Part 1 : Daily Electric Power Usage Trends of 270 clients

Data : Electric demand (power) of clients in Portugal, during 2013 and 2014

Contains 370 time series, corresponding to the electric demand for 370 clients.

Clustering techniques were used.

15 min interval time series

Features

- Data set has no missing values.
- Values are in kW of each 15 min
- Each column represent one client. Some clients were created after 2011. In these cases consumption were considered zero.

```
<class 'pandas.core.frame.DataFrame'>
Index: 140256 entries, 2011-01-01 00:15:00 to 2015-01-01 00:00:00
Columns: 370 entries, MT_001 to MT_370
dtypes: float64(370)
memory usage: 397.0+ MB
```

client1 client2 client3

	MT_001	MT_002	MT_003
2014-12-31 21:45:00	2.538071	22.048364	1.737619
2014-12-31 22:00:00	1.269036	22.048364	1.737619
2014-12-31 22:15:00	2.538071	22.048364	1.737619
2014-12-31 22:30:00	2.538071	22.048364	1.737619
2014-12-31 22:45:00	1.269036	22.048364	1.737619
2014-12-31 23:00:00	2.538071	22.048364	1.737619
2014-12-31 23:15:00	2.538071	21.337127	1.737619
2014-12-31 23:30:00	2.538071	20.625889	1.737619
2014-12-31 23:45:00	1.269036	21.337127	1.737619
2015-01-01 00:00:00	2.538071	19.914651	1.737619

10 rows × 370 columns

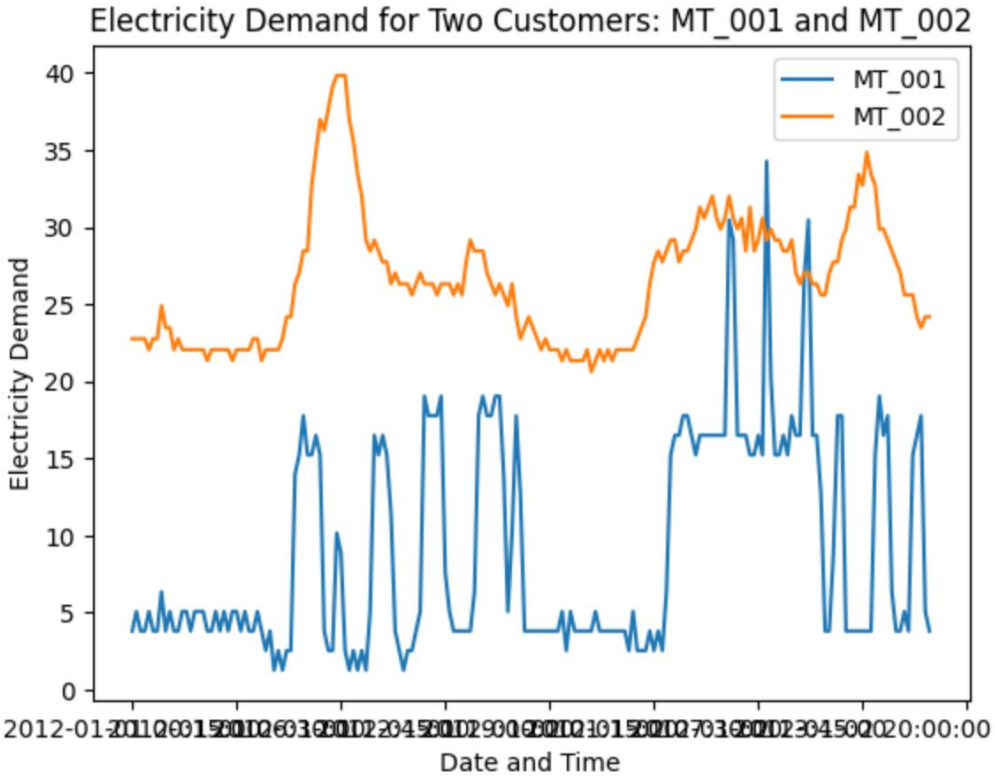
Statistical Measures

```
data.describe()
```

	MT_001	MT_002	MT_003
count	140256.000000	140256.000000	140256.000000
mean	3.970785	20.768480	2.918308
std	5.983965	13.272415	11.014456
min	0.000000	0.000000	0.000000
25%	0.000000	2.844950	0.000000
50%	1.269036	24.893314	1.737619
75%	2.538071	29.871977	1.737619
max	48.223350	115.220484	151.172893

8 rows × 370 columns

Plot the 2 days of 2012 for the first 2 clients



Main difference between the curves is the level

-> We have to normalize the curves, in order for the clustering technique to capture the behaviour of the consumption throughout the day, rather than the overall level

	MT_001	MT_002
count	140256.000000	140256.000000
mean	3.970785	20.768480

Normalize the curve

Divide each curve by its mean -> all curves have mean 1

```
average_curves_norm = average_curves/(average_curves.mean())
```

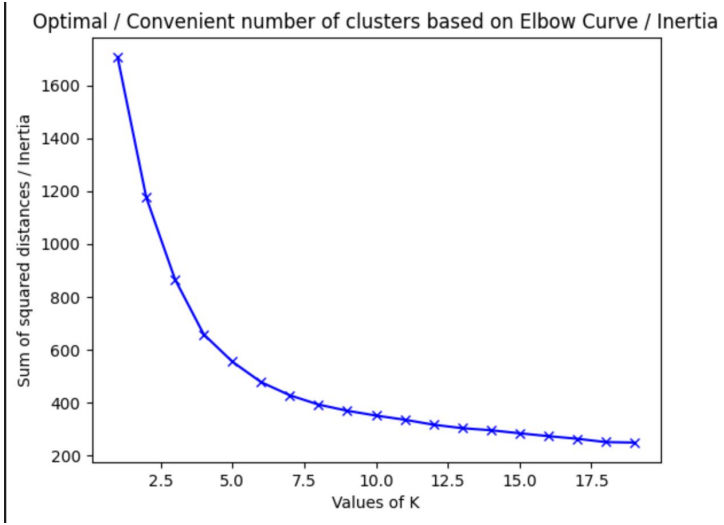
✓ 0.0s

Clustering Analysis on the average normalized curves

Within-cluster sum of squares

# of cluster(K)

Find K where: **Small** (not minimized) **Low** (not minimized)



Elbow Method :

K = 5 or 6

Plot

Make a plot for each cluster, that includes:

- The number of clients in the cluster
- All the curves in the cluster
- The curve corresponding to the centroid of the cluster (blue)

5 centroids (from 5 clusters)

24 hr * 4 (15min interval) = **96** attributes

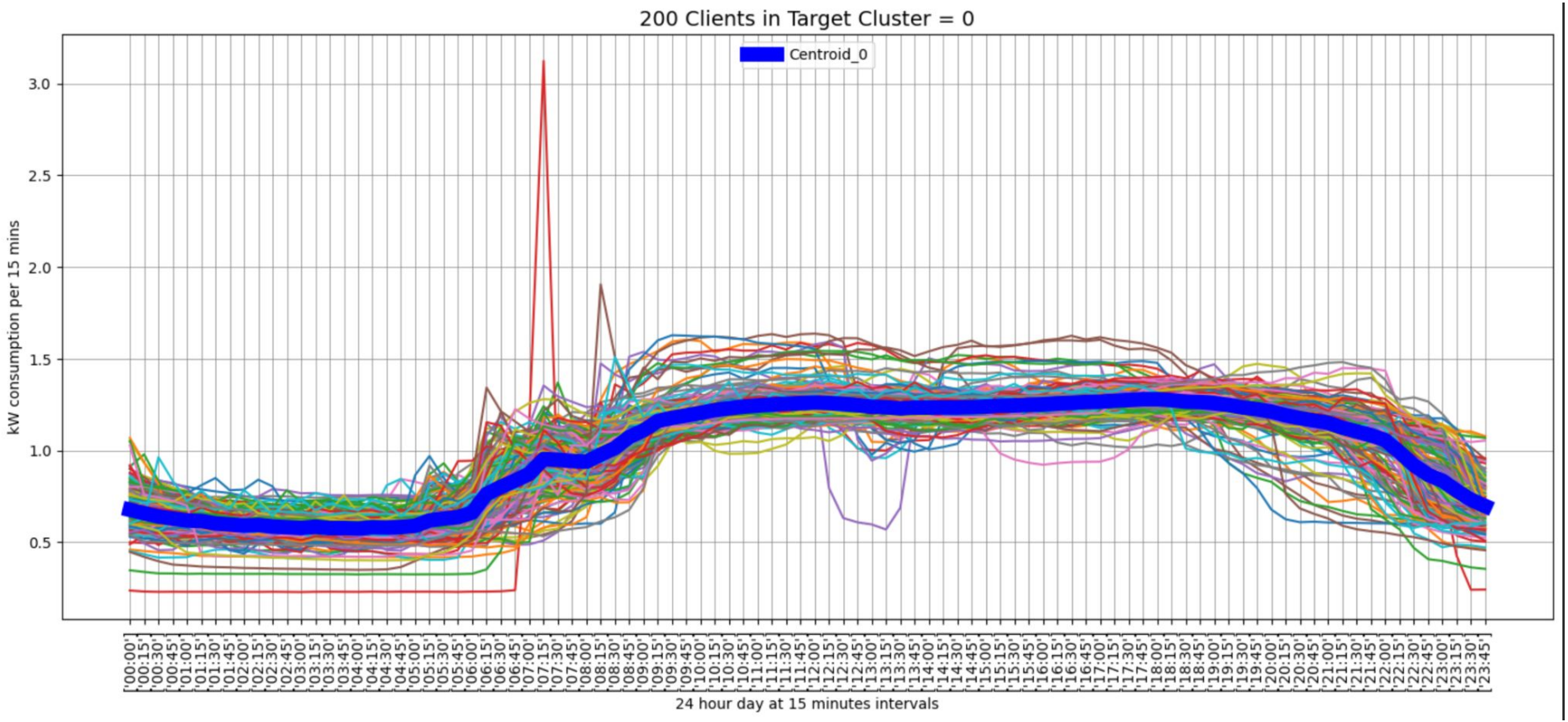
```
# Selected number of clusters
trueK = 5

# Train the model
print(X)
kmeans = KMeans(n_clusters=trueK, random_state=0).fit(X) # train the model

# Get the centroids
centroids = kmeans.cluster_centers_

# Divide each centroid by its average
centroids = centroids / np.mean(centroids)

# Inspect
centroids.shape
```

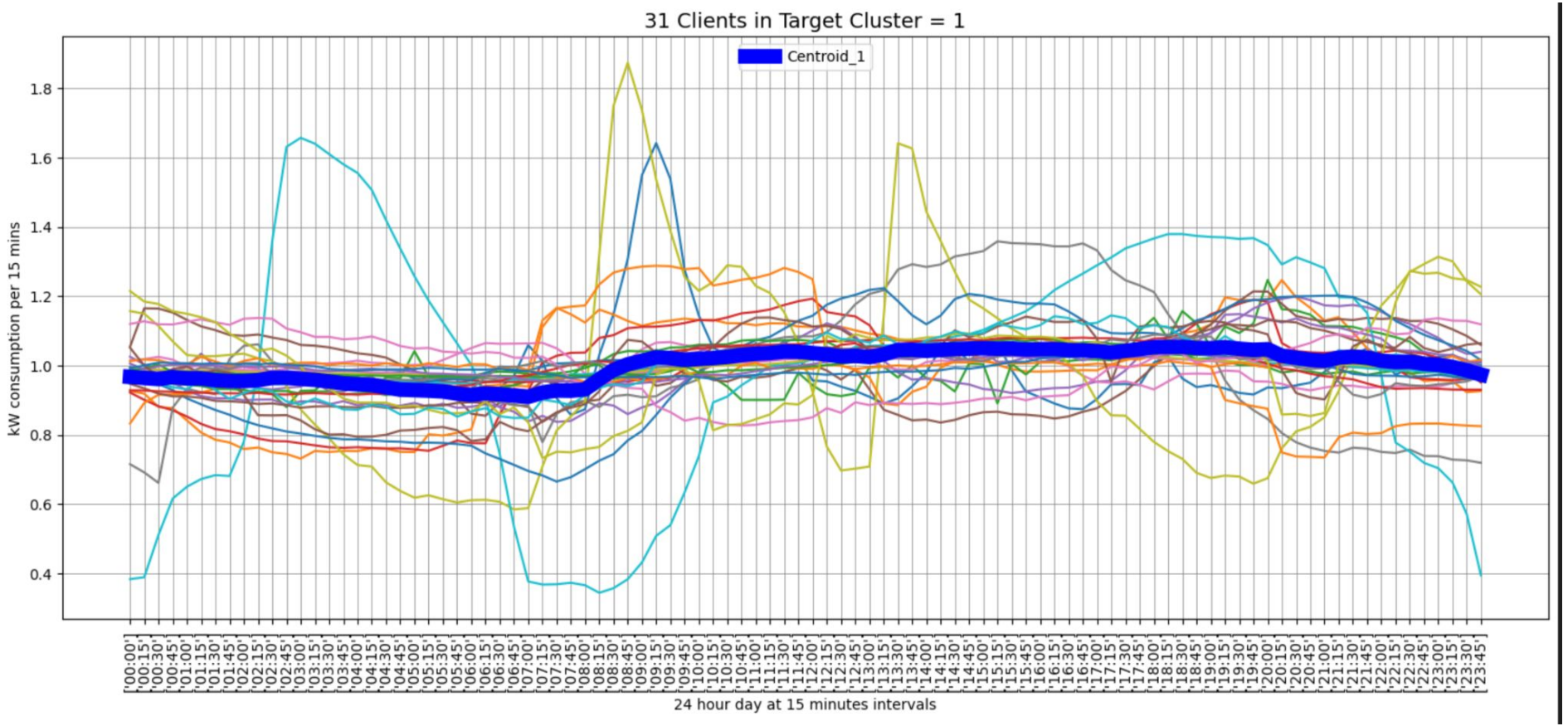
```
(5, 96)
```

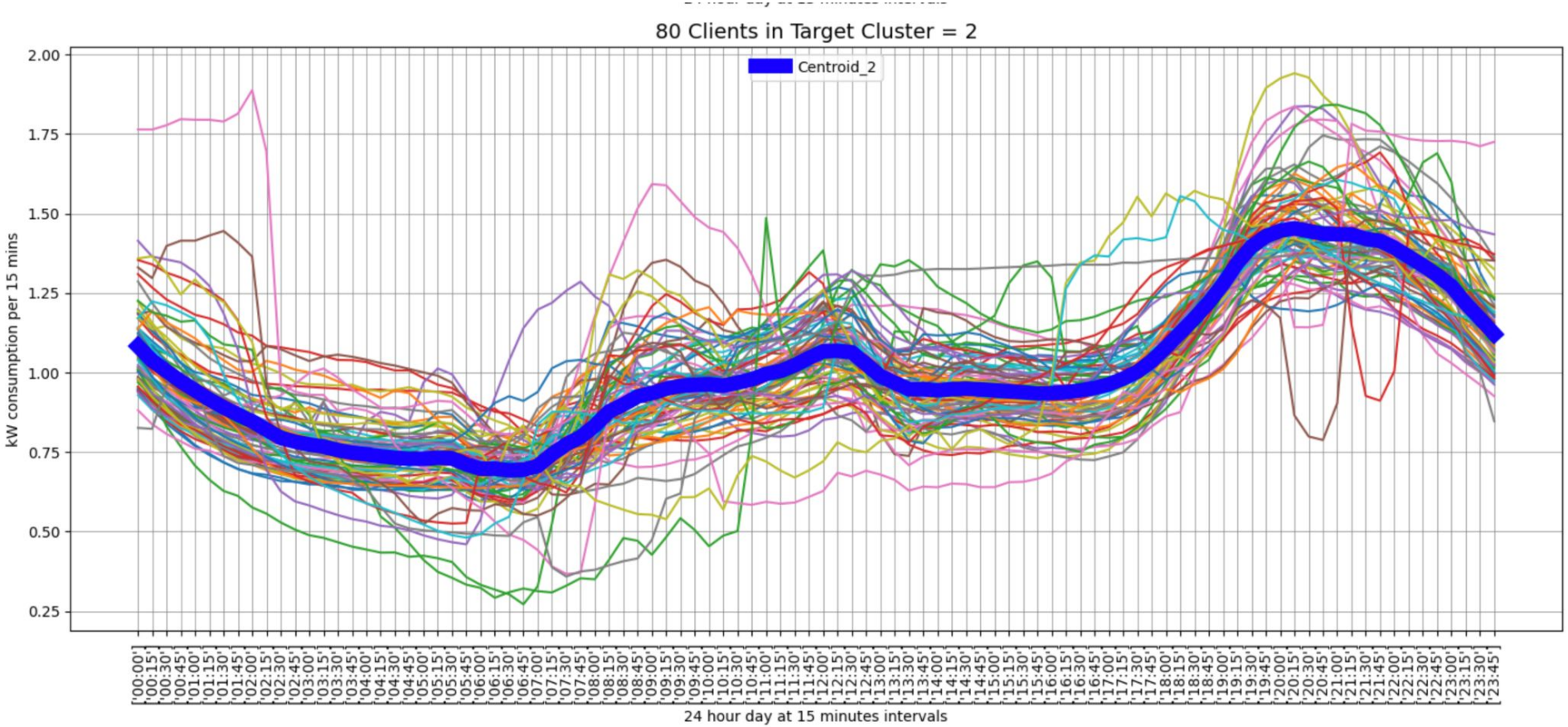
	Cluster	Client
174	0	174
218	0	218
217	0	217
216	0	216
215	0	215
...	...	...
324	3	324
115	4	115
116	4	116
326	4	326
117	4	117

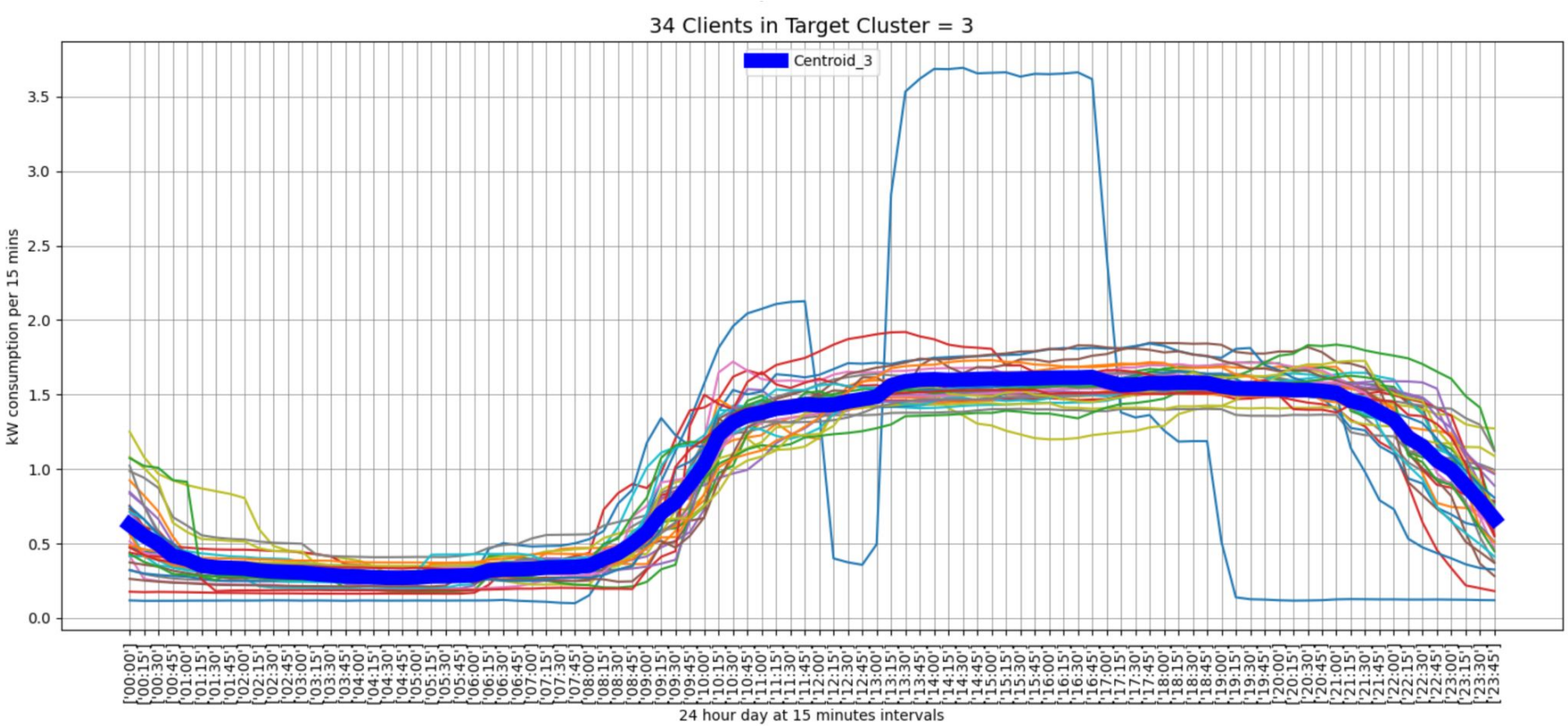
349 rows × 2 columns

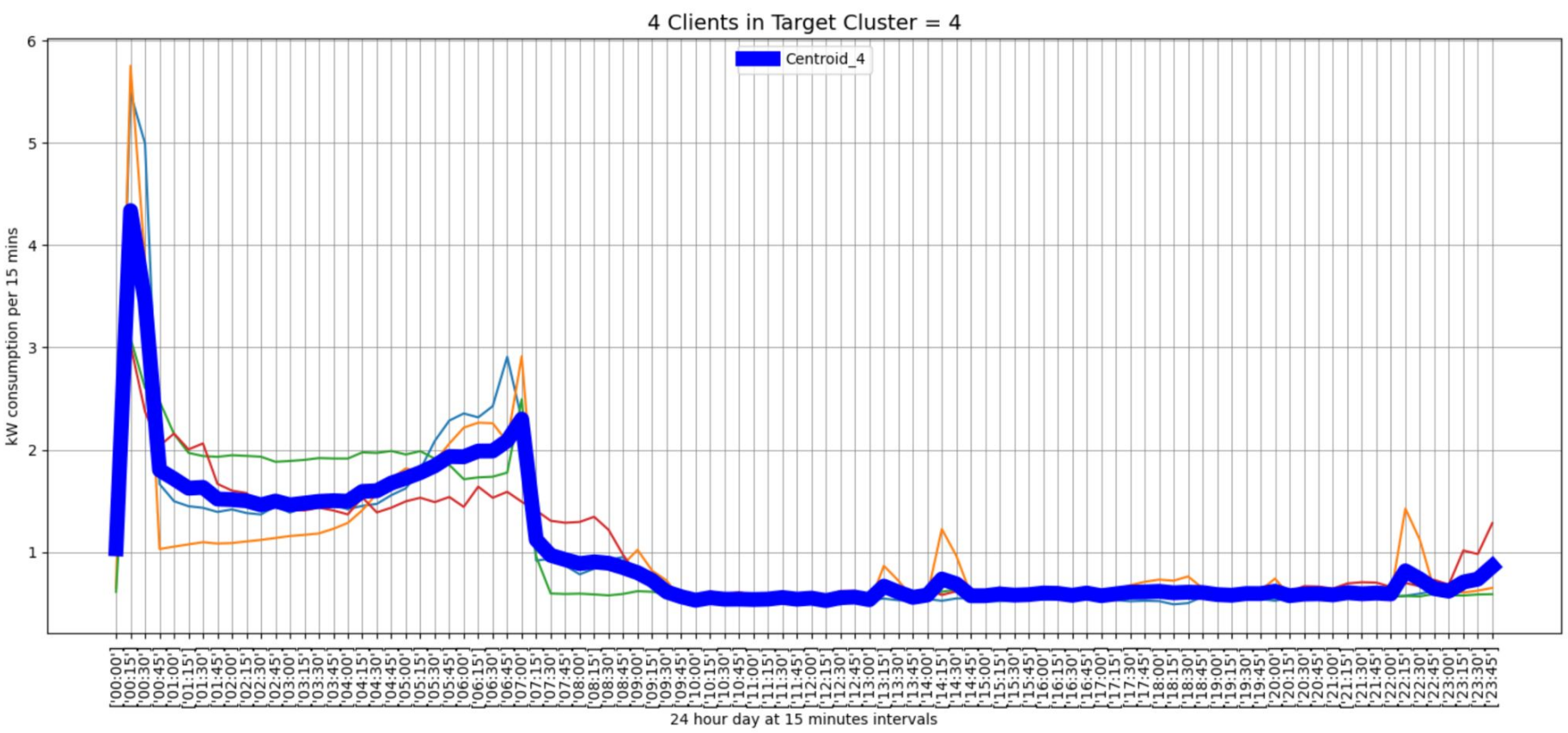
```
Cluster
0    200
1     31
2     80
3     34
4      4
Name: Client, dtype: int64
```


200 Clients in Target Cluster = 0
Centroid_0
kW consumption per 15 mins
3.0
2.5
2.0
1.5
1.0
0.5
24 hour day at 15 minutes intervals


31 Clients in Target Cluster = 1
Centroid_1
kW consumption per 15 mins
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
['00:00']
['00:15']
['00:30']
['00:45']
['01:00']
['01:15']
['01:30']
['01:45']
['02:00']
['02:15']
['02:30']
['02:45']
['03:00']
['03:15']
['03:30']
['03:45']
['04:00']
['04:15']
['04:30']
['04:45']
['05:00']
['05:15']
['05:30']
['05:45']
['06:00']
['06:15']
['06:30']
['06:45']
['07:00']
['07:15']
['07:30']
['07:45']
['08:00']
['08:15']
['08:30']
['08:45']
['09:00']
['09:15']
['09:30']
['09:45']
['10:00']
['10:15']
['10:30']
['10:45']
['11:00']
['11:15']
['11:30']
['11:45']
['12:00']
['12:15']
['12:30']
['12:45']
['13:00']
['13:15']
['13:30']
['13:45']
['14:00']
['14:15']
['14:30']
['14:45']
['15:00']
['15:15']
['15:30']
['15:45']
['16:00']
['16:15']
['16:30']
['16:45']
['17:00']
['17:15']
['17:30']
['17:45']
['18:00']
['18:15']
['18:30']
['18:45']
['19:00']
['19:15']
['19:30']
['19:45']
['20:00']
['20:15']
['20:30']
['20:45']
['21:00']
['21:15']
['21:30']
['21:45']
['22:00']
['22:15']
['22:30']
['22:45']
['23:00']
['23:15']
['23:30']
['23:45']
24 hour day at 15 minutes intervals


80 Clients in Target Cluster = 2
Centroid_2
kW consumption per 15 mins
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
24 hour day at 15 minutes intervals


34 Clients in Target Cluster = 3
Centroid_3
kW consumption per 15 mins
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
24 hour day at 15 minutes intervals


4 Clients in Target Cluster = 4
Centroid_4
kW consumption per 15 mins
24 hour day at 15 minutes intervals

Part 2 : Daily Curves of a 1 single client

Purpose

Previous Clustering : Understand electric power usage trend of multiple clients

Current Clustering : **Understand distinctive power usage pattern of an individual from their daily curves of 730 days.**

Select a single client and normalize

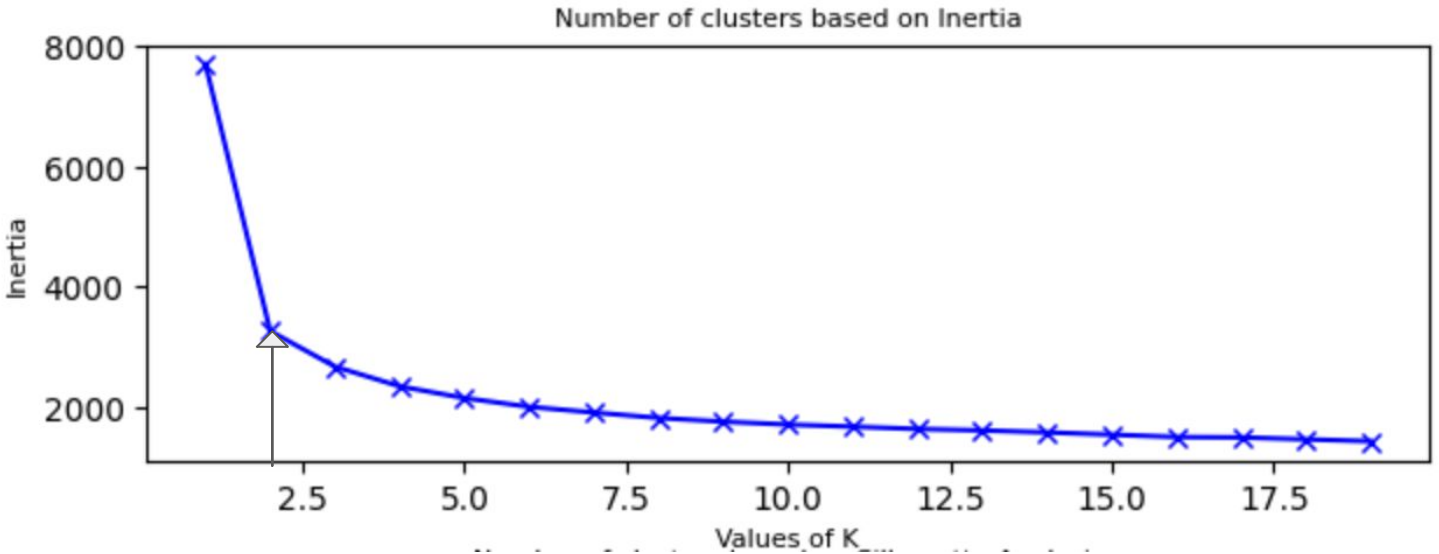
```
# Selecting a random single client
client = 'MT_022'
oneClient = data_13_14[client]

# Initialize list of arrays, each array being a normalized curve for a day
X = []
for J in range(2*365):
    X.extend([np.array(oneClient[J*96:(J+1)*96])])
X = X / np.mean(X)
```

k-Means Clustering

```
for num_clusters_2a in elbow_K_2a:
    elbow_kmns_2a = KMeans(n_clusters=num_clusters_2a)
    elbow_kmns_2a.fit(X)
    Sum_of_squared_distances_2a.append(elbow_kmns_2a.inertia_)
```

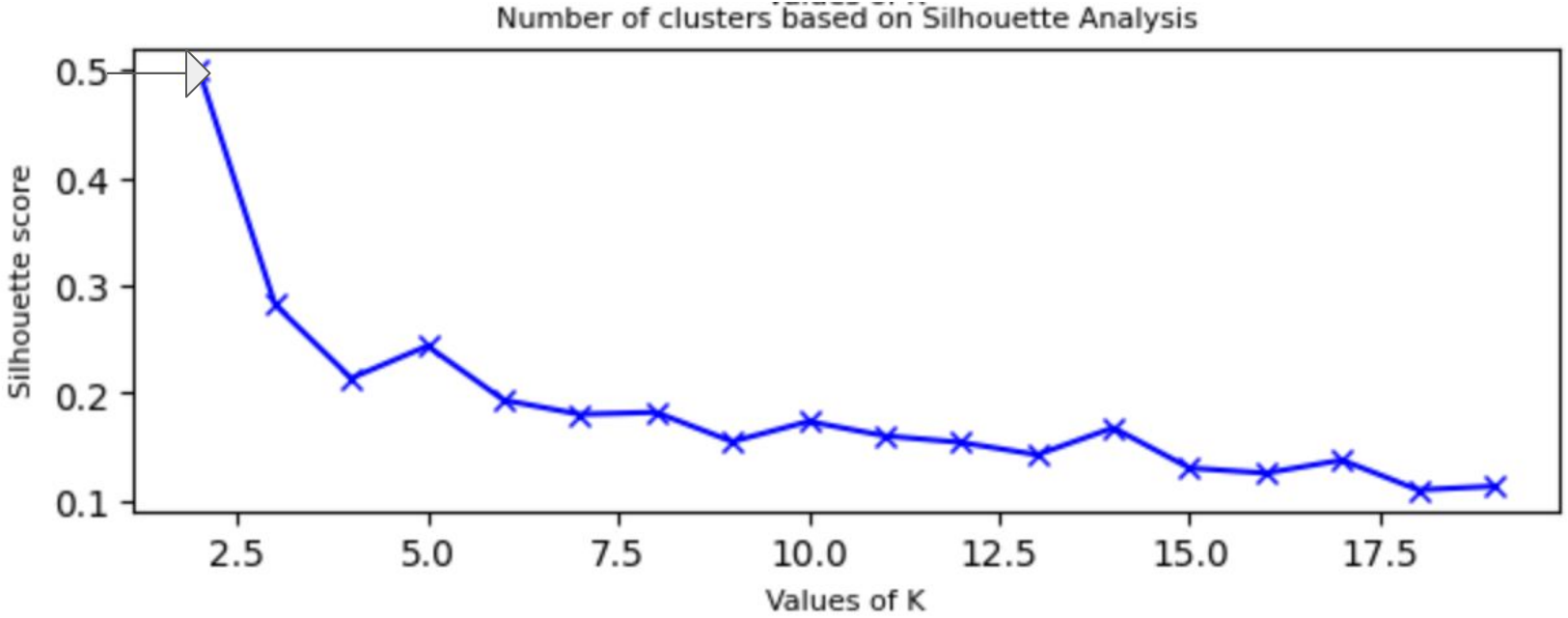
Again, use elbow method to obtain optimal “k” value

→ k = 2



Verify “k” : Silhouette Analysis

k = 2 gives the highest silhouette score



What are the clusters represent?

```
centroids_2b.shape
```

```
(2, 96)
```

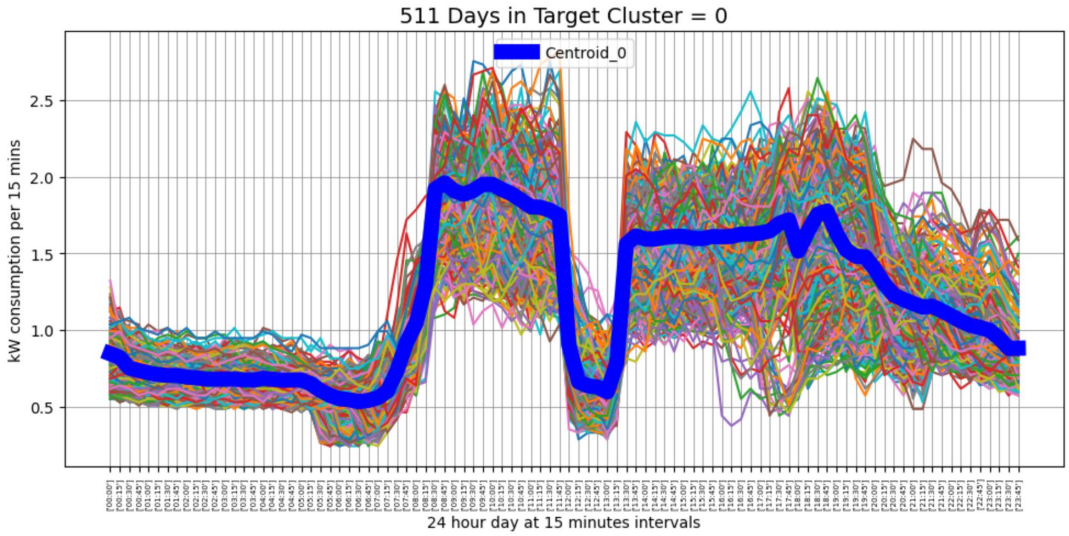
2 centroids (from 2 clusters)

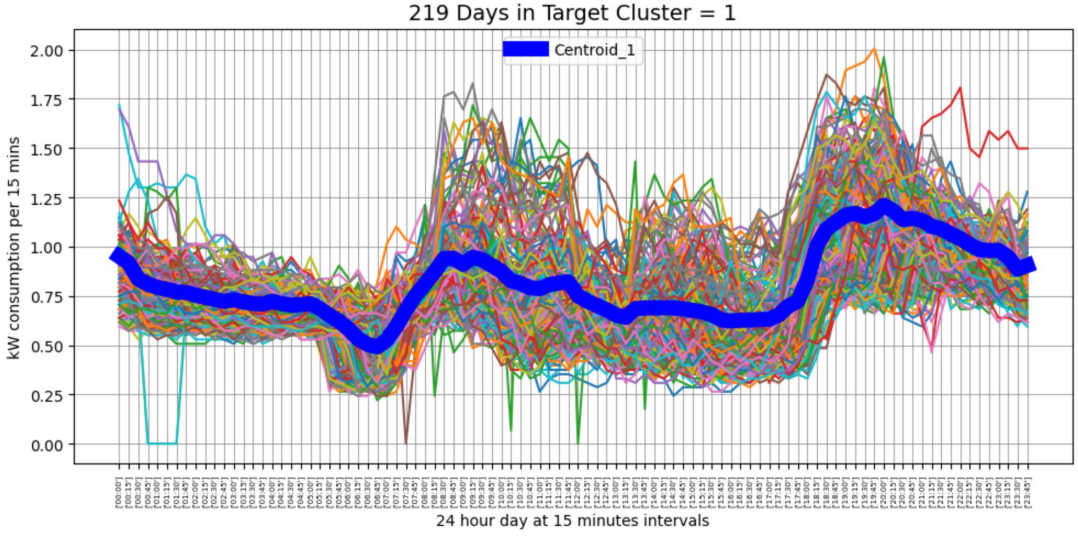
24 hr * 4 (15min interval) = **96** attributes

	Cluster	Data
279	0	279
611	0	611
728	0	728
363	0	363
612	0	612
...	...	...
348	1	348
341	1	341
334	1	334
361	1	361
729	1	729

730 rows × 2 columns

```
Cluster
0    511
1    219
Name: Data, dtype: int64
```


511 Days in Target Cluster = 0
Centroid_0
kW consumption per 15 mins
24 hour day at 15 minutes intervals


219 Days in Target Cluster = 1
Centroid_1
kW consumption per 15 mins
24 hour day at 15 minutes intervals

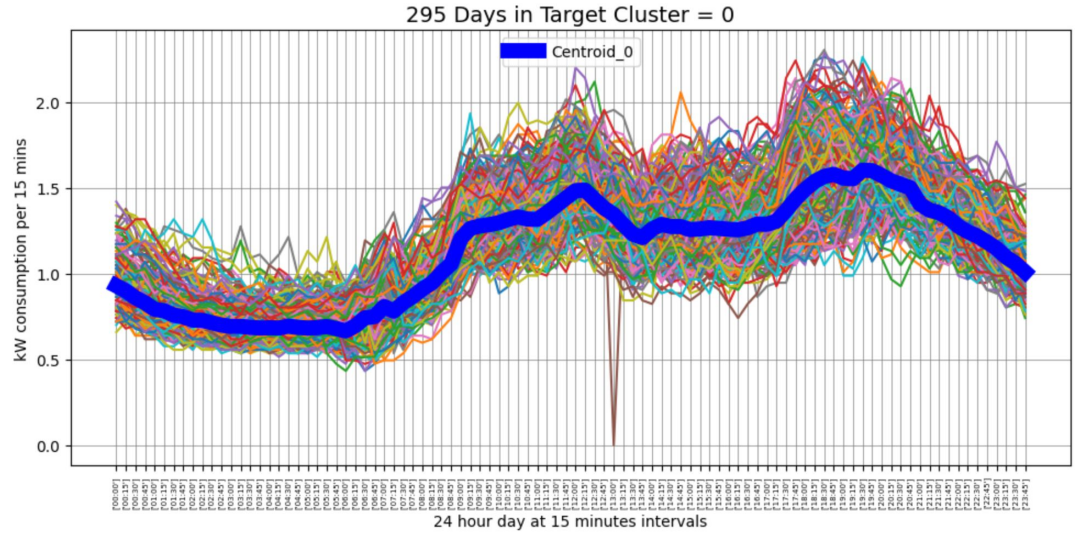
Is k-Means the best clustering method?

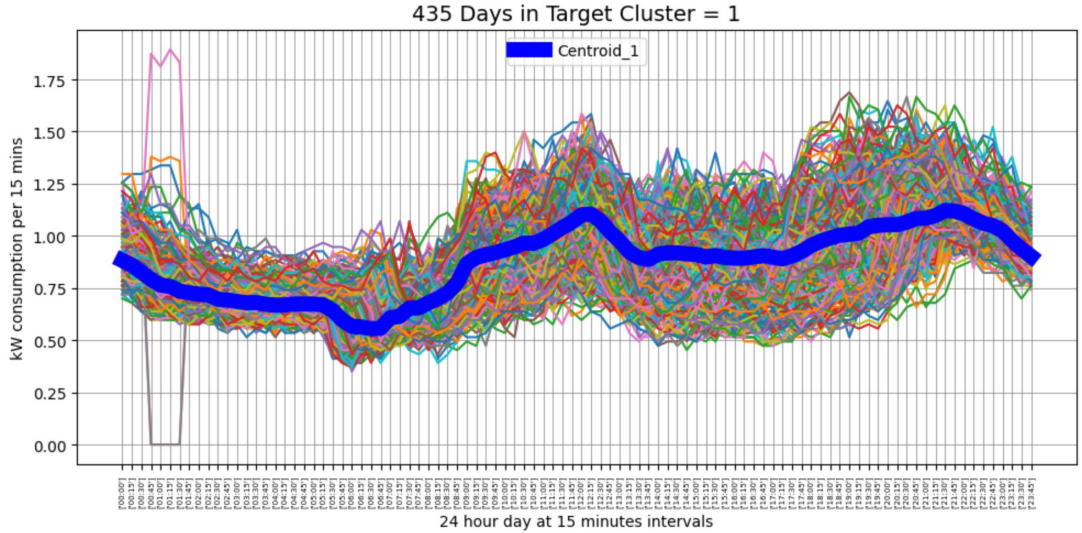
Limitation of K means

For MT_033, two clusters are similar

-Dimension of 96

-Non spherical cluster?





k-Means vs DBSCAN

time series distance / discrete wavelet transform

Criteria	**Y/N**	**k-Means**	**DBSCAN**	**Priority**
Are clusters formed in arbitrary shape?	Probably	-1	+1	1
High Dimension?	Yes	-1	-1	
Should consider noise?	Yes	-1	+1	
Is it okay to consider local region only?	No	+1	-1	2
Is it hard to predefine the #of cluster?	No	+1	N/A	
Is it hard to determine Eps and Minpts?	Yes	N/A	-1	3
Complexity?	k-Means faster	+1	-	

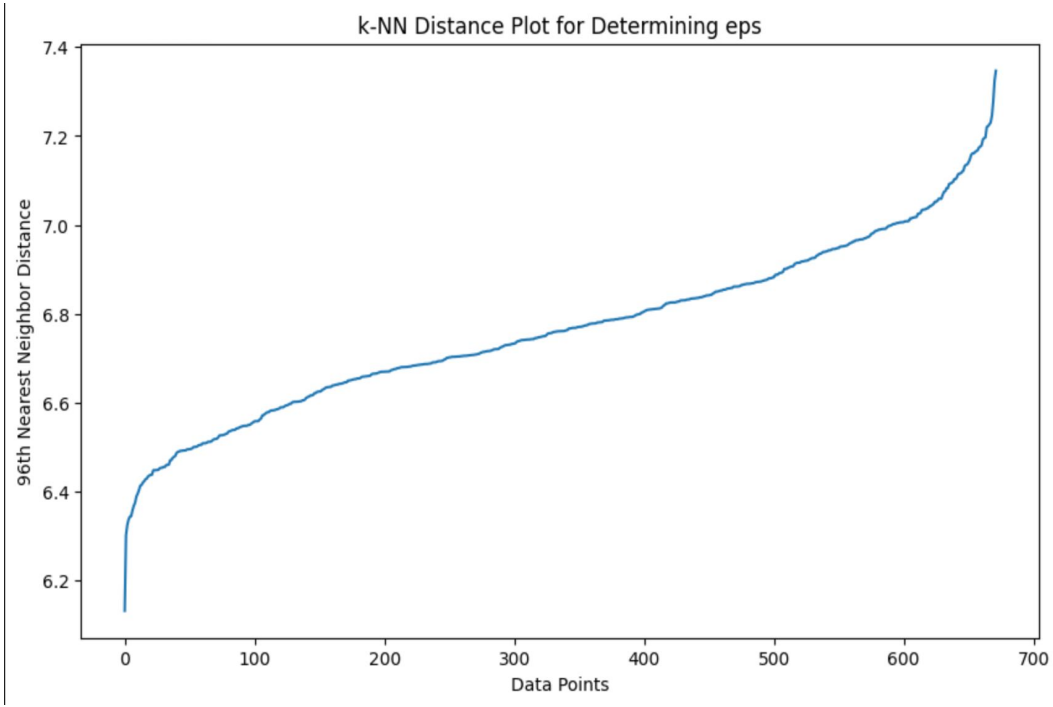
DBSCAN Implementation

Step 1 : Find parameter

min_sample = dimension

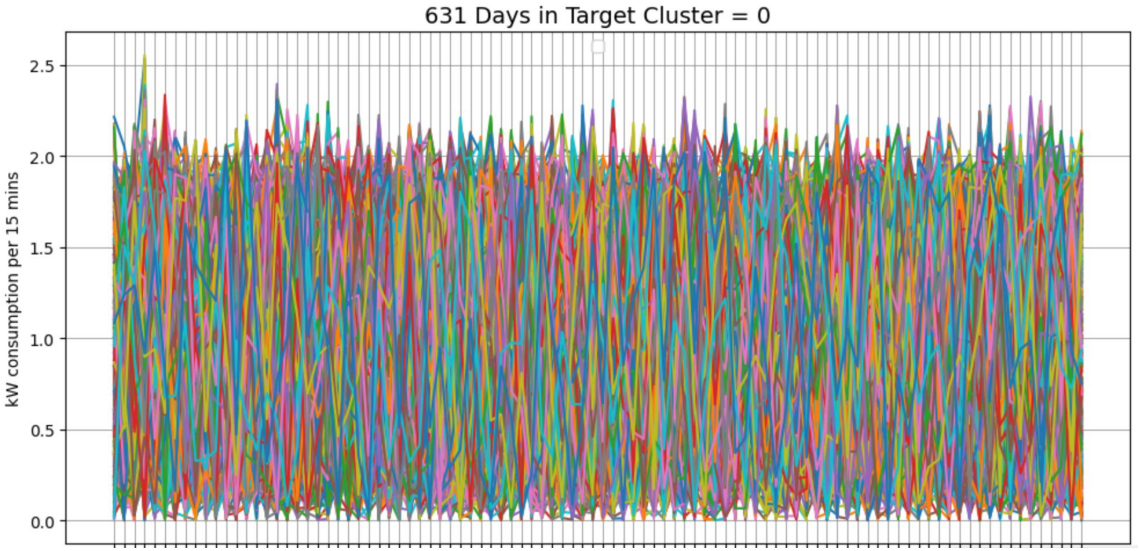
epsilon = Use k-NN distance plot

```
eps = 7.2
min_samples = 96
```



DBSCAN Implementation

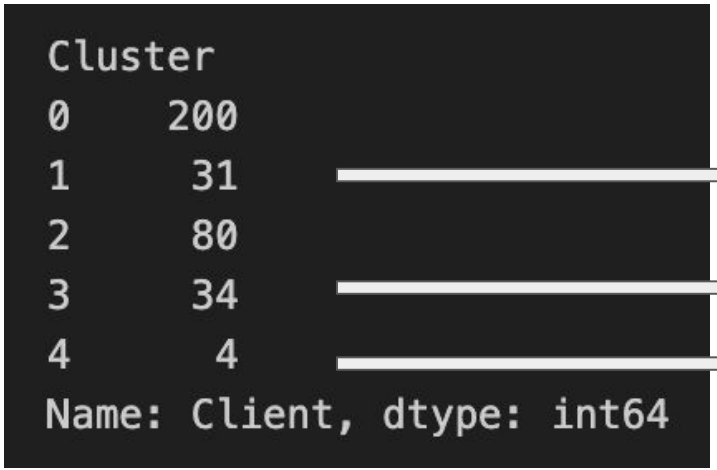
Step 2 : Implement DBSCAN and find clusters



Not effective for all range of min_pts and eps

Try DBSCAN with a more clustered dataset

Previously, in part 1, 'Daily Electric Power Usage Trends of 270 Clients' showcased 5 different clusters with k-Means clustering technique.



Previous clusters using k-Means

Setting min_pts using this,

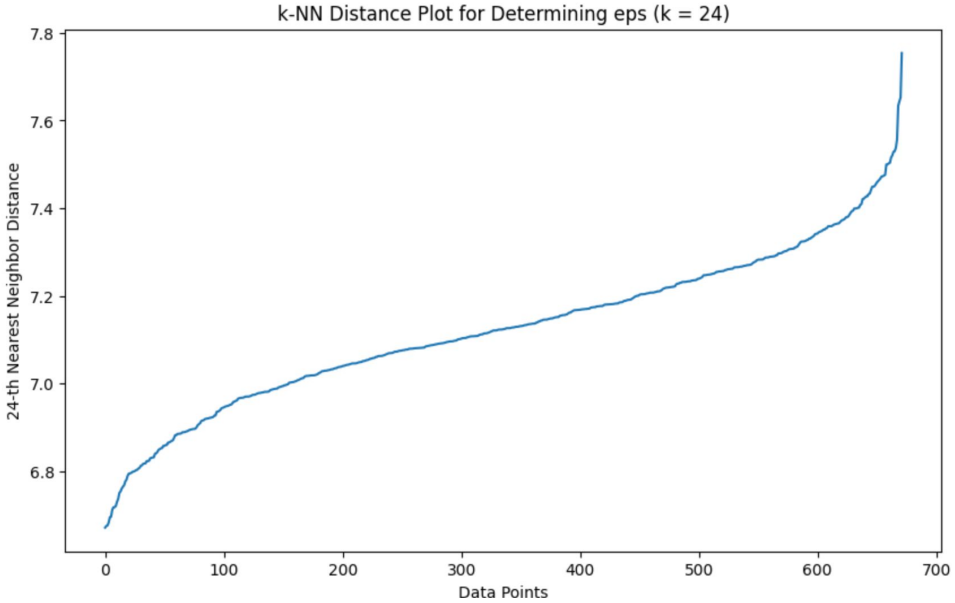
Two small clusters with dataset around 30 suggests min_pts just below 30 wil yield similar cluster

Consider this cluster as an outlier

Try DBSCAN with a more clustered dataset

```
min_samples = 25
```

```
eps = 7.5
```




```
Estimated number of clusters: 1
Total data points: 672
Clustered points: 672
Outlier points: 0
```

Again, just a single cluster

Discrete Wavelet Transform

Step 1: Transform data to time series and obtain wavelet coefficient

Step 2 : Choose the best coefficient

Step 3 : Visualization of reduced dimension by reconstruction through inverse transformation

Step 4 : Use clustering techniques with reduced dimensionality

Discrete Wavelet Transform Setup

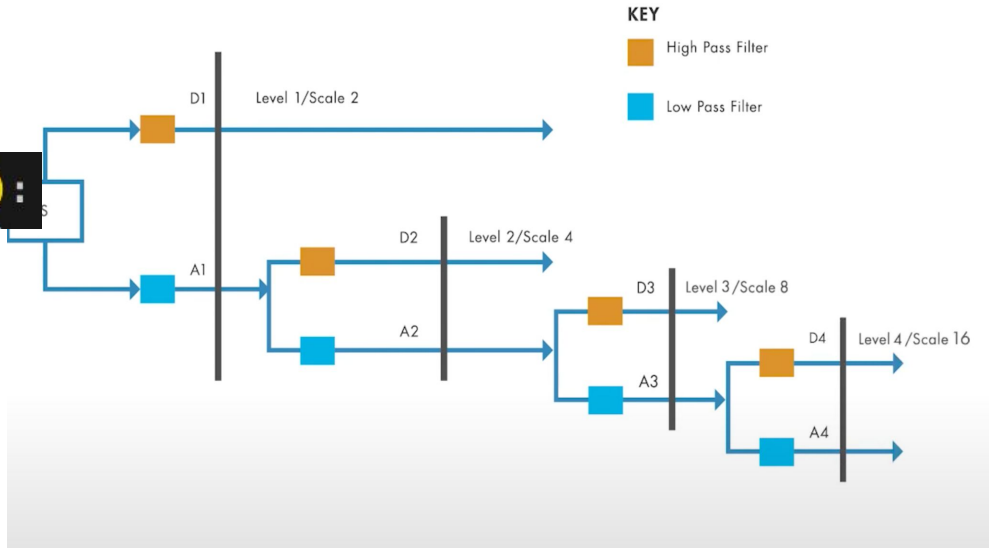
Parameter

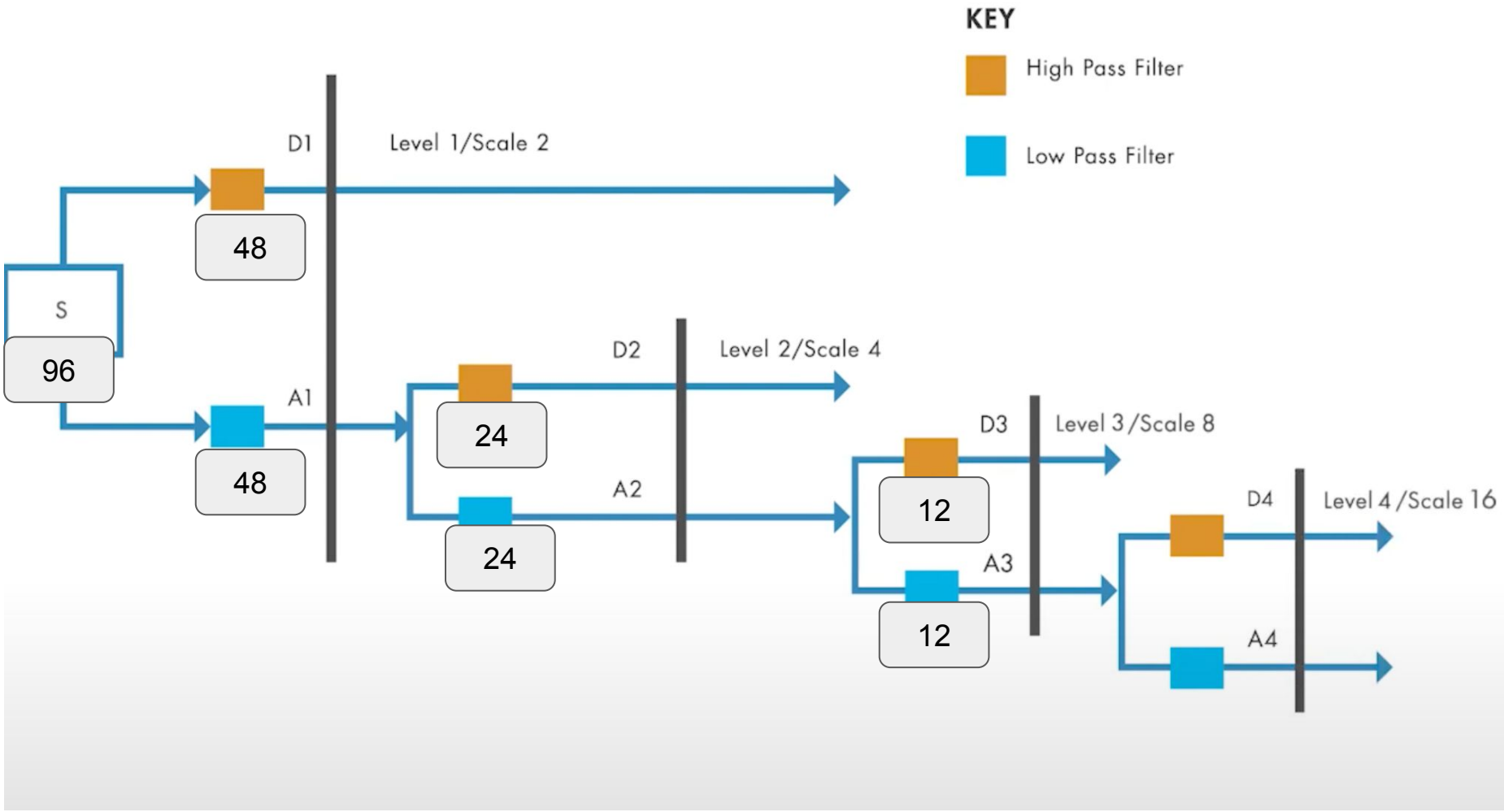
wavelet = db1 : Daubechies wavelet -> orthogonal (transformed coeff uncorrelated)

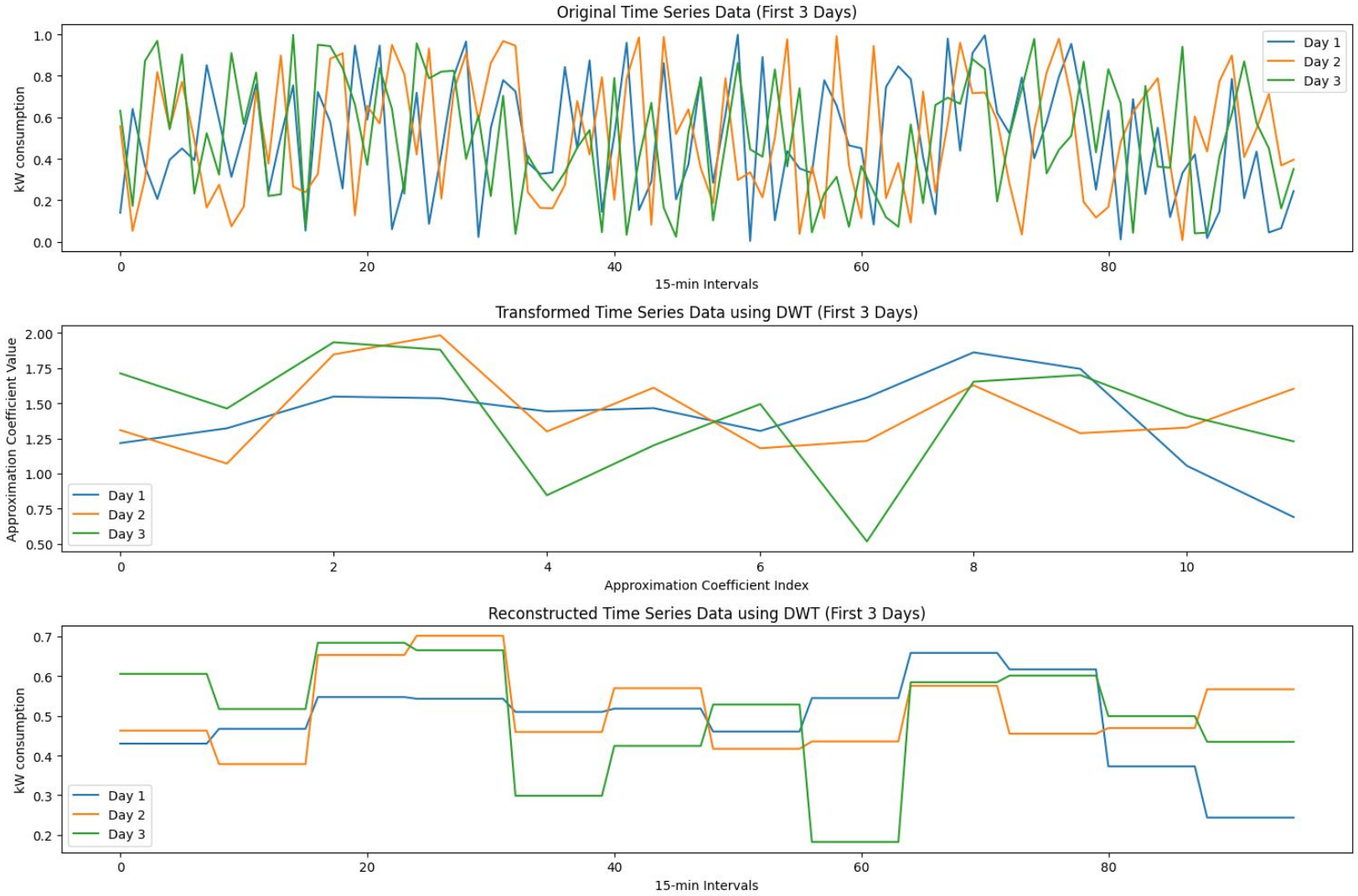
level = 3 : # of decomposition

```
def apply_dwt(X, wavelet='db1', level=3):
```

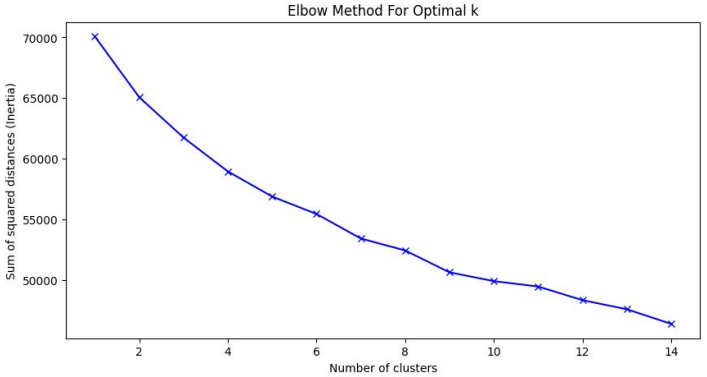
```
return transformed_data, coeffs
```

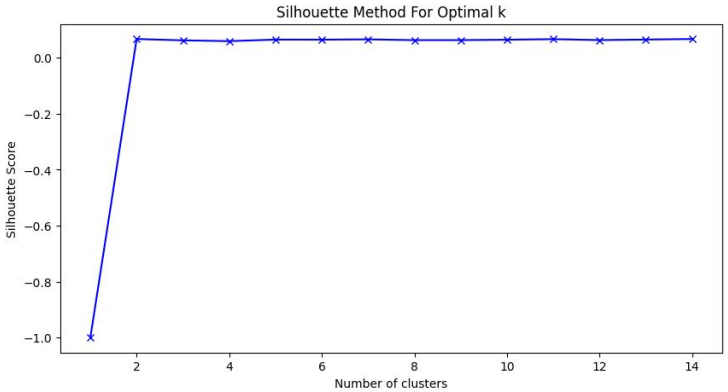



KEY
High Pass Filter
Low Pass Filter
D1
Level 1/Scale 2
48
S
96
A1
48
D2
Level 2/Scale 4
24
A2
24
D3
Level 3/Scale 8
12
A3
12
D4
Level 4/Scale 16
A4

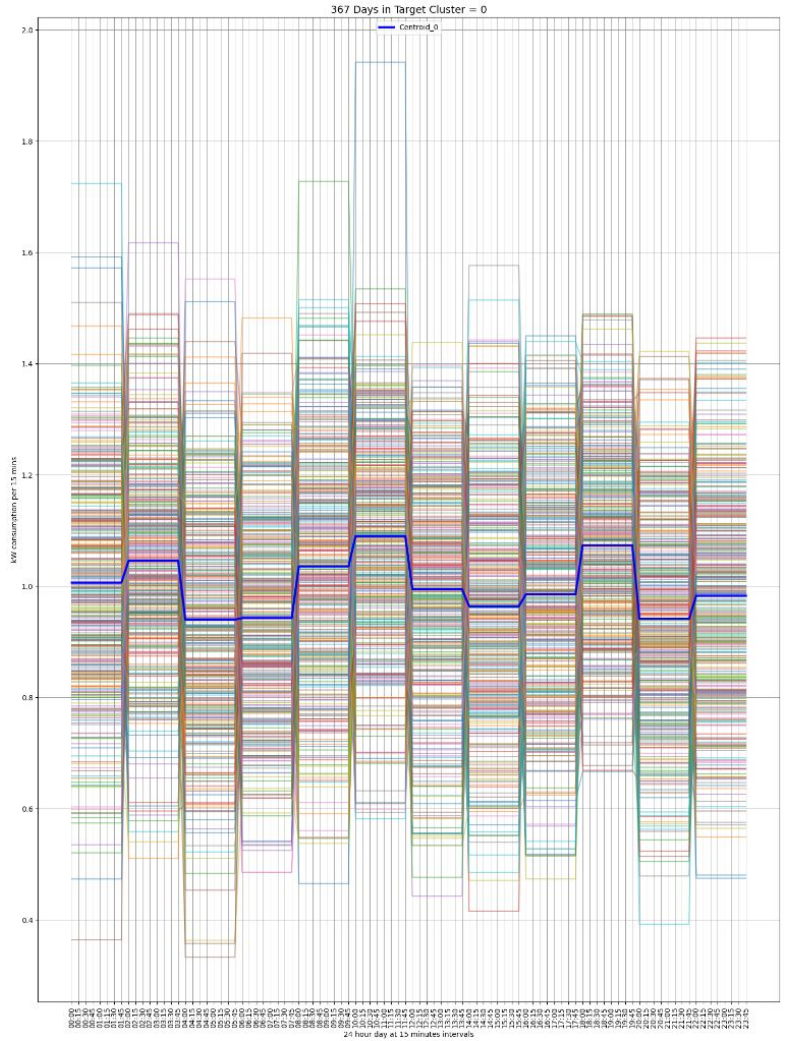

Original Time Series Data (First 3 Days)
kW consumption
15-min Intervals
Day 1
Day 2
Day 3
Transformed Time Series Data using DWT (First 3 Days)
Approximation Coefficient Value
Approximation Coefficient Index
Day 1
Day 2
Day 3
Reconstructed Time Series Data using DWT (First 3 Days)
kW consumption
15-min Intervals
Day 1
Day 2
Day 3

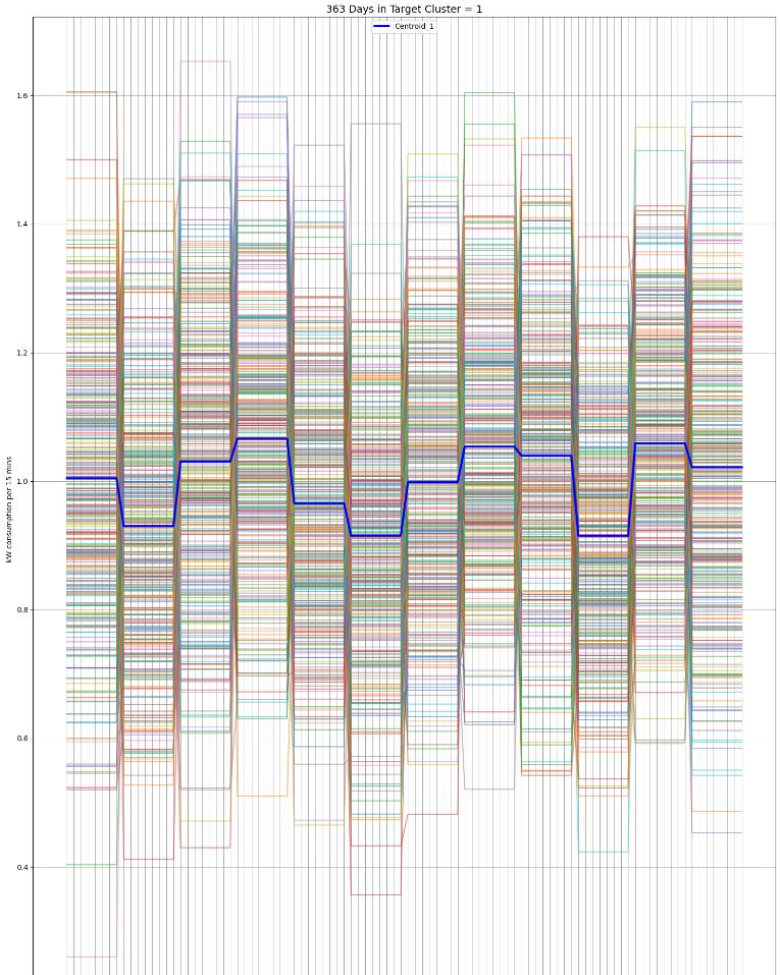
k-Means clustering after DWT

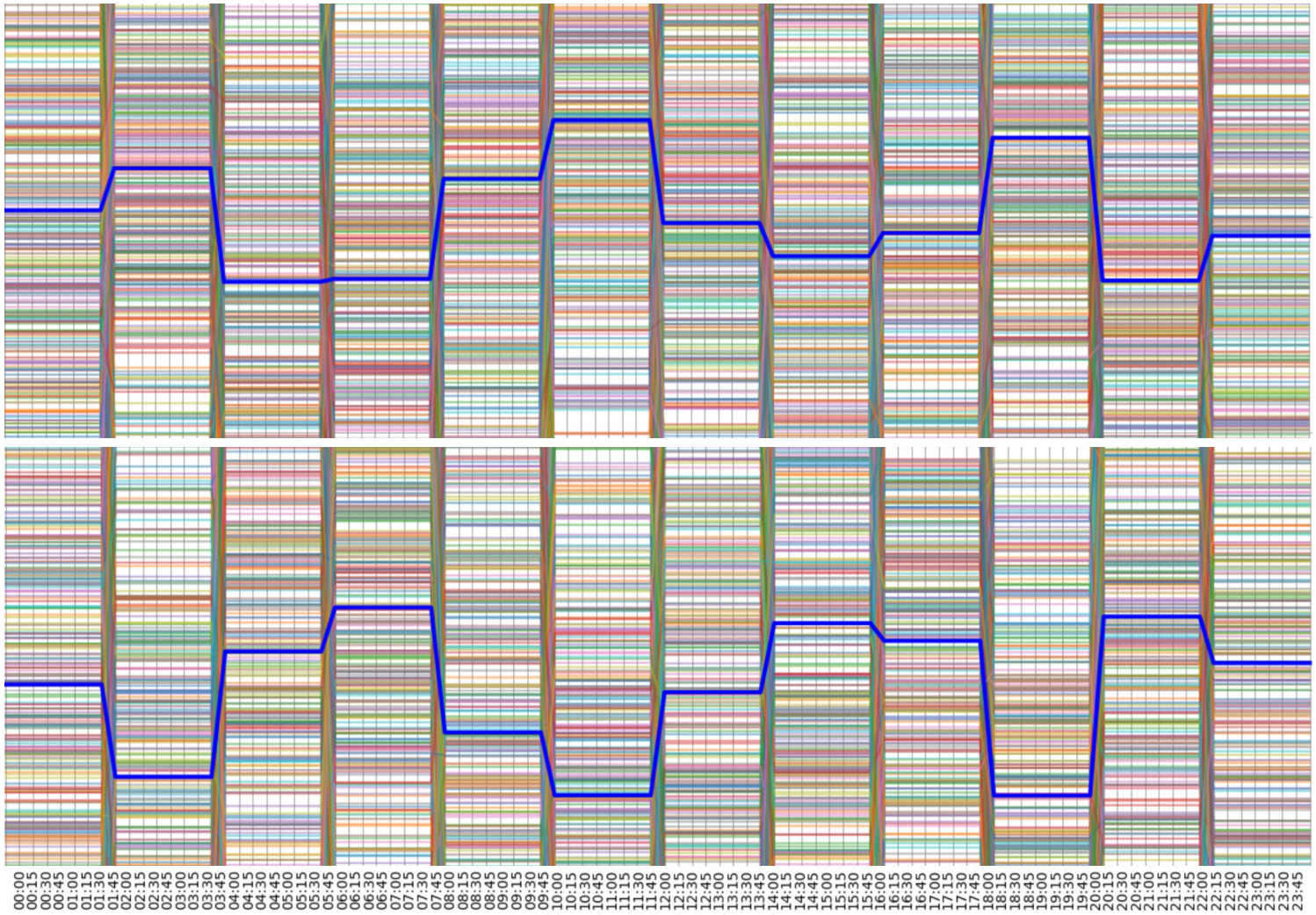


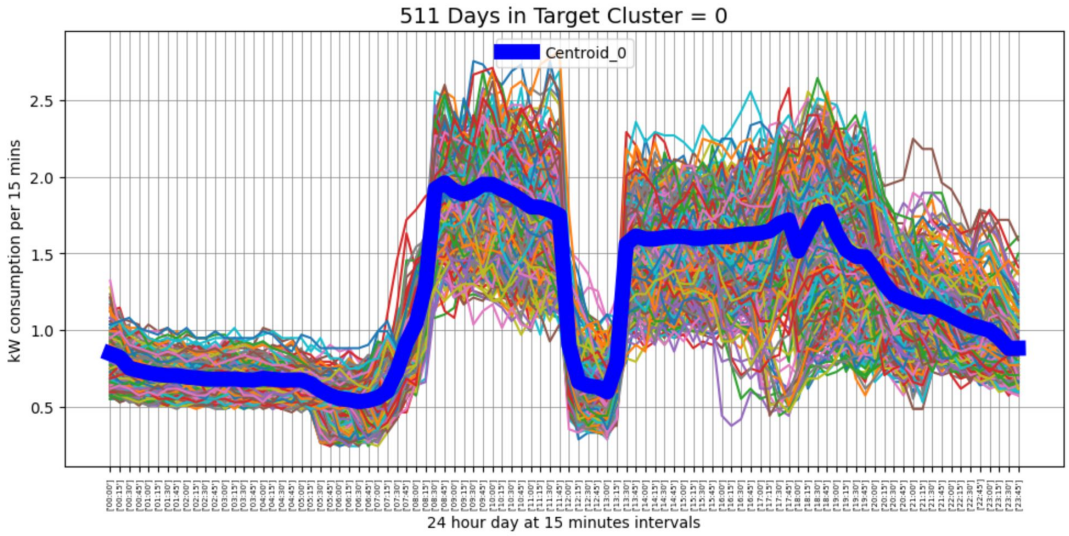


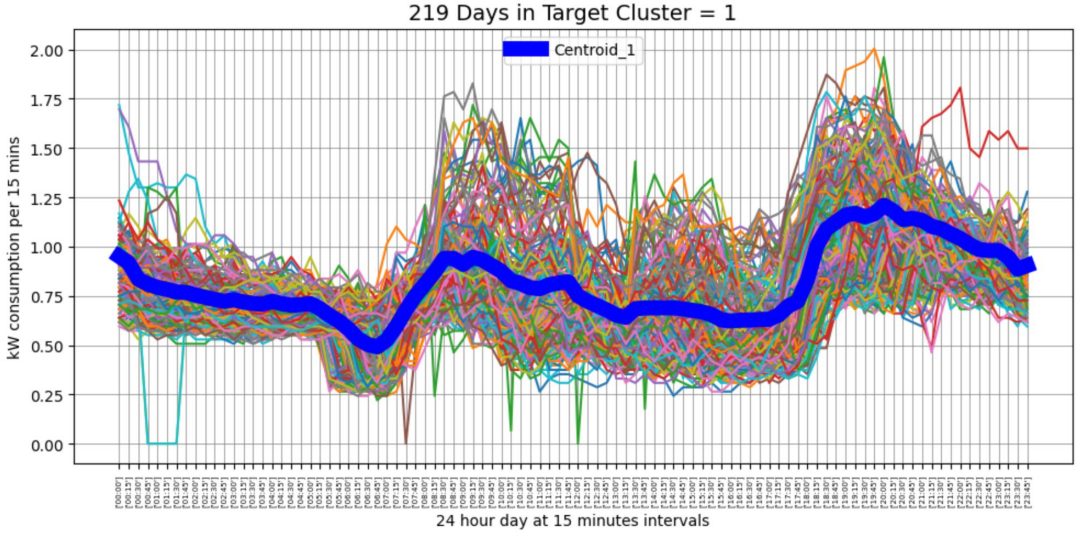

```
...   Centroids shape: (2, 96)
      Cluster  Client
0           0       0
414         0     414
411         0     411
410         0     410
409         0     409
..        ...     ...
152         1     152
153         1     153
416         1     416
521         1     521
364         1     364

[730 rows x 2 columns]
Cluster
0    367
1    363
Name: Client, dtype: int64
```


367 Days in Target Cluster = 0
Centroid_0
kW consumption per 15 mins
24 hour day at 15 minutes intervals


363 Days in Target Cluster = 1
Centroid_1
kW consumption per 15 mins


00:00
00:15
00:30
00:45
01:00
01:15
01:30
01:45
02:00
02:15
02:30
02:45
03:00
03:15
03:30
03:45
04:00
04:15
04:30
04:45
05:00
05:15
05:30
05:45
06:00
06:15
06:30
06:45
07:00
07:15
07:30
07:45
08:00
08:15
08:30
08:45
09:00
09:15
09:30
09:45
10:00
10:15
10:30
10:45
11:00
11:15
11:30
11:45
12:00
12:15
12:30
12:45
13:00
13:15
13:30
13:45
14:00
14:15
14:30
14:45
15:00
15:15
15:30
15:45
16:00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
19:15
19:30
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
21:45
22:00
22:15
22:30
22:45
23:00
23:15
23:30
23:45
24 hour day at 15 minutes intervals

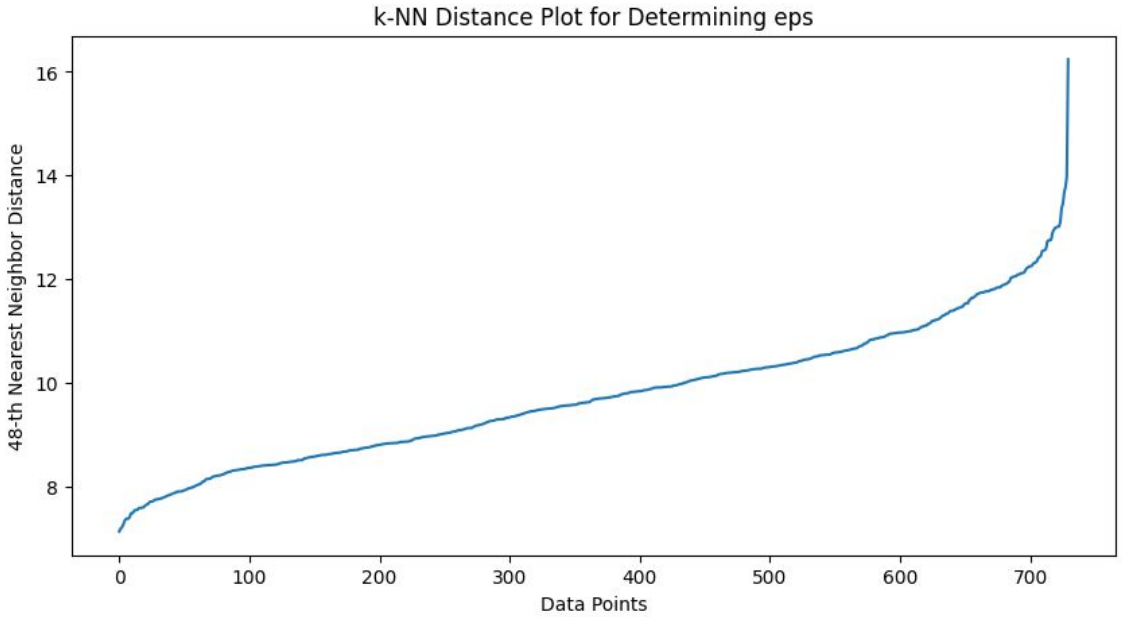

511 Days in Target Cluster = 0
Centroid_0
kW consumption per 15 mins
24 hour day at 15 minutes intervals

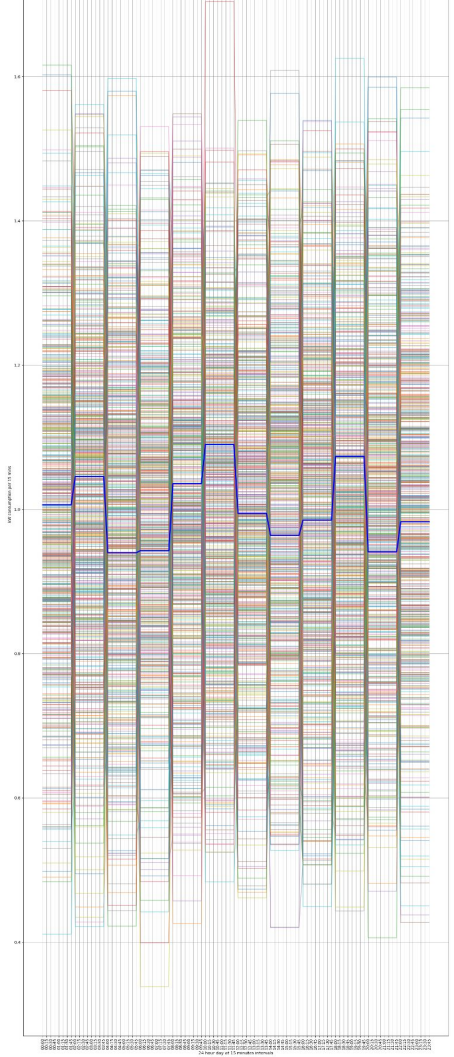

219 Days in Target Cluster = 1
Centroid_1
kW consumption per 15 mins
24 hour day at 15 minutes intervals

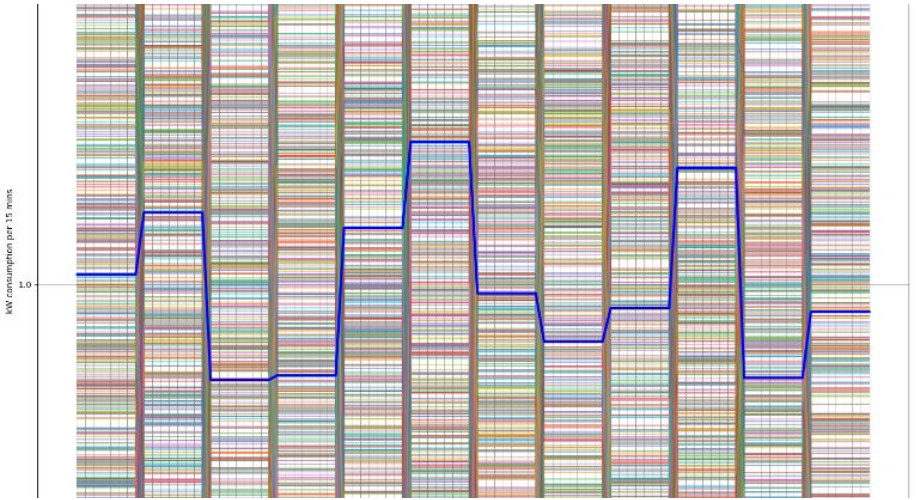
DBSCAN after DWT

```
min_samples = 24 # 2 * number of dim
```

```
eps = 12
```








```
Estimated number of clusters: 1
Total data points: 730
Clustered points: 730
Outlier points: 0
     Cluster  Client
0          0       0
481        0     481
482        0     482
483        0     483
484        0     484
..       ...     ...
246        0     246
247        0     247
248        0     248
250        0     250
729        0     729

[730 rows x 2 columns]
Cluster
0    730
Name: Client, dtype: int64
```