Forecasting_NG_Demand_Supply

May 4, 2021

1 Forcasting Natural Gas(NG) Demand/Supply

1.1 Abstract

Natural Gas(NG) is one of the valuable ones among the other energy resources. It is used as a heating source for homes and businesses through city gas companies and utilized as a raw material for power plants to generate electricity. Through this, it can be seen that various purposes of NG demand arise in the different fields. In addition, it is essential to identify accurate demand for NG as there is growing volatility in energy demand depending on the direction of the government's environmental policy.

This project focuses on building the model of forecasting the NG demand and supply amount of South Korea, which relies on imports for much of its energy sources. Datasets for training include various fields such as weather and prices of other energy resources, which are open-source. Also, those are trained by using deep learning methods such as the multi-layer perceptron(MLP) with long short-term memory(LSTM), using Tensorflow. In addition, a combination of the dataset from various factors is created by using pandas for training scenario-wise, and the results are compared by changing the variables and analyzed by different viewpoints.

[1]:
```
!pip3 install cloudmesh-common
```

```
Collecting cloudmesh-common
  Downloading https://files.pythonhosted.org/packages/08/b9/60e838cd76b05e
1991ffed2d1387c461a2fefd1e0aa09b230bff0624ff69/cloudmesh_common-4.3.65-py2.py3
-none-any.whl (80kB)
     || 81kB 3.5MB/s
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (4.41.1)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (2.8.1)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (2.23.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.7/dist-packages
(from cloudmesh-common) (5.4.8)
Collecting pyfiglet
  Downloading https://files.pythonhosted.org/packages/33/07/fcfdd7a2872f5b
348953de35acce1544dab0c1e8368dca54279b1cde5c15/pyfiglet-0.8.post1-py2.py3-none-
any.whl (865kB)
```

```
     || 870kB 10.5MB/s
Requirement already satisfied: pathlib in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (1.0.1)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (0.8.9)
Collecting python-hostlist
  Downloading https://files.pythonhosted.org/packages/2b/4f/f31dd4b4bf1a57a5c295
99e1165d0df70dbdddcfa59a7c1d04ee2ff4ccbd/python-hostlist-1.21.tar.gz
Collecting simplejson
  Downloading https://files.pythonhosted.org/packages/a8/04/377418ac1e530c
e2a196b54c6552c018fdf1fe776718053efb1f216bffcd/simplejson-3.17.2-cp37-cp37m-
manylinux2010_x86_64.whl (128kB)
     || 133kB 34.4MB/s
Collecting oyaml
  Downloading https://files.pythonhosted.org/packages/37/aa/111610d8bf5b1bb7a295
a048fc648cec346347a8b0be5881defd2d1b4a52/oyaml-1.0-py2.py3-none-any.whl
Collecting colorama
  Downloading https://files.pythonhosted.org/packages/44/98/5b86278fbbf250d239ae
0ecb724f8572af1c91f4a11edf4d36a206189440/colorama-0.4.4-py2.py3-none-any.whl
Requirement already satisfied: humanize in /usr/local/lib/python3.7/dist-
packages (from cloudmesh-common) (0.5.1)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages
(from cloudmesh-common) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-
packages (from python-dateutil->cloudmesh-common) (1.15.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7
/dist-packages (from requests->cloudmesh-common) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->cloudmesh-common)
(1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7
/dist-packages (from requests->cloudmesh-common) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->cloudmesh-common) (2.10)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from oyaml->cloudmesh-common) (3.13)
Building wheels for collected packages: python-hostlist
  Building wheel for python-hostlist (setup.py) ... done
  Created wheel for python-hostlist: filename=python_hostlist-1.21-cp37-none-
any.whl size=38931
sha256=fffc1678b9f545997dc94696eb6bff8f68bb638fffe04d5b79bbf06da0be7038
  Stored in directory: /root/.cache/pip/wheels/0b/5b/55/ddcf52288f0b10f4564ca1b2
531594ff7ccc65f487ba8dc437
Successfully built python-hostlist
Installing collected packages: pyfiglet, python-hostlist, simplejson, oyaml,
colorama, cloudmesh-common
Successfully installed cloudmesh-common-4.3.65 colorama-0.4.4 oyaml-1.0
pyfiglet-0.8.post1 python-hostlist-1.21 simplejson-3.17.2
```

[2]:
```
from cloudmesh.common.StopWatch import StopWatch
from cloudmesh.common.Shell import Shell

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, SimpleRNN, InputLayer,␣
 ↪LSTM, Dropout
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.datasets import mnist

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
```

1.2 Data Download

Reference: Using Shell.download - cloudmesh

[3]:
```
file_url_1 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/NaturalGas_Supply_per_Region.csv?raw=true'
file_url_2 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Tem_korea.csv?raw=true'
file_url_3 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Tem_seoul.csv?raw=true'
file_url_4 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Tem_daegu.csv?raw=true'
file_url_5 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Tem_busan.csv?raw=true'
file_url_6 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/precipitation.csv?raw=true'
file_url_7 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Pre_seoul.csv?raw=true'
file_url_8 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Pre_daegu.csv?raw=true'
file_url_9 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Pre_busan.csv?raw=true'
file_url_10 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Crude_Oil_Price.csv?raw=true'
file_url_11 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Coal_CIF_ARA.xls?raw=true'
file_url_12 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Coal_Kalimantan.xls?raw=true'
```

```
file_url_13 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/Coal_Richards_Bay.xls?raw=true'
file_url_14 = 'https://github.com/cybertraining-dsc/sp21-599-356/blob/main/
 ↪project/datasets/exchangerate.csv?raw=true'

destination_1 = '/content/sample_data/NaturalGas_Supply_per_Region.csv'
destination_2 = '/content/sample_data/Tem_korea.csv'
destination_3 = '/content/sample_data/Tem_seoul.csv'
destination_4 = '/content/sample_data/Tem_daegu.csv'
destination_5 = '/content/sample_data/Tem_busan.csv'
destination_6 = '/content/sample_data/precipitation.csv'
destination_7 = '/content/sample_data/Pre_seoul.csv'
destination_8 = '/content/sample_data/Pre_daegu.csv'
destination_9 = '/content/sample_data/Pre_busan.csv'
destination_10 = '/content/sample_data/Crude_Oil_Price.csv'
destination_11 = '/content/sample_data/Coal_CIF_ARA.xls'
destination_12 = '/content/sample_data/Coal_Kalimantan.xls'
destination_13 = '/content/sample_data/Coal_Richards_Bay.xls'
destination_14 = '/content/sample_data/exchangerate.csv'

Shell.download(file_url_1, destination_1, provider='system')
Shell.download(file_url_2, destination_2, provider='system')
Shell.download(file_url_3, destination_3, provider='system')
Shell.download(file_url_4, destination_4, provider='system')
Shell.download(file_url_5, destination_5, provider='system')
Shell.download(file_url_6, destination_6, provider='system')
Shell.download(file_url_7, destination_7, provider='system')
Shell.download(file_url_8, destination_8, provider='system')
Shell.download(file_url_9, destination_9, provider='system')
Shell.download(file_url_10, destination_10, provider='system')
Shell.download(file_url_11, destination_11, provider='system')
Shell.download(file_url_12, destination_12, provider='system')
Shell.download(file_url_13, destination_13, provider='system')
Shell.download(file_url_14, destination_14, provider='system')
```

```
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
INFO: Used wget
```

```
INFO: Used wget
INFO: Used wget
```

[3]:
```
'/content/sample_data/exchangerate.csv'
```

1.3 Data Pre-Process

1.3.1 Dataset Load

Load the dataset. Each dataset includes time-based monthly data. 1. The amount of natural gas supply (nine cities seperately) 2. The temperature (one national and three regional cities) 3. The precipitation (one national and three regional cities) 4. The price of crude oil (4-types) 5. The price of coal (3-types) 6. The exchange rate between US Dollars(USD) and South Korea Won(KRW)

[4]:
```
StopWatch.start("data-load")
## Natural gas dataset
ng_sup_df = pd.read_csv('sample_data/NaturalGas_Supply_per_Region.csv')
#ng_pro_df = pd.read_csv('sample_data/NaturalGas_Production.csv',␣
 ↪encoding='CP949')

## Monthly average temperature dataset
tem_total = pd.read_csv('sample_data/Tem_korea.csv', encoding='CP949')
tem_df1 = pd.read_csv('sample_data/Tem_seoul.csv', encoding='CP949')
#tem_df2 = pd.read_csv('sample_data/Tem_Incheon.csv', encoding='CP949')
#tem_df3 = pd.read_csv('sample_data/Tem_suwon_kyunggi.csv', encoding='CP949')
#tem_df4 = pd.read_csv('sample_data/Tem_wonju_gangwon.csv', encoding='CP949')
#tem_df5 = pd.read_csv('sample_data/Tem_daejeon.csv', encoding='CP949')
#tem_df6 = pd.read_csv('sample_data/Tem_gunsan_jeonbuk.csv', encoding='CP949')
#tem_df7 = pd.read_csv('sample_data/Tem_gwangju.csv', encoding='CP949')
tem_df8 = pd.read_csv('sample_data/Tem_daegu.csv', encoding='CP949')
tem_df9 = pd.read_csv('sample_data/Tem_busan.csv', encoding='CP949')

## Monthly avarage precipitation dataset
pre_total = pd.read_csv('sample_data/precipitation.csv', encoding='CP949')
pre_df1 = pd.read_csv('sample_data/Pre_seoul.csv', encoding='CP949')
#pre_df2 = pd.read_csv('sample_data/Pre_incheon.csv', encoding='CP949')
#pre_df3 = pd.read_csv('sample_data/Pre_suwon_kyunggi.csv', encoding='CP949')
#pre_df4 = pd.read_csv('sample_data/Pre_wonju_gangwon.csv', encoding='CP949')
#pre_df5 = pd.read_csv('sample_data/Pre_daejeon.csv', encoding='CP949')
#pre_df6 = pd.read_csv('sample_data/Pre_gunsan_jeonbuk.csv', encoding='CP949')
#pre_df7 = pd.read_csv('sample_data/Pre_gwangju.csv', encoding='CP949')
pre_df8 = pd.read_csv('sample_data/Pre_daegu.csv', encoding='CP949')
pre_df9 = pd.read_csv('sample_data/Pre_busan.csv', encoding='CP949')

## Crude oil price dataset
oil_df = pd.read_csv('sample_data/Crude_Oil_Price.csv')

## Coal price dataset
```

```
coal_df1 = pd.read_excel('sample_data/Coal_CIF_ARA.xls')
coal_df2  = pd.read_excel('sample_data/Coal_Kalimantan.xls')
coal_df3 = pd.read_excel('sample_data/Coal_Richards_Bay.xls')

## Exchange rate dataset
ex_df1 = pd.read_csv('sample_data/exchangerate.csv', encoding='CP949')
StopWatch.stop("data-load")
```

1.3.2 Data preprocess_1

Preprocess data using pandas and create objective dataset.

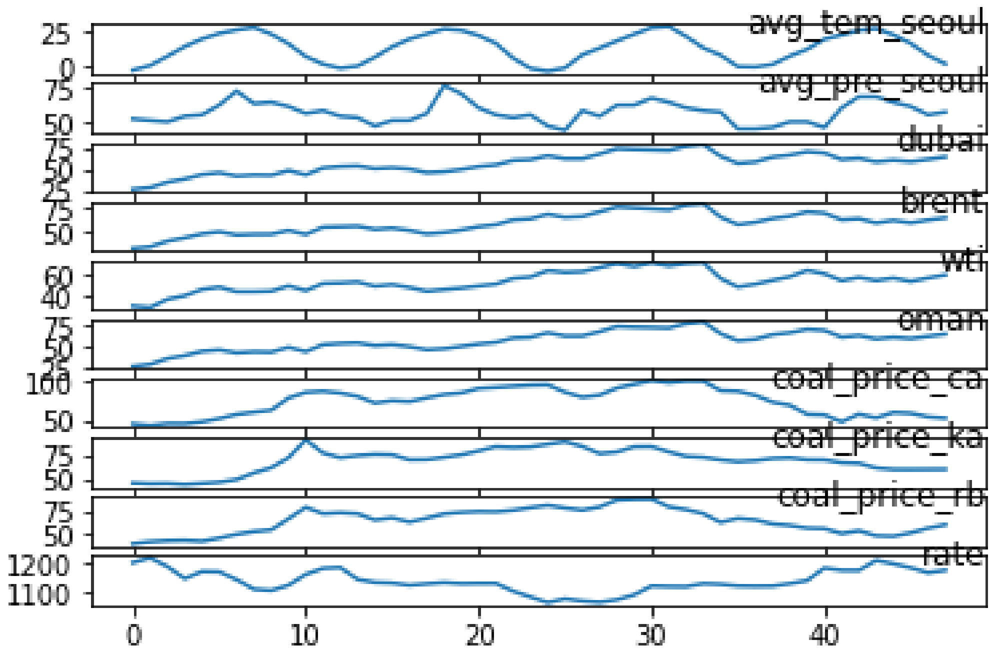
[5]:
```
StopWatch.start("data-preprocess_1")
## Change column names to lowercase
df_names = [ng_sup_df, oil_df, coal_df1, coal_df2, coal_df3]
for name in df_names:
  name.columns = name.columns.str.lower()

## Split natural gas dataset
total_ng_df = ng_sup_df.iloc[:,2:]

## Consolidate temperature dataset
tem_df = pd.concat([tem_df1['avg_tem'],
                    tem_df8['avg_tem'],
                    tem_df9['avg_tem']],axis=1)
tem_df.columns = ['avg_tem_seoul',
                  'avg_tem_daegu',
                  'avg_tem_busan']

## Consolidate precipitation dataset
pre_df = pd.concat([pre_df1['avg_precipitation'],
                    pre_df8['avg_precipitation'],
                    pre_df9['avg_precipitation']],axis=1)
pre_df.columns = ['avg_pre_seoul',
                  'avg_pre_daegu',
                  'avg_pre_busan']

## Consolidate coal price dataset
coal_df = pd.
 ↪concat([coal_df1['price'],coal_df2['price'],coal_df3['price']],axis=1)
coal_df.columns = ['coal_price_ca', 'coal_price_ka', 'coal_price_rb']

## Change exchange rate dataset shape
ex_df = ex_df1.transpose()
ex_df.columns = ['rate']
ex_df.index = list(range(48))
```

```
## Build objective dataset
df_total = pd.concat([ng_sup_df['seoul'], tem_df['avg_tem_seoul'],
                      pre_df['avg_pre_seoul'],
                      oil_df[['dubai','brent','wti','oman']],
                      coal_df, ex_df['rate']], axis=1)
df_total = df_total.iloc[:-1,:]
StopWatch.stop("data-preprocess_1")
```

[7]:
```
df_total.head()
```

[7]:
```
       seoul  avg_tem_seoul  ...  coal_price_rb     rate
0  1110948.0           -3.2  ...          40.02  1201.67
1   911323.0            0.2  ...          42.06  1217.35
2   718859.0            7.0  ...          43.16  1188.21
3   417299.0           14.1  ...          43.55  1147.51
4   354428.0           19.6  ...          42.58  1171.51

[5 rows x 11 columns]
```

[8]:
```
## Plot each column data
fig = plt.figure()
for i in range(1,df_total.shape[-1]):
  ax = fig.add_subplot(df_total.shape[-1]-1,1,i)
  ax.plot(df_total.iloc[:,i])
  ax.set_title(df_total.columns[i], y=0.5, loc='right')
plt.show()
```



[9]:
```
## Make function which can convert normalized dataset to supervised dataset
def convert_dataset(dataset, num_i, num_o, dropnan=True):
  columns = []
  col_names = []
  conv_df = pd.DataFrame(dataset)
  for i in range(num_i, 0, -1):
    columns.append(conv_df.shift(periods=i))
    for j in range(dataset.shape[-1]):
      col_names.append('column{}(t-{})'.format(j+1,i))
  for i in range(num_o):
    columns.append(conv_df.shift(periods=-i))
    if i == 0:
      for j in range(dataset.shape[-1]):
        col_names.append('column{}(t)'.format(j+1))
    else:
      for j in range(dataset.shape[-1]):
        col_names.append('column{}(t+{})'.format(j+1, i))

  new_df = pd.concat(columns, axis=1)
  new_df.columns = col_names
  if dropnan:
    new_df.dropna(inplace=True)
  return new_df
```

[10]:
```
## Make function which can make train and test dataset
def processed_dataset(dataset_norm, time_interval, boundary):
  new_df = convert_dataset(dataset_norm, time_interval, 1)
  new_df.drop(new_df.columns[list(range(dataset_norm.shape[-1]+1,␣
↪2*dataset_norm.shape[-1]))], axis=1, inplace=True)

  train = new_df.values[:boundary, :]
  test = new_df.values[boundary:,:]

  x_train, y_train = train[:, :-1], train[:, -1]
  x_test, y_test = test[:, :-1], test[:,-1]

  x_train = x_train.reshape(x_train.shape[0], 1, x_train.shape[-1])
  x_test = x_test.reshape(x_test.shape[0], 1, x_test.shape[-1])

  return x_train, y_train, x_test, y_test
```

[11]:
```
## Make function which can build the network model
def define_model(x_train, dropout, learning_rate):
  model = Sequential()
  ## LSTM Layers
  model.add(LSTM(100, input_shape = (x_train.shape[1], x_train.shape[2]),
                return_sequences=True))
  model.add(LSTM(100, dropout=dropout,
                return_sequences=False))

  ## MLP Layers
  model.add(Dense(100))
  model.add(Activation('relu'))
  model.add(Dropout(dropout))
  model.add(Dense(100))
  model.add(Activation('relu'))
  model.add(Dropout(dropout))
  model.add(Dense(1))
  model.add(Activation('relu'))

  opt = tf.keras.optimizers.Adam(learning_rate=learning_rate)
  model.compile(loss='mae', optimizer=opt)
  model.summary()

  return model
```

[12]:
```
## Make function which can train model
def train_model(model, x_train, y_train, x_test, y_test, epoch):
  history = model.fit(x_train, y_train,
                      epochs=epoch,
                      validation_data=(x_test, y_test))
  return history
```

[13]:
```
## Make function which can plot loss
def loss_plot(history):
  k = list(range(1, len(history.history['loss'])+1))
  plt.plot(k, history.history['loss'], label='Train Loss')
  plt.plot(k, history.history['val_loss'], label='Validation Loss')
  plt.ylabel('Mean Absolute Error')
  plt.xlabel('Epoch')
  plt.legend()

  return plt.show()
```

[14]:
```
## Make function which can convert dataset to original shape
def predicted_model(model, x_test, y_test, scaler):
  y_predicted = model.predict(x_test)
  x_test = x_test.reshape(x_test.shape[0], x_test.shape[2])
  x_te_re = x_test[:,1:]
  y_test = y_test.reshape(len(y_test), 1)

  inv_y_predicted = np.concatenate((y_predicted, x_te_re), axis=1)
  inv_y_predicted = scaler.inverse_transform(inv_y_predicted)[:,0]

  inv_y = np.concatenate((y_test, x_te_re), axis=1)
  inv_y = scaler.inverse_transform(inv_y)[:,0]
  rmse = np.sqrt(metrics.mean_squared_error(inv_y, inv_y_predicted))
  print('The RMSE is: %.4f' % rmse)

  plt.plot(inv_y, label='Real')
  plt.plot(inv_y_predicted, label='Prediction')
  plt.legend()

  return plt.show()
```

[15]:
```
## Make function which can convert dataset to original shape(with timesteps)
def predicted_time_model(model, x_test, y_test, scaler, time, feature):
  y_predicted = model.predict(x_test)
  x_test = x_test.reshape(x_test.shape[0], (time*feature))
  x_te_re = x_test[:,-(feature-1):]
  y_test = y_test.reshape(len(y_test), 1)

  inv_y_predicted = np.concatenate((y_predicted, x_te_re), axis=1)
  inv_y_predicted = scaler.inverse_transform(inv_y_predicted)[:,0]

  inv_y = np.concatenate((y_test, x_te_re), axis=1)
  inv_y = scaler.inverse_transform(inv_y)[:,0]

  rmse = np.sqrt(metrics.mean_squared_error(inv_y, inv_y_predicted))
  print('The RMSE is : %.4f' % rmse)

  plt.plot(inv_y, label='Real')
  plt.plot(inv_y_predicted, label='Prediction')
  plt.legend()

  return plt.show()
```

1.3.3 Data preprocess_2

Min-Max scaling

In this project, all datasets are rescaled between 0 and 1 by Min-Max scaling, one of the most common normalization methods. If there is a feature with anonymous data, The maximum value(max(x)) of data is converted to 1, and the minimum value(min(x)) of data is converted to 0. The other values between the maximum value and the minimum value get converted to x′, between 0 and 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

Reference: How to use MinMaxScaler
Reference: How to process time series dataset
Reference: How to process multivariate time series dataset

[16]:
```
StopWatch.start("data-preprocess_2")
scaler = MinMaxScaler()
```

```
dataset_norm = scaler.fit_transform(df_total)
x_train, y_train, x_test, y_test = processed_dataset(dataset_norm, 1, 12)
StopWatch.stop("data-preprocess_2")
```

1.4 Define Model

For forecasting the NG supply amount from the time series dataset, MLP with LSTM network model is designed by using Tensorflow. The first and second LSTM layers have 100 units, and a total of 3 layers of MLP follow it. Each MLP layer has 100 neurons instead of the final layer, where its neuron is 1. In addition, dropout was designated to prevent overfitting of data, the Adam is used as an optimizer, and the Rectified Linear Unit(ReLU) as an activation function.

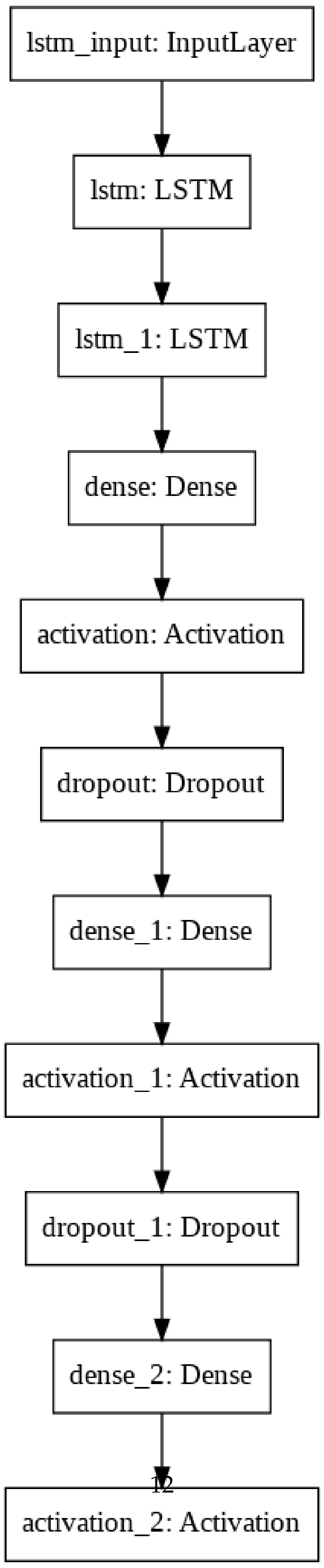
Reference: Introduction to MAE and RMSE

Reference: MLP + LSTM with Tensorflow

[17]:
```
StopWatch.start("compile")
model = define_model(x_train, 0.1, 0.0005)
StopWatch.stop("compile")
tf.keras.utils.plot_model(model)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 1, 100)            44800
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense (Dense)                (None, 100)               10100
_________________________________________________________________
activation (Activation)      (None, 100)               0
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               10100
_________________________________________________________________
activation_1 (Activation)    (None, 100)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 101
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 145,501
Trainable params: 145,501
Non-trainable params: 0
_________________________________________________________________
```

[17]:


lstm_input: InputLayer
lstm: LSTM
lstm_1: LSTM
dense: Dense
activation: Activation
dropout: Dropout
dense_1: Dense
activation_1: Activation
dropout_1: Dropout
dense_2: Dense
activation_2: Activation

1.5 Train

Train the dataset.

[18]:
```
StopWatch.start("train")
history = train_model(model, x_train, y_train, x_test, y_test, 100)
StopWatch.stop("train")
```

```
Epoch 1/100
1/1 [==============================] - 5s 5s/step - loss: 0.2666 - val_loss:
0.3747
Epoch 2/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2617 - val_loss:
0.3664
Epoch 3/100
1/1 [==============================] - 0s 95ms/step - loss: 0.2586 - val_loss:
0.3583
Epoch 4/100
1/1 [==============================] - 0s 92ms/step - loss: 0.2550 - val_loss:
0.3498
Epoch 5/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2497 - val_loss:
0.3409
Epoch 6/100
1/1 [==============================] - 0s 112ms/step - loss: 0.2447 - val_loss:
0.3317
Epoch 7/100
1/1 [==============================] - 0s 107ms/step - loss: 0.2440 - val_loss:
0.3226
Epoch 8/100
1/1 [==============================] - 0s 103ms/step - loss: 0.2350 - val_loss:
0.3132
Epoch 9/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2356 - val_loss:
0.3044
Epoch 10/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2366 - val_loss:
0.2971
Epoch 11/100
1/1 [==============================] - 0s 103ms/step - loss: 0.2356 - val_loss:
0.2909
Epoch 12/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2319 - val_loss:
0.2851
Epoch 13/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2279 - val_loss:
```

```
0.2797
Epoch 14/100
1/1 [==============================] - 0s 102ms/step - loss: 0.2339 - val_loss:
0.2747
Epoch 15/100
1/1 [==============================] - 0s 103ms/step - loss: 0.2298 - val_loss:
0.2701
Epoch 16/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2251 - val_loss:
0.2656
Epoch 17/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2253 - val_loss:
0.2613
Epoch 18/100
1/1 [==============================] - 0s 115ms/step - loss: 0.2275 - val_loss:
0.2581
Epoch 19/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2278 - val_loss:
0.2551
Epoch 20/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2275 - val_loss:
0.2521
Epoch 21/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2211 - val_loss:
0.2500
Epoch 22/100
1/1 [==============================] - 0s 92ms/step - loss: 0.2247 - val_loss:
0.2490
Epoch 23/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2215 - val_loss:
0.2487
Epoch 24/100
1/1 [==============================] - 0s 89ms/step - loss: 0.2226 - val_loss:
0.2491
Epoch 25/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2135 - val_loss:
0.2487
Epoch 26/100
1/1 [==============================] - 0s 102ms/step - loss: 0.2112 - val_loss:
0.2476
Epoch 27/100
1/1 [==============================] - 0s 112ms/step - loss: 0.2202 - val_loss:
0.2464
Epoch 28/100
1/1 [==============================] - 0s 106ms/step - loss: 0.2122 - val_loss:
0.2448
Epoch 29/100
1/1 [==============================] - 0s 91ms/step - loss: 0.2113 - val_loss:
```

```
0.2439
Epoch 30/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2141 - val_loss:
0.2427
Epoch 31/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1986 - val_loss:
0.2405
Epoch 32/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2028 - val_loss:
0.2379
Epoch 33/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2051 - val_loss:
0.2348
Epoch 34/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1989 - val_loss:
0.2309
Epoch 35/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1942 - val_loss:
0.2263
Epoch 36/100
1/1 [==============================] - 0s 115ms/step - loss: 0.1966 - val_loss:
0.2211
Epoch 37/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1943 - val_loss:
0.2157
Epoch 38/100
1/1 [==============================] - 0s 112ms/step - loss: 0.1875 - val_loss:
0.2123
Epoch 39/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1945 - val_loss:
0.2092
Epoch 40/100
1/1 [==============================] - 0s 110ms/step - loss: 0.1815 - val_loss:
0.2060
Epoch 41/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1763 - val_loss:
0.2023
Epoch 42/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1760 - val_loss:
0.1972
Epoch 43/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1563 - val_loss:
0.1931
Epoch 44/100
1/1 [==============================] - 0s 115ms/step - loss: 0.1537 - val_loss:
0.1888
Epoch 45/100
1/1 [==============================] - 0s 107ms/step - loss: 0.1431 - val_loss:
```

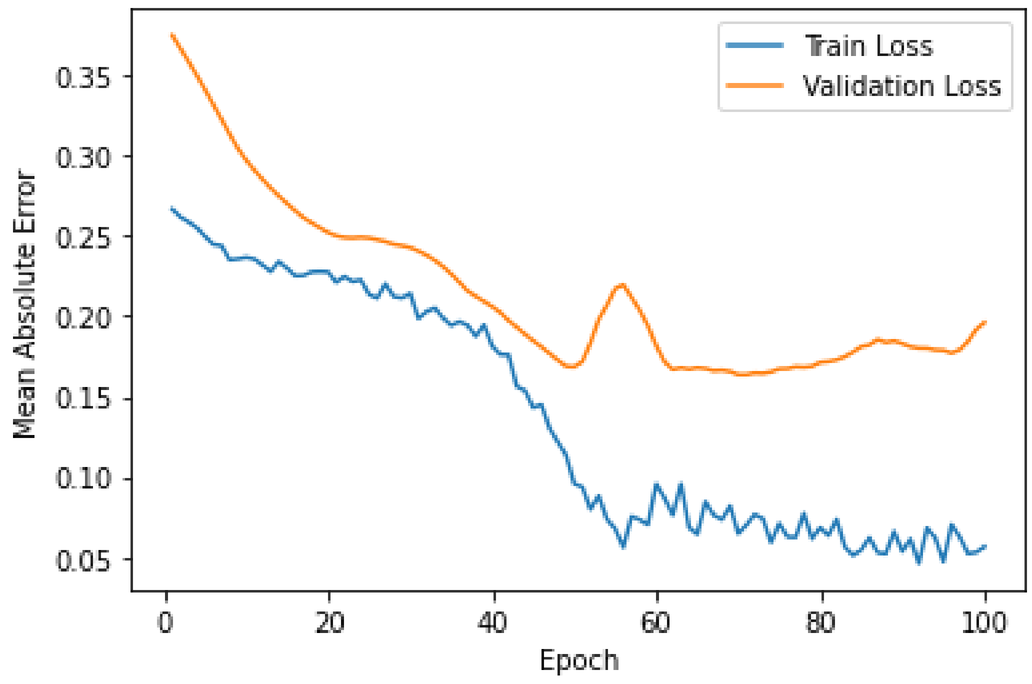
```
0.1848
Epoch 46/100
1/1 [==============================] - 0s 118ms/step - loss: 0.1450 - val_loss:
0.1810
Epoch 47/100
1/1 [==============================] - 0s 120ms/step - loss: 0.1303 - val_loss:
0.1768
Epoch 48/100
1/1 [==============================] - 0s 145ms/step - loss: 0.1217 - val_loss:
0.1727
Epoch 49/100
1/1 [==============================] - 0s 113ms/step - loss: 0.1137 - val_loss:
0.1689
Epoch 50/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0959 - val_loss:
0.1684
Epoch 51/100
1/1 [==============================] - 0s 107ms/step - loss: 0.0938 - val_loss:
0.1718
Epoch 52/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0798 - val_loss:
0.1836
Epoch 53/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0883 - val_loss:
0.1980
Epoch 54/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0737 - val_loss:
0.2068
Epoch 55/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0678 - val_loss:
0.2171
Epoch 56/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0562 - val_loss:
0.2196
Epoch 57/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0755 - val_loss:
0.2117
Epoch 58/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0734 - val_loss:
0.2036
Epoch 59/100
1/1 [==============================] - 0s 89ms/step - loss: 0.0703 - val_loss:
0.1939
Epoch 60/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0956 - val_loss:
0.1816
Epoch 61/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0869 - val_loss:
```

```
0.1717
Epoch 62/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0764 - val_loss:
0.1669
Epoch 63/100
1/1 [==============================] - 0s 110ms/step - loss: 0.0957 - val_loss:
0.1679
Epoch 64/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0685 - val_loss:
0.1671
Epoch 65/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0644 - val_loss:
0.1680
Epoch 66/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0846 - val_loss:
0.1675
Epoch 67/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0763 - val_loss:
0.1661
Epoch 68/100
1/1 [==============================] - 0s 108ms/step - loss: 0.0735 - val_loss:
0.1665
Epoch 69/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0819 - val_loss:
0.1657
Epoch 70/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0647 - val_loss:
0.1637
Epoch 71/100
1/1 [==============================] - 0s 118ms/step - loss: 0.0702 - val_loss:
0.1637
Epoch 72/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0767 - val_loss:
0.1647
Epoch 73/100
1/1 [==============================] - 0s 107ms/step - loss: 0.0741 - val_loss:
0.1645
Epoch 74/100
1/1 [==============================] - 0s 113ms/step - loss: 0.0592 - val_loss:
0.1654
Epoch 75/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0711 - val_loss:
0.1673
Epoch 76/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0628 - val_loss:
0.1674
Epoch 77/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0626 - val_loss:
```

```
0.1686
Epoch 78/100
1/1 [==============================] - 0s 107ms/step - loss: 0.0774 - val_loss:
0.1683
Epoch 79/100
1/1 [==============================] - 0s 116ms/step - loss: 0.0616 - val_loss:
0.1688
Epoch 80/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0685 - val_loss:
0.1714
Epoch 81/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0635 - val_loss:
0.1718
Epoch 82/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0738 - val_loss:
0.1729
Epoch 83/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0565 - val_loss:
0.1749
Epoch 84/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0512 - val_loss:
0.1777
Epoch 85/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0546 - val_loss:
0.1811
Epoch 86/100
1/1 [==============================] - 0s 108ms/step - loss: 0.0623 - val_loss:
0.1822
Epoch 87/100
1/1 [==============================] - 0s 108ms/step - loss: 0.0529 - val_loss:
0.1855
Epoch 88/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0522 - val_loss:
0.1837
Epoch 89/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0662 - val_loss:
0.1847
Epoch 90/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0536 - val_loss:
0.1830
Epoch 91/100
1/1 [==============================] - 0s 121ms/step - loss: 0.0614 - val_loss:
0.1811
Epoch 92/100
1/1 [==============================] - 0s 116ms/step - loss: 0.0464 - val_loss:
0.1802
Epoch 93/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0683 - val_loss:
```

```
0.1801
Epoch 94/100
1/1 [==============================] - 0s 110ms/step - loss: 0.0622 - val_loss:
0.1790
Epoch 95/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0474 - val_loss:
0.1788
Epoch 96/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0704 - val_loss:
0.1771
Epoch 97/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0626 - val_loss:
0.1792
Epoch 98/100
1/1 [==============================] - 0s 94ms/step - loss: 0.0523 - val_loss:
0.1844
Epoch 99/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0531 - val_loss:
0.1917
Epoch 100/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0566 - val_loss:
0.1960
```

[19]:
```
loss_plot(history)
```



Evaluation

Mean Absolute Error(MAE) and Root Mean Squared Error(RMSE) are applied for this time series dataset to evaluate this network model. The MAE measures the average magnitude of the errors and is presented by the formula as following, where n is the number of errors, y_i is the i^th true value, and $\hat{y}_i$ is the i^th predicted value.

$$MAE = \frac{\Sigma_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

Also, The RMSE is used for observing the differences between the real dataset and prediction values. The following is the formula of RMSE, and each value of this is same for MAE.

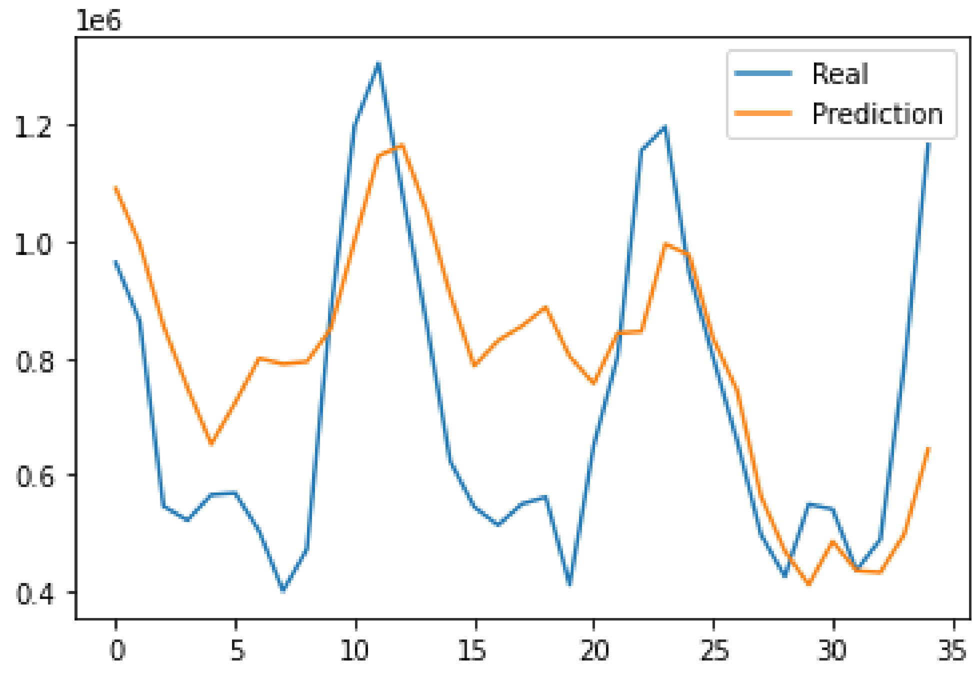
$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

1.6 Predict

Since the datasets used for the training are normalized between 0 and 1, they get converted again to a range of the ground truth values. From these rescaled datasets, it is possible to obtain the RMSE and compare the differences between a actual value and a predicted value.

[20]:
```
StopWatch.start("predict")
predicted_model(model, x_test, y_test, scaler)
StopWatch.stop("predict")
```

```
The RMSE is: 227017.8431
```



Reference: StopWatch and Benchmark

[21]:
```
StopWatch.benchmark()
```

```
+---------------------+----------------------------------------------------------
---------+
| Attribute           | Value
|
|---------------------+----------------------------------------------------------
---------|
| BUG_REPORT_URL      | "https://bugs.launchpad.net/ubuntu/"
|
| DISTRIB_CODENAME    | bionic
|
| DISTRIB_DESCRIPTION | "Ubuntu 18.04.5 LTS"
|
| DISTRIB_ID          | Ubuntu
|
| DISTRIB_RELEASE     | 18.04
|
| HOME_URL            | "https://www.ubuntu.com/"
|
| ID                  | ubuntu
|
| ID_LIKE             | debian
|
| NAME                | "Ubuntu"
|
| PRETTY_NAME         | "Ubuntu 18.04.5 LTS"
|
| PRIVACY_POLICY_URL  | "https://www.ubuntu.com/legal/terms-and-policies
/privacy-policy" |
| SUPPORT_URL         | "https://help.ubuntu.com/"
|
| UBUNTU_CODENAME     | bionic
|
| VERSION             | "18.04.5 LTS (Bionic Beaver)"
|
| VERSION_CODENAME    | bionic
|
| VERSION_ID          | "18.04"
|
| cpu_count           | 2
|
| mem.active          | 1.1 GiB
|
| mem.available       | 11.7 GiB
|
| mem.free            | 9.9 GiB
|
| mem.inactive        | 1.4 GiB
|
| mem.percent         | 8.0 %
|
| mem.total           | 12.7 GiB
|
| mem.used            | 792.3 MiB
|
| platform.version    | #1 SMP Thu Jul 23 08:00:38 PDT 2020
|
| python              | 3.7.10 (default, Feb 20 2021, 21:17:23)
|
|                     | [GCC 7.5.0]
|
| python.pip          | 19.3.1
|
| python.version      | 3.7.10
|
| sys.platform        | linux
|
| uname.machine       | x86_64
|
| uname.node          | 554b9d81d372
|
| uname.processor     | x86_64
|
| uname.release       | 4.19.112+
|
| uname.system        | Linux
|
| uname.version       | #1 SMP Thu Jul 23 08:00:38 PDT 2020
|
| user                | collab
|
+---------------------+----------------------------------------------------------
---------+

+-------------------+----------+--------+--------+---------------------+-------+
--------------+--------+-------+-------------------------------------+
| Name              | Status   |   Time |    Sum | Start               | tag   |
Node         | User   | OS    | Version                             |
|-------------------+----------+--------+--------+---------------------+-------+
--------------+--------+-------+-------------------------------------|
| data-load         | ok       |  0.093 |  0.093 | 2021-05-02 23:09:53 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| data-preprocess_1 | ok       |  0.022 |  0.022 | 2021-05-02 23:09:55 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| data-preprocess_2 | ok       |  0.037 |  0.037 | 2021-05-02 23:10:12 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| compile           | ok       |  0.984 |  0.984 | 2021-05-02 23:10:14 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| train             | ok       | 16.181 | 16.181 | 2021-05-02 23:11:05 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| predict           | ok       |  1.063 |  1.063 | 2021-05-02 23:11:22 |       |
554b9d81d372 | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
+-------------------+----------+--------+--------+---------------------+-------+
--------------+--------+-------+-------------------------------------+

# csv,timer,status,time,sum,start,tag,uname.node,user,uname.system,platform.vers
ion
# csv,data-load,ok,0.093,0.093,2021-05-02 23:09:53,,554b9d81d372,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,data-preprocess_1,ok,0.022,0.022,2021-05-02
23:09:55,,554b9d81d372,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,data-preprocess_2,ok,0.037,0.037,2021-05-02
23:10:12,,554b9d81d372,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,compile,ok,0.984,0.984,2021-05-02 23:10:14,,554b9d81d372,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,train,ok,16.181,16.181,2021-05-02 23:11:05,,554b9d81d372,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,predict,ok,1.063,1.063,2021-05-02 23:11:22,,554b9d81d372,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
```

1.7 Test1 - Climate dataset

In this test, the climate dataset is used to build a model. The dataset includes temperature data and precipitation data.

[32]:
```
StopWatch.start("test1-data-preprocess_1")
df_total_ver_2 = pd.concat([ng_sup_df['seoul'], tem_df['avg_tem_seoul'],
                      pre_df['avg_pre_seoul']], axis=1)
StopWatch.stop("test1-data-preprocess_1")
df_total_ver_2.head()
```

[32]:
```
     seoul  avg_tem_seoul  avg_pre_seoul
0  1110948           -3.2             53
1   911323            0.2             52
2   718859            7.0             51
3   417299           14.1             55
4   354428           19.6             56
```

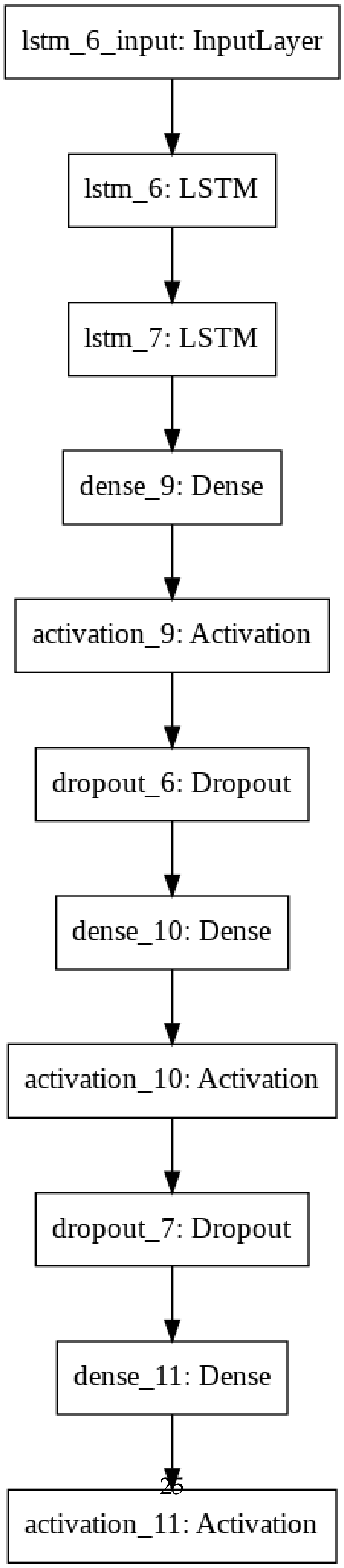
[33]:
```
StopWatch.start("test1-data-preprocess_2")
scaler2 = MinMaxScaler()
dataset_norm2 = scaler2.fit_transform(df_total_ver_2)
```

```
x_train2, y_train2, x_test2, y_test2 = processed_dataset(dataset_norm2, 1, 12)
x_train2.shape, y_train2.shape, x_test2.shape, y_test2.shape
StopWatch.stop("test1-data-preprocess_2")
```

[34]:
```
StopWatch.start("test2-compile")
model2 = define_model(x_train2, 0.1, 0.0005)
StopWatch.stop("test2-compile")
tf.keras.utils.plot_model(model2)
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_6 (LSTM)                (None, 1, 100)            41600
_________________________________________________________________
lstm_7 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense_9 (Dense)              (None, 100)               10100
_________________________________________________________________
activation_9 (Activation)    (None, 100)               0
_________________________________________________________________
dropout_6 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_10 (Dense)             (None, 100)               10100
_________________________________________________________________
activation_10 (Activation)   (None, 100)               0
_________________________________________________________________
dropout_7 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 101
_________________________________________________________________
activation_11 (Activation)   (None, 1)                 0
=================================================================
Total params: 142,301
Trainable params: 142,301
Non-trainable params: 0
_________________________________________________________________
```

[34]:


lstm_6_input: InputLayer
lstm_6: LSTM
lstm_7: LSTM
dense_9: Dense
activation_9: Activation
dropout_6: Dropout
dense_10: Dense
activation_10: Activation
dropout_7: Dropout
dense_11: Dense
activation_11: Activation

[35]:
```
StopWatch.start("test1-train")
history2 = train_model(model2, x_train2, y_train2, x_test2, y_test2, 100)
StopWatch.stop("test1-train")
```

```
Epoch 1/100
1/1 [==============================] - 4s 4s/step - loss: 0.2679 - val_loss:
0.3820
Epoch 2/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2679 - val_loss:
0.3820
Epoch 3/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2678 - val_loss:
0.3801
Epoch 4/100
1/1 [==============================] - 0s 81ms/step - loss: 0.2668 - val_loss:
0.3760
Epoch 5/100
1/1 [==============================] - 0s 81ms/step - loss: 0.2634 - val_loss:
0.3715
Epoch 6/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2606 - val_loss:
0.3671
Epoch 7/100
1/1 [==============================] - 0s 84ms/step - loss: 0.2588 - val_loss:
0.3627
Epoch 8/100
1/1 [==============================] - 0s 110ms/step - loss: 0.2571 - val_loss:
0.3582
Epoch 9/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2536 - val_loss:
0.3533
Epoch 10/100
1/1 [==============================] - 0s 83ms/step - loss: 0.2515 - val_loss:
0.3482
Epoch 11/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2488 - val_loss:
0.3428
Epoch 12/100
1/1 [==============================] - 0s 86ms/step - loss: 0.2456 - val_loss:
0.3371
Epoch 13/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2429 - val_loss:
0.3312
Epoch 14/100
1/1 [==============================] - 0s 84ms/step - loss: 0.2410 - val_loss:
0.3252
```

```
Epoch 15/100
1/1 [==============================] - 0s 88ms/step - loss: 0.2363 - val_loss:
0.3192
Epoch 16/100
1/1 [==============================] - 0s 108ms/step - loss: 0.2352 - val_loss:
0.3131
Epoch 17/100
1/1 [==============================] - 0s 81ms/step - loss: 0.2339 - val_loss:
0.3070
Epoch 18/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2312 - val_loss:
0.3012
Epoch 19/100
1/1 [==============================] - 0s 102ms/step - loss: 0.2318 - val_loss:
0.2966
Epoch 20/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2293 - val_loss:
0.2923
Epoch 21/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2339 - val_loss:
0.2884
Epoch 22/100
1/1 [==============================] - 0s 111ms/step - loss: 0.2295 - val_loss:
0.2848
Epoch 23/100
1/1 [==============================] - 1s 544ms/step - loss: 0.2275 - val_loss:
0.2814
Epoch 24/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2237 - val_loss:
0.2782
Epoch 25/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2294 - val_loss:
0.2752
Epoch 26/100
1/1 [==============================] - 0s 79ms/step - loss: 0.2264 - val_loss:
0.2723
Epoch 27/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2264 - val_loss:
0.2695
Epoch 28/100
1/1 [==============================] - 0s 91ms/step - loss: 0.2254 - val_loss:
0.2669
Epoch 29/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2262 - val_loss:
0.2644
Epoch 30/100
1/1 [==============================] - 0s 111ms/step - loss: 0.2273 - val_loss:
0.2620
```

```
Epoch 31/100
1/1 [==============================] - 0s 103ms/step - loss: 0.2251 - val_loss:
0.2606
Epoch 32/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2215 - val_loss:
0.2590
Epoch 33/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2269 - val_loss:
0.2574
Epoch 34/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2298 - val_loss:
0.2559
Epoch 35/100
1/1 [==============================] - 0s 91ms/step - loss: 0.2203 - val_loss:
0.2542
Epoch 36/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2177 - val_loss:
0.2525
Epoch 37/100
1/1 [==============================] - 0s 86ms/step - loss: 0.2268 - val_loss:
0.2511
Epoch 38/100
1/1 [==============================] - 0s 110ms/step - loss: 0.2161 - val_loss:
0.2496
Epoch 39/100
1/1 [==============================] - 0s 91ms/step - loss: 0.2174 - val_loss:
0.2492
Epoch 40/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2141 - val_loss:
0.2485
Epoch 41/100
1/1 [==============================] - 0s 92ms/step - loss: 0.2146 - val_loss:
0.2476
Epoch 42/100
1/1 [==============================] - 0s 95ms/step - loss: 0.2117 - val_loss:
0.2467
Epoch 43/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2109 - val_loss:
0.2457
Epoch 44/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2096 - val_loss:
0.2446
Epoch 45/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2069 - val_loss:
0.2433
Epoch 46/100
1/1 [==============================] - 0s 113ms/step - loss: 0.1966 - val_loss:
0.2411
```

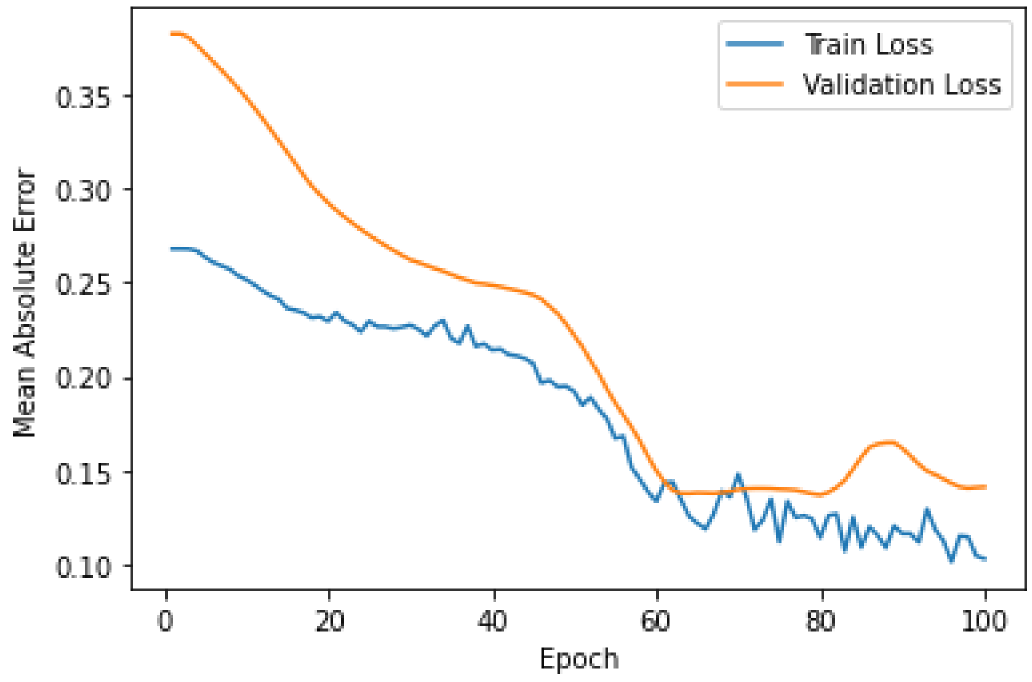
```
Epoch 47/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1980 - val_loss:
0.2373
Epoch 48/100
1/1 [==============================] - 0s 122ms/step - loss: 0.1945 - val_loss:
0.2333
Epoch 49/100
1/1 [==============================] - 0s 85ms/step - loss: 0.1949 - val_loss:
0.2279
Epoch 50/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1920 - val_loss:
0.2218
Epoch 51/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1848 - val_loss:
0.2158
Epoch 52/100
1/1 [==============================] - 0s 101ms/step - loss: 0.1890 - val_loss:
0.2084
Epoch 53/100
1/1 [==============================] - 0s 111ms/step - loss: 0.1826 - val_loss:
0.2016
Epoch 54/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1775 - val_loss:
0.1933
Epoch 55/100
1/1 [==============================] - 0s 114ms/step - loss: 0.1671 - val_loss:
0.1860
Epoch 56/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1683 - val_loss:
0.1796
Epoch 57/100
1/1 [==============================] - 0s 114ms/step - loss: 0.1517 - val_loss:
0.1731
Epoch 58/100
1/1 [==============================] - 0s 104ms/step - loss: 0.1456 - val_loss:
0.1658
Epoch 59/100
1/1 [==============================] - 0s 103ms/step - loss: 0.1389 - val_loss:
0.1576
Epoch 60/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1337 - val_loss:
0.1499
Epoch 61/100
1/1 [==============================] - 0s 106ms/step - loss: 0.1436 - val_loss:
0.1440
Epoch 62/100
1/1 [==============================] - 0s 109ms/step - loss: 0.1440 - val_loss:
0.1395
```

```
Epoch 63/100
1/1 [==============================] - 0s 103ms/step - loss: 0.1346 - val_loss:
0.1380
Epoch 64/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1258 - val_loss:
0.1379
Epoch 65/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1221 - val_loss:
0.1382
Epoch 66/100
1/1 [==============================] - 0s 80ms/step - loss: 0.1191 - val_loss:
0.1381
Epoch 67/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1270 - val_loss:
0.1380
Epoch 68/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1393 - val_loss:
0.1384
Epoch 69/100
1/1 [==============================] - 0s 84ms/step - loss: 0.1362 - val_loss:
0.1391
Epoch 70/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1480 - val_loss:
0.1399
Epoch 71/100
1/1 [==============================] - 0s 88ms/step - loss: 0.1359 - val_loss:
0.1404
Epoch 72/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1186 - val_loss:
0.1406
Epoch 73/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1239 - val_loss:
0.1406
Epoch 74/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1350 - val_loss:
0.1405
Epoch 75/100
1/1 [==============================] - 0s 112ms/step - loss: 0.1121 - val_loss:
0.1401
Epoch 76/100
1/1 [==============================] - 0s 109ms/step - loss: 0.1337 - val_loss:
0.1400
Epoch 77/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1251 - val_loss:
0.1396
Epoch 78/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1262 - val_loss:
0.1387
```

```
Epoch 79/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1244 - val_loss:
0.1378
Epoch 80/100
1/1 [==============================] - 0s 98ms/step - loss: 0.1144 - val_loss:
0.1371
Epoch 81/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1262 - val_loss:
0.1385
Epoch 82/100
1/1 [==============================] - 0s 105ms/step - loss: 0.1270 - val_loss:
0.1412
Epoch 83/100
1/1 [==============================] - 0s 110ms/step - loss: 0.1073 - val_loss:
0.1450
Epoch 84/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1254 - val_loss:
0.1508
Epoch 85/100
1/1 [==============================] - 0s 106ms/step - loss: 0.1092 - val_loss:
0.1571
Epoch 86/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1202 - val_loss:
0.1624
Epoch 87/100
1/1 [==============================] - 0s 105ms/step - loss: 0.1156 - val_loss:
0.1641
Epoch 88/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1090 - val_loss:
0.1648
Epoch 89/100
1/1 [==============================] - 0s 94ms/step - loss: 0.1207 - val_loss:
0.1649
Epoch 90/100
1/1 [==============================] - 0s 103ms/step - loss: 0.1166 - val_loss:
0.1619
Epoch 91/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1166 - val_loss:
0.1578
Epoch 92/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1120 - val_loss:
0.1533
Epoch 93/100
1/1 [==============================] - 0s 106ms/step - loss: 0.1296 - val_loss:
0.1498
Epoch 94/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1188 - val_loss:
0.1478
```

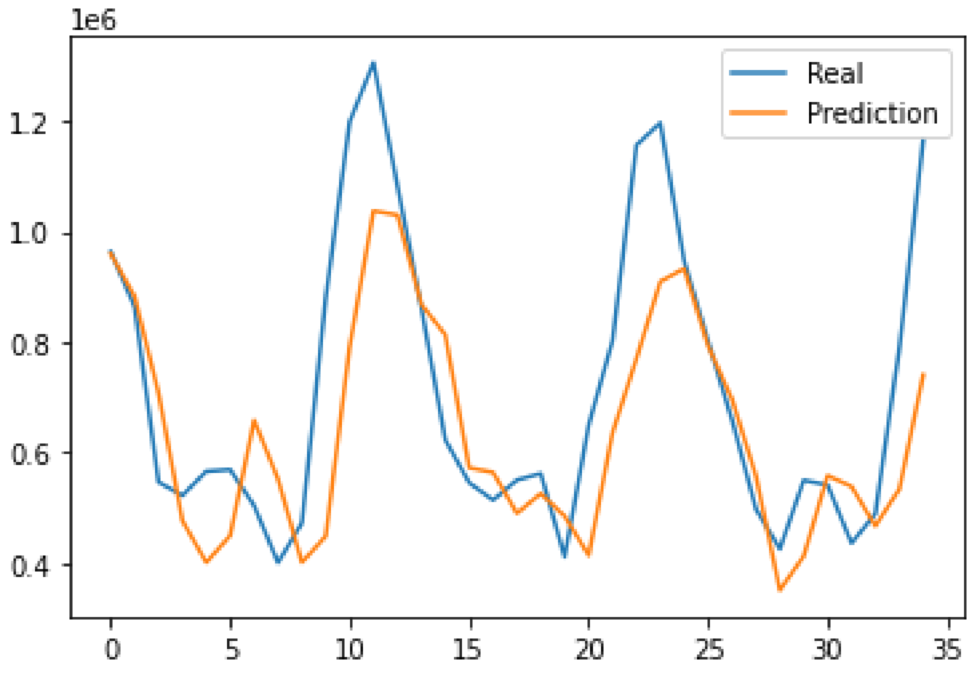
```
Epoch 95/100
1/1 [==============================] - 0s 122ms/step - loss: 0.1127 - val_loss:
0.1457
Epoch 96/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1017 - val_loss:
0.1433
Epoch 97/100
1/1 [==============================] - 0s 110ms/step - loss: 0.1154 - val_loss:
0.1414
Epoch 98/100
1/1 [==============================] - 0s 88ms/step - loss: 0.1149 - val_loss:
0.1407
Epoch 99/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1047 - val_loss:
0.1411
Epoch 100/100
1/1 [==============================] - 0s 101ms/step - loss: 0.1032 - val_loss:
0.1414
```

[36]:
```
loss_plot(history2)
```



[37]:
```
StopWatch.start("test1-predict")
predicted_model(model2, x_test2, y_test2, scaler2)
StopWatch.stop("test1-predict")
```

```
The RMSE is: 185204.7026
```



1.8 Test2 - Temperature dataset

In this test, only a temperature dataset is used to build a model. The values of this dataset show that it is low in winter and high in summer.

[41]:
```
StopWatch.start("test2-data-preprocess_1")
df_total_ver_3 = pd.concat([ng_sup_df['seoul'], tem_df['avg_tem_seoul']],␣
 ↪axis=1)
StopWatch.stop("test2-data-preprocess_1")
df_total_ver_3.head()
```

[41]:
```
     seoul  avg_tem_seoul
0  1110948           -3.2
1   911323            0.2
2   718859            7.0
3   417299           14.1
4   354428           19.6
```

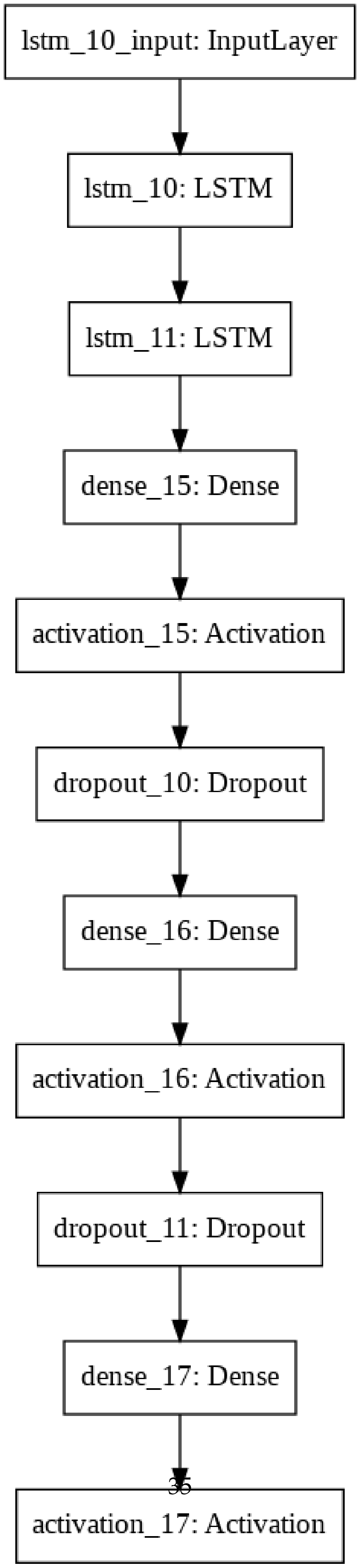
[42]:
```
StopWatch.start("test2-data-preprocess_2")
scaler3 = MinMaxScaler()
dataset_norm3 = scaler3.fit_transform(df_total_ver_3)
x_train3, y_train3, x_test3, y_test3 = processed_dataset(dataset_norm3, 1, 12)
StopWatch.stop("test2-data-preprocess_2")
x_train3.shape, y_train3.shape, x_test3.shape, y_test3.shape
```

[42]:
```
((12, 1, 2), (12,), (35, 1, 2), (35,))
```

[43]:
```
StopWatch.start("test2-compile")
model3 = define_model(x_train3, 0.1, 0.0005)
StopWatch.stop("test2-compile")
tf.keras.utils.plot_model(model3)
```

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_10 (LSTM)               (None, 1, 100)            41200
_________________________________________________________________
lstm_11 (LSTM)               (None, 100)               80400
_________________________________________________________________
dense_15 (Dense)             (None, 100)               10100
_________________________________________________________________
activation_15 (Activation)   (None, 100)               0
_________________________________________________________________
dropout_10 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_16 (Dense)             (None, 100)               10100
_________________________________________________________________
activation_16 (Activation)   (None, 100)               0
_________________________________________________________________
dropout_11 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 101
_________________________________________________________________
activation_17 (Activation)   (None, 1)                 0
=================================================================
Total params: 141,901
Trainable params: 141,901
Non-trainable params: 0
_________________________________________________________________
```

[43]:


lstm_10_input: InputLayer
lstm_10: LSTM
lstm_11: LSTM
dense_15: Dense
activation_15: Activation
dropout_10: Dropout
dense_16: Dense
activation_16: Activation
dropout_11: Dropout
dense_17: Dense
activation_17: Activation

[44]:
```
StopWatch.start("test2-train")
history3 = train_model(model3, x_train3, y_train3, x_test3, y_test3, 100)
StopWatch.stop("test2-train")
```

```
Epoch 1/100
1/1 [==============================] - 4s 4s/step - loss: 0.2677 - val_loss:
0.3768
Epoch 2/100
1/1 [==============================] - 0s 86ms/step - loss: 0.2641 - val_loss:
0.3712
Epoch 3/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2609 - val_loss:
0.3661
Epoch 4/100
1/1 [==============================] - 0s 80ms/step - loss: 0.2584 - val_loss:
0.3608
Epoch 5/100
1/1 [==============================] - 0s 86ms/step - loss: 0.2549 - val_loss:
0.3553
Epoch 6/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2527 - val_loss:
0.3494
Epoch 7/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2506 - val_loss:
0.3432
Epoch 8/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2471 - val_loss:
0.3366
Epoch 9/100
1/1 [==============================] - 0s 110ms/step - loss: 0.2435 - val_loss:
0.3297
Epoch 10/100
1/1 [==============================] - 0s 109ms/step - loss: 0.2398 - val_loss:
0.3228
Epoch 11/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2354 - val_loss:
0.3159
Epoch 12/100
1/1 [==============================] - 0s 112ms/step - loss: 0.2362 - val_loss:
0.3089
Epoch 13/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2325 - val_loss:
0.3023
Epoch 14/100
1/1 [==============================] - 0s 86ms/step - loss: 0.2324 - val_loss:
0.2962
```

```
Epoch 15/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2329 - val_loss:
0.2908
Epoch 16/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2325 - val_loss:
0.2858
Epoch 17/100
1/1 [==============================] - 0s 102ms/step - loss: 0.2323 - val_loss:
0.2813
Epoch 18/100
1/1 [==============================] - 0s 96ms/step - loss: 0.2275 - val_loss:
0.2770
Epoch 19/100
1/1 [==============================] - 0s 105ms/step - loss: 0.2296 - val_loss:
0.2729
Epoch 20/100
1/1 [==============================] - 0s 109ms/step - loss: 0.2294 - val_loss:
0.2692
Epoch 21/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2240 - val_loss:
0.2655
Epoch 22/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2294 - val_loss:
0.2629
Epoch 23/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2277 - val_loss:
0.2604
Epoch 24/100
1/1 [==============================] - 0s 131ms/step - loss: 0.2274 - val_loss:
0.2580
Epoch 25/100
1/1 [==============================] - 0s 112ms/step - loss: 0.2308 - val_loss:
0.2568
Epoch 26/100
1/1 [==============================] - 0s 100ms/step - loss: 0.2249 - val_loss:
0.2555
Epoch 27/100
1/1 [==============================] - 0s 110ms/step - loss: 0.2238 - val_loss:
0.2550
Epoch 28/100
1/1 [==============================] - 0s 107ms/step - loss: 0.2247 - val_loss:
0.2552
Epoch 29/100
1/1 [==============================] - 0s 95ms/step - loss: 0.2234 - val_loss:
0.2552
Epoch 30/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2255 - val_loss:
0.2551
```

```
Epoch 31/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2209 - val_loss:
0.2549
Epoch 32/100
1/1 [==============================] - 0s 105ms/step - loss: 0.2227 - val_loss:
0.2546
Epoch 33/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2184 - val_loss:
0.2551
Epoch 34/100
1/1 [==============================] - 0s 103ms/step - loss: 0.2200 - val_loss:
0.2554
Epoch 35/100
1/1 [==============================] - 0s 111ms/step - loss: 0.2189 - val_loss:
0.2556
Epoch 36/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2137 - val_loss:
0.2555
Epoch 37/100
1/1 [==============================] - 0s 95ms/step - loss: 0.2088 - val_loss:
0.2547
Epoch 38/100
1/1 [==============================] - 0s 109ms/step - loss: 0.2136 - val_loss:
0.2540
Epoch 39/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2060 - val_loss:
0.2528
Epoch 40/100
1/1 [==============================] - 0s 88ms/step - loss: 0.2093 - val_loss:
0.2515
Epoch 41/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2046 - val_loss:
0.2498
Epoch 42/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2072 - val_loss:
0.2477
Epoch 43/100
1/1 [==============================] - 0s 84ms/step - loss: 0.2005 - val_loss:
0.2452
Epoch 44/100
1/1 [==============================] - 0s 94ms/step - loss: 0.1962 - val_loss:
0.2401
Epoch 45/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1948 - val_loss:
0.2347
Epoch 46/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1923 - val_loss:
0.2289
```

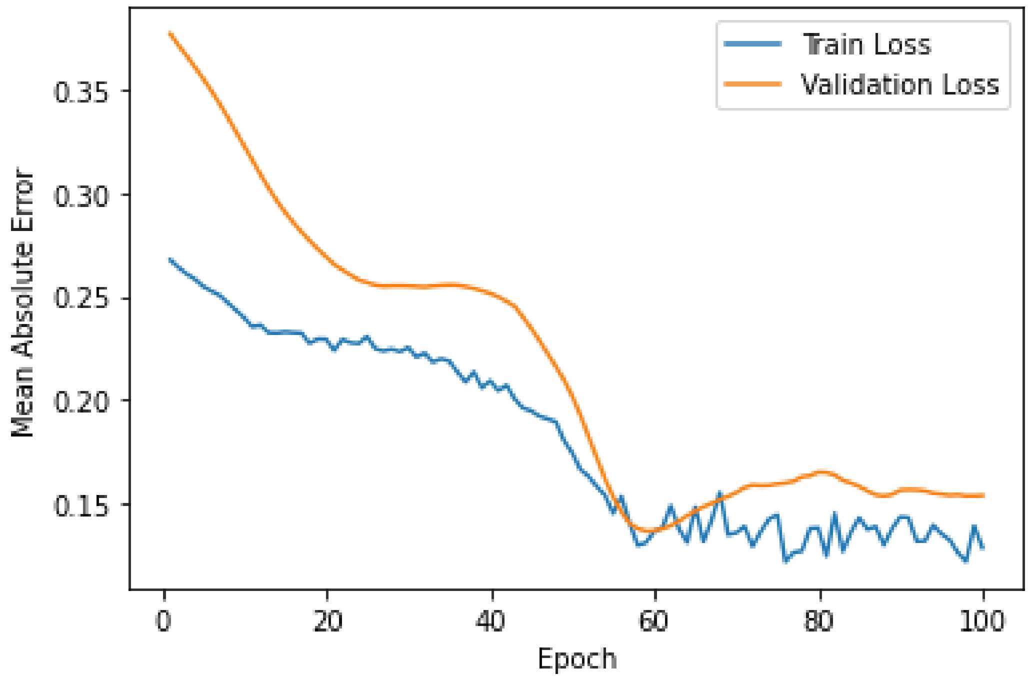
```
Epoch 47/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1909 - val_loss:
0.2225
Epoch 48/100
1/1 [==============================] - 0s 112ms/step - loss: 0.1895 - val_loss:
0.2164
Epoch 49/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1805 - val_loss:
0.2098
Epoch 50/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1746 - val_loss:
0.2020
Epoch 51/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1666 - val_loss:
0.1928
Epoch 52/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1632 - val_loss:
0.1824
Epoch 53/100
1/1 [==============================] - 0s 83ms/step - loss: 0.1584 - val_loss:
0.1722
Epoch 54/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1542 - val_loss:
0.1621
Epoch 55/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1454 - val_loss:
0.1534
Epoch 56/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1536 - val_loss:
0.1461
Epoch 57/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1409 - val_loss:
0.1405
Epoch 58/100
1/1 [==============================] - 0s 109ms/step - loss: 0.1299 - val_loss:
0.1381
Epoch 59/100
1/1 [==============================] - 0s 88ms/step - loss: 0.1312 - val_loss:
0.1368
Epoch 60/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1366 - val_loss:
0.1370
Epoch 61/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1385 - val_loss:
0.1380
Epoch 62/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1490 - val_loss:
0.1394
```

```
Epoch 63/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1380 - val_loss:
0.1417
Epoch 64/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1314 - val_loss:
0.1442
Epoch 65/100
1/1 [==============================] - 0s 94ms/step - loss: 0.1484 - val_loss:
0.1465
Epoch 66/100
1/1 [==============================] - 0s 111ms/step - loss: 0.1318 - val_loss:
0.1486
Epoch 67/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1422 - val_loss:
0.1503
Epoch 68/100
1/1 [==============================] - 0s 103ms/step - loss: 0.1554 - val_loss:
0.1520
Epoch 69/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1350 - val_loss:
0.1536
Epoch 70/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1360 - val_loss:
0.1554
Epoch 71/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1391 - val_loss:
0.1578
Epoch 72/100
1/1 [==============================] - 0s 82ms/step - loss: 0.1295 - val_loss:
0.1591
Epoch 73/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1367 - val_loss:
0.1588
Epoch 74/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1426 - val_loss:
0.1589
Epoch 75/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1447 - val_loss:
0.1596
Epoch 76/100
1/1 [==============================] - 0s 84ms/step - loss: 0.1221 - val_loss:
0.1601
Epoch 77/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1265 - val_loss:
0.1609
Epoch 78/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1272 - val_loss:
0.1630
```

```
Epoch 79/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1381 - val_loss:
0.1635
Epoch 80/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1383 - val_loss:
0.1652
Epoch 81/100
1/1 [==============================] - 0s 88ms/step - loss: 0.1247 - val_loss:
0.1650
Epoch 82/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1454 - val_loss:
0.1639
Epoch 83/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1270 - val_loss:
0.1614
Epoch 84/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1364 - val_loss:
0.1600
Epoch 85/100
1/1 [==============================] - 0s 86ms/step - loss: 0.1433 - val_loss:
0.1584
Epoch 86/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1376 - val_loss:
0.1562
Epoch 87/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1391 - val_loss:
0.1543
Epoch 88/100
1/1 [==============================] - 0s 98ms/step - loss: 0.1300 - val_loss:
0.1538
Epoch 89/100
1/1 [==============================] - 0s 112ms/step - loss: 0.1380 - val_loss:
0.1545
Epoch 90/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1437 - val_loss:
0.1566
Epoch 91/100
1/1 [==============================] - 0s 76ms/step - loss: 0.1433 - val_loss:
0.1568
Epoch 92/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1318 - val_loss:
0.1567
Epoch 93/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1320 - val_loss:
0.1563
Epoch 94/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1396 - val_loss:
0.1551
```

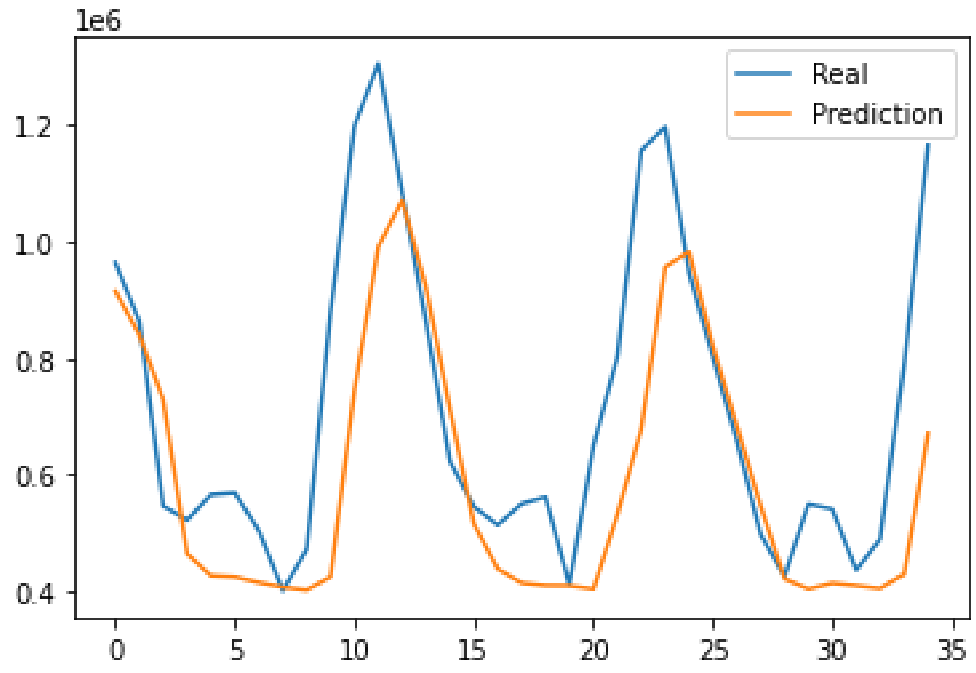
```
Epoch 95/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1355 - val_loss:
0.1547
Epoch 96/100
1/1 [==============================] - 0s 84ms/step - loss: 0.1326 - val_loss:
0.1541
Epoch 97/100
1/1 [==============================] - 0s 101ms/step - loss: 0.1264 - val_loss:
0.1544
Epoch 98/100
1/1 [==============================] - 0s 98ms/step - loss: 0.1222 - val_loss:
0.1538
Epoch 99/100
1/1 [==============================] - 0s 118ms/step - loss: 0.1393 - val_loss:
0.1538
Epoch 100/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1288 - val_loss:
0.1541
```

[45]:
```
loss_plot(history3)
```



[46]:
```
StopWatch.start("test2-predict")
predicted_model(model3, x_test3, y_test3, scaler3)
StopWatch.stop("test2-predict")
```

```
The RMSE is: 207585.1521
```



[120]:

```
```

1.9 Test3 - Applying timesteps

In this test, all dataset are used to build a model. The dataset is same to the first implementation, but timesteps are applied.

[47]:

```
StopWatch.start("test3-data-preprocess_1")
df_total_ver_4 = df_total.copy()
StopWatch.stop("test3-data-preprocess_1")
df_total_ver_4.head()
```

[47]:

```
       seoul  avg_tem_seoul  ...  coal_price_rb     rate
0  1110948.0           -3.2  ...          40.02  1201.67
1   911323.0            0.2  ...          42.06  1217.35
2   718859.0            7.0  ...          43.16  1188.21
3   417299.0           14.1  ...          43.55  1147.51
4   354428.0           19.6  ...          42.58  1171.51

[5 rows x 11 columns]
```

[48]:

```
StopWatch.start("test3-data-preprocess_2")
scaler4 = MinMaxScaler()
dataset_norm4 = scaler4.fit_transform(df_total_ver_4)
```

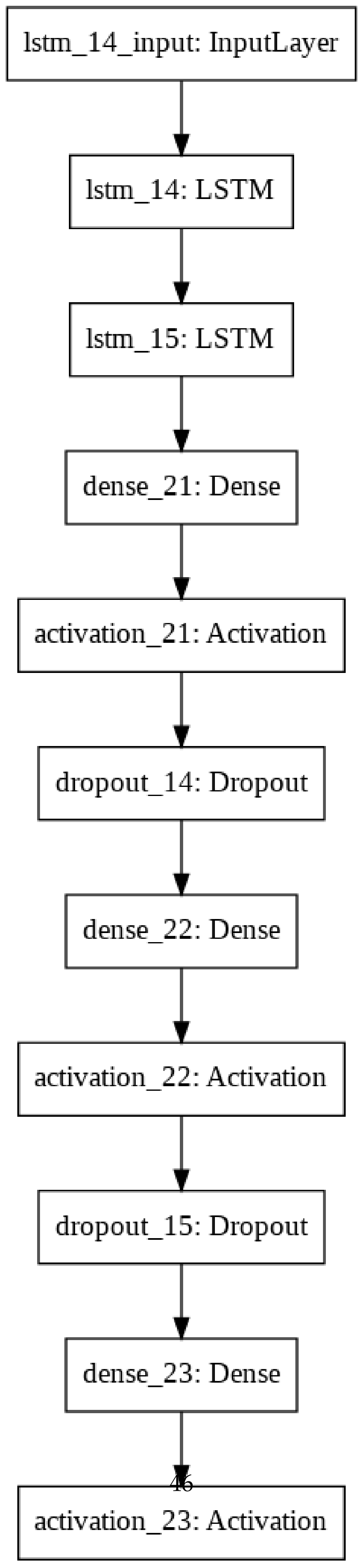
```
months = 2
features = 11
n = months*features

new_df = convert_dataset(dataset_norm4, months, 1)
values = new_df.values
train4 = values[:12, :]
test4 = values[12:,:]

x_train4, y_train4 = train4[:, :n], train4[:, -features]
x_test4, y_test4 = test4[:, :n], test4[:,-features]

x_train4 = x_train4.reshape(x_train4.shape[0], months, features)
x_test4 = x_test4.reshape(x_test4.shape[0], months, features)
StopWatch.stop("test3-data-preprocess_2")
x_train4.shape, y_train4.shape, x_test4.shape, y_test4.shape
```

[48]:
```
((12, 2, 11), (12,), (34, 2, 11), (34,))
```

[50]:
```
StopWatch.start("test3-compile")
model4 = define_model(x_train4, 0.1, 0.0005)
StopWatch.stop("test3-compile")
tf.keras.utils.plot_model(model4)
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_14 (LSTM)               (None, 2, 100)            44800
_________________________________________________________________
lstm_15 (LSTM)               (None, 100)               80400
_________________________________________________________________
dense_21 (Dense)             (None, 100)               10100
_________________________________________________________________
activation_21 (Activation)   (None, 100)               0
_________________________________________________________________
dropout_14 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_22 (Dense)             (None, 100)               10100
_________________________________________________________________
activation_22 (Activation)   (None, 100)               0
_________________________________________________________________
dropout_15 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_23 (Dense)             (None, 1)                 101
_________________________________________________________________
activation_23 (Activation)   (None, 1)                 0
=================================================================
Total params: 145,501
Trainable params: 145,501
Non-trainable params: 0
_________________________________________________________________
```

[50]:


lstm_14_input: InputLayer
lstm_14: LSTM
lstm_15: LSTM
dense_21: Dense
activation_21: Activation
dropout_14: Dropout
dense_22: Dense
activation_22: Activation
dropout_15: Dropout
dense_23: Dense
activation_23: Activation

[51]:
```
StopWatch.start("test3-train")
history4 = train_model(model4, x_train4, y_train4, x_test4, y_test4, 100)
StopWatch.stop("test3-train")
```

```
Epoch 1/100
1/1 [==============================] - 6s 6s/step - loss: 0.2645 - val_loss:
0.3351
Epoch 2/100
1/1 [==============================] - 0s 92ms/step - loss: 0.2549 - val_loss:
0.3173
Epoch 3/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2466 - val_loss:
0.3009
Epoch 4/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2388 - val_loss:
0.2860
Epoch 5/100
1/1 [==============================] - 0s 91ms/step - loss: 0.2324 - val_loss:
0.2716
Epoch 6/100
1/1 [==============================] - 0s 113ms/step - loss: 0.2287 - val_loss:
0.2590
Epoch 7/100
1/1 [==============================] - 0s 105ms/step - loss: 0.2241 - val_loss:
0.2468
Epoch 8/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2196 - val_loss:
0.2354
Epoch 9/100
1/1 [==============================] - 0s 107ms/step - loss: 0.2201 - val_loss:
0.2270
Epoch 10/100
1/1 [==============================] - 0s 95ms/step - loss: 0.2227 - val_loss:
0.2217
Epoch 11/100
1/1 [==============================] - 0s 92ms/step - loss: 0.2153 - val_loss:
0.2184
Epoch 12/100
1/1 [==============================] - 0s 104ms/step - loss: 0.2105 - val_loss:
0.2160
Epoch 13/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2153 - val_loss:
0.2139
Epoch 14/100
1/1 [==============================] - 0s 105ms/step - loss: 0.2031 - val_loss:
0.2119
```

```
Epoch 15/100
1/1 [==============================] - 0s 98ms/step - loss: 0.1998 - val_loss:
0.2099
Epoch 16/100
1/1 [==============================] - 0s 106ms/step - loss: 0.2051 - val_loss:
0.2081
Epoch 17/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1968 - val_loss:
0.2069
Epoch 18/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1882 - val_loss:
0.2054
Epoch 19/100
1/1 [==============================] - 0s 103ms/step - loss: 0.1848 - val_loss:
0.2039
Epoch 20/100
1/1 [==============================] - 0s 113ms/step - loss: 0.1738 - val_loss:
0.2027
Epoch 21/100
1/1 [==============================] - 0s 100ms/step - loss: 0.1753 - val_loss:
0.2014
Epoch 22/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1780 - val_loss:
0.2007
Epoch 23/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1615 - val_loss:
0.2002
Epoch 24/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1546 - val_loss:
0.2009
Epoch 25/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1464 - val_loss:
0.2062
Epoch 26/100
1/1 [==============================] - 0s 101ms/step - loss: 0.1373 - val_loss:
0.2172
Epoch 27/100
1/1 [==============================] - 0s 106ms/step - loss: 0.1323 - val_loss:
0.2350
Epoch 28/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1161 - val_loss:
0.2584
Epoch 29/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1125 - val_loss:
0.2838
Epoch 30/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0933 - val_loss:
0.3184
```

```
Epoch 31/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0791 - val_loss:
0.3451
Epoch 32/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0959 - val_loss:
0.3541
Epoch 33/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0937 - val_loss:
0.3545
Epoch 34/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0747 - val_loss:
0.3485
Epoch 35/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0666 - val_loss:
0.3356
Epoch 36/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0763 - val_loss:
0.3245
Epoch 37/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0884 - val_loss:
0.3193
Epoch 38/100
1/1 [==============================] - 0s 107ms/step - loss: 0.0898 - val_loss:
0.3245
Epoch 39/100
1/1 [==============================] - 0s 93ms/step - loss: 0.0927 - val_loss:
0.3392
Epoch 40/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1010 - val_loss:
0.3470
Epoch 41/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0814 - val_loss:
0.3659
Epoch 42/100
1/1 [==============================] - 0s 106ms/step - loss: 0.0962 - val_loss:
0.3750
Epoch 43/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0752 - val_loss:
0.3789
Epoch 44/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0943 - val_loss:
0.3676
Epoch 45/100
1/1 [==============================] - 0s 106ms/step - loss: 0.0819 - val_loss:
0.3547
Epoch 46/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0767 - val_loss:
0.3372
```

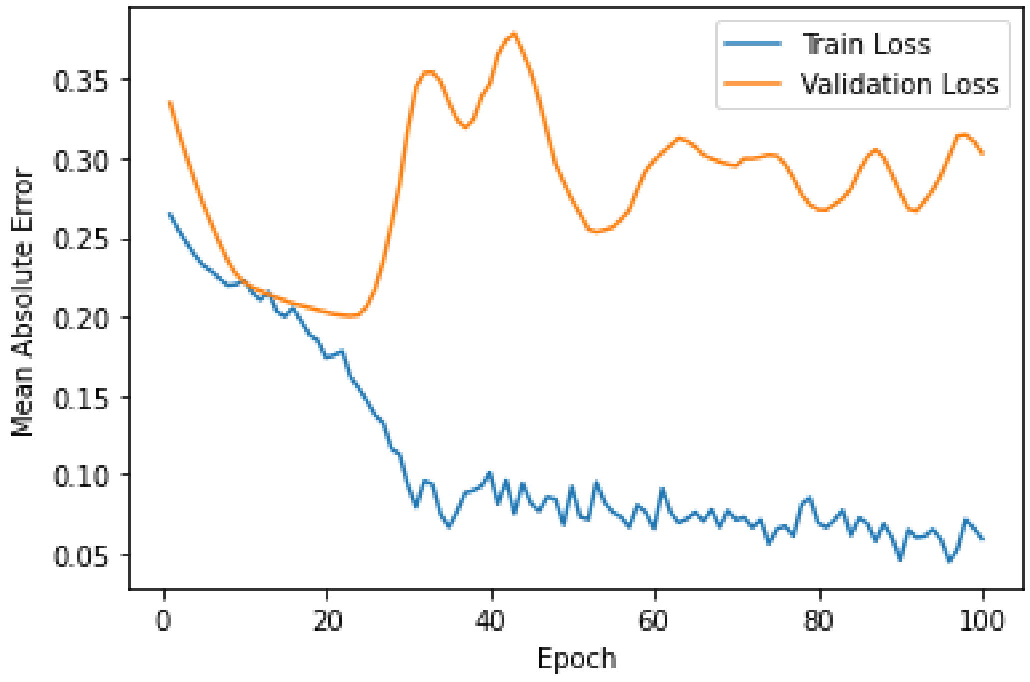
```
Epoch 47/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0856 - val_loss:
0.3164
Epoch 48/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0842 - val_loss:
0.2964
Epoch 49/100
1/1 [==============================] - 0s 87ms/step - loss: 0.0681 - val_loss:
0.2849
Epoch 50/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0922 - val_loss:
0.2738
Epoch 51/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0732 - val_loss:
0.2649
Epoch 52/100
1/1 [==============================] - 0s 82ms/step - loss: 0.0714 - val_loss:
0.2551
Epoch 53/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0950 - val_loss:
0.2533
Epoch 54/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0822 - val_loss:
0.2546
Epoch 55/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0759 - val_loss:
0.2564
Epoch 56/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0730 - val_loss:
0.2614
Epoch 57/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0665 - val_loss:
0.2673
Epoch 58/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0806 - val_loss:
0.2803
Epoch 59/100
1/1 [==============================] - 0s 106ms/step - loss: 0.0761 - val_loss:
0.2919
Epoch 60/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0656 - val_loss:
0.2985
Epoch 61/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0910 - val_loss:
0.3036
Epoch 62/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0759 - val_loss:
0.3082
```

```
Epoch 63/100
1/1 [==============================] - 0s 77ms/step - loss: 0.0694 - val_loss:
0.3126
Epoch 64/100
1/1 [==============================] - 0s 87ms/step - loss: 0.0718 - val_loss:
0.3108
Epoch 65/100
1/1 [==============================] - 0s 107ms/step - loss: 0.0759 - val_loss:
0.3074
Epoch 66/100
1/1 [==============================] - 0s 110ms/step - loss: 0.0705 - val_loss:
0.3021
Epoch 67/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0776 - val_loss:
0.2997
Epoch 68/100
1/1 [==============================] - 0s 109ms/step - loss: 0.0667 - val_loss:
0.2976
Epoch 69/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0769 - val_loss:
0.2960
Epoch 70/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0712 - val_loss:
0.2951
Epoch 71/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0726 - val_loss:
0.2997
Epoch 72/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0662 - val_loss:
0.2997
Epoch 73/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0712 - val_loss:
0.3007
Epoch 74/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0558 - val_loss:
0.3021
Epoch 75/100
1/1 [==============================] - 0s 111ms/step - loss: 0.0656 - val_loss:
0.3014
Epoch 76/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0672 - val_loss:
0.2960
Epoch 77/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0609 - val_loss:
0.2874
Epoch 78/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0810 - val_loss:
0.2770
```

```
Epoch 79/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0853 - val_loss:
0.2706
Epoch 80/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0693 - val_loss:
0.2676
Epoch 81/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0663 - val_loss:
0.2674
Epoch 82/100
1/1 [==============================] - 0s 90ms/step - loss: 0.0711 - val_loss:
0.2707
Epoch 83/100
1/1 [==============================] - 0s 106ms/step - loss: 0.0772 - val_loss:
0.2744
Epoch 84/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0613 - val_loss:
0.2806
Epoch 85/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0724 - val_loss:
0.2919
Epoch 86/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0689 - val_loss:
0.3008
Epoch 87/100
1/1 [==============================] - 0s 108ms/step - loss: 0.0575 - val_loss:
0.3056
Epoch 88/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0687 - val_loss:
0.3003
Epoch 89/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0600 - val_loss:
0.2899
Epoch 90/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0457 - val_loss:
0.2782
Epoch 91/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0649 - val_loss:
0.2680
Epoch 92/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0599 - val_loss:
0.2668
Epoch 93/100
1/1 [==============================] - 0s 84ms/step - loss: 0.0608 - val_loss:
0.2729
Epoch 94/100
1/1 [==============================] - 0s 83ms/step - loss: 0.0651 - val_loss:
0.2799
```

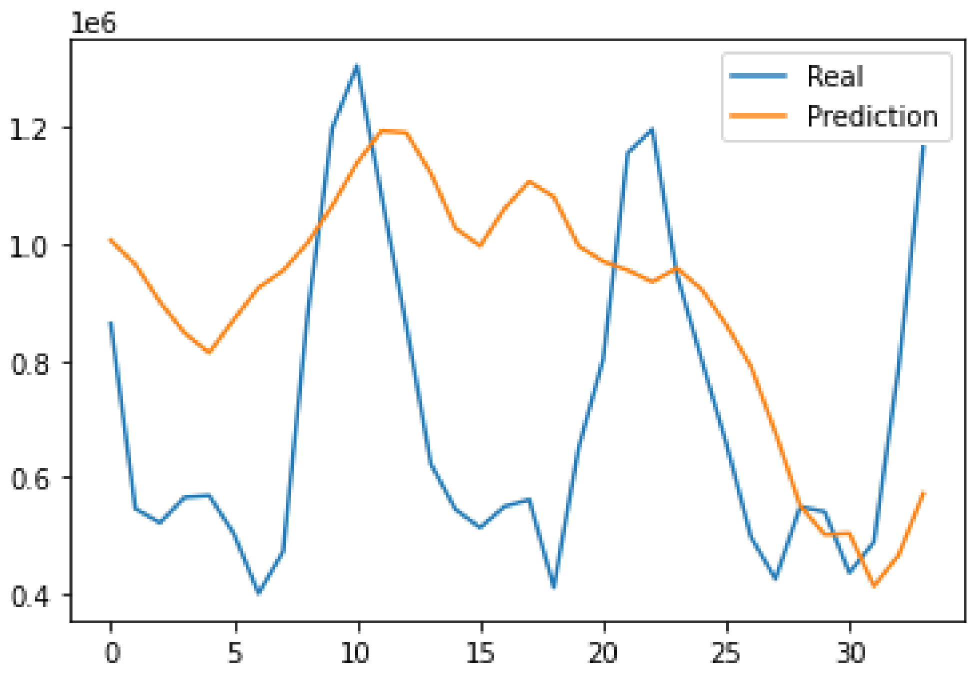
```
Epoch 95/100
1/1 [==============================] - 0s 93ms/step - loss: 0.0587 - val_loss:
0.2892
Epoch 96/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0448 - val_loss:
0.3016
Epoch 97/100
1/1 [==============================] - 0s 103ms/step - loss: 0.0526 - val_loss:
0.3140
Epoch 98/100
1/1 [==============================] - 0s 93ms/step - loss: 0.0713 - val_loss:
0.3148
Epoch 99/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0659 - val_loss:
0.3106
Epoch 100/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0591 - val_loss:
0.3032
```

[52]:
```
loss_plot(history4)
```



[53]:
```
StopWatch.start("test3-predict")
predicted_time_model(model4, x_test4, y_test4, scaler4, months, features)
StopWatch.stop("test3-predict")
```

```
WARNING:tensorflow:5 out of the last 9 calls to <function
Model.make_predict_function.<locals>.predict_function at 0x7f46a9aae5f0>
triggered tf.function retracing. Tracing is expensive and the excessive number
of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2)
passing tensors with different shapes, (3) passing Python objects instead of
tensors. For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
The RMSE is : 340842.9452
```



[155]:

```
```

1.10 Test4 - National dataset

[54]:

```
StopWatch.start("test4-data-preprocess_1")
df_total_country = pd.concat([ng_sup_df.sum(axis=1), tem_total['avg_tem'],
                              pre_total['avg_precipitation'],
                              oil_df[['dubai','brent','wti','oman']],
                              coal_df, ex_df['rate']], axis=1)
df_total_country = df_total_country.iloc[:-1,:]
df_total_country = df_total_country.rename(columns = {0:'total'})
StopWatch.stop("test4-data-preprocess_1")
df_total_country.head()
```

[54]:
```
        total  avg_tem  avg_precipitation  ...  coal_price_ka  coal_price_rb
rate
0  4250103.0      2.8               26.3  ...          46.33          40.02
1201.67
1  3559794.0      8.8               58.2  ...          45.67          42.06
1217.35
2  3182961.0     15.8              169.0  ...          45.89          43.16
1188.21
3  2125051.0     21.8              221.2  ...          44.92          43.55
1147.51
4  1978337.0     26.2              140.0  ...          45.98          42.58
1171.51

[5 rows x 11 columns]
```

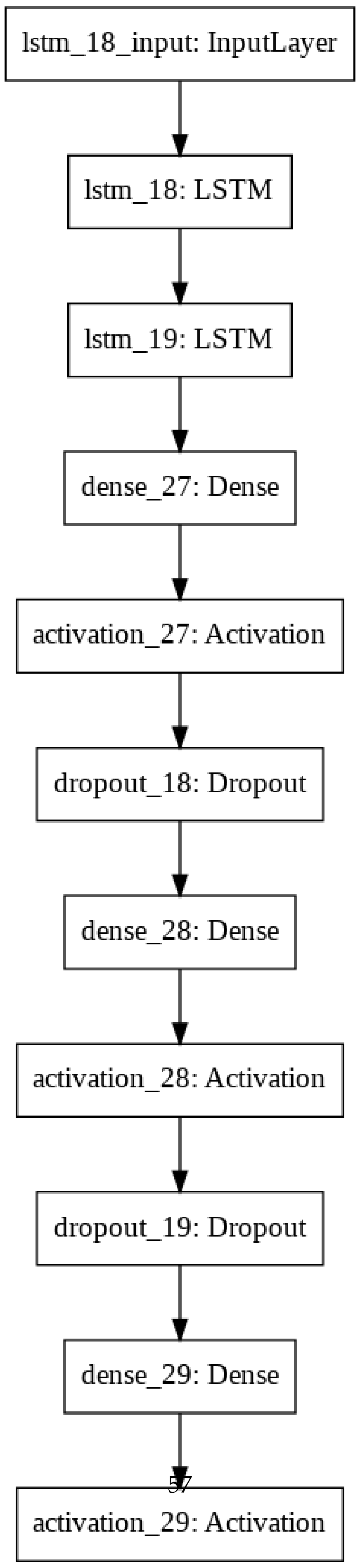
[55]:
```
StopWatch.start("test4-data-preprocess_2")
scaler5 = MinMaxScaler()
dataset_norm5 = scaler5.fit_transform(df_total_country)
x_train5, y_train5, x_test5, y_test5 = processed_dataset(dataset_norm5, 1, 12)
StopWatch.stop("test4-data-preprocess_2")
```

[57]:
```
StopWatch.start("test4-compile")
model5 = define_model(x_train5, 0.1, 0.0005)
StopWatch.stop("test4-compile")
tf.keras.utils.plot_model(model5)
```

```
Model: "sequential_9"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_18 (LSTM)               (None, 1, 100)            44800

_________________________________________________________________
lstm_19 (LSTM)               (None, 100)               80400

_________________________________________________________________
dense_27 (Dense)             (None, 100)               10100

_________________________________________________________________
activation_27 (Activation)   (None, 100)               0

_________________________________________________________________
dropout_18 (Dropout)         (None, 100)               0

_________________________________________________________________
dense_28 (Dense)             (None, 100)               10100

_________________________________________________________________
activation_28 (Activation)   (None, 100)               0

_________________________________________________________________
dropout_19 (Dropout)         (None, 100)               0

_________________________________________________________________
dense_29 (Dense)             (None, 1)                 101

_________________________________________________________________
```

```
activation_29 (Activation)   (None, 1)                 0
=================================================================
Total params: 145,501
Trainable params: 145,501
Non-trainable params: 0

_________________________________________________________________
```

[57]:


lstm_18_input: InputLayer
lstm_18: LSTM
lstm_19: LSTM
dense_27: Dense
activation_27: Activation
dropout_18: Dropout
dense_28: Dense
activation_28: Activation
dropout_19: Dropout
dense_29: Dense
activation_29: Activation

[58]:
```
StopWatch.start("test4-train")
history5 = train_model(model5, x_train5, y_train5, x_test5, y_test5, 100)
StopWatch.stop("test4-train")
```

```
Epoch 1/100
1/1 [==============================] - 5s 5s/step - loss: 0.2538 - val_loss:
0.3711
Epoch 2/100
1/1 [==============================] - 0s 81ms/step - loss: 0.2524 - val_loss:
0.3655
Epoch 3/100
1/1 [==============================] - 0s 87ms/step - loss: 0.2481 - val_loss:
0.3595
Epoch 4/100
1/1 [==============================] - 0s 89ms/step - loss: 0.2427 - val_loss:
0.3534
Epoch 5/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2390 - val_loss:
0.3472
Epoch 6/100
1/1 [==============================] - 0s 106ms/step - loss: 0.2351 - val_loss:
0.3412
Epoch 7/100
1/1 [==============================] - 0s 89ms/step - loss: 0.2302 - val_loss:
0.3349
Epoch 8/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2281 - val_loss:
0.3283
Epoch 9/100
1/1 [==============================] - 0s 93ms/step - loss: 0.2248 - val_loss:
0.3217
Epoch 10/100
1/1 [==============================] - 0s 110ms/step - loss: 0.2180 - val_loss:
0.3155
Epoch 11/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2190 - val_loss:
0.3091
Epoch 12/100
1/1 [==============================] - 0s 108ms/step - loss: 0.2150 - val_loss:
0.3024
Epoch 13/100
1/1 [==============================] - 0s 90ms/step - loss: 0.2148 - val_loss:
0.2959
Epoch 14/100
1/1 [==============================] - 0s 101ms/step - loss: 0.2124 - val_loss:
0.2893
```

```
Epoch 15/100
1/1 [==============================] - 0s 99ms/step - loss: 0.2105 - val_loss:
0.2830
Epoch 16/100
1/1 [==============================] - 0s 111ms/step - loss: 0.2059 - val_loss:
0.2770
Epoch 17/100
1/1 [==============================] - 0s 97ms/step - loss: 0.2000 - val_loss:
0.2711
Epoch 18/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2027 - val_loss:
0.2655
Epoch 19/100
1/1 [==============================] - 0s 94ms/step - loss: 0.2024 - val_loss:
0.2607
Epoch 20/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1972 - val_loss:
0.2560
Epoch 21/100
1/1 [==============================] - 0s 105ms/step - loss: 0.1957 - val_loss:
0.2513
Epoch 22/100
1/1 [==============================] - 0s 102ms/step - loss: 0.2011 - val_loss:
0.2468
Epoch 23/100
1/1 [==============================] - 0s 92ms/step - loss: 0.1961 - val_loss:
0.2424
Epoch 24/100
1/1 [==============================] - 0s 91ms/step - loss: 0.1909 - val_loss:
0.2390
Epoch 25/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1889 - val_loss:
0.2357
Epoch 26/100
1/1 [==============================] - 0s 102ms/step - loss: 0.1824 - val_loss:
0.2328
Epoch 27/100
1/1 [==============================] - 0s 88ms/step - loss: 0.1865 - val_loss:
0.2299
Epoch 28/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1783 - val_loss:
0.2270
Epoch 29/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1886 - val_loss:
0.2246
Epoch 30/100
1/1 [==============================] - 0s 95ms/step - loss: 0.1754 - val_loss:
0.2219
```

```
Epoch 31/100
1/1 [==============================] - 0s 85ms/step - loss: 0.1820 - val_loss:
0.2195
Epoch 32/100
1/1 [==============================] - 0s 96ms/step - loss: 0.1675 - val_loss:
0.2167
Epoch 33/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1724 - val_loss:
0.2141
Epoch 34/100
1/1 [==============================] - 0s 93ms/step - loss: 0.1632 - val_loss:
0.2114
Epoch 35/100
1/1 [==============================] - 0s 105ms/step - loss: 0.1611 - val_loss:
0.2085
Epoch 36/100
1/1 [==============================] - 0s 104ms/step - loss: 0.1506 - val_loss:
0.2052
Epoch 37/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1562 - val_loss:
0.2015
Epoch 38/100
1/1 [==============================] - 0s 104ms/step - loss: 0.1489 - val_loss:
0.1975
Epoch 39/100
1/1 [==============================] - 0s 109ms/step - loss: 0.1403 - val_loss:
0.1932
Epoch 40/100
1/1 [==============================] - 0s 97ms/step - loss: 0.1253 - val_loss:
0.1887
Epoch 41/100
1/1 [==============================] - 0s 90ms/step - loss: 0.1233 - val_loss:
0.1840
Epoch 42/100
1/1 [==============================] - 0s 108ms/step - loss: 0.1130 - val_loss:
0.1804
Epoch 43/100
1/1 [==============================] - 0s 99ms/step - loss: 0.1224 - val_loss:
0.1773
Epoch 44/100
1/1 [==============================] - 0s 87ms/step - loss: 0.1040 - val_loss:
0.1756
Epoch 45/100
1/1 [==============================] - 0s 89ms/step - loss: 0.1099 - val_loss:
0.1761
Epoch 46/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0817 - val_loss:
0.1813
```

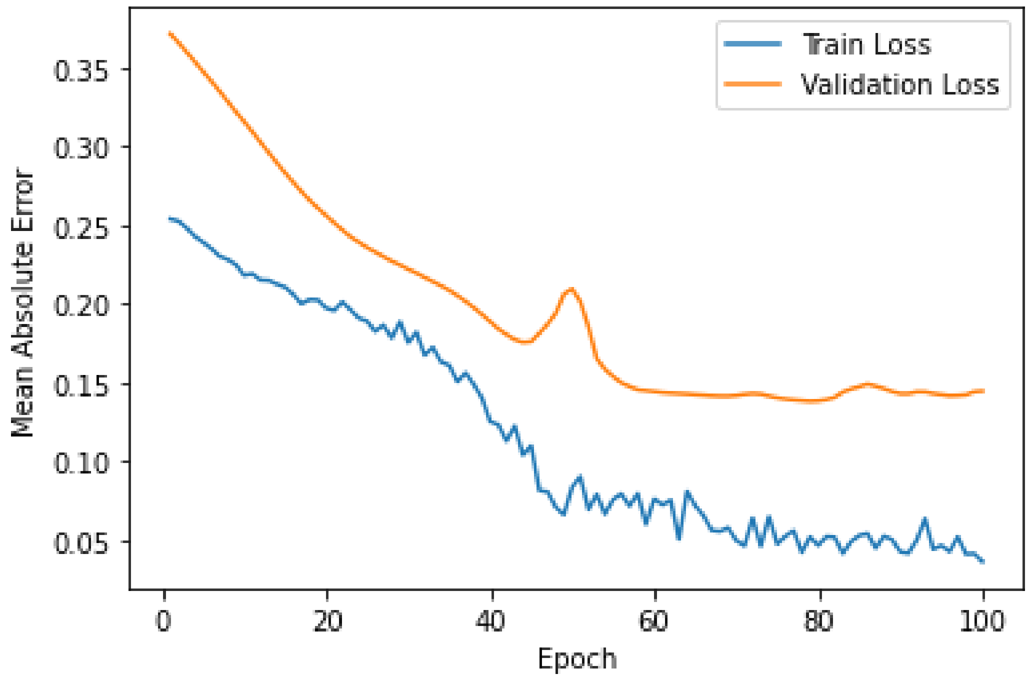
```
Epoch 47/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0809 - val_loss:
0.1870
Epoch 48/100
1/1 [==============================] - 0s 90ms/step - loss: 0.0711 - val_loss:
0.1939
Epoch 49/100
1/1 [==============================] - 0s 94ms/step - loss: 0.0664 - val_loss:
0.2061
Epoch 50/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0838 - val_loss:
0.2094
Epoch 51/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0904 - val_loss:
0.2014
Epoch 52/100
1/1 [==============================] - 0s 94ms/step - loss: 0.0695 - val_loss:
0.1851
Epoch 53/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0793 - val_loss:
0.1652
Epoch 54/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0667 - val_loss:
0.1585
Epoch 55/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0759 - val_loss:
0.1539
Epoch 56/100
1/1 [==============================] - 0s 104ms/step - loss: 0.0796 - val_loss:
0.1499
Epoch 57/100
1/1 [==============================] - 0s 90ms/step - loss: 0.0717 - val_loss:
0.1475
Epoch 58/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0798 - val_loss:
0.1455
Epoch 59/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0602 - val_loss:
0.1449
Epoch 60/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0764 - val_loss:
0.1444
Epoch 61/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0726 - val_loss:
0.1437
Epoch 62/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0756 - val_loss:
0.1434
```

```
Epoch 63/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0508 - val_loss:
0.1431
Epoch 64/100
1/1 [==============================] - 0s 92ms/step - loss: 0.0811 - val_loss:
0.1428
Epoch 65/100
1/1 [==============================] - 0s 86ms/step - loss: 0.0719 - val_loss:
0.1424
Epoch 66/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0657 - val_loss:
0.1421
Epoch 67/100
1/1 [==============================] - 0s 90ms/step - loss: 0.0564 - val_loss:
0.1417
Epoch 68/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0557 - val_loss:
0.1414
Epoch 69/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0580 - val_loss:
0.1414
Epoch 70/100
1/1 [==============================] - 0s 140ms/step - loss: 0.0499 - val_loss:
0.1418
Epoch 71/100
1/1 [==============================] - 0s 83ms/step - loss: 0.0466 - val_loss:
0.1426
Epoch 72/100
1/1 [==============================] - 0s 86ms/step - loss: 0.0642 - val_loss:
0.1431
Epoch 73/100
1/1 [==============================] - 0s 91ms/step - loss: 0.0464 - val_loss:
0.1428
Epoch 74/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0651 - val_loss:
0.1414
Epoch 75/100
1/1 [==============================] - 0s 87ms/step - loss: 0.0476 - val_loss:
0.1404
Epoch 76/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0527 - val_loss:
0.1396
Epoch 77/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0562 - val_loss:
0.1391
Epoch 78/100
1/1 [==============================] - 0s 85ms/step - loss: 0.0424 - val_loss:
0.1386
```

```
Epoch 79/100
1/1 [==============================] - 0s 94ms/step - loss: 0.0523 - val_loss:
0.1382
Epoch 80/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0468 - val_loss:
0.1385
Epoch 81/100
1/1 [==============================] - 0s 112ms/step - loss: 0.0525 - val_loss:
0.1393
Epoch 82/100
1/1 [==============================] - 0s 96ms/step - loss: 0.0520 - val_loss:
0.1406
Epoch 83/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0417 - val_loss:
0.1441
Epoch 84/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0492 - val_loss:
0.1460
Epoch 85/100
1/1 [==============================] - 0s 87ms/step - loss: 0.0531 - val_loss:
0.1474
Epoch 86/100
1/1 [==============================] - 0s 115ms/step - loss: 0.0544 - val_loss:
0.1491
Epoch 87/100
1/1 [==============================] - 0s 97ms/step - loss: 0.0450 - val_loss:
0.1478
Epoch 88/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0527 - val_loss:
0.1462
Epoch 89/100
1/1 [==============================] - 0s 99ms/step - loss: 0.0505 - val_loss:
0.1444
Epoch 90/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0428 - val_loss:
0.1430
Epoch 91/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0418 - val_loss:
0.1430
Epoch 92/100
1/1 [==============================] - 0s 111ms/step - loss: 0.0503 - val_loss:
0.1442
Epoch 93/100
1/1 [==============================] - 0s 88ms/step - loss: 0.0637 - val_loss:
0.1441
Epoch 94/100
1/1 [==============================] - 0s 101ms/step - loss: 0.0444 - val_loss:
0.1431
```

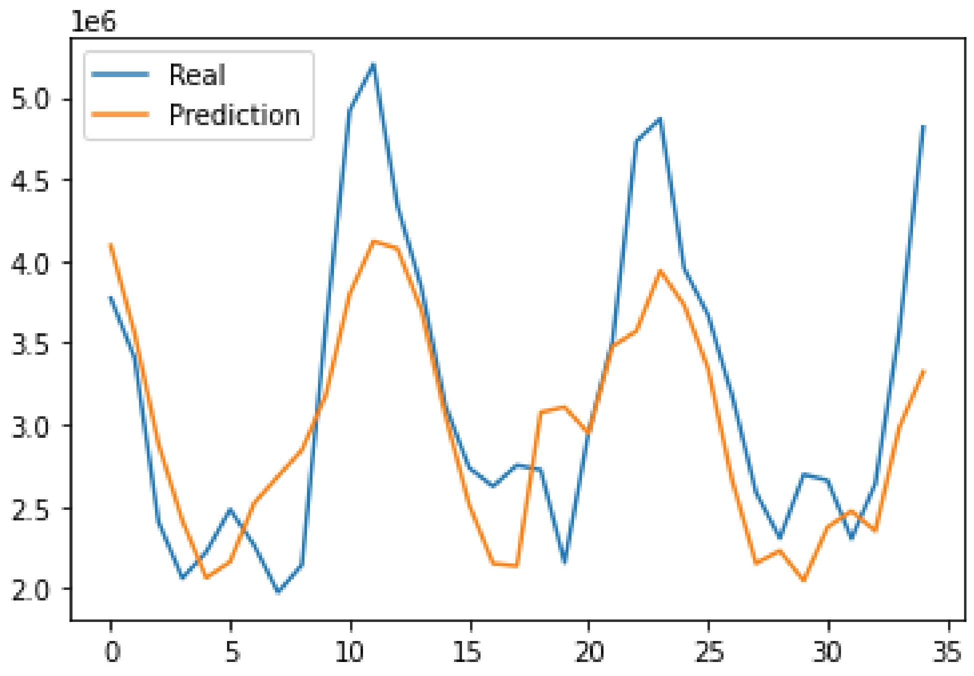
```
Epoch 95/100
1/1 [==============================] - 0s 102ms/step - loss: 0.0467 - val_loss:
0.1424
Epoch 96/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0430 - val_loss:
0.1417
Epoch 97/100
1/1 [==============================] - 0s 95ms/step - loss: 0.0523 - val_loss:
0.1419
Epoch 98/100
1/1 [==============================] - 0s 105ms/step - loss: 0.0413 - val_loss:
0.1423
Epoch 99/100
1/1 [==============================] - 0s 98ms/step - loss: 0.0416 - val_loss:
0.1442
Epoch 100/100
1/1 [==============================] - 0s 100ms/step - loss: 0.0365 - val_loss:
0.1447
```

[59]:
```
loss_plot(history5)
```



[60]:
```
StopWatch.start("test4-predict")
predicted_model(model5, x_test5, y_test5, scaler5)
StopWatch.stop("test4-predict")
```

```
WARNING:tensorflow:6 out of the last 11 calls to <function
Model.make_predict_function.<locals>.predict_function at 0x7f46b062a4d0>
triggered tf.function retracing. Tracing is expensive and the excessive number
of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2)
passing tensors with different shapes, (3) passing Python objects instead of
tensors. For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
The RMSE is: 587340.7243
```



1.11 Benchmarks

For a benchmark, the Cloudmesh StopWatch and Benchmark are used to measure the program's performance. The time spent on data load, data preprocessing, network model compile, training, and prediction was separately measured, and the overall time for execution of all scenarios is around 77 seconds. It can be seen that The training time for the fourth scenario is the longest and the one for the fifth scenario is the shortest.

[171]:
```
StopWatch.benchmark()
```

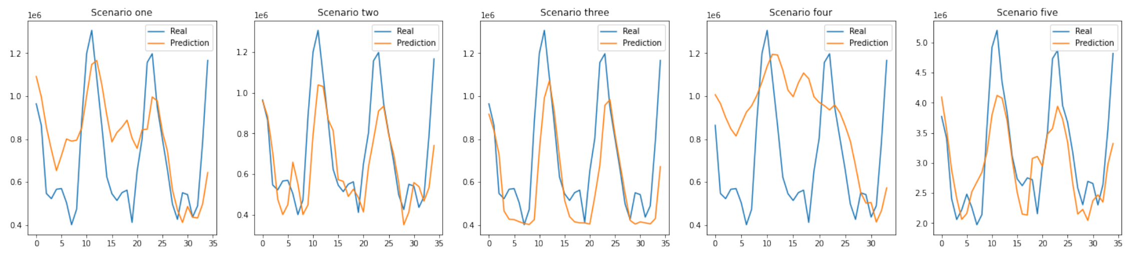
```
+---------------------+----------------------------------------------------------
---------+
| Attribute           | Value
|
|---------------------+---------------------------------------------------------
---------|
| BUG_REPORT_URL      | "https://bugs.launchpad.net/ubuntu/"
|
| DISTRIB_CODENAME    | bionic
|
| DISTRIB_DESCRIPTION | "Ubuntu 18.04.5 LTS"
|
| DISTRIB_ID          | Ubuntu
|
| DISTRIB_RELEASE     | 18.04
|
| HOME_URL            | "https://www.ubuntu.com/"
|
| ID                  | ubuntu
|
| ID_LIKE             | debian
|
| NAME                | "Ubuntu"
|
| PRETTY_NAME         | "Ubuntu 18.04.5 LTS"
|
| PRIVACY_POLICY_URL  | "https://www.ubuntu.com/legal/terms-and-policies
/privacy-policy" |
| SUPPORT_URL         | "https://help.ubuntu.com/"
|
| UBUNTU_CODENAME     | bionic
|
| VERSION             | "18.04.5 LTS (Bionic Beaver)"
|
| VERSION_CODENAME    | bionic
|
| VERSION_ID          | "18.04"
|
| cpu_count           | 2
|
| mem.active          | 1.4 GiB
|
| mem.available       | 11.5 GiB
|
| mem.free            | 9.1 GiB
|
| mem.inactive        | 2.0 GiB
|
| mem.percent         | 9.5 %
|
| mem.total           | 12.7 GiB
|
| mem.used            | 1.4 GiB
|
| platform.version    | #1 SMP Thu Jul 23 08:00:38 PDT 2020
|
| python              | 3.7.10 (default, Feb 20 2021, 21:17:23)
|
|                     | [GCC 7.5.0]
|
| python.pip          | 19.3.1
|
| python.version      | 3.7.10
|
| sys.platform        | linux
|
| uname.machine       | x86_64
|
| uname.node          | d91aa3bf059f
|
| uname.processor     | x86_64
|
| uname.release       | 4.19.112+
|
| uname.system        | Linux
|
| uname.version       | #1 SMP Thu Jul 23 08:00:38 PDT 2020
|
| user                | collab
|
+---------------------+----------------------------------------------------------
---------+

+-------------------------+----------+--------+---------+---------------------+-
------+--------------+--------+-------+-------------------------------------+
| Name                    | Status   |   Time |     Sum | Start               |
tag   | Node         | User   | OS    | Version                             |
|-------------------------+----------+--------+---------+---------------------+-
------+--------------+--------+-------+-------------------------------------|
| data-load               | ok       |  0.051 |   0.051 | 2021-05-02 08:10:29 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| data-preprocess_1       | ok       |  0.013 |   0.013 | 2021-05-02 08:10:31 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| data-preprocess_2       | ok       |  0.009 |   0.076 | 2021-05-02 08:25:42 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| compile                 | ok       |  0.635 |   5.718 | 2021-05-02 08:25:42 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| train                   | ok       | 11.979 |  97.234 | 2021-05-02 08:25:44 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| predict                 | ok       |  0.954 |   7.557 | 2021-05-02 08:25:56 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test1-data-preprocess_1 | ok       |  0.003 |   0.017 | 2021-05-02 08:26:47 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test1-data-preprocess_2 | ok       |  0.016 |   0.031 | 2021-05-02 08:26:47 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test2-compile           | ok       |  0.629 |   3.637 | 2021-05-02 08:27:21 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test1-train             | ok       | 11.914 |  35.913 | 2021-05-02 08:26:54 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test1-predict           | ok       |  0.945 |   3.791 | 2021-05-02 08:27:09 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test2-data-preprocess_1 | ok       |  0.004 |   0.007 | 2021-05-02 08:27:13 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test2-data-preprocess_2 | ok       |  0.008 |   0.014 | 2021-05-02 08:27:14 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test2-train             | ok       | 12.468 |  24.362 | 2021-05-02 08:27:23 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test2-predict           | ok       |  0.975 |   1.884 | 2021-05-02 08:27:39 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test3-data-preprocess_1 | ok       |  0.001 |   0.006 | 2021-05-02 08:29:07 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test3-data-preprocess_2 | ok       |  0.006 |   0.025 | 2021-05-02 08:29:08 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test3-compile           | ok       |  0.602 |   5.484 | 2021-05-02 08:31:36 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test3-train             | ok       | 20.282 | 138.311 | 2021-05-02 08:31:37 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test3-predict           | ok       |  0.979 |   7.888 | 2021-05-02 08:31:57 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test4-data-preprocess_1 | ok       |  0.019 |   0.032 | 2021-05-02 08:31:58 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test4-data-preprocess_2 | ok       |  0.009 |   0.016 | 2021-05-02 08:31:59 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test4-compile           | ok       |  0.576 |   2.416 | 2021-05-02 08:32:52 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test4-train             | ok       | 11.858 |  47.539 | 2021-05-02 08:32:53 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
| test4-predict           | ok       |  1.777 |   5.596 | 2021-05-02 08:33:05 |
| d91aa3bf059f | collab | Linux | #1 SMP Thu Jul 23 08:00:38 PDT 2020 |
+-------------------------+----------+--------+---------+---------------------+-
------+--------------+--------+-------+-------------------------------------+

# csv,timer,status,time,sum,start,tag,uname.node,user,uname.system,platform.vers
ion
# csv,data-load,ok,0.051,0.051,2021-05-02 08:10:29,,d91aa3bf059f,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
```

```
# csv,data-preprocess_1,ok,0.013,0.013,2021-05-02
08:10:31,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,data-preprocess_2,ok,0.009,0.076,2021-05-02
08:25:42,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,compile,ok,0.635,5.718,2021-05-02 08:25:42,,d91aa3bf059f,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,train,ok,11.979,97.234,2021-05-02 08:25:44,,d91aa3bf059f,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,predict,ok,0.954,7.557,2021-05-02 08:25:56,,d91aa3bf059f,collab,Linux,#1
SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test1-data-preprocess_1,ok,0.003,0.017,2021-05-02
08:26:47,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test1-data-preprocess_2,ok,0.016,0.031,2021-05-02
08:26:47,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test2-compile,ok,0.629,3.637,2021-05-02
08:27:21,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test1-train,ok,11.914,35.913,2021-05-02
08:26:54,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test1-predict,ok,0.945,3.791,2021-05-02
08:27:09,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test2-data-preprocess_1,ok,0.004,0.007,2021-05-02
08:27:13,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test2-data-preprocess_2,ok,0.008,0.014,2021-05-02
08:27:14,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test2-train,ok,12.468,24.362,2021-05-02
08:27:23,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test2-predict,ok,0.975,1.884,2021-05-02
08:27:39,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test3-data-preprocess_1,ok,0.001,0.006,2021-05-02
08:29:07,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test3-data-preprocess_2,ok,0.006,0.025,2021-05-02
08:29:08,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test3-compile,ok,0.602,5.484,2021-05-02
08:31:36,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test3-train,ok,20.282,138.311,2021-05-02
08:31:37,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test3-predict,ok,0.979,7.888,2021-05-02
08:31:57,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test4-data-preprocess_1,ok,0.019,0.032,2021-05-02
08:31:58,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test4-data-preprocess_2,ok,0.009,0.016,2021-05-02
08:31:59,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test4-compile,ok,0.576,2.416,2021-05-02
08:32:52,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test4-train,ok,11.858,47.539,2021-05-02
08:32:53,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
# csv,test4-predict,ok,1.777,5.596,2021-05-02
08:33:05,,d91aa3bf059f,collab,Linux,#1 SMP Thu Jul 23 08:00:38 PDT 2020
```

[66]:
```
def predicted_model(model, x_test, y_test, scaler):
  y_predicted = model.predict(x_test)
  x_test = x_test.reshape(x_test.shape[0], x_test.shape[2])
  x_te_re = x_test[:,1:]
  y_test = y_test.reshape(len(y_test), 1)

  inv_y_predicted = np.concatenate((y_predicted, x_te_re), axis=1)
  inv_y_predicted = scaler.inverse_transform(inv_y_predicted)[:,0]

  inv_y = np.concatenate((y_test, x_te_re), axis=1)
  inv_y = scaler.inverse_transform(inv_y)[:,0]

  rmse = np.sqrt(metrics.mean_squared_error(inv_y, inv_y_predicted))
  #print('The RMSE is: %.4f' % rmse)

  plt.plot(inv_y, label='Real')
  plt.plot(inv_y_predicted, label='Prediction')
  plt.legend()
```

[69]:
```
def predicted_time_model(model, x_test, y_test, scaler, time, feature):
  y_predicted = model.predict(x_test)
  x_test = x_test.reshape(x_test.shape[0], (time*feature))
  x_te_re = x_test[:,-(feature-1):]
  y_test = y_test.reshape(len(y_test), 1)

  inv_y_predicted = np.concatenate((y_predicted, x_te_re), axis=1)
  inv_y_predicted = scaler.inverse_transform(inv_y_predicted)[:,0]

  inv_y = np.concatenate((y_test, x_te_re), axis=1)
  inv_y = scaler.inverse_transform(inv_y)[:,0]

  rmse = np.sqrt(metrics.mean_squared_error(inv_y, inv_y_predicted))
  #print('The RMSE is : %.4f' % rmse)

  plt.plot(inv_y, label='Real')
  plt.plot(inv_y_predicted, label='Prediction')
  plt.legend()
```

[71]:
```
fig = plt.figure(figsize=(25,5))
plt.subplot(151)
plt.title('Scenario one')
predicted_model(model, x_test, y_test, scaler)
plt.subplot(152)
plt.title('Scenario two')
predicted_model(model2, x_test2, y_test2, scaler2)
plt.subplot(153)
plt.title('Scenario three')
predicted_model(model3, x_test3, y_test3, scaler3)
plt.subplot(154)
plt.title('Scenario four')
predicted_time_model(model4, x_test4, y_test4, scaler4, months, features)
plt.subplot(155)
plt.title('Scenario five')
predicted_model(model5, x_test5, y_test5, scaler5)
```

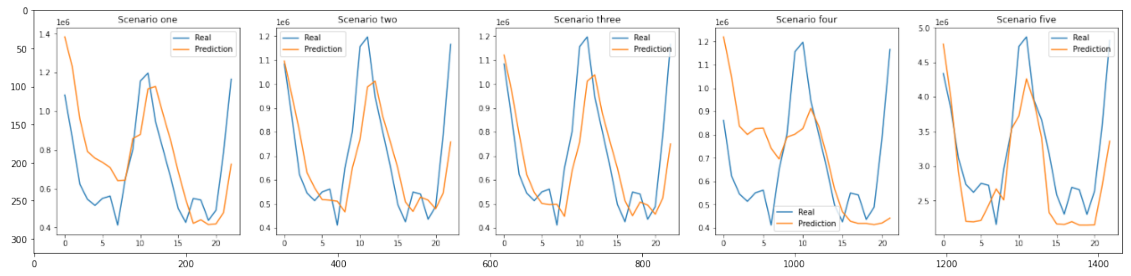


1.12 Compared results

Out of the five scenarios in total, the second and third have smaller RMSE than others, and the graphs also show relatively similar results. The first and fourth show differences in the beginning and similar trends in the last part. However, it is noteworthy that the gap at the beginning of them is very large, but it tends to shrink together at the point of decline and stretch together at the point of increase.

In the first and fifth scenarios, all data are identical except that they differ in regional scale in temperature and precipitation. It is also the same that 12 months of data are used as the training set. From the subtle differences in the shape of the resulting graph, it can be seen that the national average data cannot represent the situation in a particular region, and the amount of NG supply differs depending on the circumstances in the region.

[9]:
```
from PIL import Image
img = Image.open("sample_data/compared_prediction_2.png")
plt.figure(figsize=(25,5))
plt.imshow(img)
plt.show()
```



After changing the training set from twelve months to twenty-four months, the results are more clearly visible. The second and third prediction graphs have a more similar shape and the RMSE value decreases than the previous setting. The results of other scenarios show that the overall shape has improved; contrarily, the shape of the rapidly changing middle part is better in the previous condition.

1.13 Conclusion

From the results of this project, it can be seen that simplifying factors that have a significant impact shows better efficiency than combining various factors. For example, NG consumption tends to increase for heating in the cold weather. In addition, there is a lot of precipitation in the warm or hot weather, on the contrary, there is relatively little precipitation in the cold weather. It can be seen that these seasonal elements show relatively high consistency for affecting prediction when those are used as training datasets. Also, the predictions are derived more effectively when the seasonal datasets are combined.

However, in training set with 12 months duration, the last part of the scenario used the dataset combined with various factors, which seem to be not related to season, tends to match real data. Furthemore, when the training set is doubled on the same dataset, it can be seen that the differences between real and prediction graph is decreased than the result of smaller traing set. Based on this, it can be expected that the results could vary if a large amount of dataset with a longer period is used and the ratio of the training set is appropriately adjusted.

South Korea imports a large amount of its energy resources. Also, the plan for energy supply and demand is being made and operated through nation-led policies. Ironically, the government's plan also shows a sharp change in direction with recent environmental issues, and the volatility of demand in the energy market is greater than before. Therefore, methodologies for accurate forecasting of energy demand will need to be complemented and developed constantly to prepare for and overcome this variability.

In this project, Forecasting NG demand and supply was carried out using various data factors such as weather and price that is relatively easily obtained than the datasets which are complex economic indicators or classified as confidential. Nevertheless, state-of-the-art deep learning methods show that it has the flexibility and potential to forecast NG demand through the tendency of the results that indicate a relativaly consistent with the actual data. From this point of view, it is thought that the research on NG in South Korea should be conducted in an advanced form by utilizing various data and more specialized analysis.