In [1]:

```
import scipy.io
import seaborn as sns
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
```

In [4]:

```
# Using os.path.join to create the path
data_dir = "C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        path = os.path.join(root, file_name)
        print(path)
```

```
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA007_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA014_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA021_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR007_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR014_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR021_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR007_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR014_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR021_3.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\rN_3.mat
```

In [5]:

```
# Using f-strings for path
path = f'C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR021_3.mat'
mat = scipy.io.loadmat(path)
```

In [6]:

```
# Using mat.keys() directly
key_name = list(mat.keys())[3]
```

In [7]:

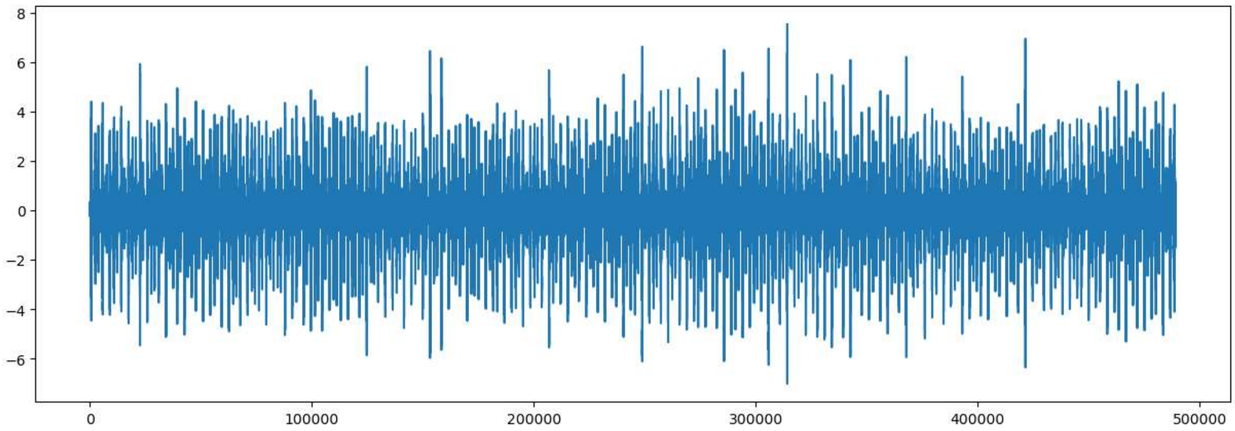
```
# Simplifying fault creation
fault = np.full((len(mat[key_name]), 1), file_name[:-4])
```

In [8]:

```
# Using DataFrame initialization directly
df_temp = pd.DataFrame({'DE_data': np.ravel(mat[key_name]), 'fault': np.ravel(fault)})
```

In [9]:

```
# Plotting directly without creating a variable
plt.figure(figsize=(15, 5))
plt.plot(df_temp.iloc[:, 0])
plt.show()
```



In [10]:

```
# Initializing df with data directly
df = pd.DataFrame(columns=['DE_data', 'fault'])
```

In [11]:

```
# Using f-strings for path
data_dir = "C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        # Check if the file has a .mat extension
        if file_name.endswith('.mat'):
            path = os.path.join(root, file_name)
            print(path)

            try:
                mat = scipy.io.loadmat(path)
                key_name = list(mat.keys())[3]
                DE_data = mat.get(key_name)
                # Simplifying fault creation
                fault = np.full((len(DE_data), 1), file_name[:-4])

                # Concatenating directly without creating df_temp
                df = pd.concat([df, pd.DataFrame({'DE_data': np.ravel(DE_data), 'fault': np.ravel(fault)})],
                print(df['fault'].unique())

            except Exception as e:
                print(f"Error processing file {file_name}: {e}")
```

```
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA007_3.mat
['BA007_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA014_3.mat
['BA007_3' 'BA014_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\BA021_3.mat
['BA007_3' 'BA014_3' 'BA021_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR007_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR014_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\IR021_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3' 'IR021_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR007_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3' 'IR021_3' 'OR007_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR014_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3' 'IR021_3' 'OR007_3'
 'OR014_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\OR021_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3' 'IR021_3' 'OR007_3'
 'OR014_3' 'OR021_3']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp\rN_3.mat
['BA007_3' 'BA014_3' 'BA021_3' 'IR007_3' 'IR014_3' 'IR021_3' 'OR007_3'
 'OR014_3' 'OR021_3' 'rN_3']
```

In [12]:

```
# Save the resulting DataFrame to a CSV file
df.to_csv('C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/3hp/3hp_all_faults.csv', index=False)
```

In [13]:

```
# Display the DataFrame
df
```

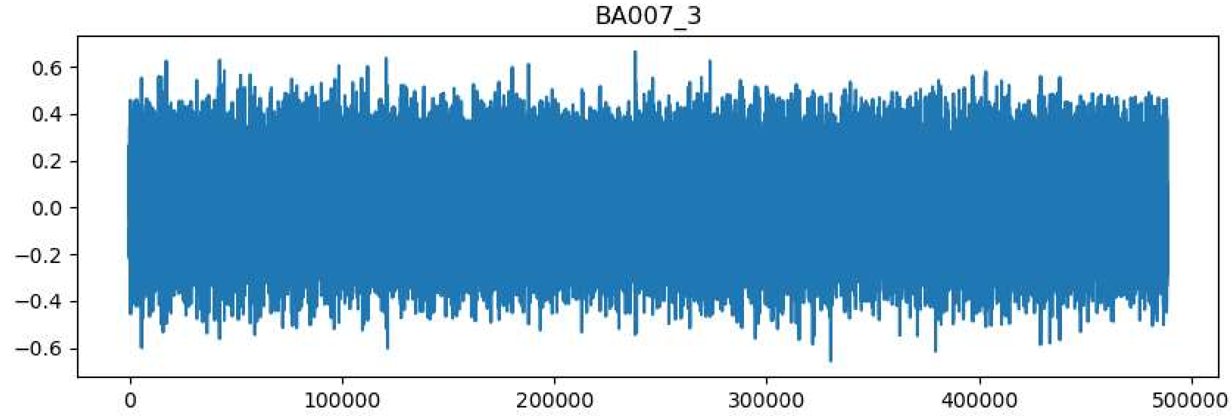
Out[13]:

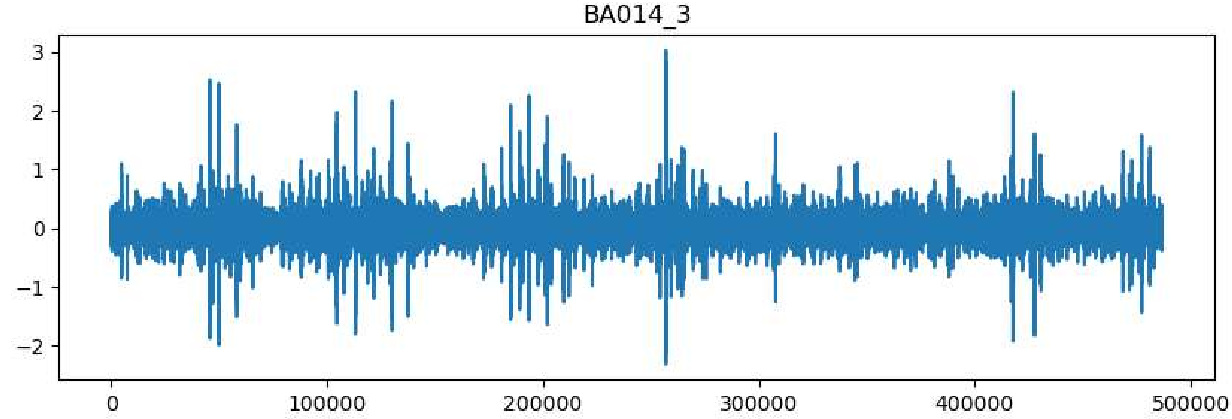
	DE_data	fault
0	0.253050	BA007_3
1	0.257431	BA007_3
2	0.238447	BA007_3
3	0.218838	BA007_3
4	0.191718	BA007_3
...	...	...
485638	0.035673	rN_3
485639	-0.004590	rN_3
485640	-0.023574	rN_3
485641	0.005215	rN_3
485642	0.047773	rN_3

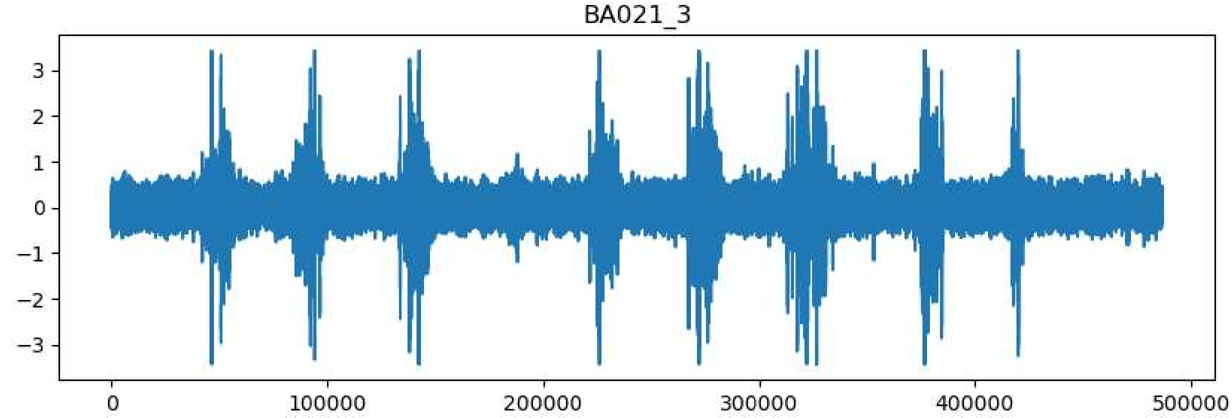
4875582 rows × 2 columns

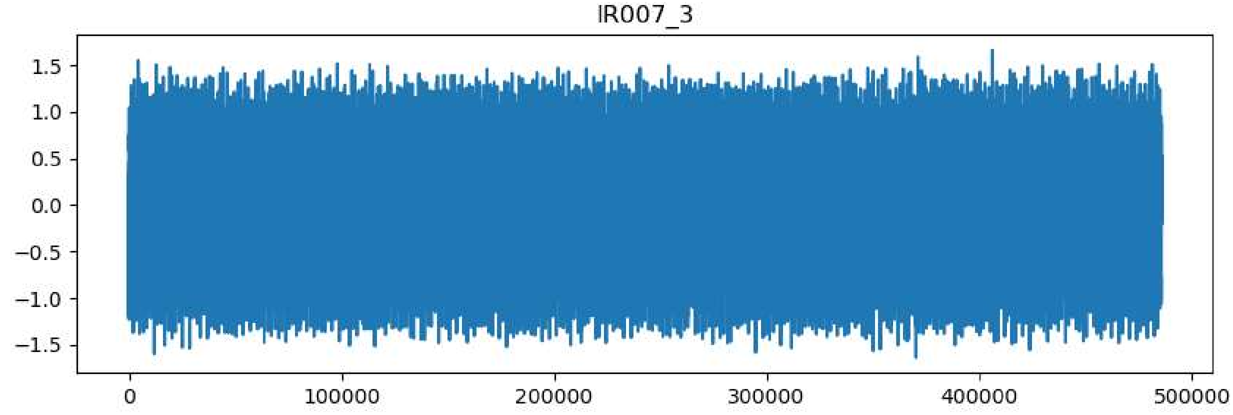
In [14]:

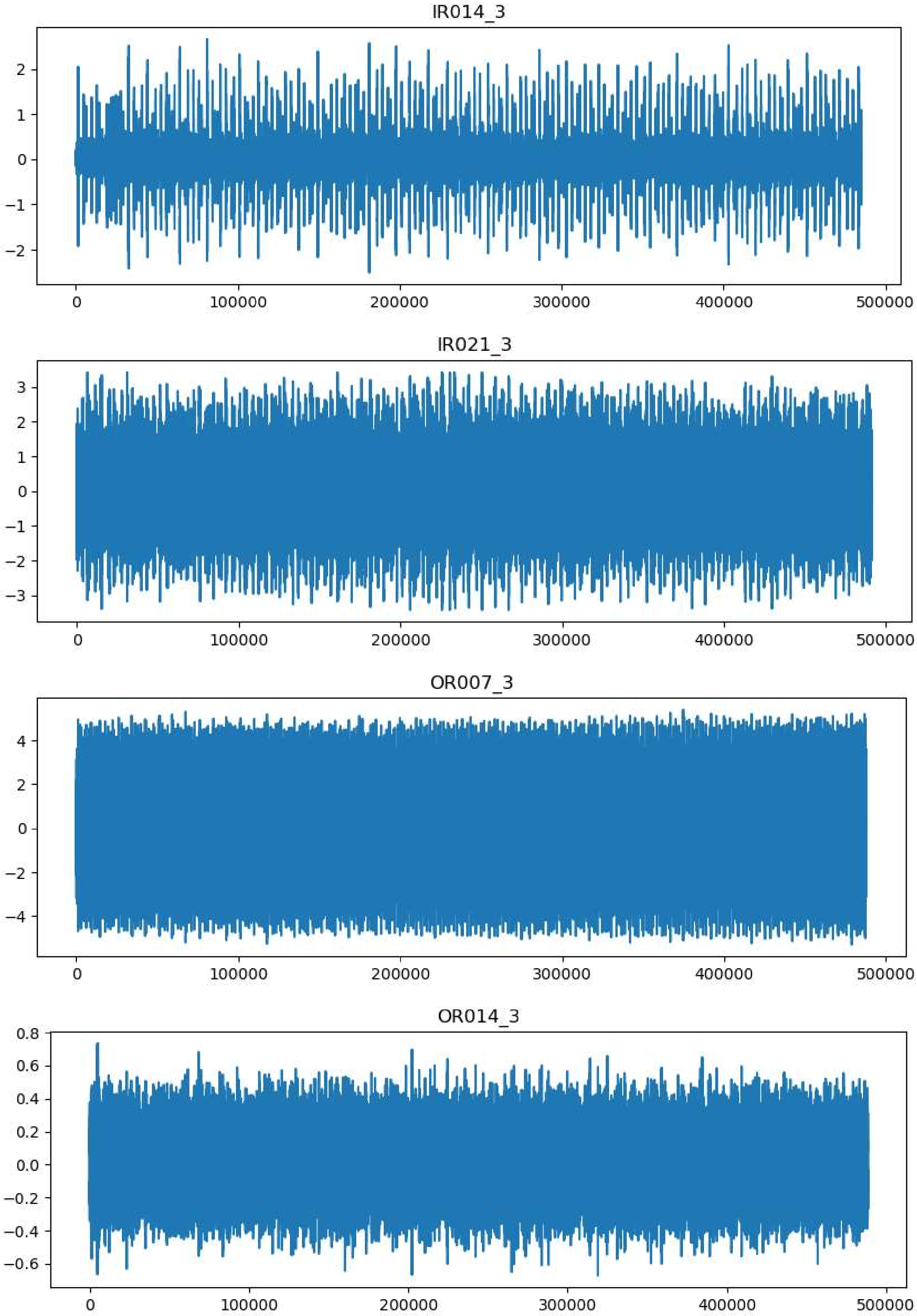
```
# Simplifying the faults loop
for f in df['fault'].unique():
    plt.figure(figsize=(10, 3))
    plt.plot(df[df['fault'] == f].iloc[:, 0])
    plt.title(f)
    plt.show()
```

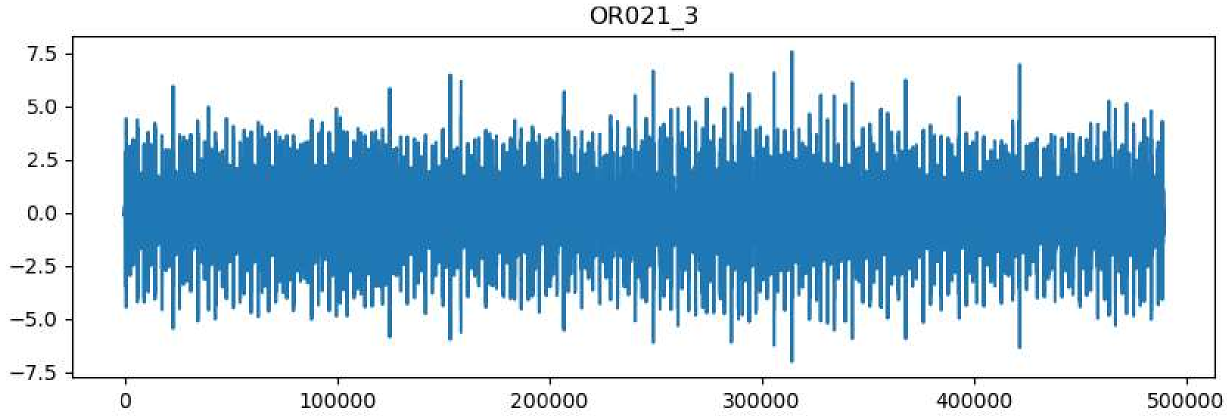

BA007_3
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
100000
200000
300000
400000
500000

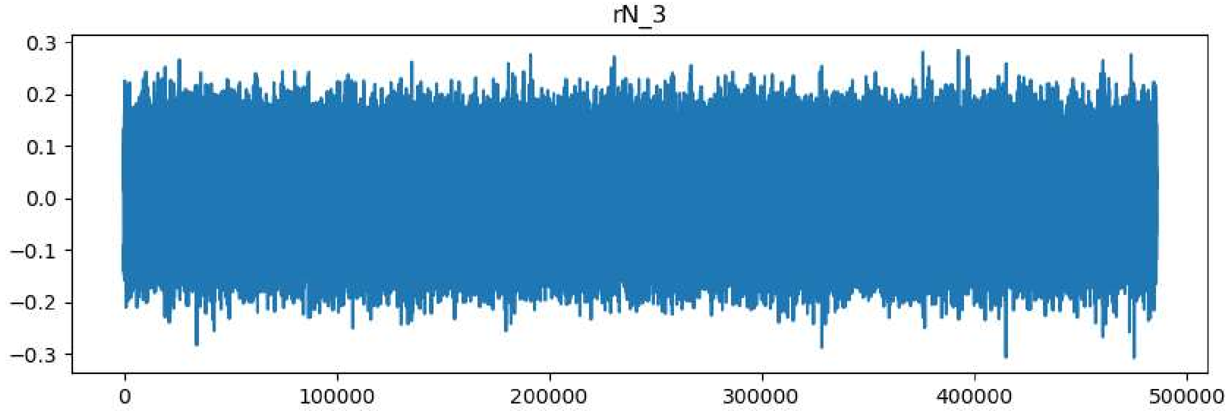

BA014_3
3
2
1
0
−1
−2
0
100000
200000
300000
400000
500000


BA021_3
3
2
1
0
−1
−2
−3
0
100000
200000
300000
400000
500000

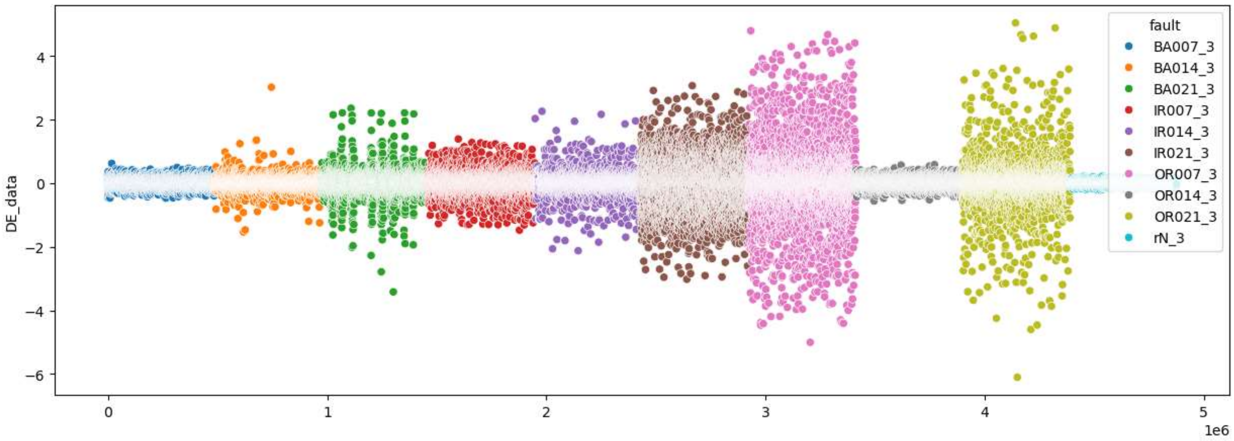

IR007_3
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
100000
200000
300000
400000
500000


IR014_3
2
1
0
−1
−2
0
100000
200000
300000
400000
500000
IR021_3
3
2
1
0
−1
−2
−3
0
100000
200000
300000
400000
500000
OR007_3
4
2
0
−2
−4
0
100000
200000
300000
400000
500000
OR014_3
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
100000
200000
300000
400000
500000





In [15]:
```
# Plotting directly without creating a variable
plt.figure(figsize=(15, 5))
sns.scatterplot(data=df.iloc[::100, :], y='DE_data', x=np.arange(0, len(df), 100), hue='fault')
plt.show()
```



In [ ]: