Visualising Smart Meter data by Manifold Learning using Dynamic Time Warping

Presented by: Maria Lim

Introduction

- Data – high-dimensional - multivariate & vast
- Explore & visualise – dimensionality reduction
- Find low-dimensional manifolds of the data hidden in their high-dimensional space
- Detect households with anomalous electricity demand
- Incorporate dynamic time warping

Outline

- Dynamic Time Warping (DTW)
- Dimensionality Reduction
- Data used and preparation
- Results
- Conclusion

Dynamic Time Warping (DTW)

- Aim: measure - similarity by finding the optimal match - yields a minimal cost (the distance) under certain restrictions:
- Let two time series, $q = [q_1, q_2, q_3, \dots, q_N]$ and $r = [r_1, r_2, r_3, \dots, r_M]$
- 1) Monotonicity condition. Eg: $q_3 \rightarrow r_3$, $q_4 \cancel{\rightarrow} r_2\ or\ r_1$
- 2) Continuity condition
- 3) Boundary condition. Eg: $q_1 \rightarrow r_1$ and $q_N \rightarrow r_M$

DTW Algorithm

1. **Initialisation:**

   for $i = 1\ to\ N$: $D_{i,0} = \infty$

   for $j = 1\ to\ M$: $D_{0,j} = \infty$

   $D_{0,0} = 0$

2. **Calculate cost matrix:**

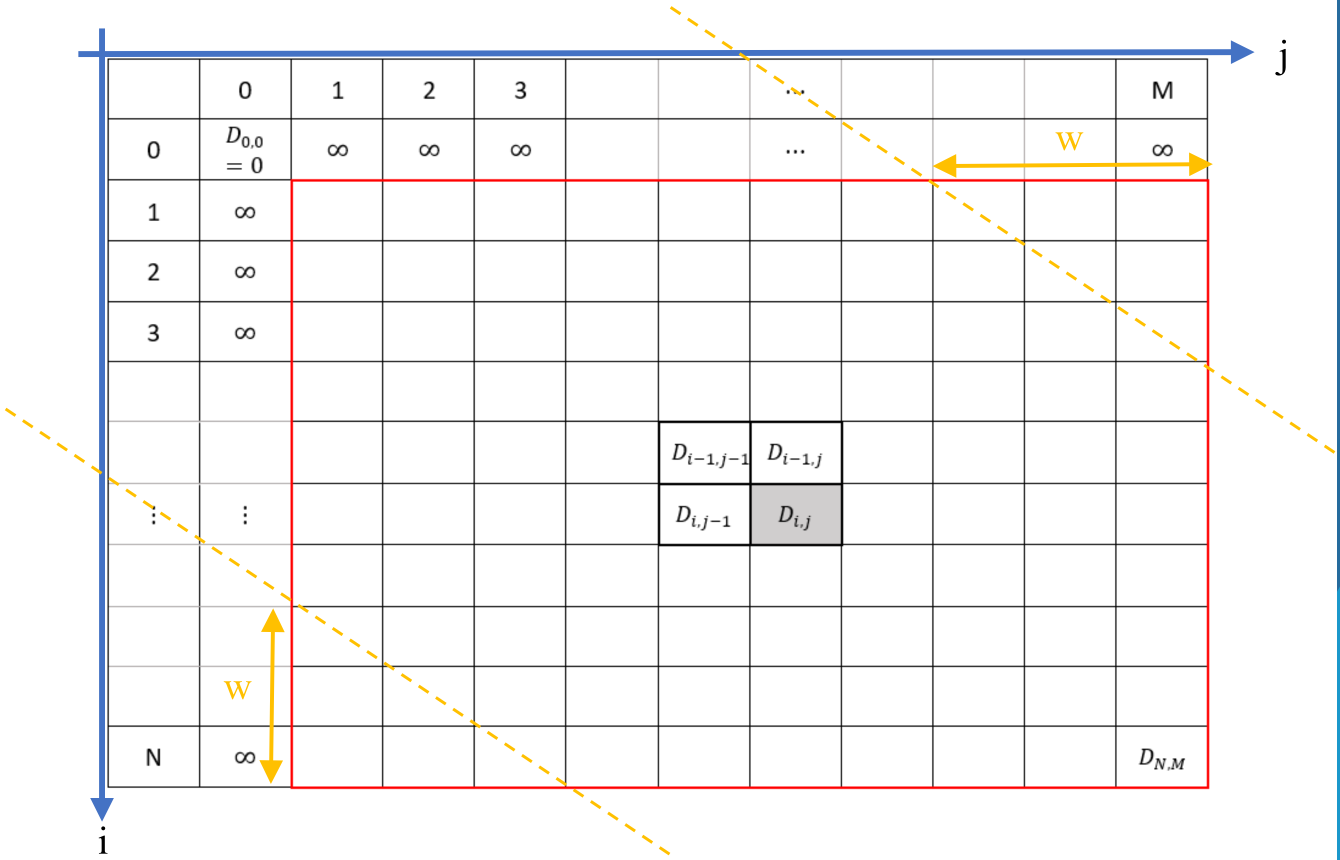
   for $i = 1\ to\ N$:

   for $j = 1\ to\ M$:

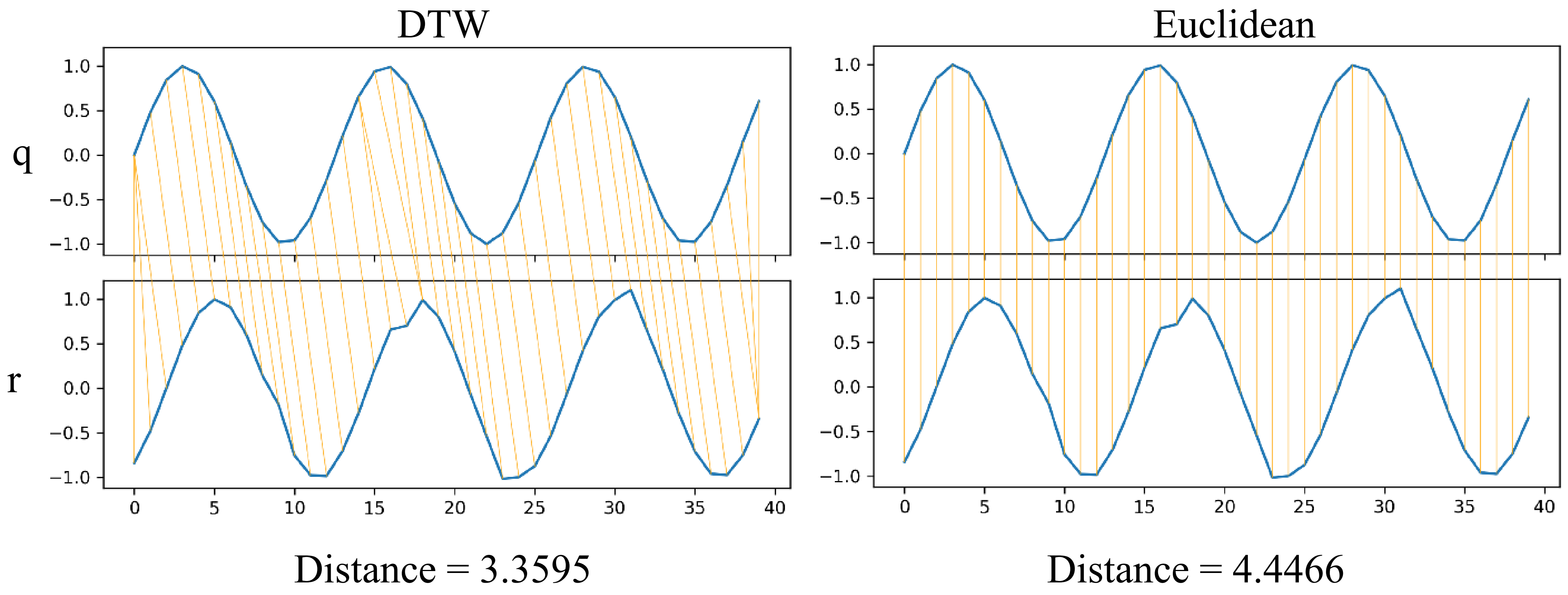
$$D_{i,j} = d(q_i, r_j) + min\begin{cases} D_{i-1,j-1}\ (match) \\ D_{i-1,j}\ (insertion) \\ D_{i,j-1}\ (deletion) \end{cases}$$

3. Warping path = Path traced back from $D_{N,M}$ to $D_{0,0}$ with minimum cost

4. Distance = $D_{N,M}$



Comparison with Euclidean distance



Dimensionality reduction

- Observations – time series - lying on metric space with distance between them – DTW
- Visualise – low dimension representation using:
- 1) classical Multidimensional scaling (MDS)
- 2) Isometric feature mapping (Isomap)
- 3) Locally Linear Embedding (LLE)

Classical MDS

- Input: $x_i \in \mathbb{R}^p \; for \; i = 1,2,3, \ldots, n$
  - Distance between x_i and $x_j = \delta_{ij}$
- Output: $y_i \in \mathbb{R}^m \; for \; i = 1,2,3, \ldots, n$
  - Distance between y_i and $y_j = d_{ij}$
- Goal: d_{ij} matches δ_{ij}

Classical MDS (Euclidean)

- Aim: minimize strain function $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ({\delta_{ij}}^2 - {d_{ij}}^2)$

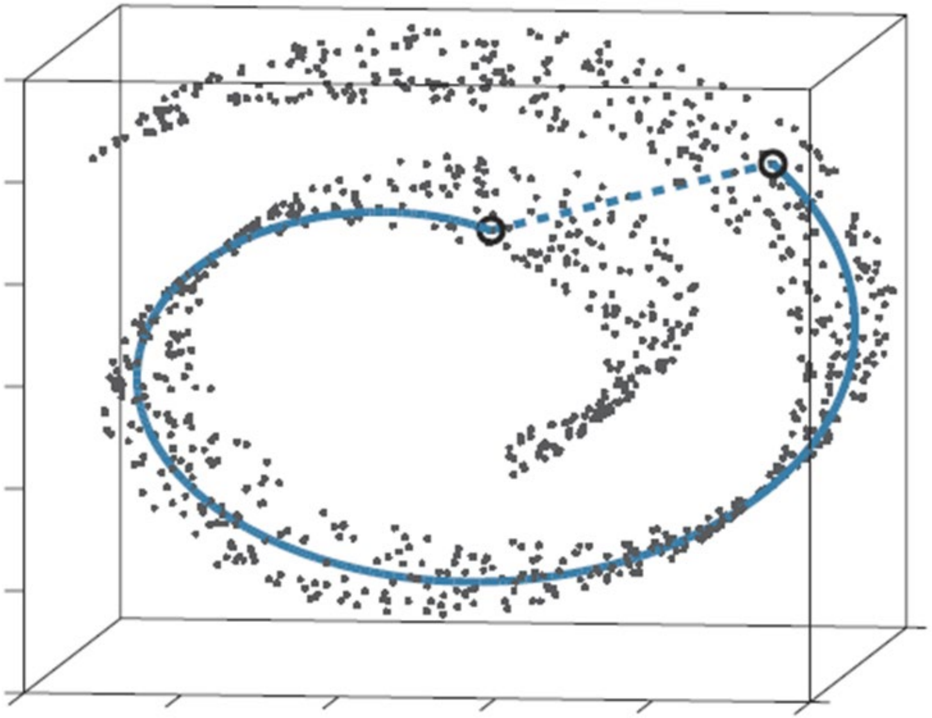
1. Set up an $n \times n$ matrix of squared interpoint distances $\Delta^{(2)} = \{{\delta_{ij}}^2\}$
2. Apply double centering:
$\mathrm{B} = H'\Delta^{(2)}H$
where centring matrix $H = I - \frac{1}{n}J_n$,
I is an $n \times n$ identity matrix, and
J_n is an $n \times n$ matrix of all ones
3. Apply eigenvalue decomposition:
$\mathrm{B} = U\Lambda\mathrm{U}'$
4. Output coordinates are given by:
$Y = U_m{\Lambda_m}^{1/2}$
where U_m is the matrix of m eigenvectors and
Λ_m is the diagonal matrix of m eigenvalues of B

Classical MDS (non-Euclidean)

- Aim: minimize $tr\{(\mathrm{B}-\mathrm{B}^{*})\}$
- Where B^{*} = doubly-centred squared (Euclidean) distance matrix in output space
- Consider only eigenvectors corresponding to non-negative eigenvalues

Isomap

- non-linear extension to classical MDS
- Instead of interpoint distances (dashed line), consider geodesic distances (solid line) between input points



Isomap algorithm

- 1) Construct neighbourhood graph G
  - Calculate Euclidean distances between points, δ_{ij}
  - Connect each point to its K-nearest neighbours
  - Edge weight = δ_{ij}
- 2) Find shortest path on G, between every pair of points, $d_G(i,j)$ and estimate geodesic distance $D_G = \{d_G(i,j)\}$
  - Use algorithms like Dijkstra's method, Floyd- Warshall algorithm
- 3) Construct embedding using MDS
  - Use D_G as input

Pros and cons of Isomap

Advantages

- exploits a dataset's geometry
- global

Disadvantages

- High complexity
- Unclear to choose K

LLE

- Preserve local properties
- Suited - non-convex manifolds

LLE algorithm

- 1) Nearest neighbour search
  - Construct ε or K-neighbourhood graph G to find nearest neighbours (nn) of each input point x_i
- 2) Constrained least-squares fits
  - Reconstruct x_i as linear combination of its nn x_j, minimise

$$\left\| x_i - \sum_{j \in \mathcal{U}_k(i)} w_{ij}\, x_j \right\|^2$$

- 3) Eigenproblem
  - Fix weight w_{ij}, find output point y_i that minimise

$$\left\| y_i - \sum_{j \in \mathcal{U}_k(i)} w_{ij}\, y_j \right\|^2$$

Pros and cons of LLE

Advantage

- Complexity – sparsity of weight matrix

Disadvantage

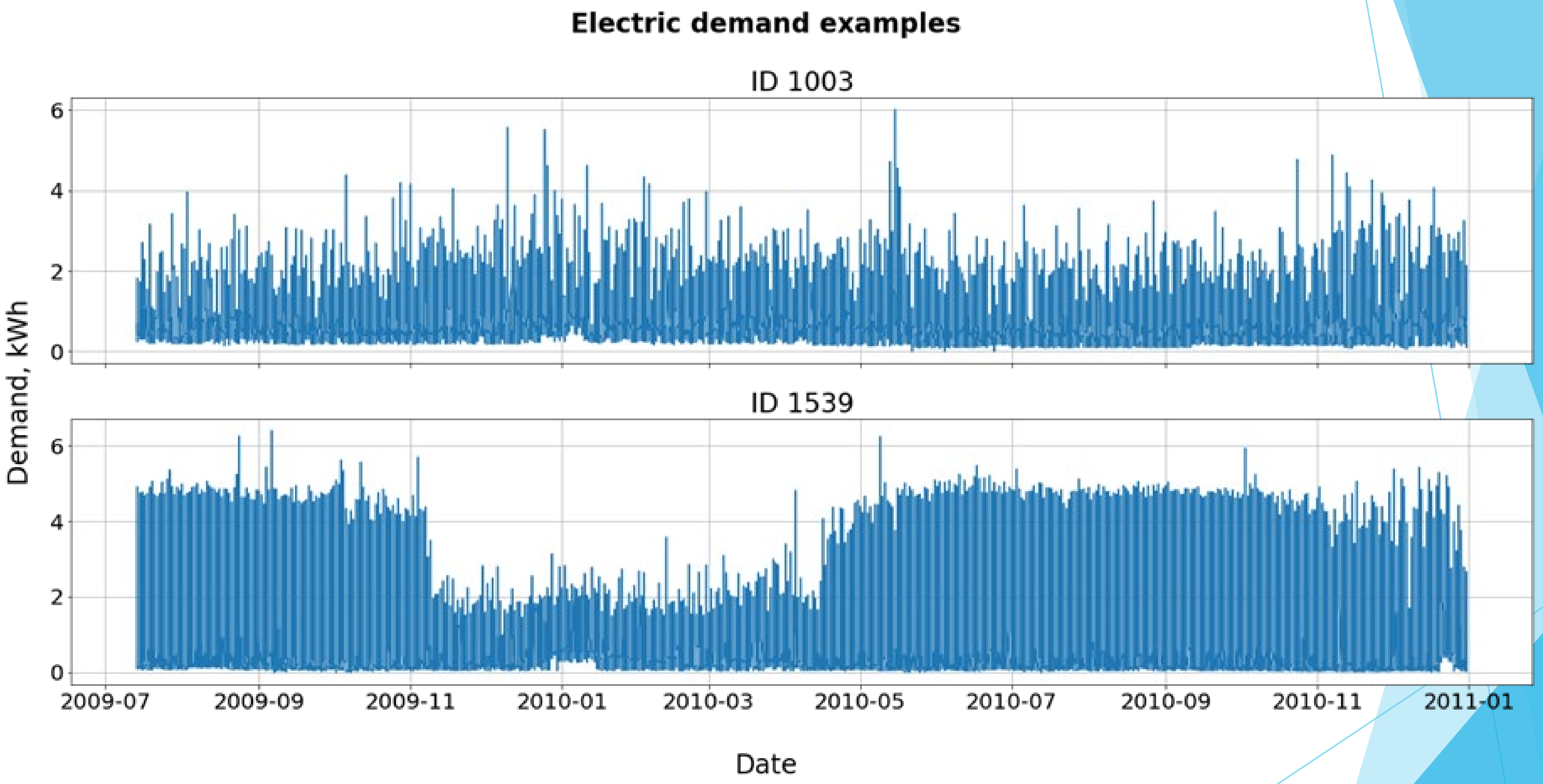
- Local based

Quality measure: Trustworthiness

- Penalise error: observations close in output space, but not close in the input space
- $T(K) = 1 - \frac{2}{G_K} \sum_{i=1}^{n} \sum_{\substack{j \in V_k(i) \\ j \notin \mathcal{U}_k(i)}} (\rho_{ij} - K)$
- $0 < T(K) < 1$

The Dataset

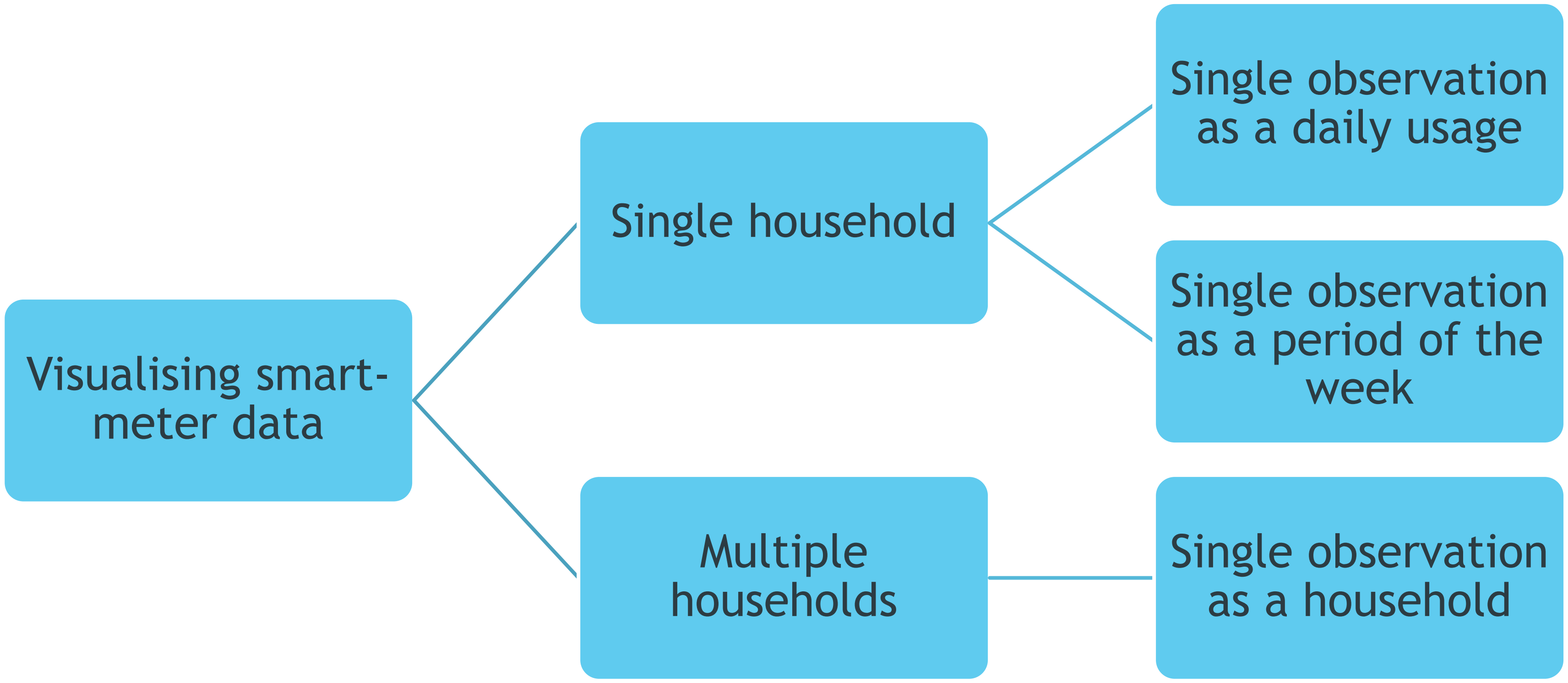
- Irish smart meter dataset
  - over 5,000 homes and businesses
  - behavioral trial conducted by Commission for Energy Regulation (CER)
  - Between 14 July 2009 and 31 December 2010
  - Half-hourly interval (denote as period) electricity usage records: one day has 48 periods

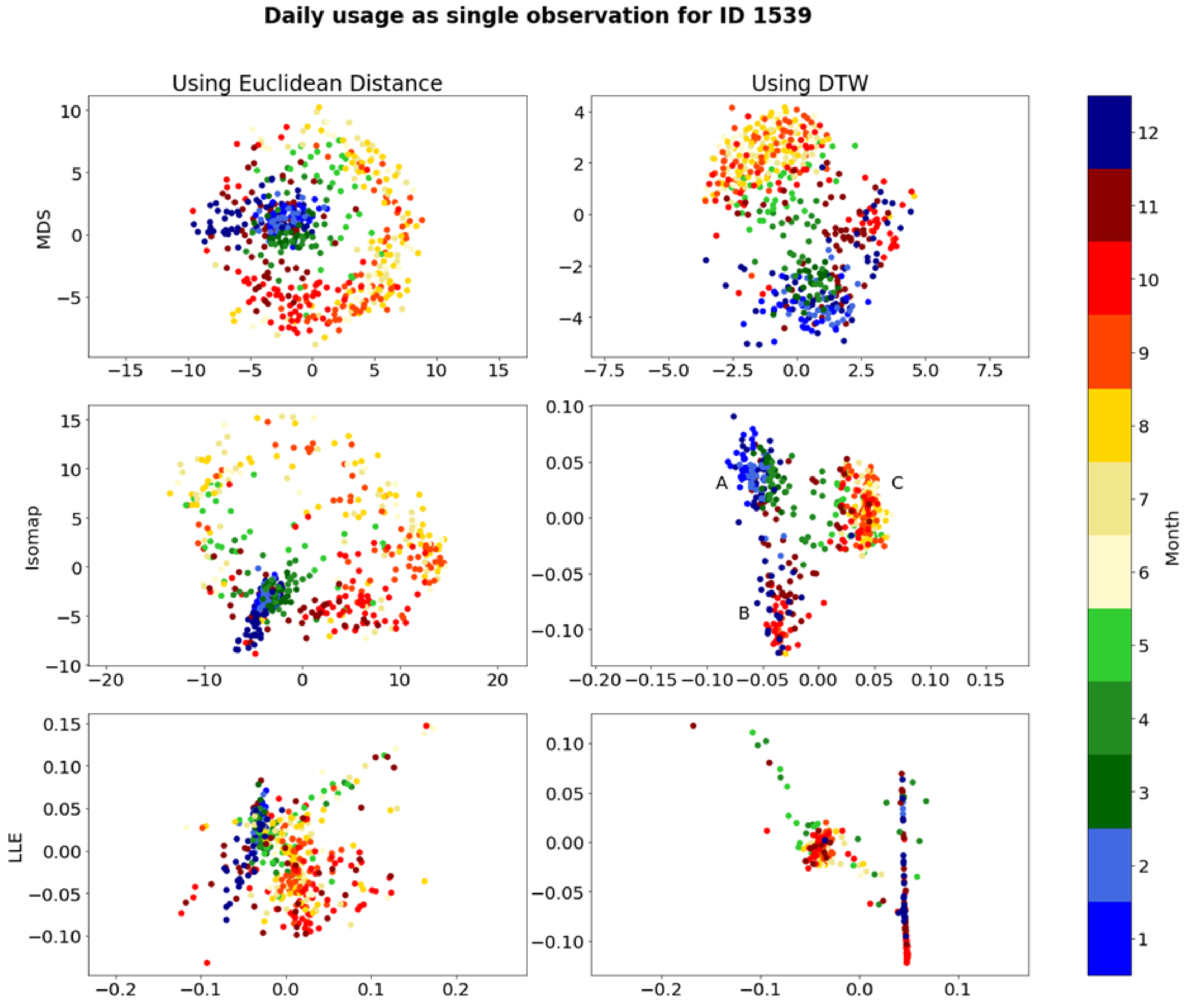

Electric demand examples
ID 1003
ID 1539
Demand, kWh
Date
0
2
4
6
2009-07
2009-09
2009-11
2010-01
2010-03
2010-05
2010-07
2010-09
2010-11
2011-01

Preparation of data, input X

- Matrix n x t
- n = number of observations, could be a day, a period of week, a household etc
- t = time of demand, could be a half-hour period, a day, a week, or a month

Results

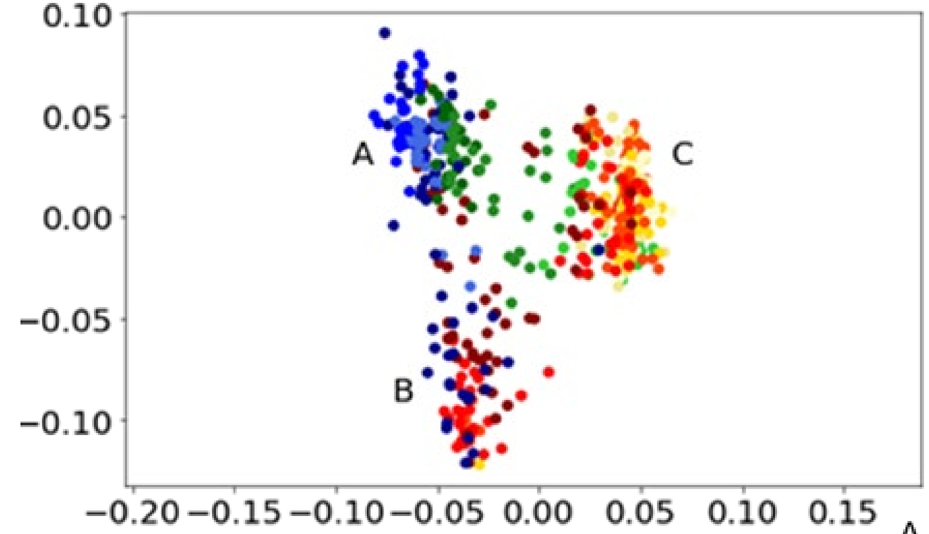

Visualising smart-meter data
Single household
Multiple households
Single observation as a daily usage
Single observation as a period of the week
Single observation as a household



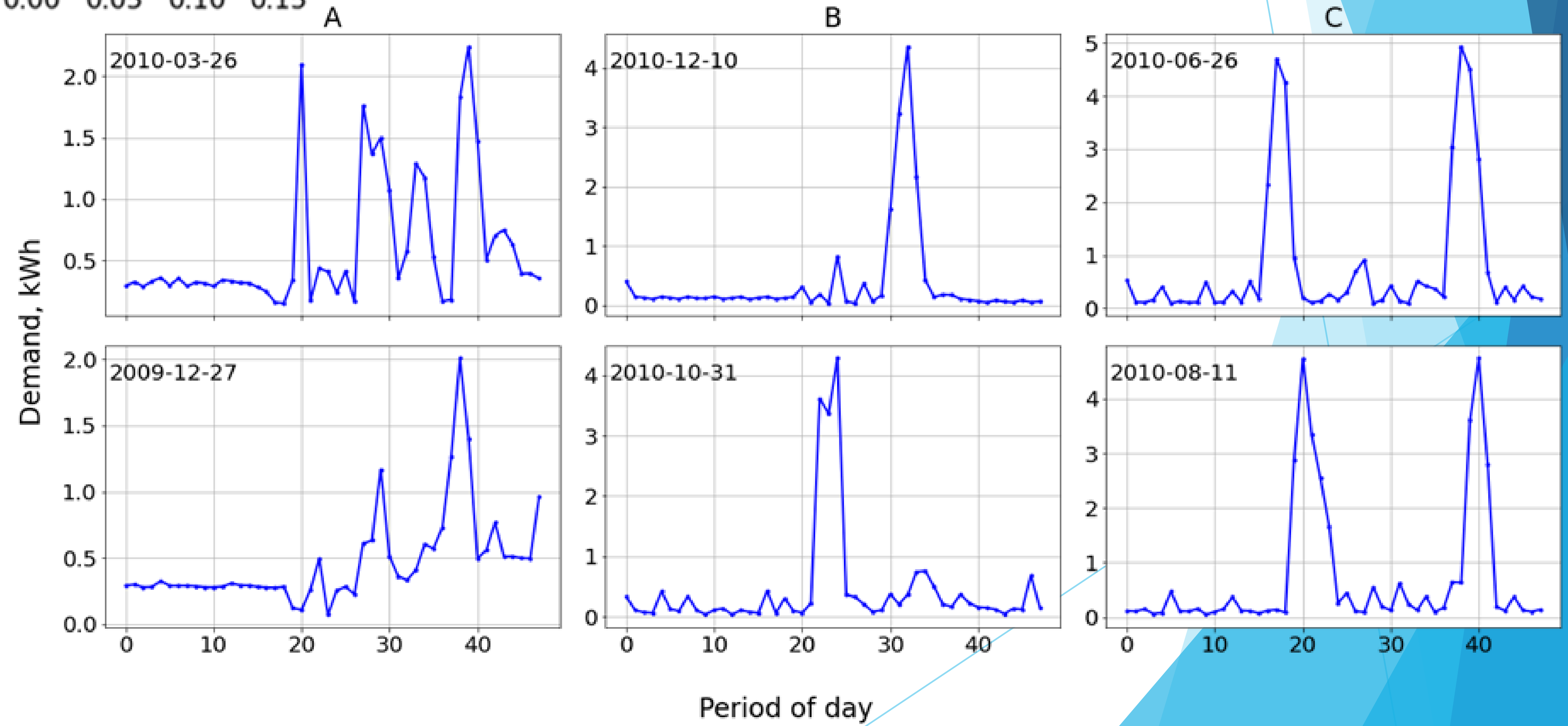
Single household: Single observation as a daily usage

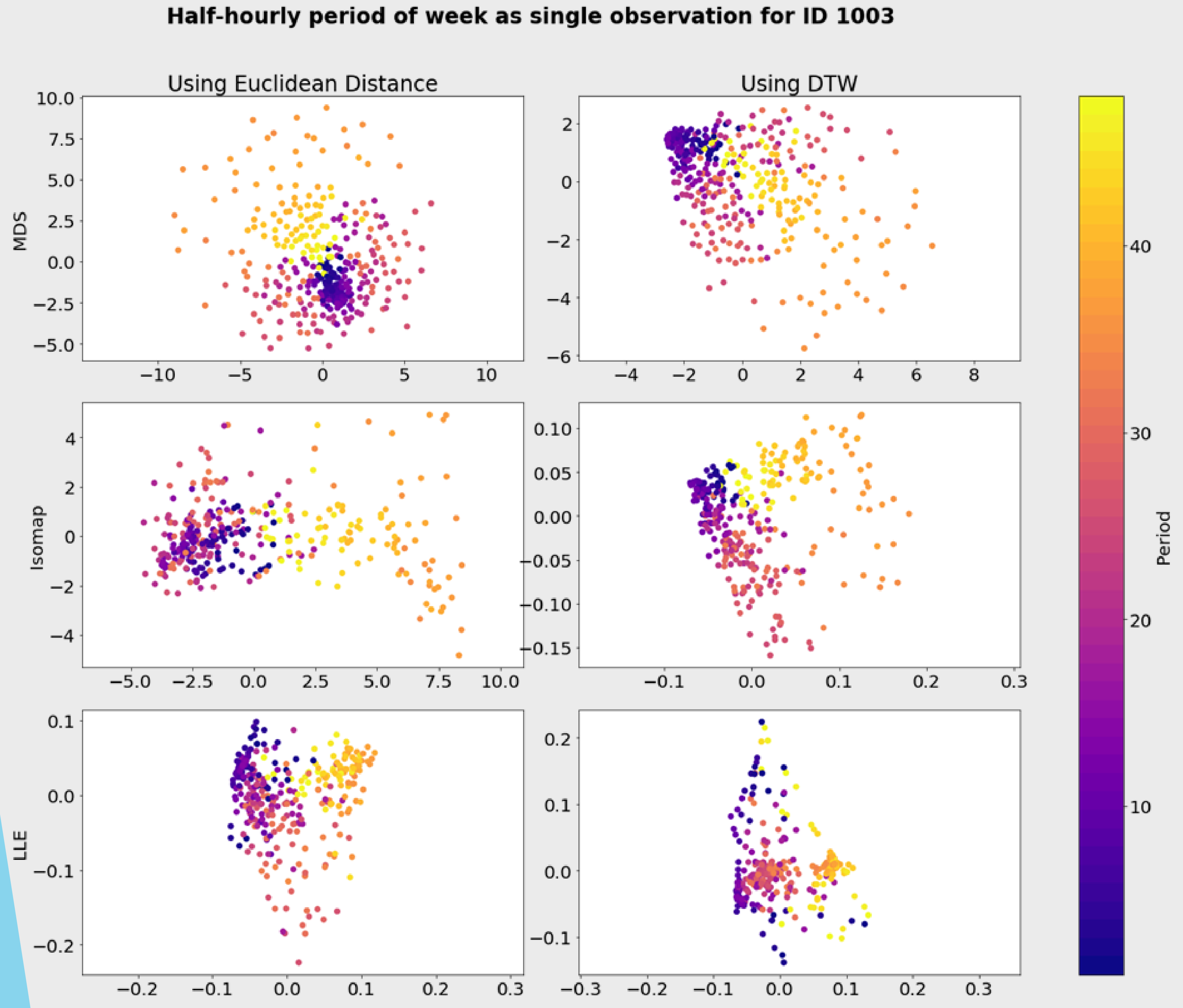
n = number of days
t = demand in a day
Input data: $X \in \mathbb{R}^{534 \times 48}$

Dimensionality reduction	Trustworthiness for DTW
MDS	**0.885**
Isomap	**0.868**
LLE	0.839



Examples of observation from each cluster





Single household: Single observation as a period of week

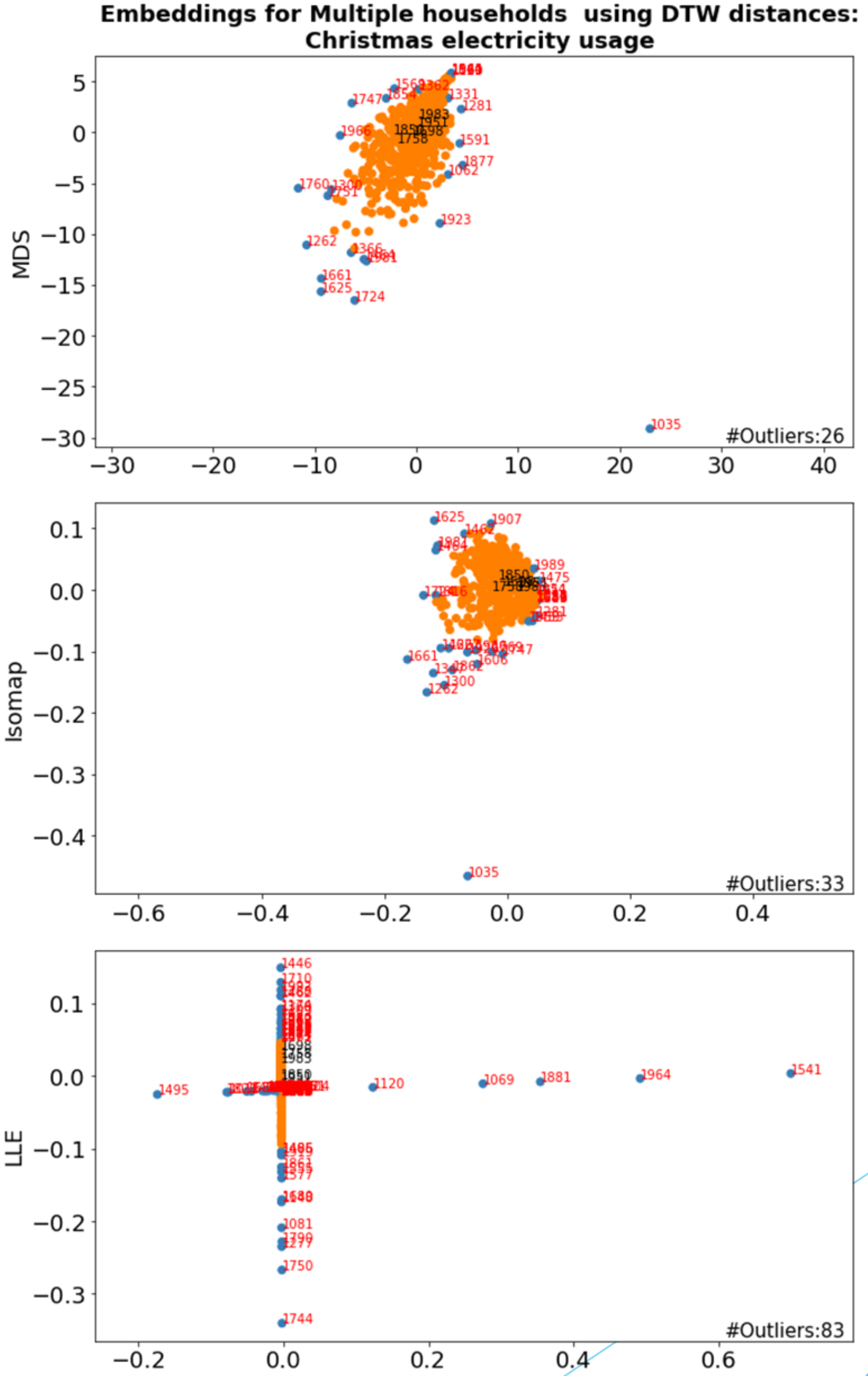
n = one period in a week
t = week of observation
Input data: $X \in \mathbb{R}^{336\times74}$

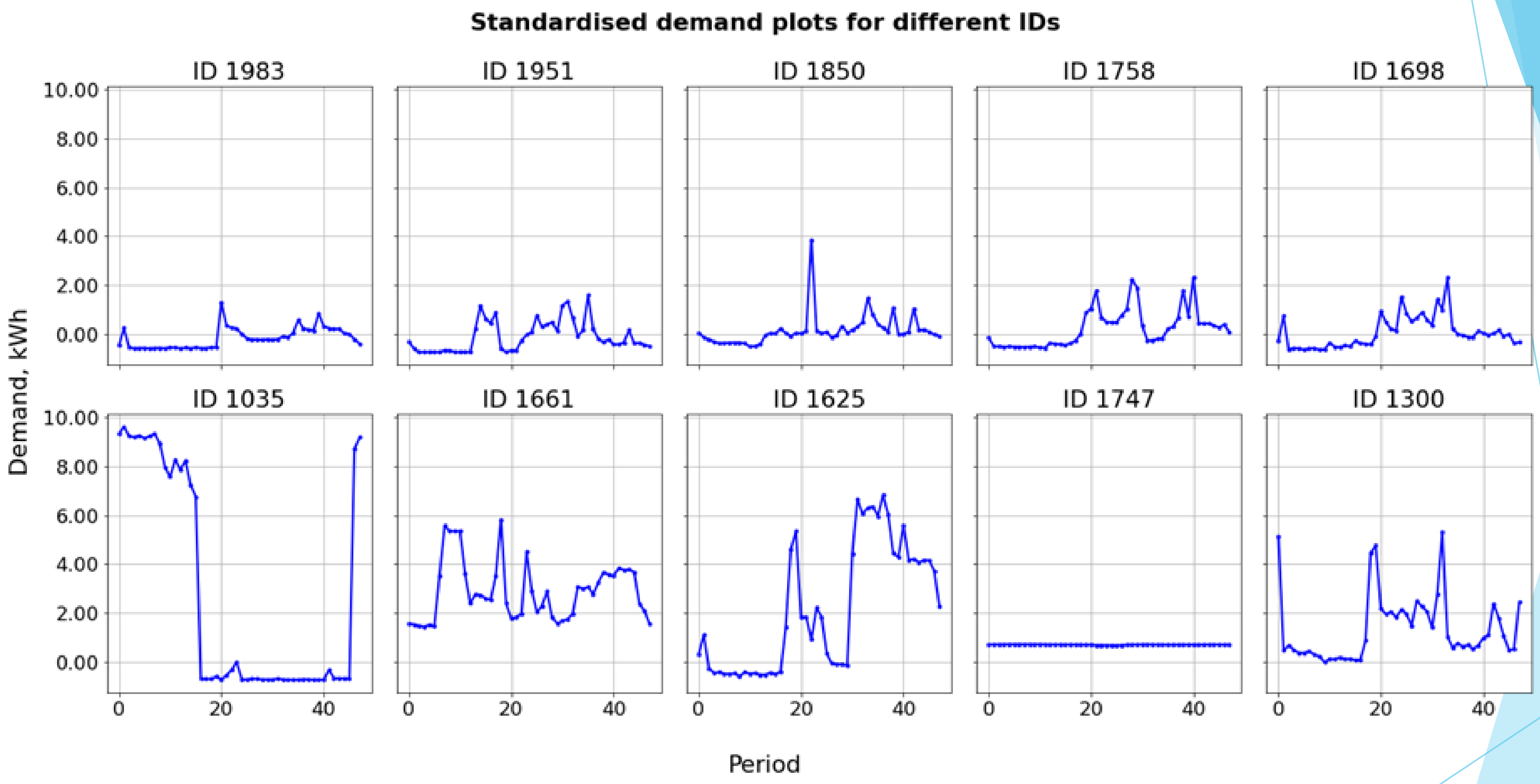
Dimensionality reduction	Trustworthiness for DTW
MDS	**0.883**
Isomap	**0.884**
LLE	0.747

Multiple households: Christmas day electricity consumptions

- n = number of households
- t = demand of Christmas day
- Input data: $X \in \mathbb{R}^{556 \times 48}$
- Local Outlier Factor (LOF): density-based outlier detection method – measures local deviation of the density - with respect to neighbours

Dimensionality reduction	MDS	Isomap	LLE
Trustworthiness	**0.911**	0.892	0.641





Typical households

Anomalous households

Conclusions

- DTW distances represent better similarity – usage pattern
- Topology of manifolds - well-preserved in - MDS and Isomap, but not LLE
- successfully identify both typical and anomalous households
- Future work: apply other manifold learning algorithms; explore application of bounding techniques in DTW to efficiently handle large data (when comparing multiple households)

Thank you