In [1]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, accuracy_score
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM, Activation
from keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
plt.style.use('ggplot')
%matplotlib inline
```

In [2]:
```
dataset_train=pd.read_csv('./sensordata/PM_train.txt',sep=' ',header=None).drop([26,27],axis=1)
col_names = ['id','cycle','setting1','setting2','setting3','s1','s2','s3','s4','s5','s6','s7','s8','s9','s10','s11','s
dataset_train.columns=col_names
print('Shape of Train dataset: ',dataset_train.shape)
dataset_train
```

```
Shape of Train dataset:  (7275, 26)
```

Out[2]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7270	37	125	0.0043	-0.0002	100.0	518.67	643.31	1590.95	1418.46	14.62	...	520.74	2388.20	8119.80	8.4733	0.03	393	2388	100
7271	37	126	-0.0021	0.0004	100.0	518.67	643.34	1594.74	1410.82	14.62	...	520.82	2388.21	8120.23	8.5147	0.03	394	2388	100
7272	37	127	-0.0019	0.0000	100.0	518.67	642.61	1593.17	1414.05	14.62	...	520.51	2388.17	8122.34	8.4422	0.03	395	2388	100
7273	37	128	-0.0023	0.0001	100.0	518.67	642.99	1595.74	1416.59	14.62	...	521.20	2388.18	8122.77	8.4684	0.03	394	2388	100
7274	37	129	0.0006	-0.0001	100.0	518.67	642.36	1592.32	1416.01	14.62	...	520.89	2388.21	8121.99	8.4877	0.03	393	2388	100

7275 rows × 26 columns

In [3]:
```
dataset_test=pd.read_csv('./sensordata/PM_test.txt',sep=' ',header=None).drop([26,27],axis=1)
dataset_test.columns=col_names
#dataset_test.head()
print('Shape of Test dataset: ',dataset_train.shape)
dataset_train.head()
```

```
Shape of Test dataset:  (7275, 26)
```

Out[3]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s19
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100.0

5 rows × 26 columns

In [4]:
```
pm_truth=pd.read_csv('./sensordata/PM_truth.txt',sep=' ',header=None).drop([1],axis=1)
pm_truth.columns=['more']
pm_truth['id']=pm_truth.index+1
pm_truth.head()
# generate column max for test data
rul = pd.DataFrame(dataset_test.groupby('id')['cycle'].max()).reset_index()
rul.columns = ['id', 'max']
rul.head()
```

Out[4]:

	id	max
0	1	31
1	2	49
2	3	126
3	4	106
4	5	98

In [5]:
```
# run to failure
pm_truth['rtf']=pm_truth['more']+rul['max']
pm_truth.head()
```

Out[5]:

	more	id	rtf
0	112	1	143.0
1	98	2	147.0
2	69	3	195.0
3	82	4	188.0
4	91	5	189.0

In [6]:
```
# generate column max for test data
rul = pd.DataFrame(dataset_test.groupby('id')['cycle'].max()).reset_index()
rul.columns = ['id', 'max']
rul.head()
```

Out[6]:

	id	max
0	1	31
1	2	49
2	3	126
3	4	106
4	5	98

In [7]:
```
pm_truth.drop('more', axis=1, inplace=True)
dataset_test=dataset_test.merge(pm_truth,on=['id'],how='left')
dataset_test['ttf']=dataset_test['rtf'] - dataset_test['cycle']
dataset_test.drop('rtf', axis=1, inplace=True)
dataset_test.head()
```

Out[7]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s13	s14	s15	s16	s17	s18	s19	s20	
0	1	1	0.0023	0.0003	100.0	518.67	643.02	1585.29	1398.21	14.62	...	2388.03	8125.55	8.4052	0.03	392	2388	100.0	38.86	2
1	1	2	-0.0027	-0.0003	100.0	518.67	641.71	1588.45	1395.42	14.62	...	2388.06	8139.62	8.3803	0.03	393	2388	100.0	39.02	2
2	1	3	0.0003	0.0001	100.0	518.67	642.46	1586.94	1401.34	14.62	...	2388.03	8130.10	8.4441	0.03	393	2388	100.0	39.08	2
3	1	4	0.0042	0.0000	100.0	518.67	642.44	1584.12	1406.42	14.62	...	2388.05	8132.90	8.3917	0.03	391	2388	100.0	39.00	2
4	1	5	0.0014	0.0000	100.0	518.67	642.51	1587.19	1401.92	14.62	...	2388.03	8129.54	8.4031	0.03	390	2388	100.0	38.99	2

5 rows × 27 columns

In [8]:
```
dataset_train['ttf'] = dataset_train.groupby(['id'])['cycle'].transform(max)-dataset_train['cycle']
dataset_train.head()
```

Out[8]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s13	s14	s15	s16	s17	s18	s19	s20	
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	2
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	2
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	2
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	2
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	2

5 rows × 27 columns

In [9]:
```
df_train=dataset_train.copy()
df_test=dataset_test.copy()
period=30
df_train['label_bc'] = df_train['ttf'].apply(lambda x: 1 if x <= period else 0)
df_test['label_bc'] = df_test['ttf'].apply(lambda x: 1 if x <= period else 0)
df_train.head()
```

Out[9]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s14	s15	s16	s17	s18	s19	s20	s21	
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190	1
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236	1
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442	1
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739	1
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044	1

5 rows × 28 columns

In [10]:
```
features_col_name=['setting1', 'setting2', 'setting3', 's1', 's2', 's3', 's4', 's5', 's6', 's7', 's8', 's9', 's10', 's
's12', 's13', 's14', 's15', 's16', 's17', 's18', 's19', 's20', 's21']
target_col_name='label_bc'
```

In [11]:
```
sc=MinMaxScaler()
df_train[features_col_name]=sc.fit_transform(df_train[features_col_name])
df_test[features_col_name]=sc.transform(df_test[features_col_name])
```

In [12]:
```
def gen_sequence(id_df, seq_length, seq_cols):
    df_zeros=pd.DataFrame(np.zeros((seq_length-1,id_df.shape[1])),columns=id_df.columns)
    id_df=df_zeros.append(id_df,ignore_index=True)
    data_array = id_df[seq_cols].values
    num_elements = data_array.shape[0]
    lstm_array=[]
    for start, stop in zip(range(0, num_elements-seq_length), range(seq_length, num_elements)):
        lstm_array.append(data_array[start:stop, :])
    return np.array(lstm_array)

# function to generate labels
def gen_label(id_df, seq_length, seq_cols,label):
    df_zeros=pd.DataFrame(np.zeros((seq_length-1,id_df.shape[1])),columns=id_df.columns)
    id_df=df_zeros.append(id_df,ignore_index=True)
    data_array = id_df[seq_cols].values
    num_elements = data_array.shape[0]
    y_label=[]
    for start, stop in zip(range(0, num_elements-seq_length), range(seq_length, num_elements)):
        y_label.append(id_df[label][stop])
    return np.array(y_label)
```

In [13]:
```
# timestamp or window size
seq_length=50
seq_cols=features_col_name
```

In [14]:
```
# generate X_train
X_train=np.concatenate(list(list(gen_sequence(df_train[df_train['id']==id], seq_length, seq_cols)) for id in df_train[
# print(X_train.shape)
# generate y_train
y_train=np.concatenate(list(list(gen_label(df_train[df_train['id']==id], 50, seq_cols,'label_bc')) for id in df_train[
# print(y_train.shape)
```

```
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
```

In [15]:
```
# generate X_test
X_test=np.concatenate(list(list(gen_sequence(df_test[df_test['id']==id], seq_length, seq_cols)) for id in df_test['id'
# print(X_test.shape)
# generate y_test
y_test=np.concatenate(list(list(gen_label(df_test[df_test['id']==id], 50, seq_cols,'label_bc')) for id in df_test['id'
# print(y_test.shape)
```

```
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is depr
ecated and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)
```

In [16]:
```
nb_features =X_train.shape[2]
timestamp=seq_length
model = Sequential()
model.add(LSTM(
input_shape=(timestamp, nb_features),
units=100,
return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(
units=50,
return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50, 100)           50000

 dropout (Dropout)           (None, 50, 100)           0

 lstm_1 (LSTM)               (None, 50)                30200

 dropout_1 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 80,251
Trainable params: 80,251
Non-trainable params: 0
_________________________________________________________________
```

In [17]:
```
# fit the network
model.fit(X_train, y_train, epochs=2, batch_size=200, validation_split=0.05, verbose=1,
callbacks = [EarlyStopping(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto')])
```

```
Epoch 1/2
35/35 [==============================] - 31s 586ms/step - loss: 0.3204 - accuracy: 0.8637 - val_loss: 0.3527 - val_a
ccuracy: 0.8867
Epoch 2/2
35/35 [==============================] - 23s 659ms/step - loss: 0.1187 - accuracy: 0.9577 - val_loss: 0.3993 - val_a
ccuracy: 0.8591
```

Out[17]:
```
<keras.callbacks.History at 0x1e39e25eb00>
```

In [18]:
```
# y_pred=model.predict(X_test)
# print('Accuracy of model on test data: ',accuracy_score(y_test,y_pred))
# print('Confusion Matrix: \n',confusion_matrix(y_test,y_pred))
```

In [19]:
```
def prob_failure(machine_id):
    machine_df=df_test[df_test.id==machine_id]
    machine_test=gen_sequence(machine_df,seq_length,seq_cols)
    m_pred=model.predict(machine_test)
    failure_prob=list(m_pred[-1]*100)[0]
    return failure_prob
```

In [27]:
```
machine_id=4
print('Probability that machine will fail within 30 days: ',prob_failure(machine_id))
```

```
1/4 [======>.....................] - ETA: 0s

C:\Users\cjtan\AppData\Local\Temp\ipykernel_14312\797777066.py:3: FutureWarning: The frame.append method is deprecat
ed and will be removed from pandas in a future version. Use pandas.concat instead.
  id_df=df_zeros.append(id_df,ignore_index=True)

4/4 [==============================] - 0s 54ms/step
Probability that machine will fail within 30 days:  0.9354318
```