080

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 35/50
63/63 [==============================] - 0s 3ms/step - loss: 8.5452e-05 -
root_mean_squared_error: 0.0092 - val_loss: 6.8330e-05 -
val_root_mean_squared_error: 0.0083
Epoch 36/50
63/63 [==============================] - 0s 3ms/step - loss: 7.6950e-05 -
root_mean_squared_error: 0.0088 - val_loss: 7.2473e-05 -
val_root_mean_squared_error: 0.0085
Epoch 37/50
63/63 [==============================] - 0s 3ms/step - loss: 8.8486e-05 -
root_mean_squared_error: 0.0094 - val_loss: 6.3727e-05 -
val_root_mean_squared_error: 0.0080

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 38/50
63/63 [==============================] - 0s 3ms/step - loss: 8.3364e-05 -
root_mean_squared_error: 0.0091 - val_loss: 6.4445e-05 -
val_root_mean_squared_error: 0.0080
```

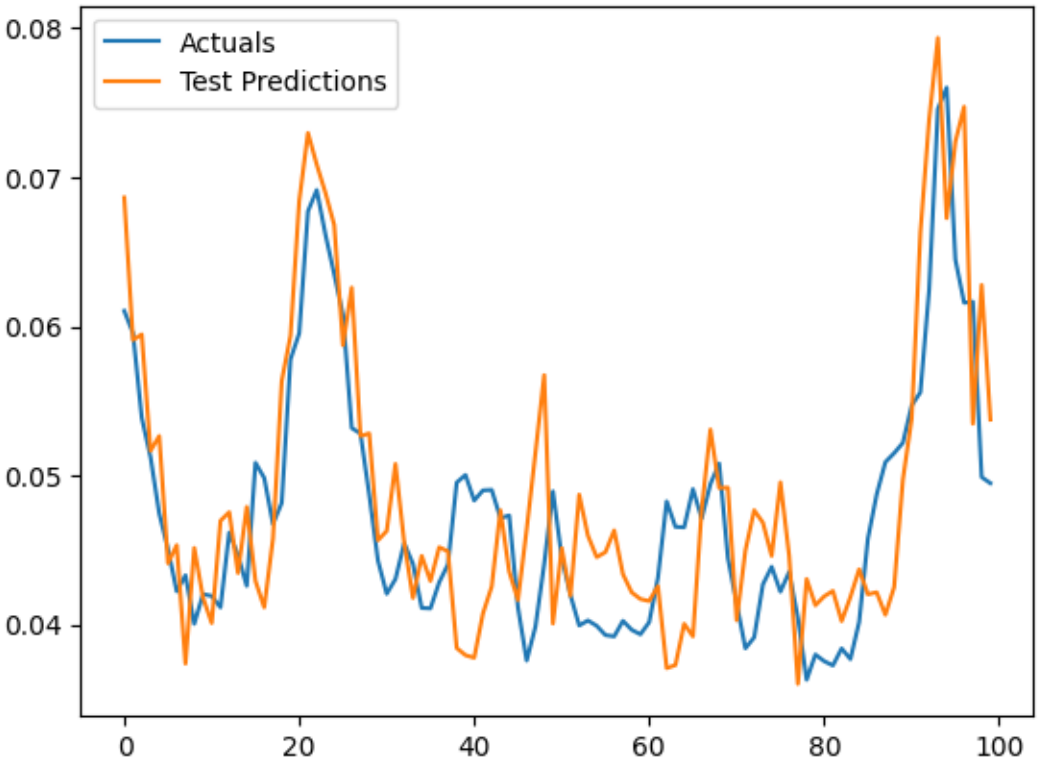
```
Epoch 39/50
63/63 [==============================] - 0s 3ms/step - loss: 7.8545e-05 -
root_mean_squared_error: 0.0089 - val_loss: 7.2400e-05 -
val_root_mean_squared_error: 0.0085
Epoch 40/50
63/63 [==============================] - 0s 3ms/step - loss: 8.1640e-05 -
root_mean_squared_error: 0.0090 - val_loss: 6.8904e-05 -
val_root_mean_squared_error: 0.0083
Epoch 41/50
63/63 [==============================] - 0s 3ms/step - loss: 7.9097e-05 -
root_mean_squared_error: 0.0089 - val_loss: 6.3777e-05 -
val_root_mean_squared_error: 0.0080
Epoch 42/50
63/63 [==============================] - 0s 3ms/step - loss: 7.7959e-05 -
root_mean_squared_error: 0.0088 - val_loss: 6.2939e-05 -
val_root_mean_squared_error: 0.0079

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 43/50
63/63 [==============================] - 0s 3ms/step - loss: 7.5628e-05 -
root_mean_squared_error: 0.0087 - val_loss: 6.5357e-05 -
val_root_mean_squared_error: 0.0081
Epoch 44/50
63/63 [==============================] - 0s 3ms/step - loss: 8.1460e-05 -
root_mean_squared_error: 0.0090 - val_loss: 6.1455e-05 -
val_root_mean_squared_error: 0.0078

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 45/50
63/63 [==============================] - 0s 3ms/step - loss: 7.6508e-05 -
root_mean_squared_error: 0.0087 - val_loss: 6.1598e-05 -
val_root_mean_squared_error: 0.0078
Epoch 46/50
63/63 [==============================] - 0s 3ms/step - loss: 7.7189e-05 -
```

```
root_mean_squared_error: 0.0088 - val_loss: 6.4897e-05 -
val_root_mean_squared_error: 0.0081
Epoch 47/50
63/63 [==============================] - 0s 3ms/step - loss: 7.3197e-05 -
root_mean_squared_error: 0.0085 - val_loss: 5.6628e-05 -
val_root_mean_squared_error: 0.0075

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 48/50
63/63 [==============================] - 0s 3ms/step - loss: 7.7215e-05 -
root_mean_squared_error: 0.0088 - val_loss: 5.4052e-05 -
val_root_mean_squared_error: 0.0074

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 49/50
63/63 [==============================] - 0s 3ms/step - loss: 7.9001e-05 -
root_mean_squared_error: 0.0089 - val_loss: 5.9220e-05 -
val_root_mean_squared_error: 0.0077
Epoch 50/50
63/63 [==============================] - 0s 3ms/step - loss: 7.6073e-05 -
root_mean_squared_error: 0.0087 - val_loss: 5.3614e-05 -
val_root_mean_squared_error: 0.0073

WARNING:absl:Found untraced functions such as lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets
```

[ ]: <tensorflow.python.keras.callbacks.History at 0x1cb4ab77b80>

[ ]:
```
plot_predictions1(model4, X2_test, y2_test)
```

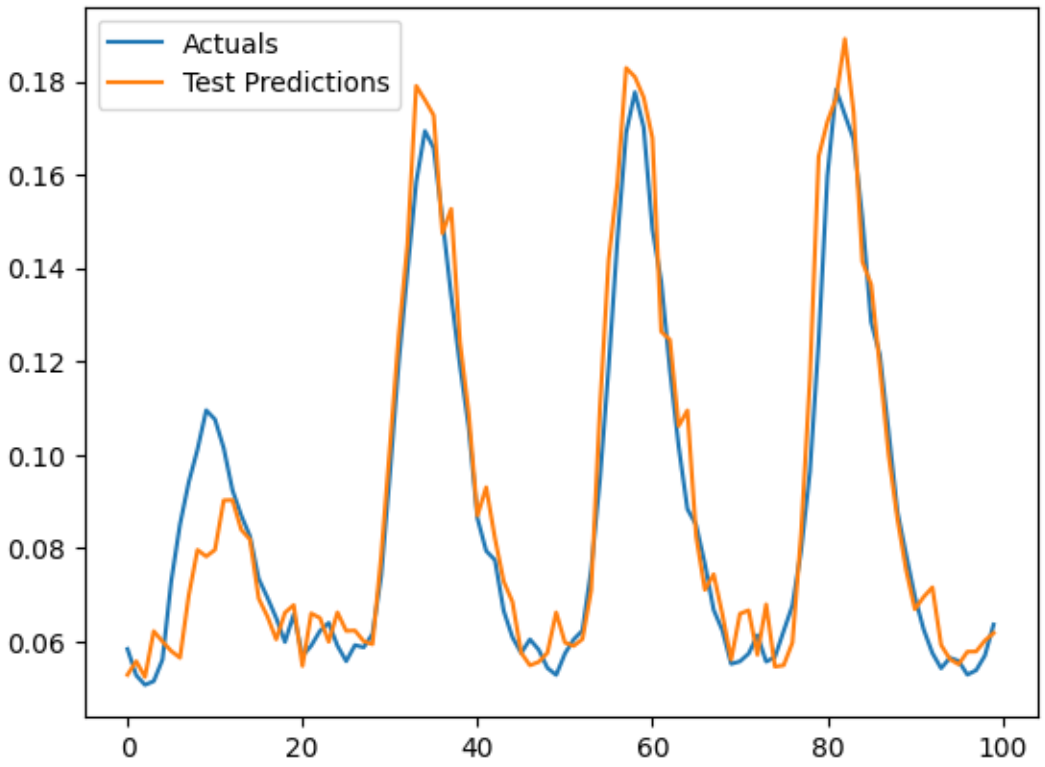
[ ]:
```
(     Predictions   Actuals
 0       0.061044  0.068645
 1       0.059552  0.059088
 2       0.053878  0.059466
 3       0.051247  0.051650
 4       0.047454  0.052649
 ..           …       …
 436     0.043955  0.052057
 437     0.044288  0.044937
 438     0.044475  0.048803
 439     0.046487  0.049585
 440     0.047514  0.046653

 [441 rows x 2 columns],
 6.332156359589317e-05)
```



[ ]:
```
plot_predictions1(model4, X2_train, y2_train)
```

[ ]:
```
(      Predictions   Actuals
 0        0.058321  0.052799
```

```
1          0.052679   0.055698
2          0.050636   0.052330
3          0.051421   0.062136
4          0.056057   0.059931
…               …          …
1995       0.048176   0.039942
1996       0.052850   0.033930
1997       0.052338   0.041539
1998       0.052410   0.051075
1999       0.058099   0.046778

[2000 rows x 2 columns],
0.00010819709736802885)
```



[ ]:
```
#Prediction without using time as a parameter (Just Temperature)
new_df= pd.DataFrame({'Power':target,"Avg Temp":df["Temp"],"High Temp":
  ↪df['Hi'],"Low Temp":df["Low"],"Combined Current":df['Total Current in Line␣
  ↪1+2+3']})
new_df
```

[ ]:
```
                     Power   Avg Temp   High Temp   Low Temp   Combined Current
DateTime
```

```
2020-08-02 00:00:00  0.056684      28.7       29.6       27.6            250.42
2020-08-02 01:00:00  0.059487      26.0       27.4       25.2            261.31
2020-08-02 02:00:00  0.052799      25.9       26.9       25.1            230.61
2020-08-02 03:00:00  0.055698      25.3       26.4       24.4            245.26
2020-08-02 04:00:00  0.052330      24.7       26.3       24.1            228.12
…                          …         …          …          …               …
2020-11-30 19:00:00  0.052057      14.4       15.1       14.1            236.15
2020-11-30 20:00:00  0.044937      13.7       14.2       13.3            203.26
2020-11-30 21:00:00  0.048803      13.4       13.7       13.2            224.24
2020-11-30 22:00:00  0.049585      13.2       13.7       12.7            228.40
2020-11-30 23:00:00  0.046653      12.8       13.4       12.6            216.55

[2878 rows x 5 columns]
```

[ ]:
```
X3, y3 = df_to_X_y2(new_df)
X3.shape, y3.shape
```

[ ]:
```
((2876, 2, 5), (2876,))
```

[ ]:
```
X3_train, y3_train = X3[:2000], y3[:2000]
X3_val, y3_val = X3[2000:2435], y3[2000:2435]
X3_test, y3_test = X3[2435:], y3[2435:]
X3_train.shape, y3_train.shape, X3_val.shape, y3_val.shape, X3_test.shape,␣
 ↪y3_test.shape
```

[ ]:
```
((2000, 2, 5), (2000,), (435, 2, 5), (435,), (441, 2, 5), (441,))
```

[ ]:
```
#Standardization for the temperatures. Dividing mean/std
avg_temp_training_mean = np.mean(X2_train[:, :, 1])
avg_temp_training_std = np.std(X2_train[:, :, 1])
hi_temp_training_mean=np.mean(X2_train[:, :, 2])
hi_temp_training_std=np.std(X2_train[:, :, 2])
low_temp_training_mean = np.mean(X2_train[:, :, 3])
low_temp_training_std=np.std(X2_train[:, :, 3])
curr_training_mean = np.mean(X2_train[:, :, 4])
curr_training_std = np.std(X2_train[:, :, 4])

def preprocess(X):
  X[:, :, 1] = (X[:, :, 1] - avg_temp_training_mean) / avg_temp_training_std
  X[:, :, 2] = (X[:, :, 2] - hi_temp_training_mean) / hi_temp_training_std
  X[:, :, 3] = (X[:, :, 3] - low_temp_training_mean) / low_temp_training_std
  X[:, :, 4] = (X[:, :, 4] - curr_training_mean) / curr_training_std
  return X

preprocess(X2_train)
preprocess(X2_val)
preprocess(X2_test)
```

```
[ ]: array([[[ 0.07168051, -1.02664378, -1.04978731, …, -0.96592583,
                0.97629601, -0.21643961],
              [ 0.07084134, -0.94738271, -1.03006985, …, -0.8660254 ,
                0.97437006, -0.22495105]],

             [[ 0.07084134, -0.94738271, -1.03006985, …, -0.8660254 ,
                0.97437006, -0.22495105],
              [ 0.06864545, -0.92756745, -0.99063492, …, -0.70710678,
                0.97236992, -0.23344536]],

             [[ 0.06864545, -0.92756745, -0.99063492, …, -0.70710678,
                0.97236992, -0.23344536],
              [ 0.05908783, -0.90775218, -0.97091746, …, -0.5       ,
                0.97029573, -0.2419219 ]],

             …,

             [[ 0.05205736, -2.21555971, -2.17368271, …,  0.25881905,
               -0.62251464,  0.78260816],
              [ 0.04493744, -2.35426657, -2.35113988, …,  0.5       ,
               -0.61566148,  0.78801075]],

             [[ 0.04493744, -2.35426657, -2.35113988, …,  0.5       ,
               -0.61566148,  0.78801075],
              [ 0.04880258, -2.41371237, -2.44972719, …,  0.70710678,
               -0.60876143,  0.79335334]],

             [[ 0.04880258, -2.41371237, -2.44972719, …,  0.70710678,
               -0.60876143,  0.79335334],
              [ 0.04958546, -2.4533429 , -2.44972719, …,  0.8660254 ,
               -0.60181502,  0.79863551]]])
```

```
[ ]: model5 = Sequential()
     model5.add(InputLayer((2, 5)))
     model5.add(LSTM(64))
     model5.add(Dense(8, 'relu'))
     model5.add(Dense(1, 'linear'))

     model5.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 64)                17920
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 520
_________________________________________________________________
```

```
dense_3 (Dense)              (None, 1)                 9
=================================================================
Total params: 18,449
Trainable params: 18,449
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
cp5 = ModelCheckpoint('model5/', save_best_only=True)
model5.compile(loss=MeanSquaredError(), optimizer=Adam(learning_rate=0.0001),␣
 ↪metrics=[RootMeanSquaredError()])
```

[ ]:
```
model5.fit(X3_train, y3_train, validation_data=(X3_val, y3_val), epochs=50,␣
 ↪callbacks=[cp4])
```

```
Epoch 1/50
63/63 [==============================] - 3s 11ms/step - loss: 0.4185 -
root_mean_squared_error: 0.6445 - val_loss: 0.1311 -
val_root_mean_squared_error: 0.3621
Epoch 2/50
63/63 [==============================] - 0s 3ms/step - loss: 0.0884 -
root_mean_squared_error: 0.2944 - val_loss: 0.0028 -
val_root_mean_squared_error: 0.0527
Epoch 3/50
63/63 [==============================] - 0s 3ms/step - loss: 0.0044 -
root_mean_squared_error: 0.0654 - val_loss: 2.9874e-04 -
val_root_mean_squared_error: 0.0173
Epoch 4/50
63/63 [==============================] - 0s 3ms/step - loss: 9.1944e-04 -
root_mean_squared_error: 0.0303 - val_loss: 1.9985e-04 -
val_root_mean_squared_error: 0.0141
Epoch 5/50
63/63 [==============================] - 0s 3ms/step - loss: 6.2340e-04 -
root_mean_squared_error: 0.0250 - val_loss: 1.6898e-04 -
val_root_mean_squared_error: 0.0130
Epoch 6/50
63/63 [==============================] - 0s 3ms/step - loss: 4.3252e-04 -
root_mean_squared_error: 0.0208 - val_loss: 1.4422e-04 -
val_root_mean_squared_error: 0.0120
Epoch 7/50
63/63 [==============================] - 0s 3ms/step - loss: 3.4575e-04 -
root_mean_squared_error: 0.0186 - val_loss: 1.4803e-04 -
val_root_mean_squared_error: 0.0122
Epoch 8/50
63/63 [==============================] - 0s 3ms/step - loss: 3.5013e-04 -
root_mean_squared_error: 0.0187 - val_loss: 1.4421e-04 -
val_root_mean_squared_error: 0.0120
Epoch 9/50
63/63 [==============================] - 0s 3ms/step - loss: 3.2149e-04 -
```

```
root_mean_squared_error: 0.0179 - val_loss: 1.3454e-04 -
val_root_mean_squared_error: 0.0116
Epoch 10/50
63/63 [==============================] - 0s 3ms/step - loss: 3.1599e-04 -
root_mean_squared_error: 0.0177 - val_loss: 1.3909e-04 -
val_root_mean_squared_error: 0.0118
Epoch 11/50
63/63 [==============================] - 0s 3ms/step - loss: 3.1317e-04 -
root_mean_squared_error: 0.0177 - val_loss: 1.5357e-04 -
val_root_mean_squared_error: 0.0124
Epoch 12/50
63/63 [==============================] - 0s 3ms/step - loss: 2.8108e-04 -
root_mean_squared_error: 0.0168 - val_loss: 1.6112e-04 -
val_root_mean_squared_error: 0.0127
Epoch 13/50
63/63 [==============================] - 0s 3ms/step - loss: 2.9964e-04 -
root_mean_squared_error: 0.0173 - val_loss: 1.4830e-04 -
val_root_mean_squared_error: 0.0122
Epoch 14/50
63/63 [==============================] - 0s 3ms/step - loss: 2.9277e-04 -
root_mean_squared_error: 0.0171 - val_loss: 1.6263e-04 -
val_root_mean_squared_error: 0.0128
Epoch 15/50
63/63 [==============================] - 0s 3ms/step - loss: 2.6987e-04 -
root_mean_squared_error: 0.0164 - val_loss: 1.4308e-04 -
val_root_mean_squared_error: 0.0120
Epoch 16/50
63/63 [==============================] - 0s 3ms/step - loss: 2.9847e-04 -
root_mean_squared_error: 0.0173 - val_loss: 1.4204e-04 -
val_root_mean_squared_error: 0.0119
Epoch 17/50
63/63 [==============================] - 0s 3ms/step - loss: 2.8832e-04 -
root_mean_squared_error: 0.0170 - val_loss: 1.4012e-04 -
val_root_mean_squared_error: 0.0118
Epoch 18/50
63/63 [==============================] - 0s 3ms/step - loss: 2.6566e-04 -
root_mean_squared_error: 0.0163 - val_loss: 1.4856e-04 -
val_root_mean_squared_error: 0.0122
Epoch 19/50
63/63 [==============================] - 0s 3ms/step - loss: 2.7711e-04 -
root_mean_squared_error: 0.0166 - val_loss: 1.3629e-04 -
val_root_mean_squared_error: 0.0117
Epoch 20/50
63/63 [==============================] - 0s 3ms/step - loss: 2.6121e-04 -
root_mean_squared_error: 0.0161 - val_loss: 1.3641e-04 -
val_root_mean_squared_error: 0.0117
Epoch 21/50
63/63 [==============================] - 0s 3ms/step - loss: 2.7026e-04 -
```

```
root_mean_squared_error: 0.0164 - val_loss: 1.3046e-04 -
val_root_mean_squared_error: 0.0114
Epoch 22/50
63/63 [==============================] - 0s 3ms/step - loss: 2.4677e-04 -
root_mean_squared_error: 0.0157 - val_loss: 1.3694e-04 -
val_root_mean_squared_error: 0.0117
Epoch 23/50
63/63 [==============================] - 0s 3ms/step - loss: 2.6986e-04 -
root_mean_squared_error: 0.0164 - val_loss: 1.1593e-04 -
val_root_mean_squared_error: 0.0108
Epoch 24/50
63/63 [==============================] - 0s 3ms/step - loss: 2.5593e-04 -
root_mean_squared_error: 0.0160 - val_loss: 1.1812e-04 -
val_root_mean_squared_error: 0.0109
Epoch 25/50
63/63 [==============================] - 0s 3ms/step - loss: 2.5411e-04 -
root_mean_squared_error: 0.0159 - val_loss: 1.2838e-04 -
val_root_mean_squared_error: 0.0113
Epoch 26/50
63/63 [==============================] - 0s 3ms/step - loss: 2.3537e-04 -
root_mean_squared_error: 0.0153 - val_loss: 1.0657e-04 -
val_root_mean_squared_error: 0.0103

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 27/50
63/63 [==============================] - 0s 3ms/step - loss: 2.5111e-04 -
root_mean_squared_error: 0.0158 - val_loss: 1.0961e-04 -
val_root_mean_squared_error: 0.0105
Epoch 28/50
63/63 [==============================] - 0s 3ms/step - loss: 2.4034e-04 -
root_mean_squared_error: 0.0155 - val_loss: 9.7536e-05 -
val_root_mean_squared_error: 0.0099

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets
```

```
Epoch 29/50
63/63 [==============================] - 0s 3ms/step - loss: 2.2513e-04 -
root_mean_squared_error: 0.0150 - val_loss: 1.0596e-04 -
val_root_mean_squared_error: 0.0103
Epoch 30/50
63/63 [==============================] - 0s 3ms/step - loss: 2.4973e-04 -
root_mean_squared_error: 0.0158 - val_loss: 1.1458e-04 -
val_root_mean_squared_error: 0.0107
Epoch 31/50
63/63 [==============================] - 0s 3ms/step - loss: 2.6756e-04 -
root_mean_squared_error: 0.0163 - val_loss: 9.6184e-05 -
val_root_mean_squared_error: 0.0098

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 32/50
63/63 [==============================] - 0s 3ms/step - loss: 2.1895e-04 -
root_mean_squared_error: 0.0148 - val_loss: 1.1448e-04 -
val_root_mean_squared_error: 0.0107
Epoch 33/50
63/63 [==============================] - 0s 3ms/step - loss: 2.1751e-04 -
root_mean_squared_error: 0.0147 - val_loss: 1.1127e-04 -
val_root_mean_squared_error: 0.0105
Epoch 34/50
63/63 [==============================] - 0s 3ms/step - loss: 2.1650e-04 -
root_mean_squared_error: 0.0147 - val_loss: 1.0547e-04 -
val_root_mean_squared_error: 0.0103
Epoch 35/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0571e-04 -
root_mean_squared_error: 0.0143 - val_loss: 1.0504e-04 -
val_root_mean_squared_error: 0.0102
Epoch 36/50
63/63 [==============================] - 0s 3ms/step - loss: 1.9599e-04 -
root_mean_squared_error: 0.0140 - val_loss: 9.8473e-05 -
val_root_mean_squared_error: 0.0099
Epoch 37/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0086e-04 -
root_mean_squared_error: 0.0142 - val_loss: 1.0190e-04 -
val_root_mean_squared_error: 0.0101
Epoch 38/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0038e-04 -
root_mean_squared_error: 0.0141 - val_loss: 1.1225e-04 -
```

```
val_root_mean_squared_error: 0.0106
Epoch 39/50
63/63 [==============================] - 0s 3ms/step - loss: 1.9818e-04 -
root_mean_squared_error: 0.0141 - val_loss: 1.1714e-04 -
val_root_mean_squared_error: 0.0108
Epoch 40/50
63/63 [==============================] - 0s 3ms/step - loss: 1.9893e-04 -
root_mean_squared_error: 0.0141 - val_loss: 9.5595e-05 -
val_root_mean_squared_error: 0.0098

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 41/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0184e-04 -
root_mean_squared_error: 0.0142 - val_loss: 1.0975e-04 -
val_root_mean_squared_error: 0.0105
Epoch 42/50
63/63 [==============================] - 0s 3ms/step - loss: 1.7770e-04 -
root_mean_squared_error: 0.0133 - val_loss: 9.7894e-05 -
val_root_mean_squared_error: 0.0099
Epoch 43/50
63/63 [==============================] - 0s 3ms/step - loss: 1.9233e-04 -
root_mean_squared_error: 0.0139 - val_loss: 1.3331e-04 -
val_root_mean_squared_error: 0.0115
Epoch 44/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0201e-04 -
root_mean_squared_error: 0.0142 - val_loss: 9.4342e-05 -
val_root_mean_squared_error: 0.0097

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 45/50
63/63 [==============================] - 0s 3ms/step - loss: 2.0066e-04 -
root_mean_squared_error: 0.0141 - val_loss: 9.1251e-05 -
val_root_mean_squared_error: 0.0096
```

```
WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 46/50
63/63 [==============================] - 0s 3ms/step - loss: 1.6242e-04 -
root_mean_squared_error: 0.0127 - val_loss: 1.2395e-04 -
val_root_mean_squared_error: 0.0111
Epoch 47/50
63/63 [==============================] - 0s 3ms/step - loss: 1.8658e-04 -
root_mean_squared_error: 0.0136 - val_loss: 9.3398e-05 -
val_root_mean_squared_error: 0.0097
Epoch 48/50
63/63 [==============================] - 0s 3ms/step - loss: 1.6124e-04 -
root_mean_squared_error: 0.0127 - val_loss: 8.9519e-05 -
val_root_mean_squared_error: 0.0095

WARNING:absl:Found untraced functions such as lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses, lstm_cell_1_layer_call_fn,
lstm_cell_1_layer_call_and_return_conditional_losses,
lstm_cell_1_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: model4\assets

INFO:tensorflow:Assets written to: model4\assets

Epoch 49/50
63/63 [==============================] - 0s 3ms/step - loss: 1.6063e-04 -
root_mean_squared_error: 0.0127 - val_loss: 9.1382e-05 -
val_root_mean_squared_error: 0.0096
Epoch 50/50
63/63 [==============================] - 0s 3ms/step - loss: 1.7035e-04 -
root_mean_squared_error: 0.0130 - val_loss: 1.3130e-04 -
val_root_mean_squared_error: 0.0115
```

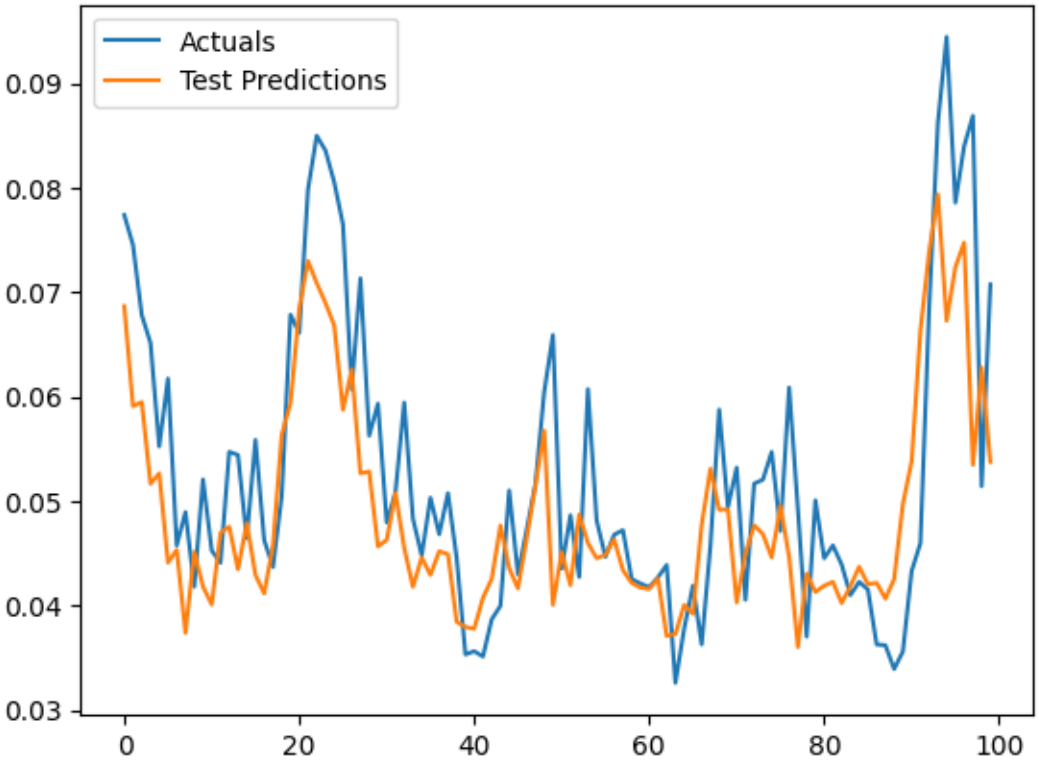
[ ]:
```
<tensorflow.python.keras.callbacks.History at 0x1cb451caa60>
```

[ ]:
```
final, mse = plot_predictions1(model5, X3_test, y3_test)
final
```

[ ]:

	Predictions	Actuals
0	0.077377	0.068645
1	0.074510	0.059088
2	0.067808	0.059466

```
3        0.065130  0.051650
4        0.055228  0.052649
..            …         …
436      0.045758  0.052057
437      0.055592  0.044937
438      0.039222  0.048803
439      0.048076  0.049585
440      0.047218  0.046653

[441 rows x 2 columns]
```



[ ]:
```
final=final.set_index(df['DateTime'][2437:].index)
print(final)
```

```
                     Predictions   Actuals
DateTime
2020-11-12 15:00:00     0.077377  0.068645
2020-11-12 16:00:00     0.074510  0.059088
2020-11-12 17:00:00     0.067808  0.059466
2020-11-12 18:00:00     0.065130  0.051650
2020-11-12 19:00:00     0.055228  0.052649
…                            …         …
```

```
2020-11-30 19:00:00     0.045758  0.052057
2020-11-30 20:00:00     0.055592  0.044937
2020-11-30 21:00:00     0.039222  0.048803
2020-11-30 22:00:00     0.048076  0.049585
2020-11-30 23:00:00     0.047218  0.046653

[441 rows x 2 columns]
```

[ ]: