In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import precision_score, recall_score, f1_score
```

```
C:\Users\Dell\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning:
A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (d
etected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:

```
df = pd.read_excel('Data.xlsx')

df['Date'] = pd.to_datetime(df['Date'])

df = df.sort_values(['Part ID', 'Date'])
```

Remove Part IDs with less than 50 days of data

In [3]:

```
part_counts = df['Part ID'].value_counts()
df_filtered = df[df['Part ID'].isin(part_counts[part_counts >= 50].index)]
```

Grouping data week by week

In [4]:

```
def resample_7d(group):
    cycle_col = group['Cycle'].max()
    mean_cols = group.drop(columns=['Cycle']).set_index('Date').resample('
    mean_cols['Cycle'] = cycle_col
    return mean_cols
df_resampled_filtered = df_filtered.groupby('Part ID').apply(resample_7d).
```

Making binary Values and creating a window

In [5]:

```
df_resampled_filtered['response'] = np.where(df_resampled_filtered['Total

feature_columns = ['Feature_1', 'Feature_2', 'Feature_3', 'Feature_4', 'Fe
                   'Feature_6', 'Feature_7', 'Feature_8', 'Feature_9', 'Fe
                   'Feature_11', 'Feature_12']

df_resampled_filtered[feature_columns] = df_resampled_filtered[feature_col
df_resampled_filtered = df_resampled_filtered.dropna()
```

In [6]:

```
df_resampled_filtered.head()
```

Out[6]:

	Date	Part ID	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Fe
5	2010-02-02	1.0	15.5	4.5	0.0	0.0	0.0	0.0	4.5	
19	2010-05-11	1.0	0.0	0.0	0.0	3.5	1.0	1.5	4.5	
22	2010-06-01	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
43	2010-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
44	2010-11-02	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

In [7]:

```
# df_resampled_filtered.to_excel("data new.xlsx")
```

Dataset Preparation

In [8]:

```
X = df_resampled_filtered[feature_columns]
y = df_resampled_filtered['response']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, s

```

In [9]:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

X_train_scaled = X_train_scaled.reshape((X_train_scaled.shape[0], 1, X_tra
X_test_scaled = X_test_scaled.reshape((X_test_scaled.shape[0], 1, X_test_s
```

Model

In [10]:

```
model = Sequential()
model.add(LSTM(units=128, return_sequences=False, input_shape=(X_train_sca
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['a
```

In [16]:

```
history = model.fit(X_train_scaled, y_train, epochs=20, batch_size=32, val
Epoch 20/20
95/95 [==============================] - 0s 5ms/step - loss: 0.5221 - accurac
y: 0.7174 - val_loss: 0.4626 - val_accuracy: 0.7697
```

In [12]:

```
loss, accuracy = model.evaluate(X_test_scaled, y_test)
print(f'Test Accuracy: {accuracy:.4f}')
```

```
24/24 [==============================] - 0s 2ms/step - loss: 0.4644 - accurac
y: 0.7671
Test Accuracy: 0.7671
```

Prediction

In [13]:

```
X_scaled = scaler.transform(X)
X_scaled = X_scaled.reshape((X_scaled.shape[0], 1, X_scaled.shape[1]))

y_pred_all = model.predict(X_scaled)
y_pred_all_classes = (y_pred_all > 0.5).astype("int32")

df_resampled_filtered['predict'] = y_pred_all_classes.flatten()

# df_resampled_filtered.to_excel('predictions_future_observation.xlsx', in

# print("Predictions for all data have been added and the file has been sa

```

```
119/119 [==============================] - 1s 2ms/step
```

In [14]:

```
df_resampled_filtered.head()
```

Out[14]:

	Date	Part ID	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Fe
5	2010-02-02	1.0	15.5	4.5	0.0	0.0	0.0	0.0	4.5	
19	2010-05-11	1.0	0.0	0.0	0.0	3.5	1.0	1.5	4.5	
22	2010-06-01	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
43	2010-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
44	2010-11-02	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	