

temp

In [1]:

```
# imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import Ridge
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error
from sklearn.model_selection import cross_val_score
```

C:\Users\juanaq\AppData\Local\Temp\ipykernel_18772\2329317007.py:2: DeprecationWarning: Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 2.0), to allow more performant data types, such as the Arrow string type, and better interoperability with other tools. but was not found to be installed on your system. If this would cause problems for you, please provide us feedback at <https://github.com/pandas-dev/pandas/issues/54466>

```
import pandas as pd
```

In [2]:

```
# dataset phat
```

```
DATA_PHAT = '../dataset/'
DATA_PATH_TRUE = 'DS-5-1-GAP-0-1-N-0_v2.csv'
DATA_PATH_NOISE1 = 'DS-5-1-GAP-1-1-N-1_v2.csv'
DATA_PATH_NOISE2 = 'DS-5-1-GAP-5-1-N-3_v2.csv'
```

In [3]:

```
df_true = pd.read_csv(DATA_PHAT + DATA_PATH_TRUE, header=None)
df_noise1 = pd.read_csv(DATA_PHAT + DATA_PATH_NOISE1, header=None)
df_noise2 = pd.read_csv(DATA_PHAT + DATA_PATH_NOISE2, header=None)
```

In [4]:

```
df_true.head(5)
```

```
Out[4]:
```

	0	1	2
0	0.00	17.49	17.04
1	2.12	17.65	17.17
2	3.06	17.70	17.24
3	4.16	17.73	17.33
4	4.93	17.75	17.39

```
In [5]:
```

```
df_noise1.head(5)
```

```
Out[5]:
```

	0	1	2	3	4	5	6	7	8	9	...	191	192	193
0	0.00	17.49	17.50	17.49	17.49	17.50	17.49	17.50	17.49	17.49	...	17.04	17.04	17.05
1	2.12	17.65	17.65	17.64	17.64	17.65	17.65	17.65	17.64	17.65	...	17.16	17.17	17.18
2	3.06	17.69	17.70	17.70	17.69	17.69	17.70	17.69	17.69	17.70	...	17.25	17.24	17.24
3	4.16	17.74	17.73	17.74	17.73	17.74	17.74	17.74	17.74	17.73	...	17.33	17.32	17.33
4	4.93	17.75	17.74	17.73	17.74	17.74	17.75	17.75	17.75	17.74	...	17.38	17.39	17.39

5 rows \times 201 columns

```
In [6]:
```

```
df_noise2.head(5)
```

```
Out[6]:
```

	0	1	2	3	4	5	6	7	8	9	...	191	192	193
0	0.00	17.61	17.55	17.48	17.46	17.43	17.53	17.35	17.66	17.60	...	17.16	17.03	17.12
1	2.12	17.71	17.55	17.70	17.52	17.67	17.62	17.76	17.73	17.63	...	17.24	17.17	17.24
2	3.06	17.68	17.77	17.61	17.72	17.73	17.78	17.80	17.81	17.68	...	17.42	17.27	17.22
3	4.16	17.62	17.72	17.66	17.69	17.75	17.65	17.82	17.78	17.86	...	17.32	17.28	17.25
4	4.93	17.80	17.54	17.66	17.71	17.82	17.69	17.70	17.81	17.82	...	17.33	17.46	17.39

5 rows \times 201 columns

```
In [7]:
```

```

import numpy as np

# Se cargan los datos
d1 = np.loadtxt('../dataset/DS-5-1-GAP-0-1-N-0_v2.dat')
d2 = np.loadtxt('../dataset/DS-5-1-GAP-1-1-N-1_v2.dat')
d3 = np.loadtxt('../dataset/DS-5-1-GAP-5-1-N-3_v2.dat')

# Se calcula la media y la varianza de las muestras de datos ruidosos
d2_mean = np.mean(d2[:, 1:101], axis=1)
d3_mean = np.mean(d3[:, 1:101], axis=1)
d2_variance = np.std(d2[:, 1:101], axis=1)
d3_variance = np.std(d3[:, 1:101], axis=1)

# Se inicializa el sesgo
d2_bias = np.zeros_like(d2_mean)
d3_bias = np.zeros_like(d3_mean)

# Se calcula el sesgo
j = 0
for i in range(len(d1)):
    if d1[i, 0] == d2[j, 0]:
        d2_bias[j] = np.abs(d1[i, 1] - d2_mean[j])
        j += 1

j = 0
for i in range(len(d1)):
    if d1[i, 0] == d3[j, 0]:
        d3_bias[j] = np.abs(d1[i, 1] - d3_mean[j])
        j += 1

# Se calcula el sesgo promedio y la varianza promedio
print('\nd2 Bias =', np.mean(d2_bias), ' Variance =', np.mean(d2_variance))
print('d3 Bias =', np.mean(d3_bias), ' Variance =', np.mean(d3_variance))

d2 Bias = 0.0004907455555553497  Variance = 0.005269364493017032
d3 Bias = 0.0050828799999996475  Variance = 0.08272162705566029

In [ ]:

```