

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Elettrica



Analisi delle curve di carico: Confronti tra metodi di clustering con dati di ingresso basati sulle curve di durata

Relatore: Prof. Gianfranco Chicco

Candidato: Richard Andree Toledo Barraza

Anno accademico 2020/2021

Sommario

1	Clustering	2
1.1	Introduzione al clustering	2
1.1.1	Classificazione degli algoritmi di clustering	3
1.2	Algoritmi di clustering	4
1.2.1	K-means	4
1.2.2	Clustering gerarchico	5
1.2.3	DBSCAN	9
1.2.4	EM-GMM	11
1.2.5	Spectral clustering	11
1.3	Clustering validation	15
2	Pre-clustering	20
2.1	Misura dei dati e condizioni di carico	23
2.2	Pulizia dei dati ed estrazione delle curve rappresentative	25
2.3	Selezione delle caratteristiche	28
3	CONDUCTS	32
3.1	Descrizione della metodologia	32
3.2	Fasi dell'approccio CONDUCTS	32
4	Identificazione delle anomalie su dataset reali	37
4.1	Dataset	37
4.2	Techniche per la identificazione delle anomalie	38
4.2.1	Z-score	38
4.2.2	Median absolute deviation (MAD)	42
4.2.3	DBSCAN	43
4.2.4	Cumulative empiriche	45
4.2.5	Risultati	50
5	Applicazione di CONDUCTS	64
5.1	Calcolo dei nove decili	64
5.2	Scelta del numero K di cluster	66
5.3	K-means	68
5.4	Clustering gerarchico	74

5.4.1	Single linkage	74
5.4.2	Complete linkage	80
5.4.3	Average linkage	86
5.4.4	Centroid linkage	92
5.4.5	Ward linkage	93
5.5	Spectral clustering	105
5.5.1	Spectral clustering (L)	105
5.5.2	Spectral clustering (S)	111
5.6	Indici di validità	117
6	Conclusioni	118

Introduzione

L'impatto che l'energia elettrica ha nelle vite delle persone al giorno d'oggi è considerevole e nel futuro lo sarà ancora di più, per questo motivo sono sempre più necessari strumenti capaci di dirci come il consumo energetico evolve negli anni in modo da adottare le misure necessarie ed anticipare tale evoluzione. Tra questi strumenti troviamo le tecniche di clustering, il cui utilizzo nel mondo elettrico è stato agevolato dalla introduzione di nuove tecnologie, come gli smartmeter, che mettono a disposizione una elevata quantità di dati. Grazie alla capacità di creare classi di utenti in base al loro consumo, le tecniche di clustering hanno trovato un ampio utilizzo in diversi ambiti del settore elettrico, infatti la conoscenza di gruppi di utenti può agevolare l'analisi della domanda di energia, migliorare i risultati delle previsioni di carico, permettere la creazione di tariffe specifiche nel mercato elettrico liberalizzato, facilitare il system planning e le operazioni di rete soprattutto nelle reti di distribuzione dove abbiamo le curve di carico variabili degli utenti. Data la importanza delle tecniche di clustering l'obiettivo di questo elaborato è quello di confrontare gli algoritmi di clustering tradizionali con una metodologia introdotta recentemente, chiamata CONDUCTS (CONsumption DURATION Cuves Times Series). CONDUCTS propone un modo innovativo di estrarre informazioni dalla sola forma dei consumi ignorando la variabile temporale ed è pertanto adatto all'analisi di carichi elettrici in cui il tipo di apparecchiature utilizzate è più importante dei tempi in cui le apparecchiature sono utilizzate.

Più nel dettaglio la tesi presenta un primo capitolo introduttivo al clustering, dove saranno anche descritti alcuni degli algoritmi di clustering che si utilizzeranno per verificare l'efficacia di CONDUCTS introducendo anche gli indicatori di validità utili per valutare i risultati ottenuti. Nel secondo capitolo si vedranno i passaggi a cui sono normalmente sottoposti i dati prima del clustering nel caso in cui le serie temporali siano trattate con le metodologie più classiche, questo per rendere più evidenti le caratteristiche che distinguono CONDUCTS, le quali saranno introdotte nel capitolo 3.

Il quarto capitolo riporta alcuni metodi per la identificazione delle anomalie all'interno delle serie temporali, insieme ai risultati delle applicazioni su due dataset derivanti da progetti inglesi. Questi dati serviranno da input per CONDUCTS, la cui metodologia verrà applicata ad alcuni degli algoritmi di clustering più classici nel capitolo 5.

Infine nell'ultimo capitolo si andranno a discutere la partizione degli utenti ottenuta analizzando anche i valori degli indici di validità.

Capitolo 1

Clustering

1.1 Introduzione al clustering

Il clustering è un processo che ha l'obiettivo di analizzare un determinato dataset e raggruppare gli elementi simili tra di loro in classi chiamate cluster. Questo processo viene denominato anche "unsupervised classification" perché ai cluster non vengono assegnate etichette ma vengono considerati semplicemente come gruppi aventi elementi simili.

La similarità tra gli elementi è determinata utilizzando vari criteri, che possono anche differire significativamente sia in termini di caratteristiche necessarie che deve avere un elemento per appartenere ad un gruppo sia in termini del modello utilizzato per formare il cluster.

Esempi di criteri con i quali gli elementi possono essere raggruppati nei cluster sono:

- Distanza minima tra elementi.
- Aree con elevate densità di dati nello spazio dei dati.
- Elementi aventi una particolare distribuzione statistica.

Allo stesso tempo si deve cercare di avere i cluster abbastanza dissimili tra loro, ragione per cui il processo di clustering può essere visto come un problema di ottimizzazione multi-obiettivo. Il metodo appropriato dipende principalmente dal tipo di dati che si deve analizzare e da come verrà utilizzato il risultato finale, quindi va scelto caso per caso.

Esistono diversi algoritmi di clustering, ma generalmente si basano su due grandi filoni:

1. Metodi aggregativi: in questi metodi il punto di partenza è quello di considerare ogni singolo elemento come un cluster distinto, successivamente i cluster vicini verranno raggruppati fino a che non si raggiunge il numero di cluster voluto o fino a che la distanza tra i cluster non rispetti il limite minimo.
2. Metodi divisivi: nei metodi divisivi inizialmente tutti gli elementi vengono considerati appartenenti ad un unico cluster, che poi viene spezzato più volte al fine di ottenere gruppi sempre più omogenei. La divisione avviene fino a che non si raggiunge il numero di cluster voluto o non si raggiunge un criterio prefissato.

1.1.1 Classificazione degli algoritmi di clustering

Gli algoritmi di clustering esistenti sono molti, e il modo per classificarli dipende dalla caratteristica alla quale si vuole dare più peso, quindi non è unico. Uno dei modi più comuni di catalogare gli algoritmi di clustering è quello di prendere in considerazione il modello adottato per il cluster, in quanto permette più facilmente di intuire l'idea alla base dell'algoritmo, in questa categorizzazione possiamo trovare:

- **Connectivity-based clustering:**

Questa famiglia di algoritmi si basa sul principio che gli elementi più vicini nello spazio dei dati hanno più similarità rispetto agli elementi più lontani. Possono appartenere sia alla categoria aggregativa che divisiva. Il cluster in questo caso è formato da tutti gli elementi che rispettano una certa condizione di distanza, la cui definizione non è unica ed è quella che distingue gli algoritmi di questa famiglia.

Questi algoritmi hanno il vantaggio che il modello dei cluster è facile da interpretare in quanto la partizione dei dati che si ottiene è un albero gerarchico, dal quale poi si andranno a scegliere i cluster più adatti. Gli svantaggi invece riguardano la gestione di grandi dataset, e i valori anomali. Questi ultimi in certi casi causano un raggruppamento forzato oppure un aumento del numero di cluster risultante.

- **Centroid-based clustering:**

Gli algoritmi di questa famiglia sono del tipo iterativo e i cluster sono rappresentati da un vettore centrale, che può non appartenere all'insieme di dati da analizzare. Gli elementi vengono assegnati ai cluster cercando di minimizzare la distanza quadratica tra il centro e gli elementi stessi.

Questi algoritmi richiedono una previa conoscenza dei dati analizzati in quanto il numero di cluster deve essere definito all'inizio.

Il vantaggio principale di questa categoria risiede nella semplicità della implementazione, ma presentano lo svantaggio che la suddivisione dei cluster può non essere quella ottimale, cioè che non si raggiunga l'ottimo globale.

- **Distribution-based clustering:**

In questa tipologia i cluster vengono modellati come distribuzioni di probabilità, la assegnazione dei punti avviene quando questi possono essere considerati come appartenenti a tale distribuzione.

Molto spesso si usa la stessa distribuzione per tutti i cluster (ad esempio distribuzioni gaussiane). Questi algoritmi presentano il vantaggio di avere una forma dei cluster meno restrittiva rispetto ad altre tipologie e quindi possono rappresentare meglio i dati, ma hanno il problema che ai cluster possono essere assegnati elementi che non hanno alcun significato (overfitting), inoltre la scelta della distribuzione non è banale.

- **Density-based clustering:**

I cluster vengono definiti come aree ad alta densità di elementi, ovvero si vanno a cercare nello spazio dei dati regioni in cui la densità di elementi è particolarmente elevata, il limite di densità dipende dall'impostazione di certi parametri, queste regioni andranno poi isolate e considerate come cluster. Gli algoritmi di questa categoria non richiedono di una previa conoscenza dei dati e quindi di specificare il numero di cluster. Tipicamente questi algoritmi sono in grado di identificare i valori anomali, che sono considerati come rumore e vengono lasciati fuori dei cluster. Lo svantaggio di questa categoria sta nella gestione di dataset appartenenti a spazi ad elevata dimensione e dati a densità variabile.

- **Graph-based clustering:**

L'idea di base di questi algoritmi risiede nell'utilizzo dell'algebra lineare per la costruzione di grafi in grado di rappresentare gli elementi del dataset in uno spazio dimensionale minore. I collegamenti presenti nel grafo rappresentano la similarità tra i vari elementi e sono quelli che permettono di identificare i cluster. Tipicamente questa famiglia di algoritmi viene utilizzata quando si hanno dati appartenenti a uno spazio di dimensioni elevate, in quanto l'algebra lineare rende i calcoli più agevoli.

1.2 Algoritmi di clustering

Di seguito verranno introdotti gli algoritmi di clustering più classici all'interno delle categorie citate precedentemente evidenziando i passaggi che compongono i loro algoritmi e le principali caratteristiche che gli distinguono.

1.2.1 K-means

Il K-means è un algoritmo di clustering iterativo appartenente alla famiglia dei centroid-based model, che ha come obiettivo la ottimizzazione di una funzione obiettivo, cioè minimizzare la distanza tra i punti e il centro del cluster. L'algoritmo del K-means può essere riassunto come :

1. Si decide il numero "K" di cluster in cui verrà suddiviso il data set.
2. Si scelgono i "K" centroidi (diversi) in modo casuale nello spazio dei dati, e ci si assicura che siano abbastanza lontani tra di loro, questa condizione è utile per la convergenza.
3. Viene calcolata la distanza di ogni oggetto del dataset rispetto ai centroidi. La distanza più utilizzata è quella euclidea, ma si possono utilizzare altre definizioni di distanza (Appendice).
4. Gli oggetti dei cluster vengono assegnati in base alla vicinanza ai centroidi.

5. Si calcolano i nuovi centroidi considerando gli elementi assegnati ai cluster, come la media delle posizioni di tutti gli oggetti appartenenti al cluster.
6. Si itera a partire dal punto 3 fino a quando non ci sono più variazioni.

Il risultato raggiunto con il K-means dipende dalle condizioni iniziali, ovvero dai centroidi scelti all'inizio della iterazione. Dal punto di vista della qualità della soluzione, il raggiungimento dell'ottimo globale non viene garantito, come è possibile vedere nelle Figure 1.1 e 1.2, la scelta dei centroidi iniziali può portare a una partizione dei dati differente.

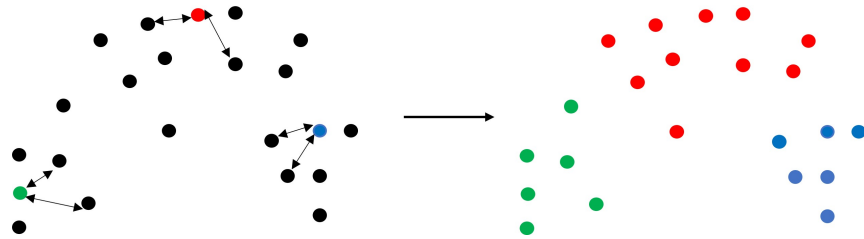


Figura 1.1: Esempio di risultato ottenuto con il K-means nel caso di $K=3$.

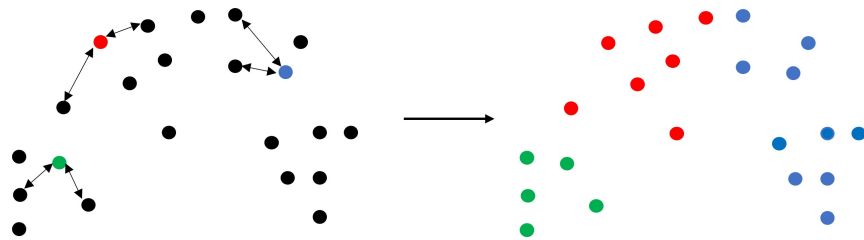


Figura 1.2: Altro esempio di risultato ottenuto con il K-means nel caso di $K=3$.

Una buona pratica è quella di eseguire l'algoritmo molte volte e comparare le variazioni tra i risultati. Il clustering si ritiene corretto se la distanza tra i cluster è grande e se la varianza dentro il cluster è piccola.

1.2.2 Clustering gerarchico

Il clustering gerarchico appartiene alla tipologia dei metodi connectivity-based e può essere di 2 tipi:

1. Agglomerativo: ogni elemento appartiene ad un cluster distinto nel primo livello, i quali verranno fusi a coppie in base alla vicinanza nei livelli successivi fino a quando non si ottiene un'unica configurazione gerarchica che racchiude tutti i cluster precedenti.

2. divisivo: si inizia considerando un unico cluster che racchiude tutti gli elementi del dataset, al livello successivo invece questo viene diviso in due cluster, così ogni cluster del livello corrente viene suddiviso in due in base ad una metrica di distanza.

Tipicamente la implementazione più utilizzata è quella di tipo agglomerativo, quindi d'ora in poi si considera solo questa tipologia.

La struttura nidificata che si ottiene con questo algoritmo, chiamata anche albero binario (perché i raggruppamenti avvengono a coppie), può essere rappresentata attraverso un dendrogramma, Figura 1.3, dove possiamo osservare come la unione di due cluster avvenga su livelli diversi.

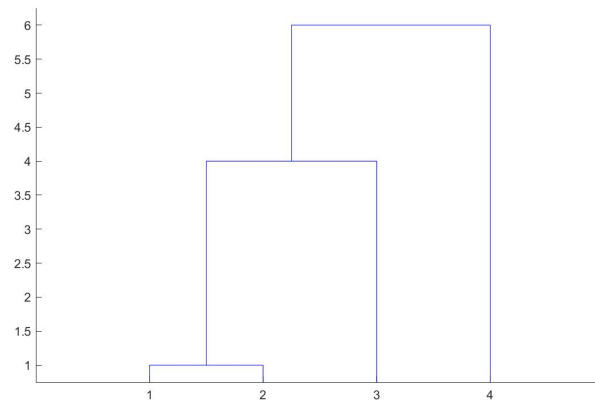


Figura 1.3: Esempio di Dendrogramma con 4 elementi.

L'algoritmo può essere riassunto nei seguenti passaggi:

1. Ogni elemento viene inizialmente trattato come un cluster unico.
2. Si sceglie la definizione di distanza e si calcola la vicinanza tra ciascun paio di cluster.
3. I punti più vicini tra loro vengono raggruppati a coppie e formano un nuovo cluster.
4. Si ricalcola la distanza tra i nuovi cluster.
5. Si itera a partire dal terzo passaggio fino a quando tutti i cluster non sono raggruppati in uno solo.

Il modo di calcolare la distanza tra cluster (linkage), può essere diverso, infatti oltre alla definizione di distanza si deve specificare anche il metodo di calcolo, ovvero gli elementi da tenere in considerazione durante il calcolo, di seguito verranno specificate le varie tipologie di linkage accoppiate al clustering gerarchico:

- Single-linkage:

Il single linkage criterion (chiamato anche criterio della minima distanza) aggrega due cluster considerando la distanza minima tra gli elementi che compongono i

cluster. Definiti due insiemi di dati $\mathbf{R}$ ed $\mathbf{S}$ si considera come distanza tra cluster il minimo tra gli elementi r_i ed s_i che compongono i due insiemi.

$$d(\mathbf{R}, \mathbf{S}) = \min\{d(r, s) : r_i \in \mathbf{R}, s_i \in \mathbf{S}\} \quad (1.1)$$

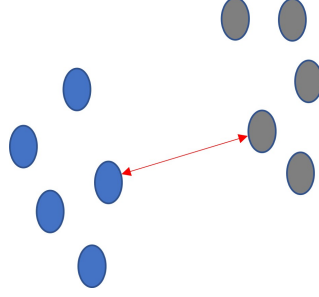


Figura 1.4: Misura della distanza nel caso di single linkage.

- Complete-linkage:

In questo caso l'aggregazione dei cluster che formano l'albero binario avviene considerando i componenti più lontani in entrambi i cluster. Considerando gli insiemi di dati $\mathbf{R}$ ed $\mathbf{S}$ la distanza di interesse è tra r_i ed s_i tale per cui $d(\mathbf{R}, \mathbf{S})$ è massima

$$d(\mathbf{R}, \mathbf{S}) = \max\{d(r, s) : r_i \in \mathbf{R}, s_i \in \mathbf{S}\} \quad (1.2)$$

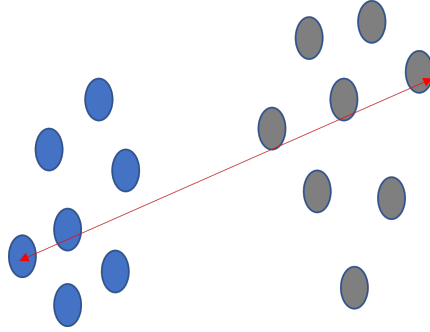


Figura 1.5: Misura della distanza nel caso di complete linkage.

- Average-linkage:

A differenza dei primi due criteri, in questo caso vengono presi in considerazione tutti i punti dei potenziali cluster da unificare, infatti per ciascuna coppia di punti viene calcolata la distanza, che sarà poi utilizzata per calcolare la distanza media tra i cluster. Dati i due insieme di dati $\mathbf{R}$ ed $\mathbf{S}$ aventi corrispondentemente n_r ed

n_s elementi, la distanza tra cluster è data dalla media delle distanze dei punti r_i e s_i .

$$d(\mathbf{R}, \mathbf{S}) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} d(r_i, s_j) \quad (1.3)$$

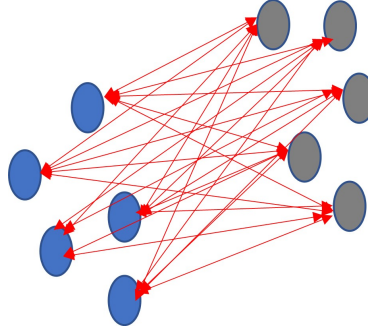


Figura 1.6: Misura della distanza nel caso di average linkage.

- Centroid-linkage:

Dai cluster alla base del albero binario si risale ai centroidi rappresentativi, in base alla distanza tra i centroidi si inizierà a fondere i cluster fino ad averne uno solo. Considerando due insiemi di dati $\mathbf{R}$ ed $\mathbf{S}$ si definiscono i centroidi $\bar{r}$ e $\bar{s}$ che li caratterizzano come la media dei loro elementi:

$$\bar{r} = \frac{1}{n_r} \sum_{i=1}^{n_r} r_i \quad (1.4)$$

$$\bar{s} = \frac{1}{n_s} \sum_{i=1}^{n_s} s_i \quad (1.5)$$

noti i centroidi possiamo calcolare la distanza tra gli insiemi come la distanza euclidea tra i due centroidi.

$$d(\mathbf{R}, \mathbf{S}) = \|\bar{r} - \bar{s}\|^2 \quad (1.6)$$

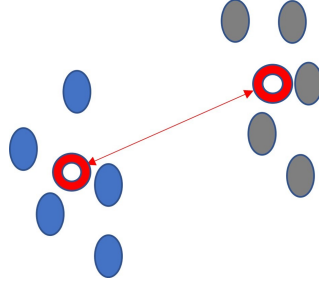


Figura 1.7: Misura della distanza nel caso centroid linkage.

- Ward-linkage:

La formazione del albero binario avviene raggruppando i cluster iniziali in base al potenziale incremento che si avrebbe, nella somma degli errori al quadrato, dalla unione di due cluster. I cluster la cui somma presenta il valore più basso sono raggruppati. Noti due insiemi di dataset $\mathbf{R}$ ed $\mathbf{S}$ aventi rispettivamente n_r ed n_s elementi e centroidi $\bar{\mathbf{r}}$ ed $\bar{\mathbf{s}}$, la distanza viene definita come:

$$d(\mathbf{R}, \mathbf{S}) = \sqrt{\frac{2n_r n_s}{n_r + n_s}} \|\bar{\mathbf{r}} - \bar{\mathbf{s}}\|^2 \quad (1.7)$$

- Weighted linkage:

Questo criterio utilizza una definizione ricorsiva per il calcolo della distanza tra 2 cluster, ovvero la distanza viene calcolata considerando i 2 componenti subito sotto il cluster di interesse. Se consideriamo il cluster $\mathbf{R}$, generato dalla unione dei cluster $\mathbf{F}$ e $\mathbf{G}$, la distanza $d(\mathbf{R}, \mathbf{S})$ viene definita come la media delle distanze $d(\mathbf{F}, \mathbf{S})$ e $d(\mathbf{G}, \mathbf{S})$

$$d(\mathbf{R}, \mathbf{S}) = \frac{d(\mathbf{F}, \mathbf{S}) + d(\mathbf{G}, \mathbf{S})}{2} \quad (1.8)$$

Uno dei vantaggi del clustering gerarchico è che non richiede di definire il numero di cluster a priori, ma "tagliando" il dendrogramma al livello giusto si può ottenere il numero di cluster desiderato. Dall'altro canto però c'è molta difficoltà a gestire dataset enormi e sensibilità al rumore e ai valori anomali, la sensibilità a questi ultimi però può essere attenuata utilizzando un linkage di tipo single.

1.2.3 DBSCAN

Tra gli algoritmi più conosciuti della famiglia dei density-based model troviamo il DBSCAN. L'idea sulla quale si basa il DBSCAN è quella di andare a cercare, per ciascun punto, un'area circostante (neighbourhood) che deve contenere un numero minimo di punti (spesso indicati come MinPts), dato uno specifico raggio (spesso indicato come Eps). I

punti con densità superiore ad un certo numero di MinPts vengono chiamati punti *core*. Se la densità, attorno ad un punto, non supera il numero MinPts richiesto e se un punto *core* si trova a una distanza minore di Eps, allora questi punti vengono classificati come punti limite. I punti che non entrano in queste classificazioni vengono considerati come rumore.

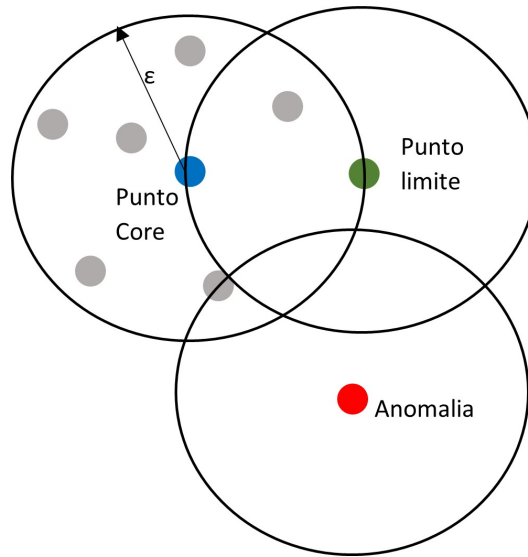


Figura 1.8: Esempio di classificazione dei punti core e limite nel caso di $\epsilon = 5$.

L'algoritmo segue la seguente procedura:

1. Dato il dataset si impostano il numero minimo di punti vicini e il raggio di ricerca.
2. Si prende in considerazione un punto per volta, se all'interno del raggio ci sono abbastanza punti allora il punto viene classificato come punto *core* e diventa il primo punto del cluster, altrimenti il punto viene classificato come rumore. i punti classificati come rumore in questo passaggio possono essere eliminati oppure essere riclassificati e inseriti in un cluster ad un'altra iterazione.
3. Tutti i punti nel neighborhood del punto *core* vengono anche essi assegnati allo stesso cluster.
4. Si verifica che i punti del neighbourhood siano punti *core* a loro volta, se lo sono allora si aggiungono anche i loro punti neighbors al cluster.
5. Il punto precedente si ripete fino a quando si arriva ai punti limiti che sono gli ultimi ad essere aggiunti al cluster.
6. Si vanno a visitare altri punti non ancora classificati e si ripete a partire dal punto 2.

Questo algoritmo è vantaggioso per dataset avente una particolare rumorosità, in più è in grado di generare cluster con forme e dimensioni differenti senza indicarne il numero. Non è consigliabile usare dataset appartenenti a uno spazio con dimensionalità elevata, in quanto in questo caso è difficile definire il concetto di densità a causa della elevata sparsità. Non è neanche consigliabile per dataset con densità variabili.

1.2.4 EM-GMM

L'algoritmo EM-GMM appartiene alla famiglia dei distribution-based clustering. L'EM-GMM assume che i dati sono distribuiti secondo una gaussiana, in questo modo si hanno due parametri per descrivere i cluster, il valore medio e la deviazione standard. I valori ottimali di valore medio e deviazione standard per ciascun cluster, che possono essere diversi, si trovano attraverso un algoritmo di ottimizzazione chiamato expectation-maximization (EM).

La procedura del clustering è riassunta come segue:

1. Si indica il numero di cluster che si vuole e poi si inizializzano i parametri delle gaussiane (in modo casuale o cercando di stimare i valori ottimali) per ciascun cluster.
2. Si calcola la probabilità che ciascun oggetto appartenga a ciascun cluster.
3. Con le probabilità ottenute, si calcolano dei nuovi parametri per le distribuzioni gaussiane in modo da massimizzare le probabilità che i dati appartengano a uno dei cluster. Questi nuovi parametri vengono calcolati facendo una somma pesata delle posizioni dei dati, usando come peso la probabilità che un dato appartenga ad un determinato cluster.
4. Vengono ripetuti i punti 2 e 3 fino a quando si hanno cambiamenti notevoli da una iterazione ad un'altra.

Uno dei vantaggi di questo algoritmo consiste nell'avere una buona flessibilità nella forma del cluster, inoltre per ciascun punto viene calcolata la probabilità di appartenenza ai vari cluster, cosa che può essere utile nel caso in cui un punto ha elevate probabilità di appartenenza a più cluster. Uno degli svantaggi che riguardano questo metodo risiede nella complessità di esecuzione e nell'elevato livello computazionale che richiede, soprattutto per spazi di dati a elevate dimensioni.

1.2.5 Spectral clustering

Lo spectral clustering è un algoritmo appartenente al graph-based clustering, capace di ridurre la complessità di dataset con dimensione elevata utilizzando informazioni provenienti dagli autovalori e autovettori di matrici particolari costruite a partire dal grafo di similarità o direttamente dal dataset, in modo da riportare i dati in uno spazio dimensionale minore dove verrà poi eseguito il clustering.

La costruzione del grafo di similarità non è unica, tutti quanti però devono rispettare la relazione locale tra i vari dati. Tra le tipologie di grafi abbiamo:

- ϵ -neighbourhood graph:

In questo caso vengono connessi tutti i punti la cui distanza non supera una certa soglia ϵ . Per via della impostazione di questo limite non c'è bisogno di attribuire un peso ai collegamenti in quanto l'ordine di grandezza sarà a grandi linee lo stesso.

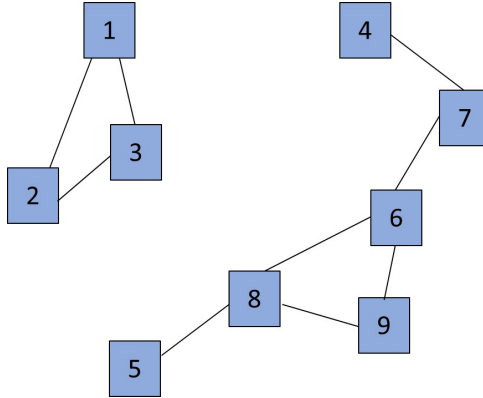


Figura 1.9: Esempio di ϵ -neighbourhood graph.

- k-nearest graph:

Per questa tipologia di grafo la connessione tra due nodi avviene se almeno uno di essi si trova tra i k nodi più vicini del altro nodo, si ignora quindi la potenziale direzionalità dei collegamenti in quanto ai fini raggruppamento non è rilevante nel nostro caso. Successivamente vengono attribuiti i pesi ai collegamenti in base alla similarità tra i nodi.

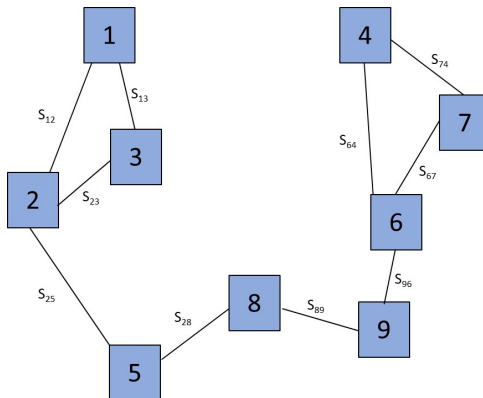


Figura 1.10: Esempio di k-nearest graph.

- fully connected graph:

Il grafo viene costruito collegando tutti i punti e successivamente attribuendo dei pesi ai collegamenti, questo grafo richiede l'utilizzo di una funzione di similarità in grado di rappresentare la relazione locale dei dati.

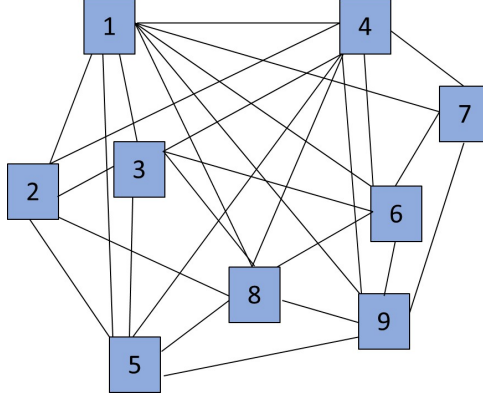


Figura 1.11: Esempio di fully connected graph.

Indifferentemente dalla tipologia di grafo questo può essere rappresentato da una matrice ($\mathbf{A}$) di adiacenza, chiamata anche matrice di similarità, dove le righe e le colonne rappresentano i nodi, le connessioni invece sono rappresentate all'interno della matrice dai rispettivi pesi (oppure da un semplice 1 se ε neighbourhood graph), se due nodi non sono connessi allora il valore all'interno della matrice sarà nullo.

Dal grafo è possibile derivare anche la Degree matrix ($\mathbf{D}$), che è una matrice diagonale, che fornisce informazioni sul numero di collegamenti che ha un determinato nodo.

Attraverso l'utilizzo di queste matrici è possibile ricavare la matrice Laplaciana, tuttavia la sua definizione non è unica, di seguito si riportano le varie tipologie di matrici Laplaciane:

- Matrice Laplaciana non normalizzata ($\mathbf{L}$):

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (1.9)$$

- Matrice Laplaciana normalizzata simmetrica.

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \quad (1.10)$$

- Matrice Laplaciana normalizzata non simmetrica.

$$\mathbf{L}_{rw} = \mathbf{D}^{-1} \mathbf{L} \quad (1.11)$$

Gli autovalori nulli della matrice Laplaciana invece indicano quanti gruppi di elementi connessi abbiamo nel grafo, si può usare questa informazione nel caso non si sapesse

a priori il numero di cluster in cui dividere i dati. Gli autovalori non nulli, ma con valori vicini a zero ci dicono se ci sono dei collegamenti molto "deboli" che eventualmente possono portare alla creazione di ulteriori cluster. Di solito è buona norma cercare il primo dislivello tra autovalori per avere una idea del numero di cluster da usare. Gli autovettori associati ai "k" più piccoli autovalori andranno a definire lo spazio dimensionale nel quale i dati reali verranno mappati per poi eseguire il clustering

La scelta del tipo di grafo e della matrice laplaciana con cui rappresentarlo non segue delle regole rigide, per questo motivo si devono vedere caso per caso quella che meglio si adatta al dataset, la descrizione dei passaggi che formano l'algoritmo in questa trattazione è stata fatta considerando il caso semplice di grafo non direzionale con una matrice laplaciana non normalizzata.

Vediamo ora l'algoritmo di clustering:

1. Per ciascun punto viene calcolata la distanza da tutti gli altri punti.
2. Dalle distanze viene costruito il grafo e le rispettive matrici di adiacenza e Degree matrix.
3. Si calcola la matrice Laplaciana.
4. Si calcolano i k autovettori, associati ai k autovalori più piccoli, della matrice laplaciana, che vengono inseriti in una matrice per colonne. Gli elementi di questa matrice andranno a rappresentare i dati reali nello spazio ridotto dei dati.
5. Si estraggono i cluster dalla matrice degli autovettori appoggiandosi ad un altro algoritmo, ad esempio il k-means.
6. I dati reali vengono assegnati in base al cluster in cui è stato messo l'elemento della matrice degli autovettori che li rappresenta.

I vantaggi dello spectral clustering risiedono nel fatto che non vengono fatte assunzioni riguardo alla forma del cluster, può essere implementato facilmente, ha tempi di calcolo relativamente veloci in quanto la maggior parte dei calcoli è solo algebra lineare, ed è capace di gestire dati con dimensioni elevate. Tuttavia la scelta del grafo non è semplice e il risultato può essere particolarmente sensibile alla scelta dei parametri.

La seguente tabella riassume le principali caratteristiche degli algoritmi di clustering.

Tabella 1.1: Riassunto algoritmi e rispettive caratteristiche.

Algoritmo	Dati input	Richiesta n° di clusters	Forma del cluster	Gestisce valori anomali	Spazio dimensionale elevato
K-means	distanze tra oggetti, centroidi	si	sferoidali	no	si
Gerarchico	distanza tra oggetti	no	arbitraria	si con il linkage adatto	si
DBSCAN	distanze e dati campionati	no	arbitraria	si	no
EM-GMM	dati campionati	si	Dipende dalla distribuzione di probabilità	si	si, ma molto complesso
Spectral	dati campionati o matrice di similarità	si, ma l'algoritmo può fornirne una indicazione	arbitraria	si	si

1.3 Clustering validation

Lo scopo degli algoritmi di clustering è quello di andare a trovare dei gruppi significativi in cui andare ad allocare i dati, non è detto però che i cluster in cui i dati sono divisi dal nostro algoritmo siano quelli ottimali, o che il numero di cluster impostato da noi, nel caso l'algoritmo lo preveda, sia corretto. Quindi è necessario andare a validare il risultato ottenuto qualitativamente.

I criteri più comuni per la valutazione del algoritmo di clustering sono:

- **Compattezza:** indica quanto vicini sono gli oggetti all'interno dello stesso cluster, più compatti sono gli elementi del cluster è più il risultato può essere considerato affidabile.

- Separazione: ci dice quanto sono distanti i cluster tra di loro. In questo caso una maggiore separazione tra i cluster può indicare una suddivisione dei dati migliore.

In termini generali, gli approcci con i quali si possono validare i cluster ottenuti sono di 3 tipi:

1. **Relative clustering validation:** è una famiglia di indici che misura la costanza di un algoritmo, comparando i risultati ottenuti variando ogni volta i valori dei parametri usati nel clustering, oppure confrontando risultati di algoritmi diversi.

Tra gli indici del *relative clustering validation* troviamo l'**indice di stabilità**, che misura l'abilità di un dataset già sottoposto al clustering, di predire il risultato del clustering di un altro dataset proveniente dalla stessa sorgente di quello precedente.

Per il calcolo di questo indice bisogna dividere un determinato dataset in due parti (in modo causale). L'algoritmo da verificare viene applicato a una di quelle parti e il risultato viene utilizzato per addestrare un algoritmo classificatore. Successivamente la tecnica di clustering e il classificatore vengono applicati all'altra parte dei dati generando due risultati diversi, la discrepanza tra i due risultati, mediata su più divisioni del dataset, definisce la stabilità dell'algoritmo.

2. **External clustering validation:** consiste nel confrontare l'esito del clustering con una struttura stabilita a priori da noi e che riteniamo corretta. Può essere considerata come una sorta di calcolo dell'errore tra dati ottenuti e aspettative.

Uno degli indici più semplici che possiamo trovare in questa categoria è l'**indice di purezza**, che ci dà una indicazione della percentuale di elementi che sono stati classificati correttamente. Per determinare il valore dell'indice come prima cosa bisogna andare a calcolare la purezza di ogni singolo cluster come rapporto tra gli elementi che sono stati correttamente allocati e il numero totale di elementi.

$$P_i = \frac{1}{N_i} \max(N_i \cap C_i) \quad (1.12)$$

- N_i elementi del cluster ottenuto con l'algoritmo.
- C_i elementi della struttura corretta .

La purezza totale della soluzione del clustering si trova facendo una somma pesata delle frazioni di punti appartenenti ad un cluster moltiplicata per il rispettivo indice di purezza.

$$P = \frac{1}{N} \sum_{i=1}^k N_i P_i \quad (1.13)$$

- N numero di elementi totali.
- k numero di cluster.

Il valore massimo di indice di purezza corrisponde alla unità, che indica che tutti i dati sono stati allocati correttamente rispetto alla struttura prestabilita.

3. **Internal clustering validation:** gli indici interni suppongono che l'algoritmo di clustering costruisce i cluster usando elementi vicini tra di loro mentre cluster diversi sono lontani. Il calcolo degli indici interni avviene utilizzando solo quantità e caratteristiche interne all'algoritmo per il calcolo degli indici.

Tra gli indici più conosciuti di questa categoria troviamo:

Silhouette index: indice in grado di valutare quali punti sono stati assegnati in modo corretto ai cluster e quali invece no. Il valore di silhouette di ogni punto viene calcolato come rapporto avente a numeratore la differenza tra la distanza minima tra il punto considerato e tutti gli altri cluster a cui il punto non appartiene " D_i " e la distanza media tra il punto considerato e tutti gli altri punti appartenenti allo stesso cluster " L_i ", al denominatore invece si ha il valore massimo tra minuendo e sottraendo del numeratore.

$$P = \frac{D_i - L_i}{\max(D_i, L_i)} \quad (1.14)$$

Se il valore di silhouette tende a 1 vuol dire che l'elemento è stato assegnato al gruppo giusto.

Se il valore di silhouette invece è piccolo allora probabilmente vuol dire che l'elemento poteva appartenere a più cluster.

Silhouette pari a -1 invece indica che elemento è stato inserito nel cluster sbagliato.

Dunn index: l'indice viene ottenuto comparando la estensione dei cluster (ad esempio il diametro) con le distanze tra di essi. La comparazione corrisponde al rapporto tra la minima distanza tra due cluster e la dimensione del cluster più esteso.

$$P = \frac{\min.dist}{\max.dim} \quad (1.15)$$

Il modo di calcolare la distanza e le dimensioni dei cluster non è unico, a seconda della definizione di distanza adottata si ottiene un Dunn index diverso. Il clustering si ritiene "buono" se l'indice assume valori alti.

Mean index adequacy (MIA): l'indice MIA valuta quanto un cluster è compatto, per componenti molto vicini tra di loro MIA assume valori molto bassi. Dato l'insieme $\mathbf{C} = \{\mathbf{c}^{(k)}, k = 1, \dots, K\}$ delle curve rappresentative dei K cluster trovati e l'insieme $\mathbf{L}^{(k)}$ degli elementi appartenenti al cluster k -esimo, l'indicatore viene calcolato come:

$$MIA^{(K)} = \sqrt{\frac{1}{K} \sum_{k=1}^K d^2(\mathbf{c}^{(k)}, \mathbf{L}^{(k)})} \quad (1.16)$$

Cluster dispersion indicator (CDI): questo indicatore viene calcolato tenendo in considerazione la distanza tra i membri dello stesso cluster $\mathbf{L}^{(k)}$ (come nel caso di MIA) e anche la distanza tra le curve rappresentative di ciascun cluster $\mathbf{c}^{(k)}$. Quindi l'indicatore CDI dà una idea sia della compattezza dei cluster e anche quanto sono diversi tra di loro i cluster. Valori più bassi di CDI indicano una suddivisione migliore.

$$CDI^{(K)} = \frac{1}{d(\mathbf{C})} \sqrt{\frac{1}{K} \sum_{k=1}^K d^2(\mathbf{L}^{(k)})} \quad (1.17)$$

Similarity matrix indicator (SMI2):

Per il calcolo di questo indicatore viene generata una matrice di similarità avente dimensioni $\mathbf{K} \times \mathbf{K}$, i cui elementi (α_{ij}) sono calcolati come:

$$\alpha_{ij} = \left(1 - \frac{1}{\ln[d(\mathbf{c}^{(i)}, \mathbf{c}^{(j)})]} \right)^{-1} \quad (1.18)$$

tali valori dicono quanto i centroidi di ciascun cluster sono simili, l'indicatore SMI è definito come il massimo dei valori α_{ij} .

$$SMI^{(K)} = \max_{i>j} \{\alpha_{ij}\} \quad (1.19)$$

In modo da evidenziare la differenza tra valori numerici definiamo SMI2 come la radice quadrata di SMI.

$$SMI2 = \sqrt{SMI} \quad (1.20)$$

SMI2 dà informazione su come i centroidi dei cluster, trovati dall'algoritmo di clustering, sono diffusi ma non dice niente per quanto riguarda la compattezza. La particolarità di questo indicatore sta nel fatto che richiede solo informazione riguardante i centroidi, quindi è particolarmente utile nel caso si abbia una numero notevole di curve di carico. Risultati di clustering migliori presentano valori di SMI minori.

Scatter index (SI): lo scatter index permette di capire la dispersione delle curve di carico e la dispersione tra i centroidi, trovati dall'algoritmo di clustering, rispetto al vettore $\mathbf{p}$ che rappresenta la dispersione totale del dataset (pooled scatter).

$$\mathbf{p} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}^{(m)} \quad (1.21)$$

$$SI^{(K)} = \left(\sum_{m=1}^M d^2(\mathbf{X}^{(m)}, \mathbf{p}) \right) \left(\sum_{k=1}^K d^2(\mathbf{c}^{(k)}, \mathbf{p}) \right)^{-1} \quad (1.22)$$

Anche in questo caso i risultati del clustering sono migliori per valori bassi di SI.

Le distanze tra vettori utilizzate dagli indici riportati sono descritte nell'appendice.

Di solito il metodo più usato è l'internal clustering validation sia perché permette di capire la bontà del risultato utilizzando solo quantità e caratteristiche appartenenti al dataset preso in considerazione, sia perché sono semplici da interpretare e i risultati che si ottengono sono affidabili. E' degno di nota che questi indicatori possono essere utilizzati sia con le serie temporali sia con qualunque vettore avente caratteristiche diverse, ma in ogni caso il confronto deve avvenire tra dati appartenenti alla stessa tipologia.

Capitolo 2

Pre-clustering

L'arrivo del mercato libero in molti Paesi ha portato con sé la necessità di conoscere meglio i propri clienti, questo ha favorito la sempre più crescente diffusione di apparecchi come gli smartmeter, anche per clienti di piccola taglia, in grado di registrare dati con una frequenza maggiore rispetto ai metodi tradizionali ed a intervalli di tempo regolari, enormi quantità di dati riguardanti i consumi dei singoli clienti.

Anche se la diffusione degli smartmeter è in continuo aumento, la installazione su tutti gli utenti non è praticabile, in quanto la quantità di informazione da trasmettere e immagazzinare sarebbe esagerata e richiederebbe molte risorse e quindi costi elevati.

Nonostante solo una frazione di clienti utilizzi lo smartmeter, la presenza di dati è comunque di dimensioni notevoli, anche quando la frequenza delle misure non è tra le più alte. Ad esempio nel caso in cui le misure avvengano ogni 30 minuti, il consumo giornaliero di energia viene rappresentato da vettori composti da 48 valori per una singola giornata.

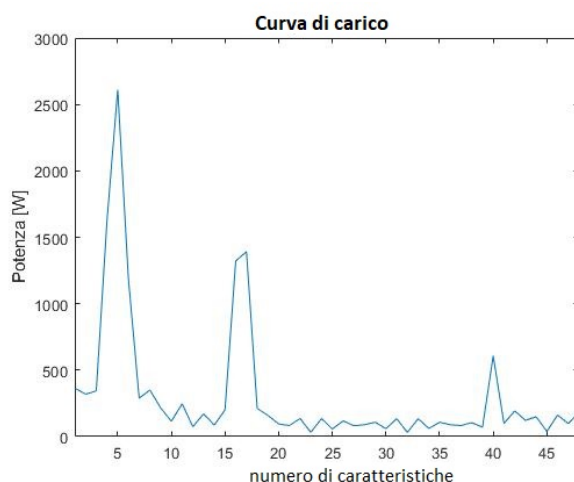


Figura 2.1: Curva di carico di una giornata estiva.

In un anno di osservazioni per un singolo cliente si hanno 365 curve, ognuna con 48 valori.

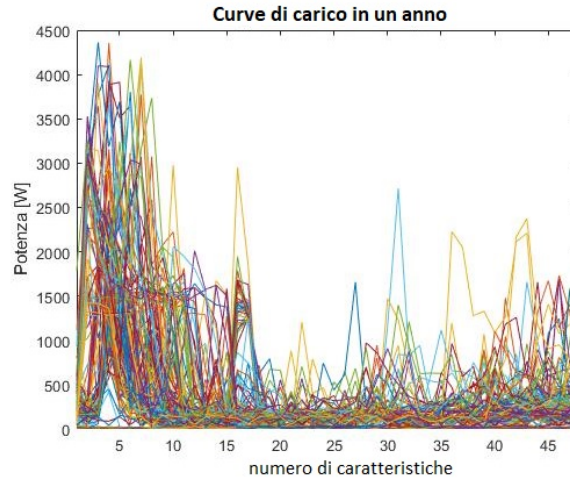


Figura 2.2: Curve di carico in 1 anno (solo giorni lavorativi).

Analizzare tutte queste curve per classificare ogni singolo cliente è molto difficoltoso, ragione per cui prima di usare questi dati come input del clustering bisogna considerare che ci saranno delle fasi in cui i dati saranno selezionati, ripuliti e processati.

Il clustering quindi sarà preceduto da alcune fasi mirate ad avere un insieme di dati con informazione affidabile. Oltre a suddividere i dati in periodi rilevanti in cui il consumo degli utenti possa ritenersi poco soggetto a fattori esterni, si dovrà verificare che nelle misure non ci siano valori derivanti da errori, guasti oppure dati mancanti. Tra gli altri motivi per cui le fasi di pre-clustering sono fondamentali troviamo la selezione delle caratteristiche (valori da utilizzare per il clustering), perchè molte volte non si lavora con tutti i valori delle curve di carico, ma si sceglie di ricavare delle caratteristiche rappresentative in modo da smaltire la quantità di dati che l'algoritmo di clustering deve processare in quanto molti algoritmi (specialmente quelli tradizionali) hanno difficoltà a gestire dataset appartenenti a spazi dimensionali grandi. Bisogna però stare attenti che oltre a benefici di accumulo e tempi di calcolo, la perdita di informazione utile può pesare nella formazione delle classi finali. Quindi un uso efficiente dei dati a disposizione è fondamentale per ricavare il profilo del consumo energetico dei clienti che poi saranno usati nel processo di classificazione dei clienti.

Normalmente le fasi pre-clustering possono essere suddivise in:

- Misura dei dati: Acquisizione dei valori di potenza per cliente in base alla scelta della frequenza di misura, che sarà un compromesso tra capacità di accumulo dei dati e fedeltà di riproduzione del consumo.
- Condizioni di carico: Selezione del periodo di tempo in cui si vogliono analizzare le curve di carico, considerando anche la tipologia di giorno che si vuole utilizzare (festivi, lavorativi, ecc).

- Pulizia dei dati: Si esaminano i dati registrati dallo smartmeter in modo da modificare o eliminare dati con valori anomali, oppure si escludono clienti che presentino dei buchi di informazioni considerevoli e quindi pochi dati utili.
- Estrazione delle curve rappresentative per cliente: Per ogni singolo cliente si andranno a utilizzare i valori delle curve giornaliere per calcolare un'unica curva rappresentativa del consumo del cliente date le condizioni di carico.
- Selezione delle caratteristiche: Scelta o calcolo delle caratteristiche che saranno utilizzate come input del clustering, partendo da quelle che compongono la curva rappresentativa del cliente. In questa fase è possibile anche eseguire riduzione delle dimensioni del dataset per via di mappature a spazi dimensionali ridotti, portando ulteriori vantaggi legati all'accumulo e ai tempi di calcolo, ma possono anche causare delle perdite di informazione.

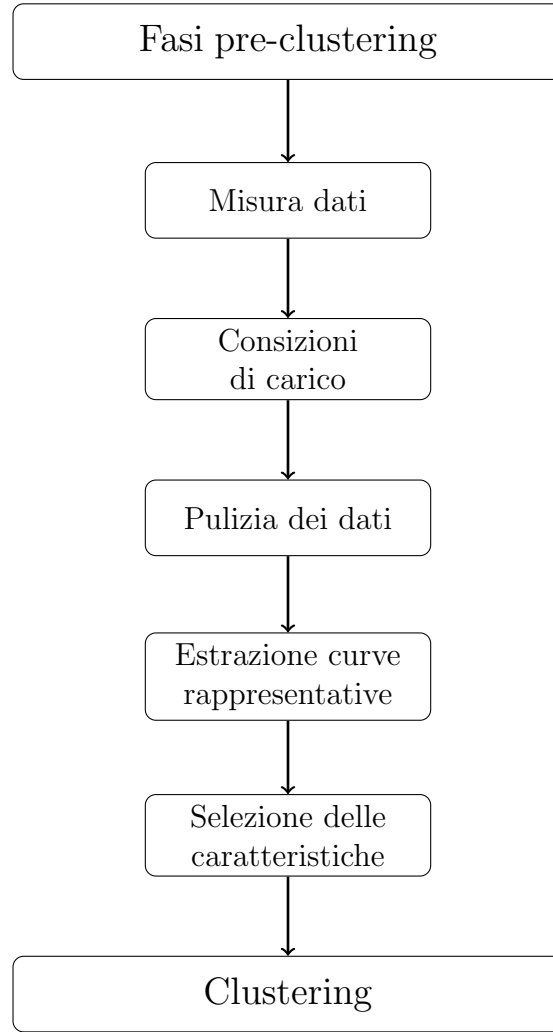


Figura 2.3: Fasi pre-clustering.

2.1 Misura dei dati e condizioni di carico

Dalla macro-categoria di interesse (residenziali, industriale, ecc) vengono selezionati un insieme di N clienti sui quali verrà eseguito il monitoraggio, che deve avvenire per tempi sufficientemente lunghi in modo da avere una quantità sufficiente di dati (es. 2-3 settimane, 1 mese, 1 anno). La scelta del passo temporale di analisi sarà il prodotto del compromesso tra la accuratezza con la quale si vuole riprodurre il consumo energetico e dallo spazio di accumulo dei dati che si vuole a disposizione.

I valori tipici sono:

- 1 min, curve di carico con 1440 punti.

- 15 min, curve di carico con 96 punti.
- 30 min, curve di carico con 48 punti.
- 1 ora, curve di carico con 24 punti.

Tutto il tempo di monitoraggio deve essere suddiviso e classificato a sua volta in archi temporali in cui il consumo del cliente risenta poco di fattori non elettrici come possono esserlo le condizioni climatiche. Ad esempio una prima suddivisione può avvenire considerando le stagioni in cui il clima ha variazioni contenute e quindi con poco effetto sulle abitudini del consumatore, successivamente si può eseguire una partizione interna in cui si escludono alcune tipologie di giorni come ad esempio i giorni festivi, nei quali si è abbastanza sicuri che le curve giornaliere abbiano un andamento diverso dal resto dei giorni.

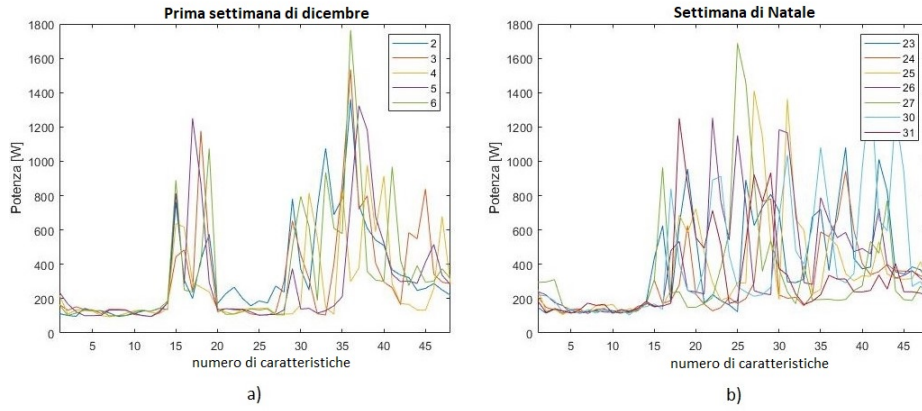


Figura 2.4: Confronto tra curve di carico appartenenti a 2 settimane diverse del mese di dicembre: a) prima settimana di dicembre b) settimana di Natale.

Come è possibile vedere in Figura 2.4, dove si hanno curve di carico appartenenti a settimane diverse dello stesso mese, una settimana normale e l'altra appartenente al periodo natalizio. Nella prima si può notare un andamento abbastanza regolare, il che indica che l'utente segue la sua routine regolarmente. Nella settimana delle festività invece possiamo vedere un comportamento più "casuale" a dimostrazione che per via delle festività le abitudini dell'utente sono state influenzate.

La partizione stagionale del periodo di monitoraggio non è l'unica, ma può essere spezzata anche in intervalli di tempo più piccoli e quindi con un pattern più "solido", ma mantenendo sempre un numero di dati sufficiente a rappresentare il consumo energetico.

2.2 Pulizia dei dati ed estrazione delle curve rappresentative

Una volta selezionata la tipologia di dati con la quale si vuole lavorare si procede alla verifica della completezza e integrità dei dati, ovvero che appartengano a condizioni operative normali.

Occorre però prima definire le tipologie di anomalie più comuni:

- Anomalie puntuali: Come indicato dal nome, si intendono i singoli punti che deviano significativamente dal resto del gruppo. Alcuni esempi possono essere punti delle serie temporali in cui il valore di potenza va oltre la potenza contrattuale, oppure tensioni oltre i valori stabiliti dalla norma (nel caso sia possibile monitorare le tensioni).
- Anomalie collettive: In questa categoria rientrano gli insiemi di punti che hanno un comportamento diverso dal altri insiemi. Nel nostro caso questo può accadere in giornate che pur avendo consumi energetici nella norma, l'energia totale giornaliera supera di molto la energia consumata nel resto delle giornate.
- Anomalie contestuali: Sono i punti o insiemi di punti il cui comportamento è considerato anomalo in determinate condizioni. Un esempio di anomalie di questo tipo sono i consumi di energia delle giornate festive, nel caso in cui si voglia lavorare con diverse condizioni di carico (come giornate lavorative). Nel nostro caso questo tipo di anomalie viene filtrato nel momento in cui si scelgono le condizioni di carico, quindi non rientrano nella fase di pulizia dei dati.

Oltre ai valori anomali occorre verificare anche che il dataset sia "completo", cioè andare a vedere se per qualche motivo vi sono dei buchi all'interno delle serie temporali. Dopo che questi valori sono stati identificati occorre fare una scelta tra eliminare oppure ripulire i dati. La eliminazione dal dataset avviene in caso buona parte dei valori misurati abbiano scarsa informazione sul comportamento del cliente, come ad esempio avviene con i dati provenienti da anomalie collettive. I dati invece possono essere ripuliti in caso le caratteristiche delle curve di carico presentino valori mancanti o anomalie puntuali non eccessive in numero, ai quali si può rimediare andando a "correggere" questi valori con tecniche di interpolazione, sostituendoli con una delle caratteristiche adiacenti, oppure ignorarli utilizzando solo i dati attendibili disponibili.

Come è possibile vedere in Figura 2.5 ci sono spezzoni di dataset in cui un certo numero di caratteristiche è mancante in giorni diversi. Nel primo caso (sinistra) questo numero è elevato perché corrisponde a 6 giornate intere senza aver registrato dati, ragione per cui se le condizioni di carico comprendono questi giorni, la soluzione ottimale potrebbe essere quella di eliminare i dati appartenenti a questo utente. Nella figura di destra invece abbiamo due giornate incomplete, la prima avente pochissime caratteristiche può essere ignorata del tutto per la formazione delle curve rappresentative; la seconda invece presenta solo 3 valori mancanti (dalle ore 00:00 alle ore 01:30) in questo caso invece una soluzione potrebbe essere ignorare questi 3 buchi nella formazione delle curve rappresentative.

03-Jun-2013 20:30:00	631	10-Sep-2013 23:00:00	359
03-Jun-2013 21:00:00	235	10-Sep-2013 23:30:00	108
03-Jun-2013 21:30:00	337	11-Sep-2013 00:00:00	85
03-Jun-2013 22:00:00	393	11-Sep-2013 00:30:00	85
03-Jun-2013 22:30:00	311	11-Sep-2013 01:00:00	87
03-Jun-2013 23:00:00	72	12-Sep-2013 01:30:00	94
03-Jun-2013 23:30:00	185	12-Sep-2013 02:00:00	81
04-Jun-2013 00:00:00	89	12-Sep-2013 02:30:00	98
04-Jun-2013 00:30:00	76	12-Sep-2013 03:00:00	67
04-Jun-2013 01:00:00	169	12-Sep-2013 03:30:00	100
10-Jun-2013 00:00:00	57	12-Sep-2013 04:00:00	64
10-Jun-2013 01:00:00	117	12-Sep-2013 04:30:00	94

a)

b)

Figura 2.5: Esempio di serie temporali con dati mancanti: a) utente con 6 giorni mancanti b) utente con poche caratteristiche mancanti.

Tutti questi passaggi portano ad avere un set di informazioni affidabile per ciascun cliente.

Una volta ripuliti i dati raccolti è necessario identificare un solo profilo di carico che andrà a rappresentare il singolo cliente nella condizione di carico. Questa curva è la Typical Daily Load Pattern (TDLP) ricavata andando a combinare tutte le curve di carico rimaste (dalla fase precedente) del singolo cliente. I metodi statistici sono di solito quelli più utilizzati per risalire alla TDLP, tra i quali la operazione di media è la più comune.

Ad esempio se per un insieme di N clienti vengono registrate M curve di carico ciascuno, si avrà l'insieme dei dati $\mathbf{X}^n$ dell'enne-esimo cliente definito come:

$$\mathbf{X}^n = [\mathbf{x}_m^{(n)}, m = 1, \dots, M]$$

dove ogni vettore $\mathbf{x}^n$ a sua volta sarà composto da un certo numero H di caratteristiche (in base alla frequenza delle misure):

$$\mathbf{x}_m^n = [x_{hm}^n, h = 1, \dots, H]$$

Le caratteristiche della TDLP saranno ricavate andando a fare la media di tutti i valori appartenenti allo stesso istante della giornata, in questo modo si ottiene un unico vettore con lo stesso numero di caratteristiche dei vettori di partenza:

$$\bar{\mathbf{x}}^n = [\bar{x}_1^n, \bar{x}_2^n, \dots, \bar{x}_H^n]$$

Il risultato sarà una TDLP per cliente meno soggetta a valori occasionali che possono accadere in un determinato giorno e quindi più rappresentativa del cliente in condizioni normali.

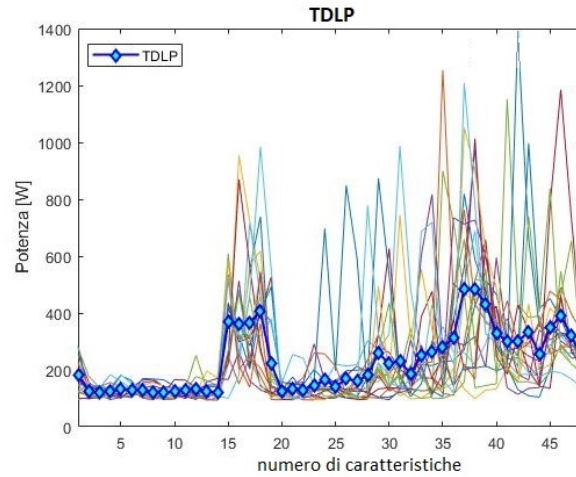


Figura 2.6: Esempio Typical Daily Load Patten.

Come è possibile vedere nella Figura 2.6, molte giornate presentano dei picchi in posizioni diverse, che sono caratteristici dei clienti residenziali per via dell'utilizzo di elettrodomestici in modo poco regolare. Con la operazione di media i picchi vengono attenuati e viene fuori un pattern rappresentativo che riassume il comportamento durante queste giornate.

Dalle TDLP l'informazione più rilevante per il clustering è la forma della curva, sarà questa caratteristica quella più utile ai fini del calcolo della distanza tra due pattern rappresentativi, per determinare quanto simili sono due curve.

La normalizzazione delle TDLP è lo strumento che ha lo scopo di rimuovere l'influenza del valore delle ampiezze permettendo di arrivare a un Representative Load Pattern (RLP) per cliente. Il valore di potenza normalizzante si chiama potenza di riferimento e può avere valori diversi:

- Relativa al valore di potenza media della serie, che permette di mantenere anche informazioni sull'energia totale consumata dal cliente.
- Normalizzazione usando il valore di picco raggiunto durante tutto il periodo di monitoraggio. Questa scelta però può risultare sconveniente in quanto il picco è frutto di situazioni anomale.
- Usando la potenza contrattuale, permette l'identificazione di valori oltre la norma.
- Usando il valore massimo della TDLP, in questo caso i picchi anomali sono stati attenuati dall'operazione di media ottenendo anche delle curve che variano tra 0 e 1. In alcuni casi si preferisce utilizzare anche il valore minimo in modo di avere il valore nullo in tutte le RLP. In questo caso, però, la curva di carico viene ad essere deformata e questo può alterare la natura dei confronti.

La scelta tipica ricade nella normalizzazione utilizzando il valore massimo della TDLP, in questo modo otteniamo un insieme di vettori $\mathbf{D}$ composto dalle N RLP:

$$\mathbf{D} = [\mathbf{d}_n, n = 1, \dots, N]$$

$$\mathbf{d}_n = \bar{\mathbf{x}}_n / \max(\bar{\mathbf{x}}_n)$$

dove $\mathbf{d}_n$ ha lo stesso numero di caratteristiche delle curve di carico giornaliera di partenza

$$\mathbf{d}_n = [d_{hn}, h = 1, \dots, H]$$

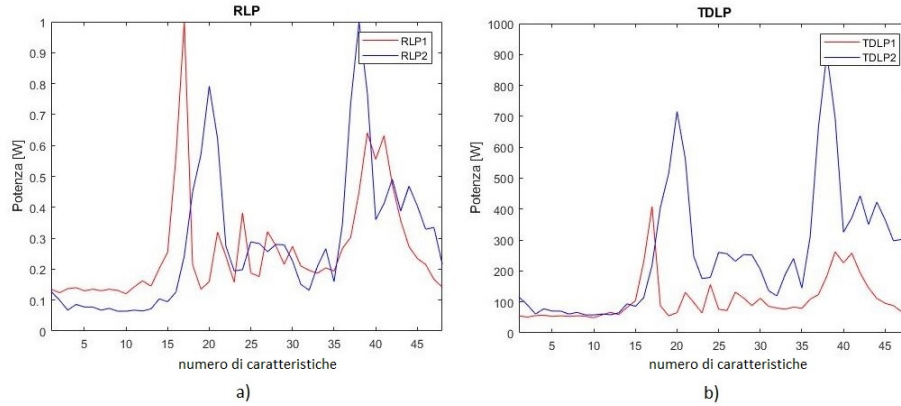


Figura 2.7: a)RLP vs b)TDLP.

Con la normalizzazione abbiamo ottenuto due curve in cui si dà più peso alla forma rispetto al valore di potenza (Figura 2.7). Questo è particolarmente utile quando la similitudine si basa sul calcolo di distanza, in quanto se l'influenza del valore di potenza non viene eliminata due curve molto simili nella forma possono risultare in classi diverse perché considerate molto distanti, proprio per i livelli di potenza raggiunti dai due utenti.

2.3 Selezione delle caratteristiche

La selezione delle caratteristiche è un aspetto molto importante in quanto riguarda la scelta dei dati di ingresso per il clustering. Partendo dalle RLP precedentemente ricavate è possibile fare uno stadio di estrazione di nuove caratteristiche oppure utilizzare direttamente quelle delle RLP.

Caratterizzare la forma del consumo attraverso tutti i punti è uno dei modi più semplici in quanto non richiede altri passi di calcolo sia prima che dopo il clustering. In questo caso si parlerà di H fattori di forma diretti, il cui numero dipende dalla frequenza di campionamento. L'alternativa ad usare fattori di forma diretti è quella di andare a ricavare nuovi fattori di forma (da quelli di partenza), chiamati fattori di forma indiretti e che metteranno in mostra aspetti particolari delle RLP.

Tra i fattori di forma indiretti più comuni troviamo:

- fattori di forma a base giornaliera, i cui valori variano nell'intervallo $[0 \ 1]$, alcuni esempi sono:

$$SF_1 = \frac{P_{av}}{P_{max}} \quad (2.1)$$

$$SF_2 = \frac{P_{av,daylight}}{P_{max.daylight}} \quad (2.2)$$

$$SF_3 = \frac{P_{min}}{P_{av}} \quad (2.3)$$

$$SF_4 = \frac{P_{av,night}}{P_{av}} \quad (2.4)$$

$$SF_5 = \frac{P_{av,lunchtime}}{P_{av.daylight}} \quad (2.5)$$

$$SF_6 = \frac{P_{min,daylight}}{P_{av.daylight}} \quad (2.6)$$

- Una variante dei fattori di forma precedenti sono gli *average alignment degree*, che sono ottenuti dividendo l'asse dei tempi in quattro sottointervalli, in ciascuno di questi sottointervalli si calcolerà la potenza media che verrà poi divisa per la potenza media giornaliera:

$$AAD_{dt} = \frac{P_{av,dt}}{P_{av,daily}} \quad (2.7)$$

- Coefficienti di non uniformità dati dal rapporto tra minimo e massimo della potenza giornaliera.

$$NUC = \frac{P_{min,daily}}{P_{max,daily}} \quad (2.8)$$

- Coefficienti di modulazione dati dal rapporto tra le potenze di picco orarie e la potenza media giornaliera

$$MC = \frac{P_{av,peakhour}}{P_{av,daily}} \quad (2.9)$$

Oltre ai fattori di forma nel dominio del tempo è possibile usare dei coefficienti nel dominio della frequenza, derivanti da trasformazioni quali la trasformata di Fourier e la trasformata *wavelet*.

La estrazione di nuove caratteristiche porta con sé i vantaggi poter utilizzare meno spazio per l'accumulo dei data e di avere dei tempi di calcolo minori in quanto gli algoritmi di clustering lavorano con una quantità di dati inferiore.

Ulteriori riduzioni dello spazio dimensionale si possono ottenere attraverso metodi di proiezione quali Principal Component Analysis (PCA), Curvilinear Component Aanalysis (CCA), Sammon maps:

1. Principal Component Analysis (PCA): è un metodo di riduzione lineare che proietta i dati originali H -dimensionali in uno spazio R -dimensionale, dove $R \ll H$. L'idea dietro questo metodo è quella di andare a ricavare delle nuove variabili non correlate tra loro (ottenute come combinazione lineare di quelle di partenza) attraverso gli autovalori e autovettori della matrice di covarianza, la informazione totale di partenza sarà divisa tra queste nuove variabili in percentuali diverse (in base all'autovalore, che indica la quantità di varianza che la nuova variabile ha con sé).

Dato che le nuove variabili (chiamate anche variabili principali) non contengono la stessa percentuale di informazione, ma tutta l'informazione viene distribuita in modo decrescente tra di esse, la riduzione delle dimensioni si può fare considerando solo le variabili che massimizzano la varianza (informazione).

Questo metodo presenta una buona compressione dei dati ma ha lo svantaggio di essere solo lineare e quindi non può rappresentare bene le correlazioni non lineari tra variabili (non preserva strutture complesse generalmente), in più bisogna stare attenti alla perdita di informazione.

2. Sammon map: Questo algoritmo si basa su una mappatura dei vettori di partenza in uno spazio dimensionale minore in modo tale da mantenere la struttura dei dati iniziali. Per preservare la struttura iniziale questo metodo cerca di preservare le distanze dello spazio iniziale in modo tale che elementi vicini nello spazio di partenza siano vicini anche nello spazio ridotto senza modificare la topologia. La fedeltà della riproduzione della struttura viene quantificata dalla seguente funzione di errore:

$$E = \frac{1}{\sum_{i=1}^{M-1} \sum_{j=i+1}^M d(\mathbf{x}_i, \mathbf{x}_j)} \sum_{i=1}^{M-1} \sum_{j=i+1}^M \frac{[d(\mathbf{x}_i, \mathbf{x}_j)] - [d(\mathbf{y}_i, \mathbf{y}_j)]^2}{d(\mathbf{x}_i, \mathbf{x}_j)} \quad (2.10)$$

dove $d(\mathbf{x}_i, \mathbf{x}_j)$ e $d(\mathbf{y}_i, \mathbf{y}_j)$ sono le distanze nello spazio originario e in quello ridotto, rispettivamente. Questa funzione dà una indicazione della quantità di "struttura persa" durante la trasformazione. Molto spesso come definizione di distanza si usa la distanza euclidea, ma è possibile utilizzarne altre. Da notare che in entrambe le dimensioni il numero di punti è lo stesso, ciò che cambia è la dimensione dei vettori.

3. Curvilinear Component Analysis (CCA): E' un *self organizing neural network* che proietta i dati H -dimensionali in uno spazio ridotto preservando la topologia dello spazio originale, come la Sammon map è in grado di preservare anche strutture non lineari dello spazio originale.

Questo metodo porta con sé alcuni miglioramenti rispetto alle Sammon map dovuti a una riproduzione migliore delle distanze. CCA in questo caso dà priorità alla riproduzione delle distanze piccole preservando la topologia locale.

$$E = \frac{1}{2} \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M [d(\mathbf{x}_i, \mathbf{x}_j)] - [d(\mathbf{y}_i, \mathbf{y}_j)]^2 F(d(\mathbf{y}_i, \mathbf{y}_j), \lambda) \quad (2.11)$$

La funzione di errore "E" ha una nuova componente, la funzione F, che è una funzione monotonamente decrescente utile a preservare la topologia locale avente come indice di decadenza λ . Inoltre CCA usa la quantizzazione vettoriale che permette di ridurre il carico computazione quando si calcolano le distanze.

In questo modo si ottiene un insieme di caratteristiche costruite in uno spazio vettoriale ridotto, ma che contengono certe caratteristiche delle RLP di partenza.

I passaggi descritti in questo capitolo seguono l'ordine maggiormente impiegato dalle tecniche di clustering più tradizionali, ci sono casi in cui le fasi dopo la pulizia dei dati possono seguire un ordine diverso o anche inserire altre fasi intermedie.

Capitolo 3

CONDUCTS

3.1 Descrizione della metodologia

Una delle problematiche legate all'utilizzo della metodologia classica per ricavare le curve rappresentative degli utenti risiede nella dipendenza del tempo che esse presentano ancora, infatti le RLP presentano i picchi negli istanti in cui la maggior parte delle curve giornaliere li hanno. Se si utilizza la distanza euclidea per confrontare due RLP con gli stessi picchi ma spostati su istanti diversi della giornata, è molto probabile che il risultato del clustering le metta in gruppi diversi.

Per superare tale problematica si può utilizzare una metodologia abbastanza recente chiamata CONDUCTS, la quale considera le serie temporali in modo diverso all'approccio classico. CONDUCTS (CONsumption DURATION Curve Time Series) sfrutta la manipolazione di datastream e una *unsupervised machine learning technique* per identificare e raggruppare il consumo energetico evitando problemi legati al tempo. Infatti attraverso l'utilizzo delle cumulative si ricaveranno delle nuove caratteristiche con un numero di punti ridotto e ordinato, in grado di rappresentare il consumo dell'utente nel periodo di tempo considerato. Applicare i concetti di distanza euclidea sui nuovi punti è più appropriato perché stiamo confrontando delle curve monotonamente crescenti e quindi in ciascun punto vengono confrontati valori normalizzati seguendo la stessa sequenza.

Al suo interno CONDUCTS racchiude sia le fasi di pre-clustering necessarie ad arrivare ai nuovi punti sia le fasi di valutazione dei risultati. Sostanzialmente le fasi di pre-clustering sono quelle riportate nel capitolo 2, tranne per lo stadio della estrazione delle curve rappresentative, in quanto CONDUCTS non utilizza le RLP classiche, in più è richiesta maggior attenzione ad alcune fasi rispetto ad altre.

3.2 Fasi dell'approccio CONDUCTS

Gli stadi che compongono l'approccio CONDUCTS sono elencati di seguito:

- Misura e pulizia dei dati: questi stadi sono identici a quelli illustrati nei capitoli precedenti.

- Normalizzazione dei dati: la normalizzazione ha un ruolo molto importante per quanto riguarda l'output del clustering, infatti in base alla normalizzazione le curve possono avere un "aspetto" diverso e quindi essere raggruppate in cluster differenti, CONDUCTS utilizza la normalizzazione rispetto alla potenza contrattuale per rimuovere gli effetti dei livelli di consumo senza modificare la forma delle curve di carico.
- Finestra temporale e condizioni di carico: anche in questo caso l'intervallo di tempo deve essere scelto in modo strategico, per limitare la influenza di parametri esterni come la temperatura e condizioni climatiche avendo un numero adeguato di punti. Inoltre è richiesta la differenziazione tra giornate lavorative e giornate festive o weekend, le quali devono essere analizzate separatamente.
- Creazione dei dati di input: questo stadio può essere paragonato a quello della estrazione delle curve rappresentative e selezione delle caratteristiche del capitolo 2, ma la manipolazione dei dati è completamente diversa.

I dati degli N utenti appartenenti alla stessa finestra temporale sono disposti in modo crescente per ottenere un unico vettore per utente in cui i valori del consumo seguono una sequenza ordinata, tali vettori prenderanno il nome di "curve di durata". Come è possibile vedere in Figura 3.1 la costruzione delle curve di durata richiede l'utilizzo di tutti i punti della finestra temporale considerata, in questo caso abbiamo delle curve con 480 punti ciascuna perché consideriamo 2 settimane lavorative con frequenza di campionamento pari a 30 min.

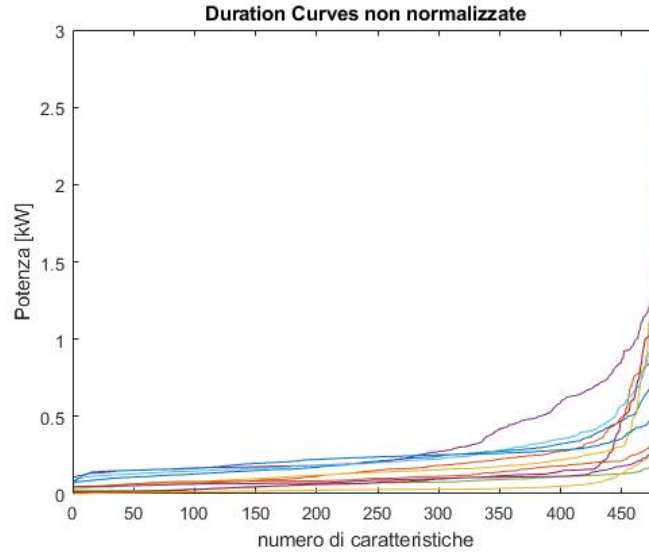


Figura 3.1: Curve di durata non normalizzate di 10 utenti considerando i dati di 2 settimane.

Utilizzando le curve di durata di tutti i clienti viene calcolato un unico vettore delle differenze medie tra valori adiacenti. Tali valori sono a loro volta ordinati e

normalizzati rispetto al valore della differenza massima, per ricavare la funzione di distribuzione del vettore delle differenze medie (CDF).

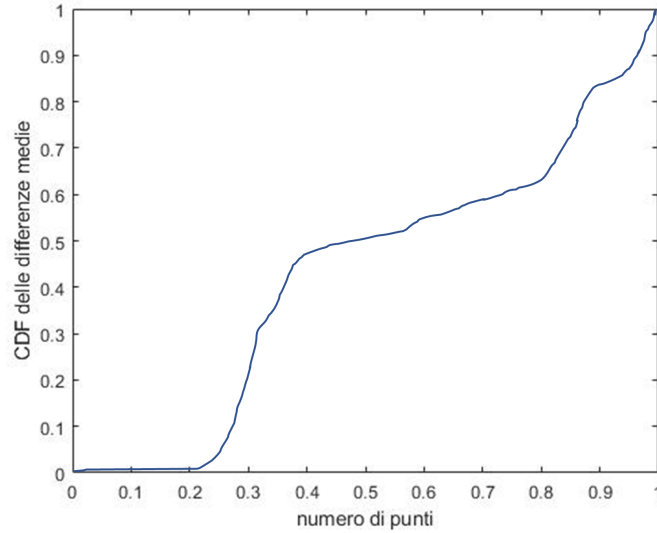


Figura 3.2: Esempio di CDF delle differenze medie.

A partire dalla CDF sono identificati i decili che dividono l'asse verticale in 10 intervalli proporzionali. I rispettivi valori sull'asse orizzontale sono ricavati dall'incrocio tra i decili e la CDF, tali valori indicheranno la posizione dei punti da prendere nelle curve di durata normalizzate, che rappresenteranno gli utenti e saranno utilizzati come input nel processo di clustering.

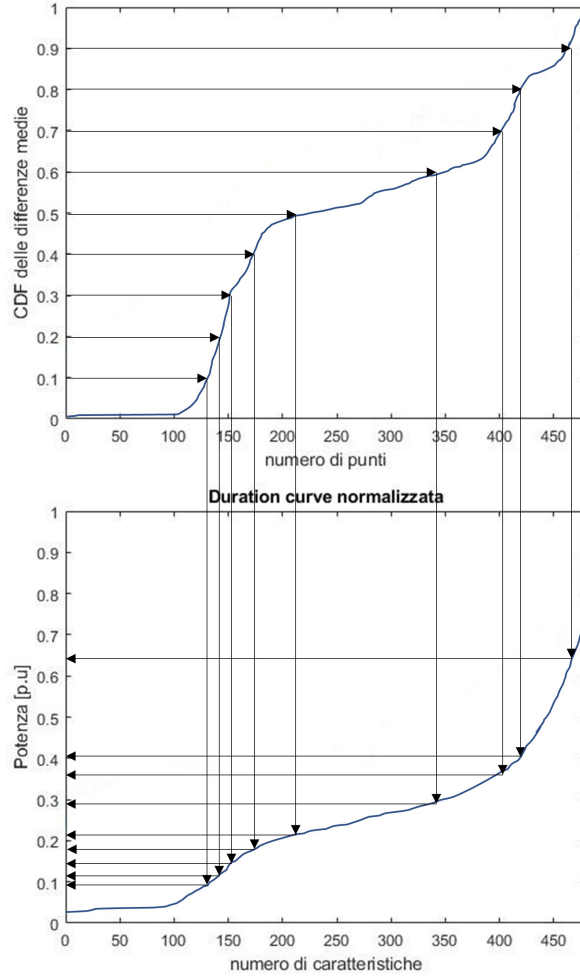


Figura 3.3: Procedura di estrazione dei decili.

- Auto regolazione: ricavate le caratteristiche necessarie si procede con l'individuazione del numero K di profili nei quali gli utenti vengono classificati. Per risolvere questa problematica utilizziamo l'indicatore SSE (sum of squared errors), che dà una idea di quanto sono compatti i cluster.

$$SSE = \sum_{k=1}^K \sum_{j=i}^{n_k} [d(\mathbf{c}_k, \mathbf{l}_j)]^2 \quad (3.1)$$

dove:

- $\mathbf{c}_k$ è il centroide del cluster K .
- n_k è il numero di utenti appartenenti al cluster k .
- $\mathbf{l}_j$ è il vettore dei decili che rappresenta l'utente j .

Applicando l'algoritmo di clustering per un certo range di valori K , possiamo costruire il grafico $SSE-K$ che mostra le variazioni dell'indicatore di compattezza, il K ottimale corrisponderà al valore per cui non si osservano più grosse variazioni di SSE.

- caratterizzazione dei cluster: una volta formati i K cluster, CONDUCTS permette di utilizzare diversi metodi per analizzare la qualità e la natura dei cluster e anche la natura dei consumi:
 1. Una rapida rappresentazione dei cluster identificati è ottenuta dall'utilizzo dei centroidi.
 2. L'utilizzo dei boxplot (o whiskers plot) permette di visualizzare i gruppi dei dati attraverso i loro quartili.
 3. L'utilizzo degli scatterplot permette di visualizzare la relazione tra il massimo consumo nella finestra temporale e il consumo medio per ciascun cluster.
- Valutazione dei cluster: CONDUCTS integra gli indici di validità basati sul valore di silhouette per esaminare la aggregazione dei punti all'interno dei cluster, la distanza tra cluster e la distribuzione degli utenti tra tutti i cluster:
 1. Average silhouette index (ASI), calcolata come la media dei valori di silhouette per l'intera popolazione m di utenti.

$$ASI = \frac{1}{M} \sum_{i=1}^n s_i \quad (3.2)$$

2. Global silhouette index (GSI), che dà una indicazione di quanto sono bilanciati i cluster in termini di numero di utenti.

$$S_k = \frac{1}{n_k} \sum_{i=1}^{n_k} s_i \quad (3.3)$$

$$GSI = \frac{1}{K} \sum_{k=1}^K S_k \quad (3.4)$$

Per entrambi questi indicatori, valori più grandi sono migliori.

Capitolo 4

Identificazione delle anomalie su dataset reali

4.1 Dataset

Nei capitoli precedenti abbiamo visto le varie fasi che portano alla formazione dell'input per l'algoritmo di clustering sia nel caso classico sia nella metodologia proposta da CONDUCTS. Per entrambi gli approcci la procedura di pulizia dei dati è la stessa, quindi vedremo di utilizzare i vari passaggi su due dataset reali derivanti da due progetti inglesi:

- Low Carbon London project, progetto condotto dalla UK Power Networks e durato 4 anni che aveva come obiettivo principale quello di monitorare l'impatto delle tecnologie low-carbon sulla rete di distribuzione londinese e sulle emissioni di CO₂. Tra i problemi che si cercava di risolvere c'era anche il *network planning*, che richiedeva il monitoraggio del consumo elettrico di un certo numero di residenze dell'area londinese. Più precisamente il dataset comprende il consumo di energia elettrica di oltre 5000 residenze dell'area nord-est di Londra, registrate ogni 30 minuti partendo dalla fine del 2011 fino a febbraio 2014. Il campione di utenti è stato scelto in modo da rappresentare in modo bilanciato la popolazione londinese. All'interno del dataset c'è una suddivisione legata al prezzo dell'energia, 1100 utenti sono stati sottoposti a una tariffa variabile, mentre il resto a una tariffa fissa. Per ridurre i tempi di calcolo si utilizzeranno 500 utenti sottoposti alla tariffa fissa.
- L'altro dataset utilizzato deriva dal progetto *Thames Valley Vision*, che è stato guidato dalla Scottish and Southern Energy Power Distribution (SSEPD). L'obiettivo di questo progetto è di monitorare e di modellizzare la rete elettrica utilizzando dati provenienti da utenti residenziali e da sottostazioni di bassa tensione in modo da poter capire la domanda attuale di energia, prevedere come questa evolverà nel futuro, implementare un controllo smart per ridurre i picchi di domanda e bilanciare la rete. Le aree coperte dal progetto sono quelle di Bracknell e i dintorni della Thames Valley area. Il dataset è composto da 220 utenti monitorati tra il 2013 e il 2014.

L'attenzione verrà concentrata unicamente sulle anomalie e non sui dati mancanti, in quanto solo una piccola parte degli utenti ha dati incompleti.

4.2 Tecniche per la identificazione delle anomalie

I dataset saranno sottoposti a 4 metodi diversi di pulizia dei dati, i risultati saranno successivamente messi a confronto in modo da determinare il metodo che si addice di più ai dataset.

L'attenzione verrà concentrata maggiormente sulle anomalie di tipo collettivo, cioè si andranno ad utilizzare informazioni derivanti da gruppi di caratteristiche piuttosto che dai singoli, poichè una analisi puntuale delle anomalie in dataset di grosse dimensioni, come in questo caso, risulterebbe computazionalmente dispendiosa, inoltre i dataset non forniscono i valori di potenza contrattuale.

La partizione dei dataset, ai fini dell'identificazione delle anomalie, che adotteremo sarà del tipo mensile, questa scelta riduce l'influenza di fattori climatici e abitudinari che possono risultare nella identificazione di falsi outlier, come ad esempio può accadere quando si mettono insieme giornate estive (in cui il consumo di energia è aumentato a causa della presenza delle apparecchiature di condizionamento) con quelle primaverili.

Come informazioni utili si è deciso di considerare le funzioni di distribuzione dell'energia durante la giornata e l'energia giornaliera totale in ogni mese, quest'ultima ci permette di ridurre ulteriormente il carico computazionale, infatti ogni giornata sarà caratterizzata soltanto da un valore. A seconda della tecnica utilizzeremo una delle due come input.

Le tecniche proposte sono elencate di seguito:

- Z-score
- Median Absolute deviation
- DBSCAN
- Analisi delle cumulative

In tutti questi casi si dovrà affrontare anche l'impostazione dei parametri di soglia che, non esistendo un modello a cui fare riferimento, devono essere trovati in modo empirico.

4.2.1 Z-score

Il Z-score è un metodo del tipo statistico parametrico capace di misurare la differenza di un insieme di osservazioni dal valore più probabile, ovvero il valore medio. Più precisamente il valore di Z_{score} esprime il numero di deviazioni standard di distanza che un certo campione presenta rispetto alla media, il cui calcolo è definito come:

$$Z_{score} = \frac{\bar{x}_i - \mu}{\sigma} \quad (4.1)$$

Come è possibile vedere, i parametri usati in questo caso sono valore medio e standard deviation (SD), le cui definizioni sono riportate di seguito:

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.2)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n - 1}} \quad (4.3)$$

dove:

- $\bar{X}$ è il valore medio.
- σ è il valore di deviazione standard.
- x_i è il valore della variabile nel punto considerato.
- n indica il numero di dati.

L'applicazione del Z_{score} richiede che gli input non appartengano ad uno spazio dimensionale elevato, ragione per cui utilizziamo le energie giornaliere come valori di x_i .

Maggiormente questo approccio si utilizza per distribuzioni del tipo Gaussiane o simili (simmetriche), in cui si ha una idea del range di SD dove si trova la maggior parte della popolazione. Lo scarto degli outlier avviene quindi impostando una soglia fissa pari a 3 SD.

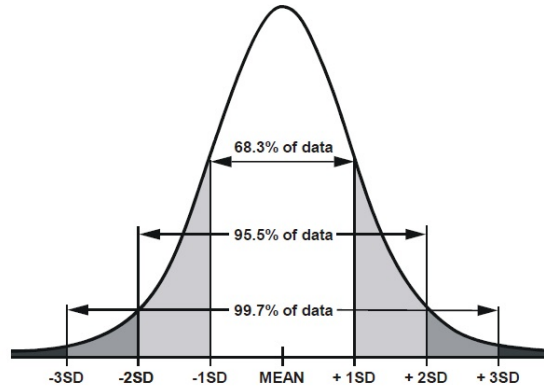


Figura 4.1: Esempio di distribuzione normale.

Come si può vedere in Figura 4.1 nel caso di distribuzione normale meno dell'1% dei dati si trova oltre 3 SD dalla media, porzione in cui si ipotizza si trovino le anomalie in quanto poco frequenti.

Nel caso dei nostri dati le distribuzioni che si hanno sono diverse da mese a mese e molto spesso sono tutt'altro che distribuzioni normali. Come si può vedere in Figura 4.2 la maggior parte delle distribuzioni non somigliano a quella normale, la distribuzione, d) è quella che si avvicina di più.

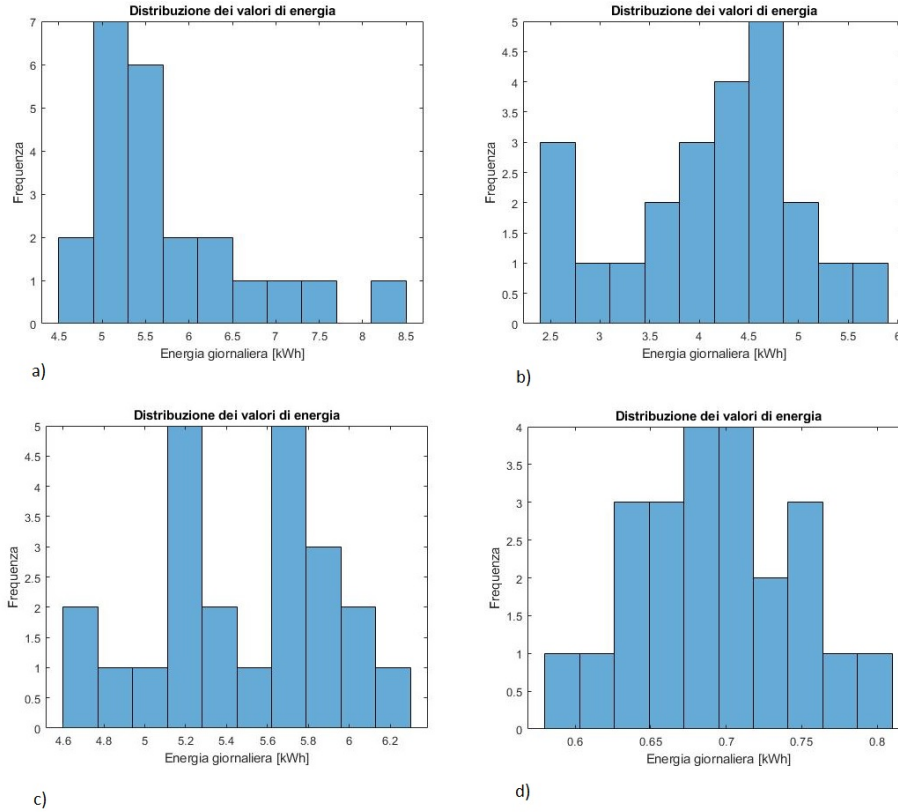


Figura 4.2: Esempi di distribuzioni delle energie totali del primo utente.

Tuttavia essendo computazionalmente poco impegnativo si è deciso di applicarlo e vedere come si comporta. La regola di 3 SD non sarà considerata come soglia fissa ma si adotterà un approccio del tipo *elbow* ("ginocchio") per determinare il multiplo di SD da adottare, questo valore non sarà unico per cliente ma cambierà di mese in mese.

Come è possibile vedere in Figura 4.3 il valore di soglia da adottare è stato determinato visualizzando il numero di outlier che si ha in funzione del multiplo di SD, e corrisponde al primo multiplo "C" per il quale il numero di outlier non subisce grandi variazioni.

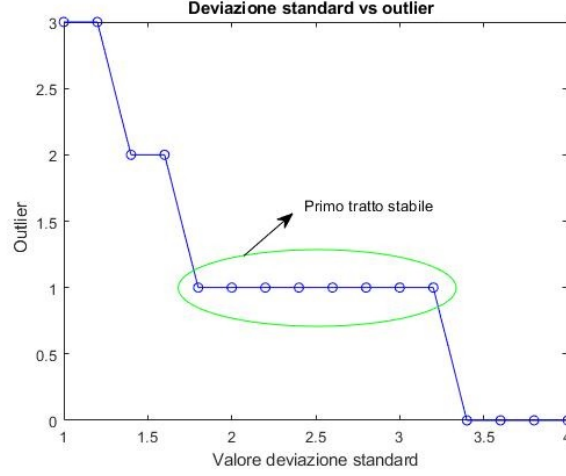


Figura 4.3: Ginocchio del secondo mese, utente 1.

Dal valore del ginocchio ricaviamo il limite superiore (UL) e inferiore (LL) che determinano l'area oltre la quale i campioni vengono classificati come outlier.

$$UL = \bar{X} + C\sigma \quad (4.4)$$

$$LL = \bar{X} - C\sigma \quad (4.5)$$

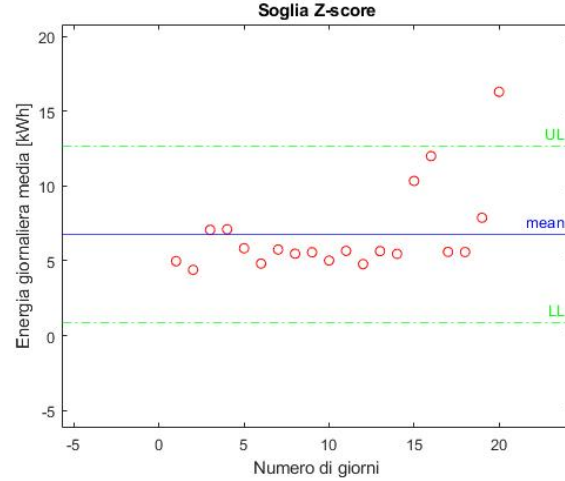


Figura 4.4: Esempio di soglia nel caso del primo utente, mese di febbraio.

Lo svantaggio principale di questo metodo tuttavia sta nel fatto che la media e la deviazione standard risentono molto del valore dei dati, quindi per anomalie estreme si tende ad aumentare il valore dei parametri rendendo il metodo innecessariamente più tollerante verso le anomalie.

4.2.2 Median absolute deviation (MAD)

MAD è un altro metodo statistico parametrico che come nel caso precedente si applica a dati del tipo univariante, ragione per cui l'input utilizzato è l'energia giornaliera. Pur avendo una applicazione simile al metodo precedente viene considerato più robusto del Z-score, specialmente per dataset piccoli, in quanto non fa ipotesi sulla distribuzione dei dati e utilizza la mediana. Infatti, dato un insieme ordinato di valori, la mediana viene definita come il valore corrispondente alla frequenza relativa cumulata pari a 0.5 (ovvero il 50-esimo percentile):

$$M(x) = \begin{cases} \frac{x_{n+1}}{2} & \text{se } n \text{ dispari} \\ \frac{x_{\frac{n}{2}} + (x_{\frac{n}{2}+1})}{2} & \text{se } n \text{ pari} \end{cases} \quad (4.6)$$

Da questa definizione si nota che, a differenza della media, la mediana non dipende dal valore assunto dal resto dei dati ed è quindi poco sensibile agli outlier estremi.

Applicando la formula 4.6 all'insieme ordinato $\mathbf{x} = [x_1, x_2, \dots, x_n]$ delle energie giornaliere del mese otteniamo il valore della mediana $M(x)$, che ci permette di ricavare il vettore $\mathbf{d}$ delle differenze assolute tra i campioni e la mediana.

$$\mathbf{d} = |x_i - M(x)| \quad (4.7)$$

L'informazione di interesse del vettore $\mathbf{d}$ è ancora la mediana $M(D)$, che moltiplicata per il coefficiente b , che tiene conto della distribuzione dei dati, permette di ottenere il valore di MAD.

$$MAD = bM(D) \quad (4.8)$$

Il criterio di reiezione degli outlier viene affidato a un multiplo "C" del MAD, il cui valore non è noto a priori ma deve essere determinato attraverso l'individuazione del ginocchio (Figura 4.5), partendo da 1 MAD fino a 5 MAD con incrementi di 0.5.

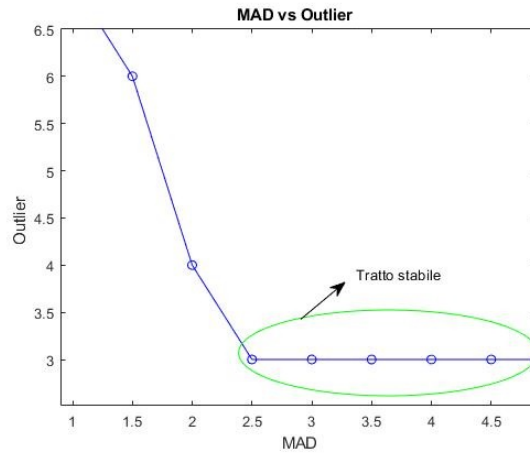


Figura 4.5: Ginocchio del secondo mese, utente 1.

Nella Figura 4.5 possiamo vedere come aumentando la soglia il numero di outlier diminuisca rapidamente, segno che questi punti sono ancora molto vicini ai punti di partenza, fino a quando il fattore moltiplicativo ' C ' diventa 2.5, tratto oltre il quale il numero di outlier rimane fisso, indicando la lontananza dei punti rimanenti che possono essere quindi classificati come outlier. Il valore di C delimiterà quindi la zona di normalità:

$$UL = M(x) + C * MAD \quad (4.9)$$

$$LL = M(x) - C * MAD \quad (4.10)$$

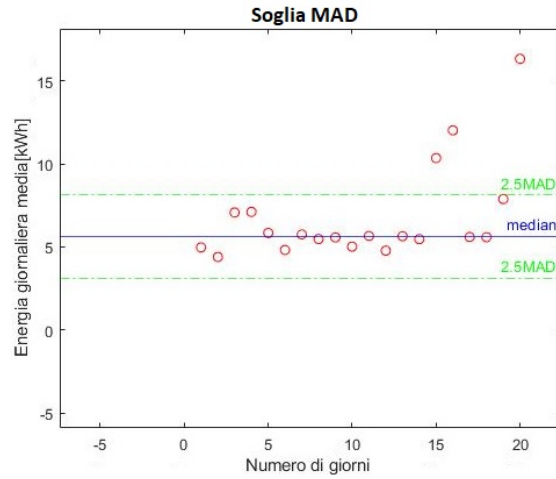


Figura 4.6: Soglia nel caso del primo utente, mese di febbraio.

Come nel caso di prima la soglia non l'abbiamo impostata fissa per tutti i mesi, ma varia in base al mese considerato.

4.2.3 DBSCAN

Nel primo capitolo avevamo presentato il DBSCAN solo come algoritmo di clustering, ma molti algoritmi di clustering possono essere utilizzati per identificare le anomalie per via della loro capacità di raggruppamento, infatti le anomalie potrebbero dar luogo a cluster singoli o non appartenere a nessuno di essi.

Essendo che ci siamo riportati a uno spazio dimensionale ridotto, con le energie, è possibile applicare questo algoritmo di clustering. La procedura che adotta questo metodo è già stata illustrata nel primo capitolo ed è riassunta nella Figura 4.7.

L'idea principale del DBSCAN è che i punti dei cluster risiedono nelle aree dense mentre gli outlier sono lontani da esse. La difficoltà quindi si trova nel dover impostare correttamente i due parametri " Eps " (raggio) e " $Minpts$ " (vicini), che definiscono la densità minima e l'estensione che l'area deve avere.



Figura 4.7: Passaggi elementari del DBSCAN.

Per impostare questi due parametri abbiamo utilizzato la procedura proposta da Matlab in quanto si utilizza l'algoritmo implementato nella versione r2021b.

Il numero MinPts deve essere scelto in modo che sia maggiore o uguale al numero delle dimensioni dei dati +1. Come valore minimo quindi si ha che MinPts deve essere almeno 2 (perché con le energie utilizziamo solo una caratteristica per giorno), in questo caso però si è deciso di adottare un valore pari a 4 in quanto prendendo in considerazione un mese per volta si ritiene che una condizione di normalità sufficiente viene raggiunta se i valori di energia sono simili almeno per 5 giorni (settimana lavorativa).

Il raggio di ricerca viene impostato utilizzando il valore di MinPts scelto precedentemente. Per ciascun punto dei dati viene calcolata la distanza dal quarto (derivante dal valore di MinPts) punto più vicino.

Queste distanze verranno visualizzate nel grafico della k -esima distanza (Figura 4.8) e si individuerà il valore del ginocchio della curva come Eps da usare. Questo perché il quarto punto più vicino rappresenta la condizione minima per la formazione di un cluster attorno a un punto *core*, se la distanza da questo punto aumenta considerevolmente allora è molto probabile che ci si avvicini al campo delle anomalie.

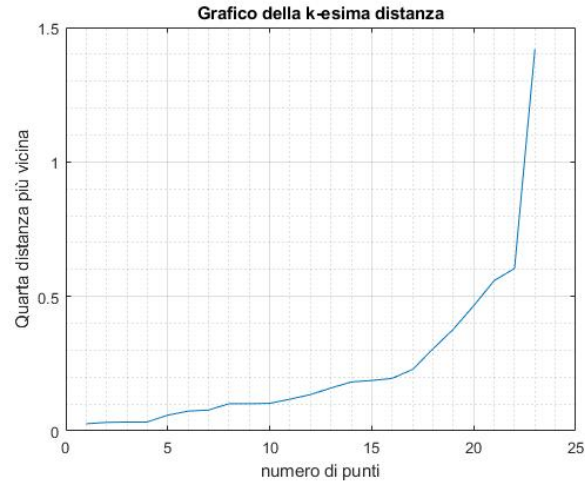


Figura 4.8: Grafico Distanze nel mese di febbraio, utente 1.

Ricavati i parametri di interesse abbiamo applicato la procedura di DBSCAN a tutti i mesi degli utenti ottenendo risultati simili a quelli riportati nella Figura 4.9.

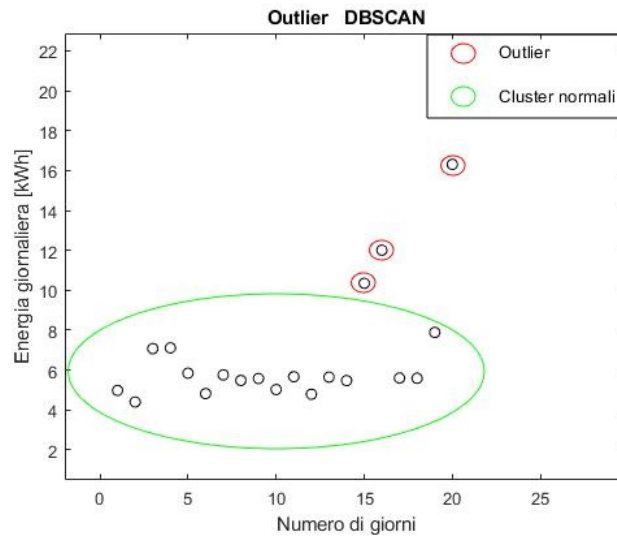


Figura 4.9: Cluster e outlier nel mese di febbraio, utente 1.

Anche in questo caso il valore del ginocchio non sarà lo stesso per tutti ma cambierà in base all'utente e al mese considerato.

4.2.4 Cumulative empiriche

Con questo approccio torniamo nello spazio dimensione iniziale, ovvero trattiamo di nuovo vettori con 48 feature. L'obiettivo principale è quello di andare a confrontare le curve di

carico giornaliero del mese tra loro, attraverso l'utilizzo delle cumulative empiriche. La Cumulativa empirica è una funzione di distribuzione generata dai campioni misurati che indica come sono distribuiti i consumi durante la giornata, ed è definita formalmente come:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(x_i \leq x) \quad (4.11)$$

dove I è una funzione indicatore, che assume il valore pari a 1 quando l'evento $x_i \leq x$ si è verificato, altrimenti è zero.

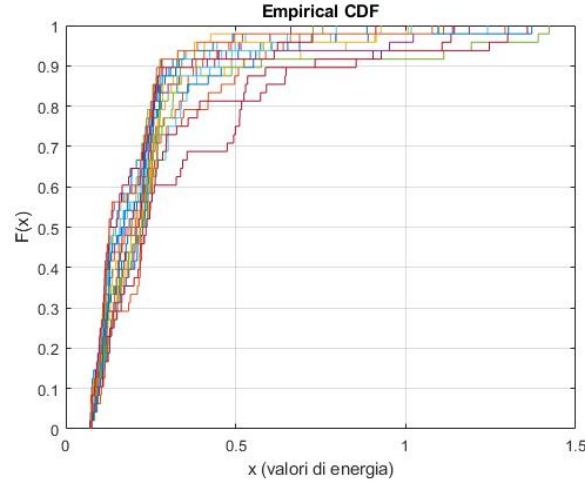


Figura 4.10: Esempi di distribuzioni.

Osservando le cumulative possiamo capire qual è la probabilità che si abbia un valore minore o uguale a x (valore di energia nel nostro caso).

Per la determinazione degli outlier si sono seguiti due approcci diversi:

- Distanze euclidee tra vettori.
- Kolmogorov-Smirnov test (KS-test).

Distanza euclidea

Un primo tentativo per andare a determinare le anomalie del mese è stato quello di utilizzare le distanze euclidee tra vettori.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\frac{1}{D} \sum_{d=1}^D (\mathbf{x}_{i,d} - \mathbf{x}_{j,d})^2} \quad (4.12)$$

dove:

- $\mathbf{x}_i$ e $\mathbf{x}_j$ rappresentano i due vettori

- D indica le dimensioni dei 2 vettori

Per poter eseguire questo calcolo si è deciso di mettere le cumulative in modo da avere sull'asse verticale il valore delle energie normalizzato rispetto al valore massimo del mese e sull'asse orizzontale il numero di feature, Figura 4.11.

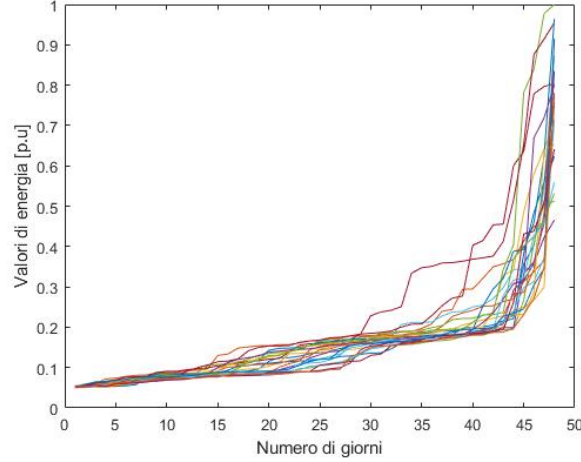


Figura 4.11: Esempi di distribuzioni con asse verticale corrispondente alle ampiezze normalizzate.

La distanza di soglia è stata ricavata in modo empirico, ovvero una volta calcolate le distanze tra tutte le cumulative si è andati a individuare le distanze maggiori per i primi 20 utenti. La soglia successivamente è stata individuata mediando i valori. Tale valore rimarrà fisso per tutti gli utenti in quanto le ampiezze delle cumulative sono state normalizzate rispetto al valore massimo.

Nella Figura 4.12 è riportata una parte della matrice delle distanze tra le varie cumulative, dove sono evidenziate le curve con le distanze più grandi (colonne 12 e 14) che andranno poi a contribuire alla media che poi sarà adottata come soglia.

11	12	13	14	15
0.2787	0.8665	0.2496	0.9848	0.4235
0.4794	0.8113	0.4538	0.7941	0.3437
0.2510	1.0926	0.1409	1.1876	0.5121
0.3960	1.0498	0.2661	1.1391	0.5985
0.3394	1.0147	0.2777	1.1307	0.5867
0.2857	0.8158	0.3295	0.8643	0.2547
0.9257	0.5234	0.9629	0.3952	0.6322
0.2076	0.7719	0.2873	0.9279	0.3392
0.2501	0.9657	0.1774	1.0191	0.3975
0.4474	0.6016	0.4743	0.7406	0.3142
0	0.9217	0.1837	1.0625	0.4104
0.9217	0	1.0187	0.5008	0.6564
0.1837	1.0187	0	1.1228	0.4775
1.0625	0.5008	1.1228	0	0.7665
0.4104	0.6564	0.4775	0.7665	0
0.1889	0.8881	0.2435	1.0061	0.2975
0.3475	0.9436	0.3160	1.0306	0.4418
0.4700	0.6127	0.5343	0.7512	0.3204
0.4158	0.9034	0.3223	0.9744	0.4771
0.3190	0.9512	0.2423	1.0509	0.5032
0.2503	0.8550	0.2573	1.0020	0.4574
0.2011	1.0253	0.2156	1.1232	0.4375
0.1922	1.0989	0.1344	1.2067	0.5440

Figura 4.12: Parte della matrice con le distanze euclidee che presenta le distanze maggiori.

KS-test

Il KS-test è un test non parametrico che permette il confronto tra due distribuzioni senza la necessità di fare ipotesi sul tipo di distribuzione che si sta trattando. Si possono avere due tipologie di test.

- KS-test a una coda, in questo caso il confronto avviene tra una distribuzione empirica e una distribuzione teorica di riferimento, permettendo di determinare la tipologia di distribuzione con la quale si ha a che fare.

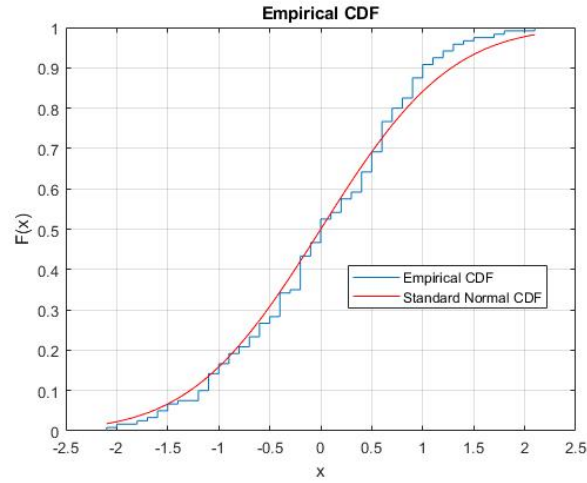


Figura 4.13: Esempio di test a una coda.

- KS-test a due code, per il confronto tra due distribuzioni empiriche, l'informazione che si vuole determinare è se le due distribuzioni appartengono alla stessa tipologia indipendentemente da quale essa sia.

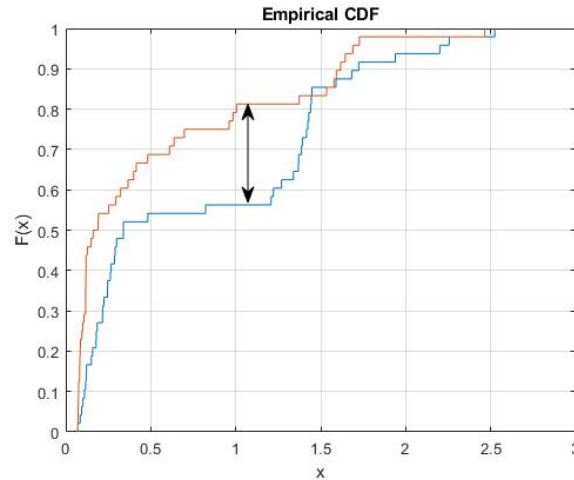


Figura 4.14: Esempio di test a due code.

In ogni caso si deve verificare se i dati provengono dalla stessa distribuzione, che rappresenta anche l'ipotesi nulla H_0 :

$$H_0 : F(x) = F_0(x) \rightarrow \forall x$$

L'ipotesi alternativa è che entrambi i dati appartengono a distribuzioni diverse H_1 :

$$H_1 : F(x) \neq F_0(x) \rightarrow \text{per qualche } x$$

Noi utilizzeremo il test a due code in quanto ciò che si vuole determinare è se le distribuzioni dei dati possono essere considerate simili. L'accettazione o il rifiuto della ipotesi H_0 viene determinata in base alla distanza massima tra le due. Date due distribuzioni $F_{1,n}(x)$ $F_{2,m}(x)$, ricavate da due insiemi di dati aventi rispettivamente n ed m campioni, si definisce la distanza D tra due cumulative come:

$$D_{n,m} = \max |F_{1,n}(x) - F_{2,m}(x)| \quad (4.13)$$

Occorre definire anche il valore critico $C(\alpha)$:

$$C(\alpha) = \sqrt{-\ln\left(\frac{\alpha}{2}\right)\frac{1}{2}} \quad (4.14)$$

che dipende dal livello di significatività che si vuole avere. Tipicamente i valori di $C(\alpha)$ sono tabulati in base al valore del livello di significatività. Noi useremo le tabelle implementate da Matlab.

Noti questi due valori, la reiezione della H_0 avviene nel caso in cui la distanza $D(n, m)$ superi il valore critico $C(\alpha)$:

$$D_{n,m} > C(\alpha) \sqrt{\frac{n+m}{nm}} \quad (4.15)$$

Nel caso del nostro dataset dobbiamo verificare la similarità tra circa 23 ECDFs, due per volta, e si andrà a vedere per quante di esse l'ipotesi nulla verrà rifiutata. Essendo che stiamo trattando cumulative legate al consumo di energia residenziale, che in ogni caso presenteranno un po' di variazione di giorno in giorno, si è deciso di adottare un livello di significatività pari a 0.005 per determinare gli outlier più significativi e quindi con distribuzione molto diversa rispetto al resto. Le anomalie saranno identificate se per una curva l'ipotesi nulla è stata rifiutata per la maggior parte delle curve.

4.2.5 Risultati

Per confrontare il comportamento dei 4 metodi abbiamo riportato il numero di anomalie totali trovate nell'anno 2013 per 20 utenti, presi in ordine sparso, in ciascun dataset (Tabelle 4.1 e 4.2). Questi numeri ci permettono di avere una idea sul comportamento globale del metodo, ad esempio numeri più bassi possono indicare dei metodi molto tolleranti verso gli outlier e quindi non del tutto affidabili, oppure metodi con numeri elevati possono indicare una innecessaria rigidità nel classificare gli outlier. I dati riportati nelle tabelle saranno integrati da immagini che riportano il comportamento dei 4 metodi più nel dettaglio.

LCL dataset

Nel caso di questo dataset Z-score e MAD hanno riportato risultati molto simili in termini di outlier totali, con un massimo che si aggira attorno alla decina per entrambi. Per quanto riguarda gli altri tre metodi, questi presentano numeri molto più grandi (più del doppio

Tabella 4.1: Classifica dei valori massimi.

Numero di outlier per metodo					
Utente	Z-score	MAD	DBSCAN	Distanza euclidea	KS-test
1	3	3	12	3	13
2	2	3	7	13	9
3	9	9	12	17	20
4	0	0	8	24	15
5	1	1	12	23	16
6	2	5	9	15	38
7	4	4	14	17	11
8	7	10	16	31	24
9	8	8	17	30	21
10	10	12	18	31	23
11	8	6	13	22	16
12	4	10	29	12	17
13	5	6	17	18	11
14	2	5	20	14	12
15	5	7	14	17	9
16	2	2	16	6	15
17	2	8	19	36	12
18	8	12	21	30	24
19	4	6	19	9	6
20	4	3	25	46	17

nella maggior parte degli utenti), soprattutto nel caso della distanza euclidea che presenta i valori massimi nella quasi totalità dei casi.

In modo da dare un maggior senso ai dati in tabella, riportiamo anche le curve anomale identificate da ciascun metodo in casi di utenti con comportamento molto diverso, da Figura 4.15 a 4.20.

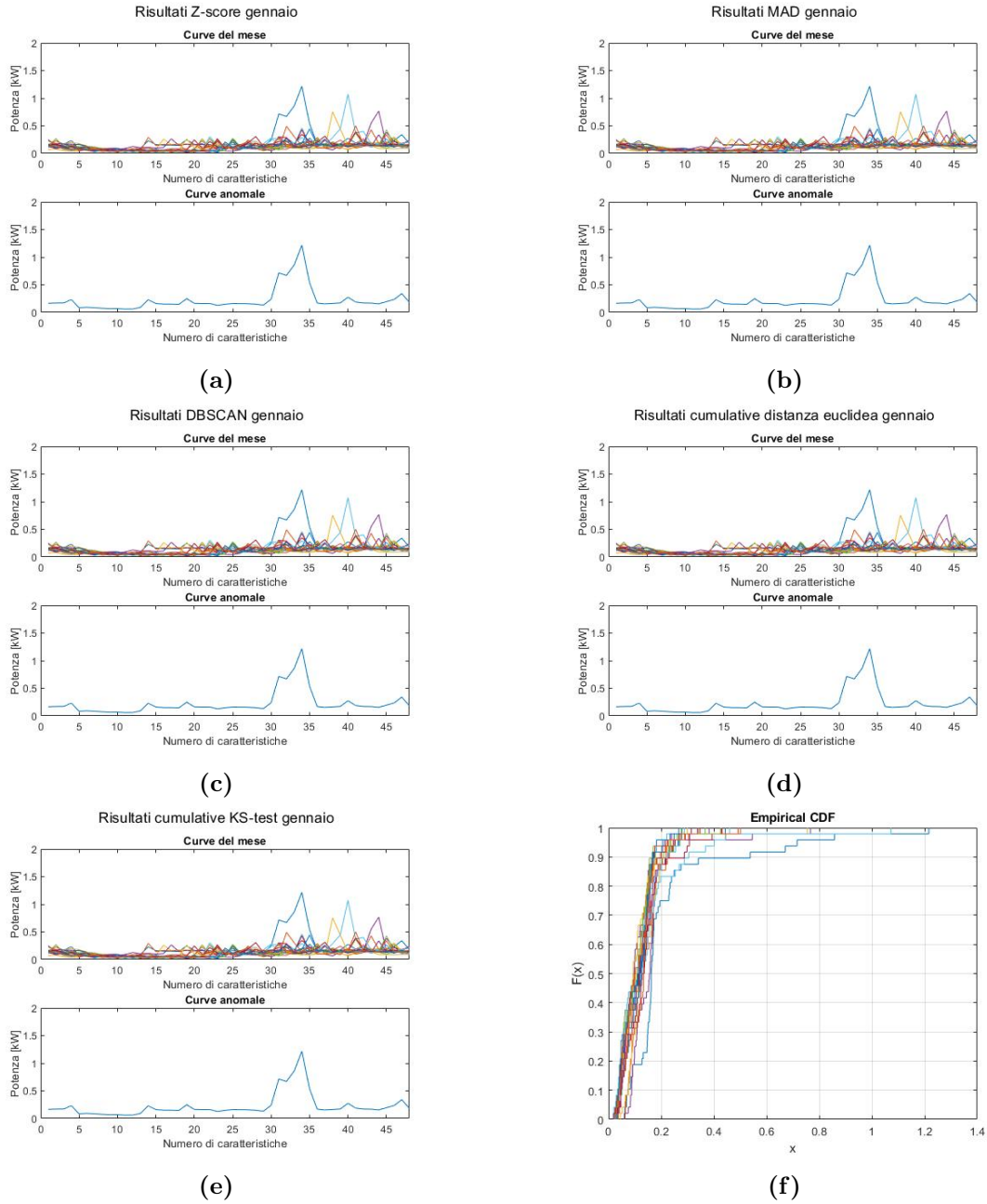


Figura 4.15: Confronto tra le anomalie riscontrate.

In caso di outlier estremi, Figura 4.15, possiamo notare che tutti i metodi sono riusciti nella identificazione degli outlier senza problemi.

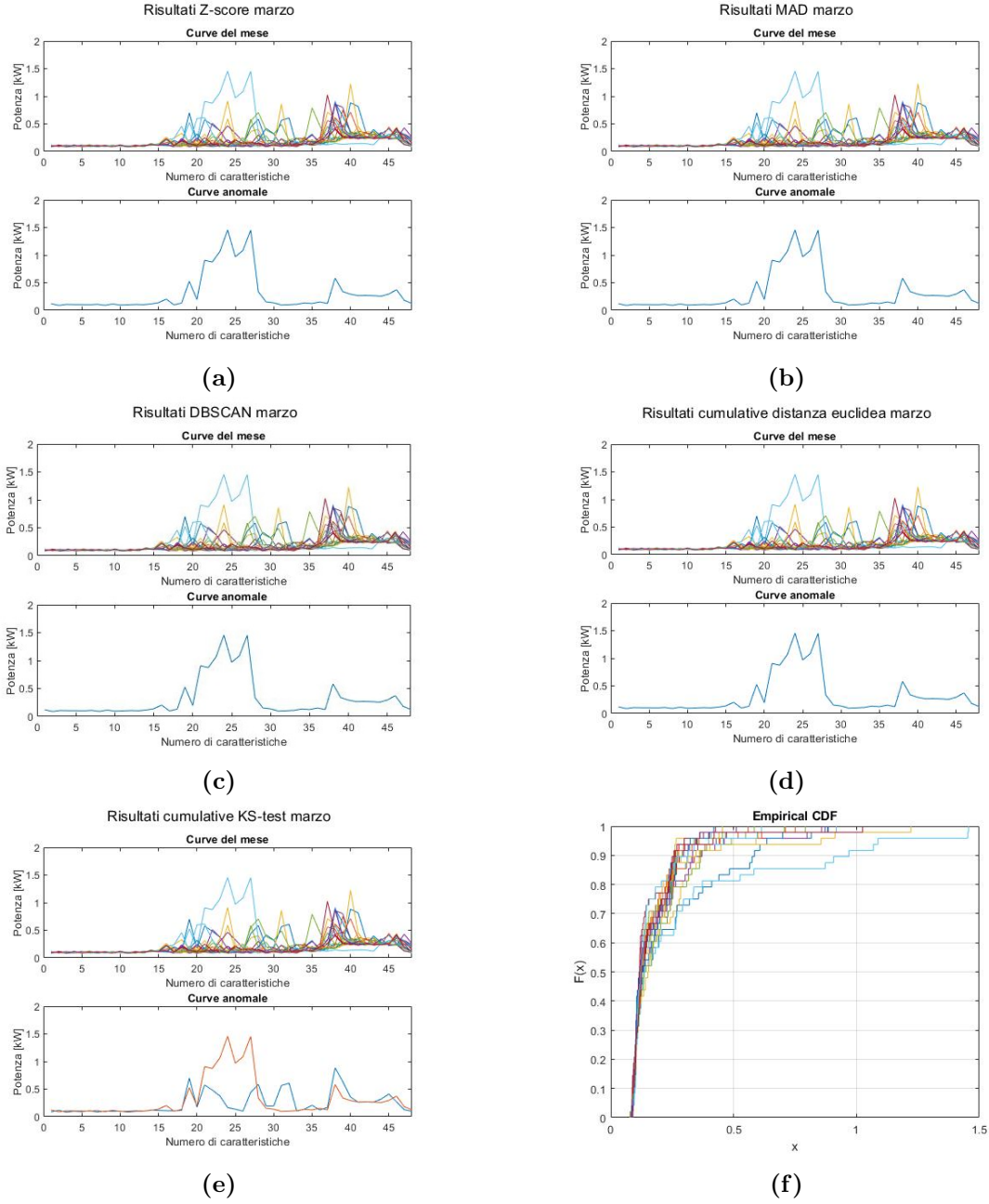


Figura 4.16: Confronto tra le anomalie riscontrate.

Un caso simile viene presentato nella Figura 4.16, come è possibile vedere dalle cumulative si evidenzia un secondo consumo energetico il cui andamento si discosta dalla maggioranza, Figura 4.16f, ma solo il KS-test è riuscito nella identificazione. Questo può indicare il fatto che la perdita della informazione dovuta all'utilizzo delle energie influenza l'identificazione delle anomalie, mentre nel caso della distanza euclidea questo può essere giustificabile dalla definizione di distanza euclidea stessa che tiene in considerazione il valore delle ampiezze massime e non solo la forma.

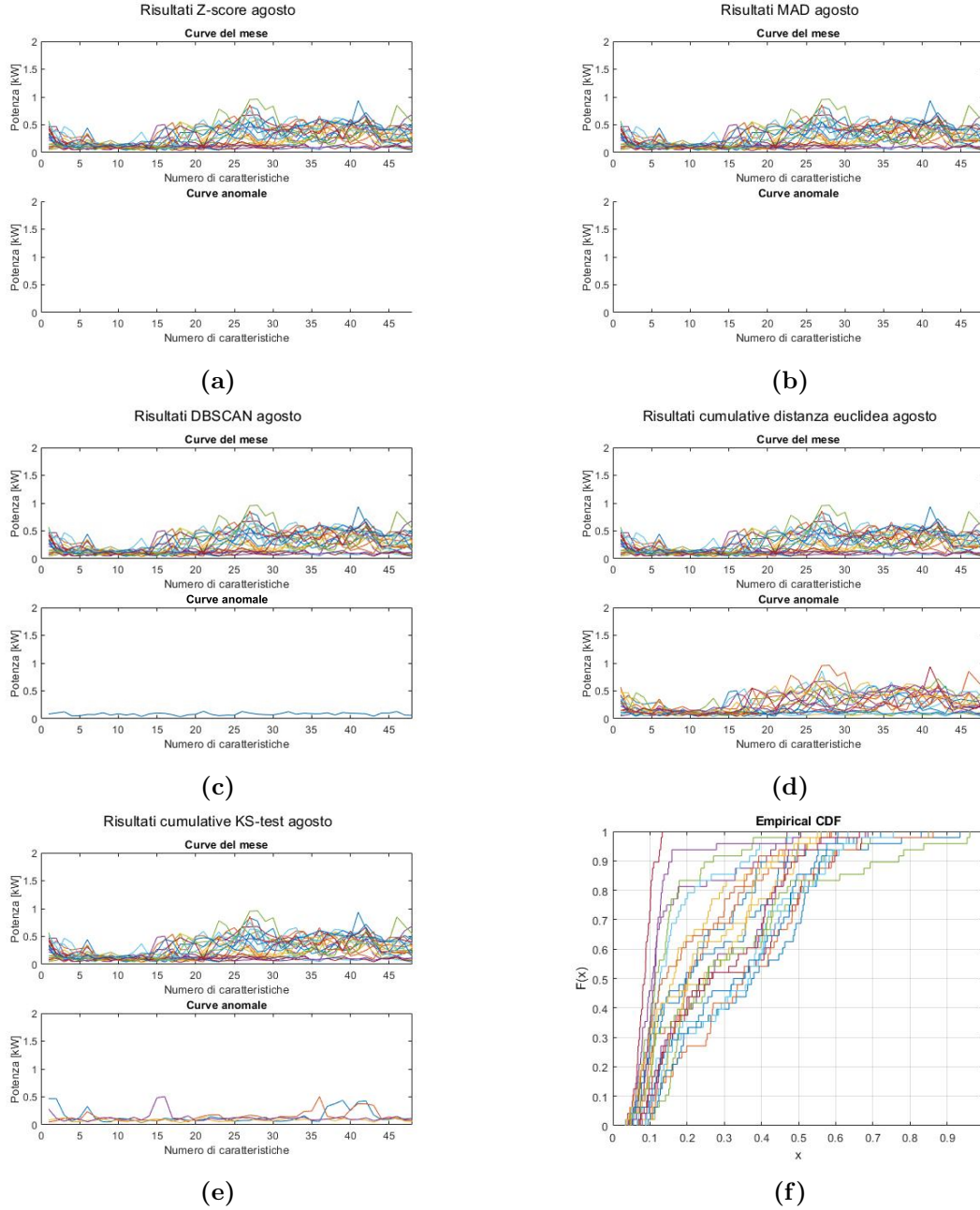


Figura 4.17: Confronto tra le anomalie riscontrate.

Nel caso in cui l'utente presenti dei consumi molto variabili, nelle Figure 4.17 e 4.18 sia Z-score che MAD sono i metodi che hanno più difficoltà nella identificazione delle anomalie. Come è possibile vedere nelle figure indicate, in tutti e tre i casi presentati non hanno identificato nessun outlier. Risultato simile anche per il DBSCAN applicato alle energie totali. Applicando la distanza euclidea alle cumulative invece, sempre nel caso di consumi molto variabili nel mese, si è riscontrato un numero elevato di anomalie, penalizzando molto questa tipologia di utente, Figura 4.17d.

Le cumulative con il KS-test sono quelle che hanno presentato risultati più costanti con questa tipologia di utente, presentando un numero di outlier contenuto i cui consumi si allontanano visibilmente dal resto.

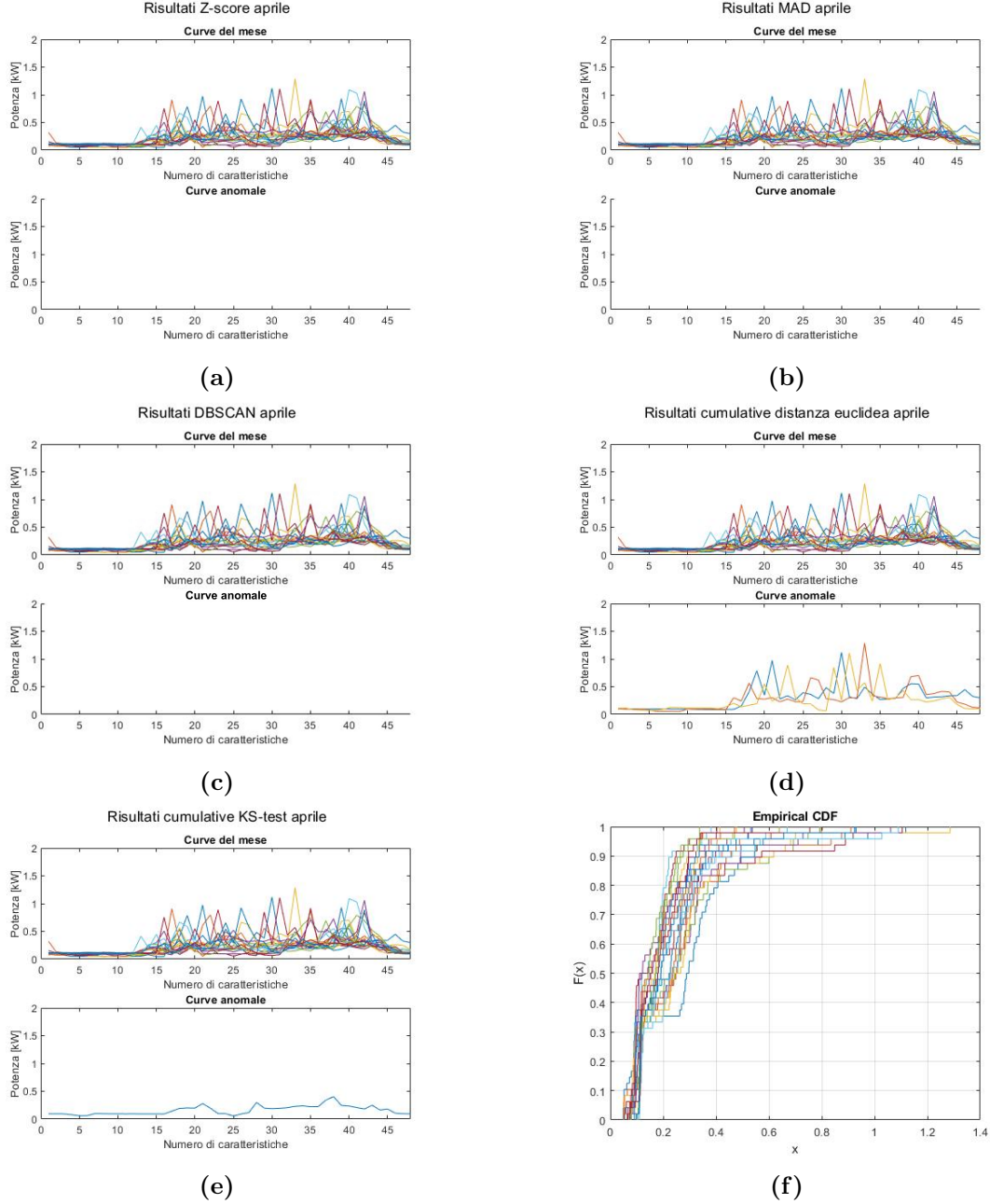


Figura 4.18: Confronto tra le anomalie riscontrate.

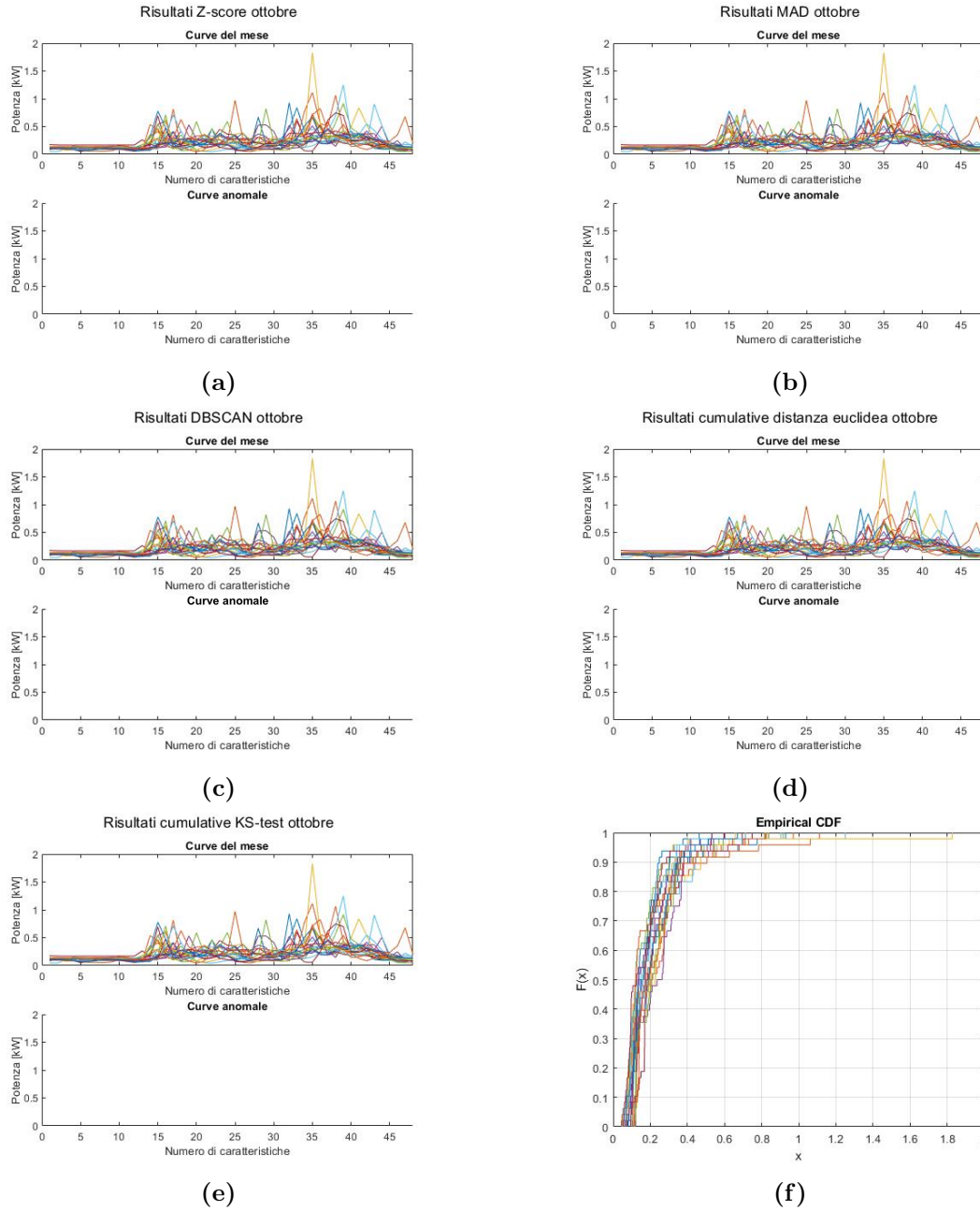


Figura 4.19: Confronto tra le anomalie riscontrate.

Infine per gli utenti con consumi molto regolari (Figura 4.19) tutti i metodi non hanno identificato outlier, cosa che può essere ritenuta ragionevole in quanto in casi simili sia la energia totale che la distribuzione del consumo non presentano grandi variazioni.

Thames Valley dataset

Risultati analoghi sono stati ottenuti anche per il dataset derivante dal progetto *Thames Valley vision* dove sia Z-score che MAD hanno rilevato il minor numero di anomalie, mentre l'analisi delle cumulative con la distanza euclidea ha presentato il maggior numero di anomalie.

Tabella 4.2: Classifica dei valori massimi.

Numero di outlier per metodo					
Utente	Z-score	MAD	DBSCAN	Distanza euclidea	KS-test
1	3	7	19	31	22
2	7	10	17	20	15
3	1	2	23	41	22
4	8	9	27	35	27
5	3	8	12	23	16
6	4	4	13	31	16
7	1	4	15	10	9
8	1	2	12	21	10
9	3	4	13	33	14
10	3	6	11	21	9
11	2	2	8	17	8
12	5	4	15	26	21
13	1	3	20	10	19
14	4	11	21	36	28
15	7	13	19	42	25
16	3	8	12	37	15
17	4	7	16	12	20
18	9	9	28	21	24
19	6	9	12	37	10
20	5	9	13	19	22

Le anomalie identificate da tutti i metodi per tipologie di utenti rilevanti sono riportate nelle Figure 4.20-4.24.

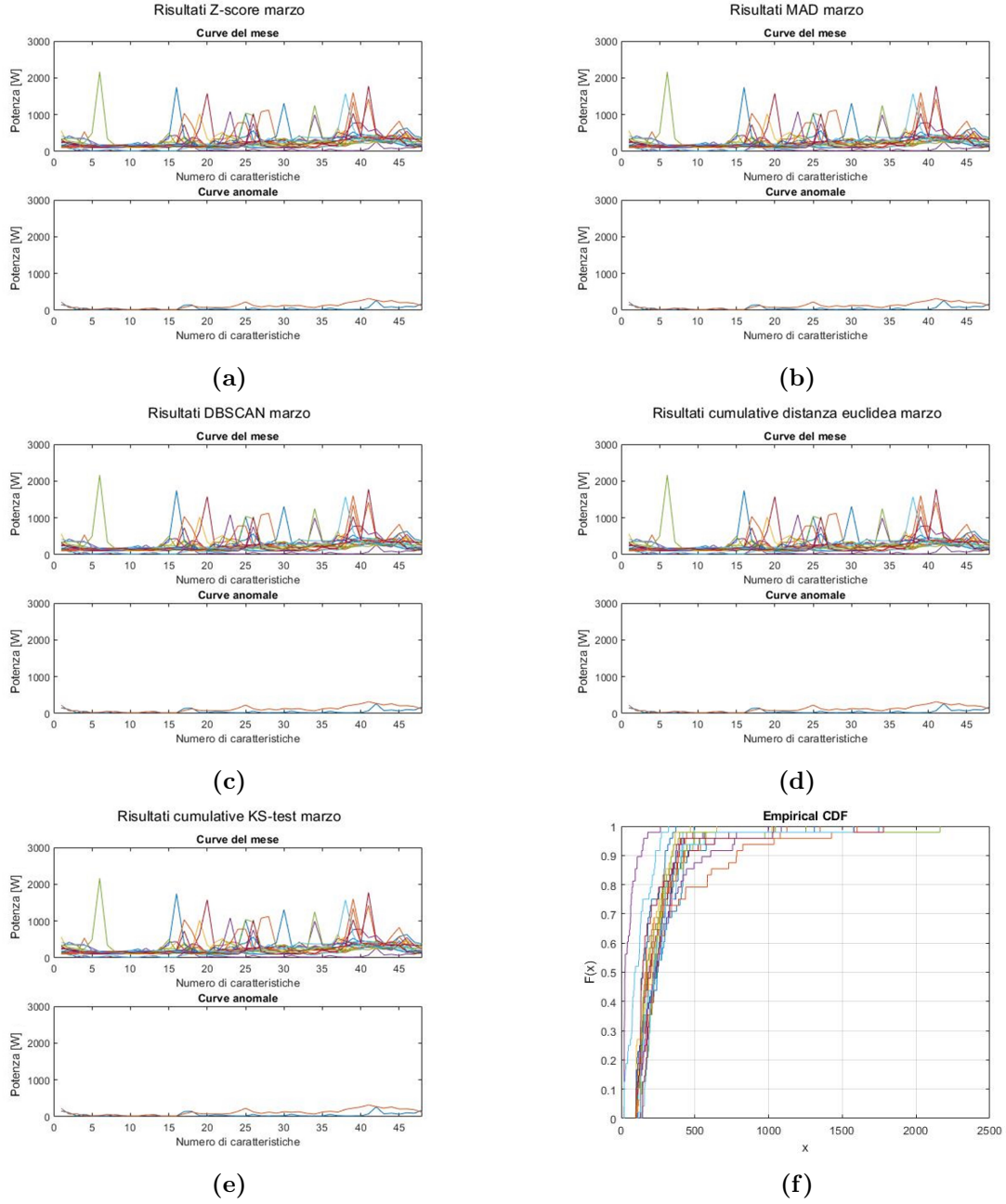


Figura 4.20: Confronto tra le anomalie riscontrate.

Le figure 4.20 e 4.21 mostrano due casi molto distinti di anomalie rilevanti, infatti nel primo caso ci sono due giornate caratterizzate dai consumi molto bassi, mentre nel secondo si apprezza una giornata con consumi superiori alla media. In entrambi i casi tutti i metodi non hanno avuto problemi ad identificare queste anomalie.

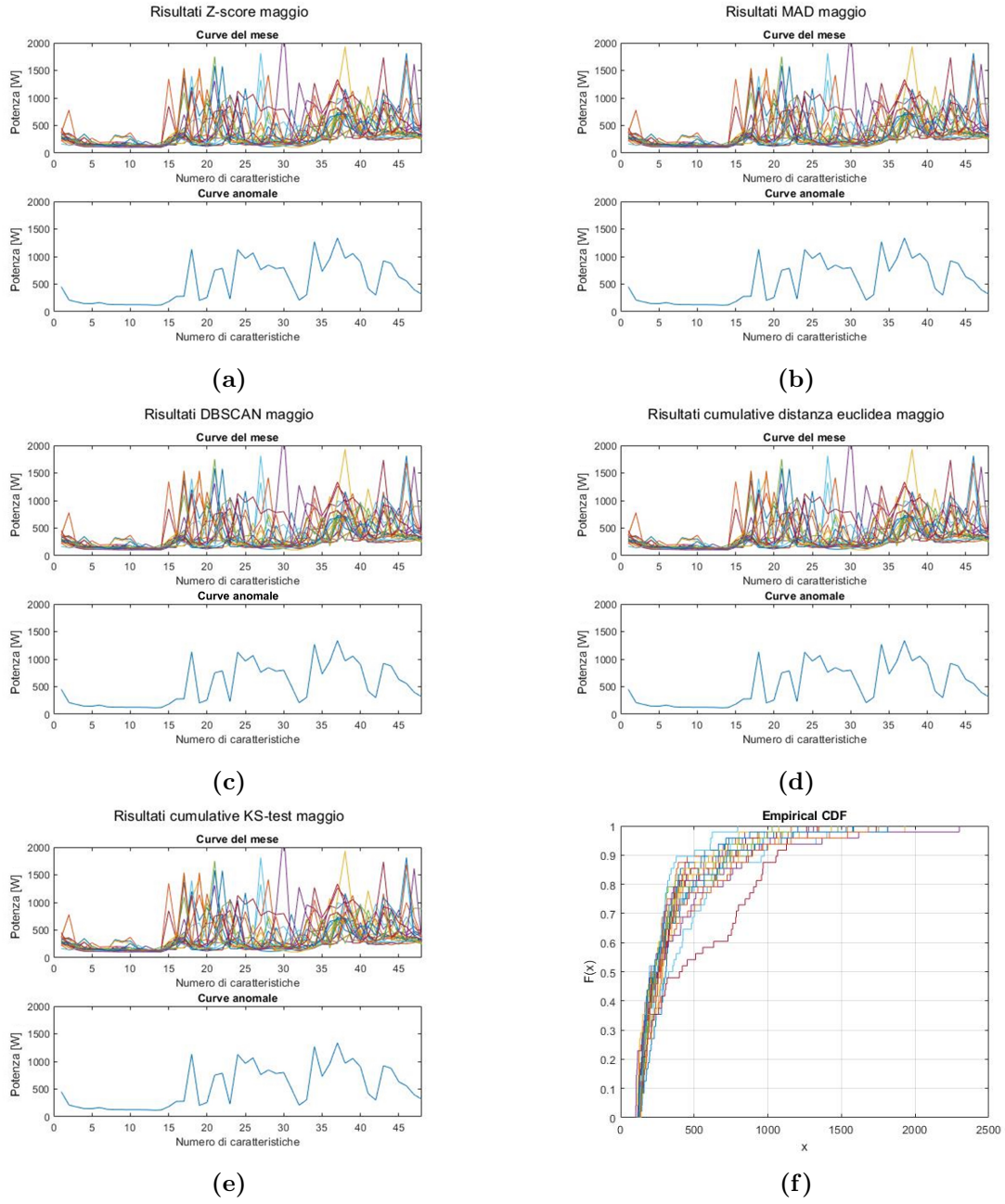


Figura 4.21: Confronto tra le anomalie riscontrate.

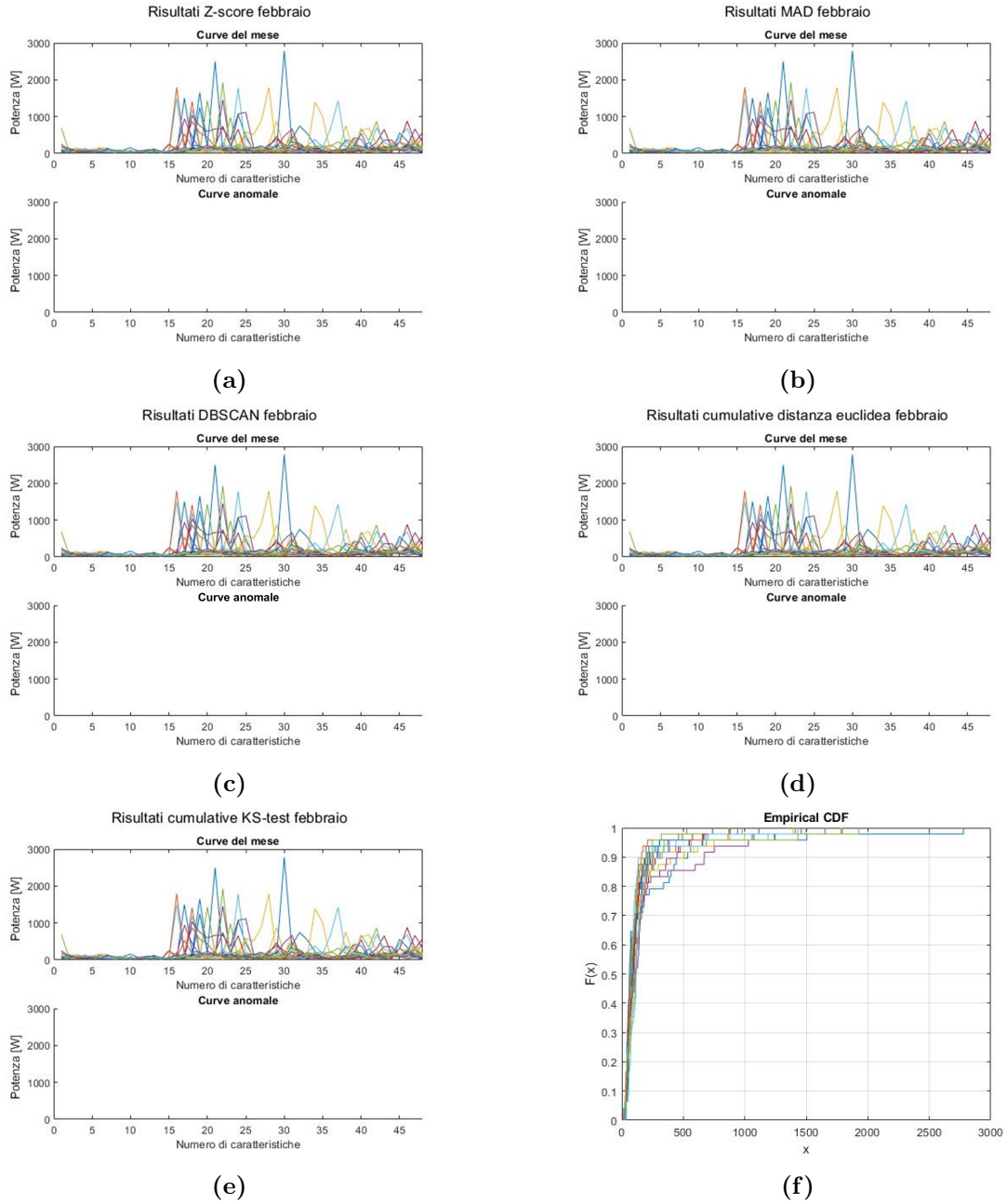


Figura 4.22: Confronto tra le anomalie riscontrate.

Figura 4.22 riporta il caso di un utente con consumi abbastanza regolari, dove nessun algoritmo ha identificato false anomalie.

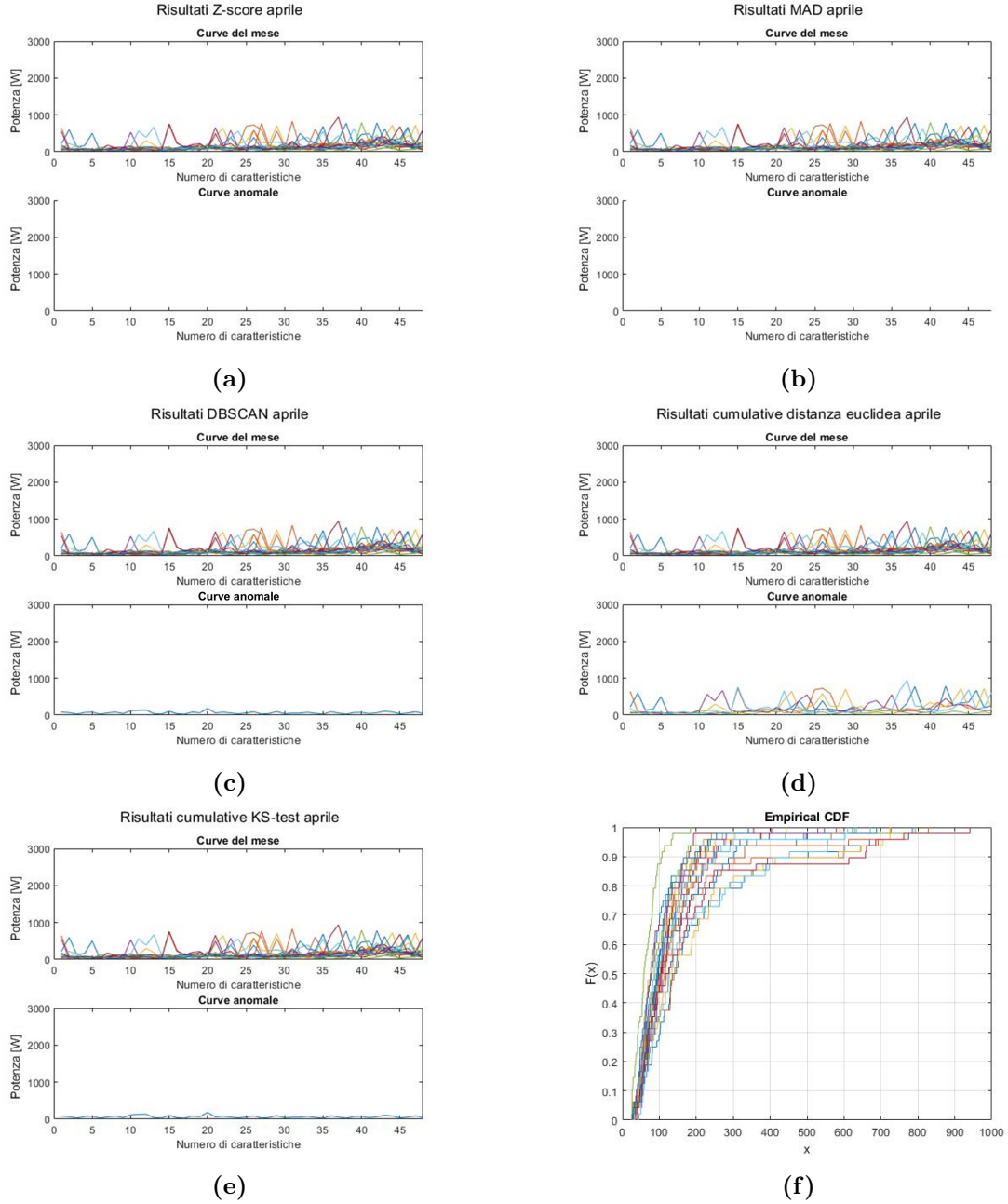


Figura 4.23: Confronto tra le anomalie riscontrate.

Nel caso in cui gli utenti presentino un consumo molto variabile (Figure 4.23 e 4.24) anche in questo caso sia Z-score che MAD non sono riusciti a rilevare outlier, come si nota dalla Figura 4.23f, pur avendo giornate con consumi molto diversi tra loro si può notare una curva che risalta di più rispetto al resto del fascio (la curva più a sinistra), ma questa non è stata rilevata. Sempre nello stesso caso, l'analisi delle cumulative attraverso le distanze euclidee ha avuto un comportamento eccessivo, riportando come outlier quasi la metà delle cumulative, mentre sia DBSCAN sia l'analisi delle cumulative utilizzando

il KS-test l'hanno identificata. Il DBSCAN tuttavia non ha avuto un comportamento molto costante per casi simili. Come si può notare in Figura 4.24, si ha un utente simile al precedente dove la identificazione di outlier è più complicata, in questo caso però ha segnalato un numero elevato di curve che, considerando il tipo di utente, potrebbero essere considerate normali.

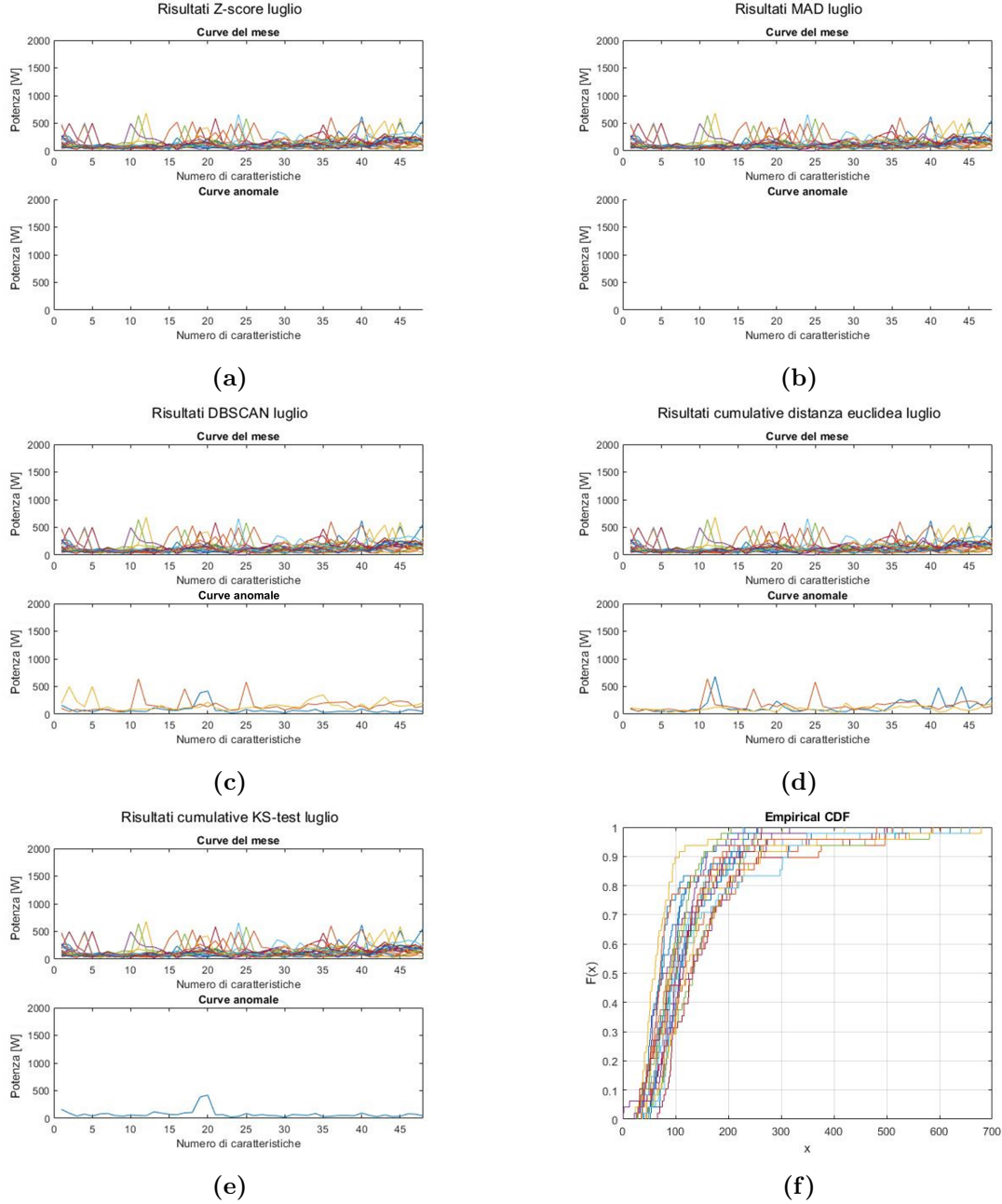


Figura 4.24: Confronto tra le anomalie riscontrate.

Oltre al paragone degli outlier, la Tabella 4.3 presenta il confronto dal punto di vista dei tempi di calcolo, da cui risulta che come ci si aspettava i metodi semplici hanno dei tempi più bassi (di circa un ordine di grandezza), mentre i metodi più complessi si aggirano su valori simili, di questa tipologia il DBSCAN presenta il valore più basso, cosa che si può attribuire all'utilizzo del numero limitato di punti che deve trattare in quanto si utilizzano le energie giornaliere totali.

Tabella 4.3: Tempi medi per ciascun metodo.

Metodo	Tempo medio identificazione outlier [s]
Z-score	0.0019
MAD	0.0025
DBSCAN	0.0194
Cumulative distanza euclidea	0.0543
Cumulative KS-test	0.0429

La Tabella 3.3 riassume i risultati per ciascun metodo.

Tabella 4.4: Riassunto metodi.

Campi di valutazione	Z-score	MAD	DBSCAN	D-euc	KS-test
Costo computazionale	Basso	Basso	Alto	Medio	Medio
Complessità	Bassa	Bassa	Elevata	Media	Media
Dimensioni elevate	no	no	no	sì	sì
Efficacia con i data set proposti	Bassa	Bassa	Media	Bassa	Buona

Capitolo 5

Applicazione di CONDUCTS

5.1 Calcolo dei nove decili

Lo scopo che si vuole raggiungere con questa tesi è determinare con quale algoritmo di clustering la filosofia di CONDUCTS riporta i risultati migliori. La sperimentazione verrà eseguita sui 2 dataset ripuliti nel capitolo precedente in cui si è adottata come finestra temporale le prime 2 settimane del mese di aprile del 2013, solo giornate lavorative, con frequenza di campionamento pari a 30 minuti. Per ogni utente avremo quindi delle *duration curves* composte da $m=480$ punti (dieci giorni composti da 48 punti ciascuno) e i rispettivi vettori delle differenze avente $m-1$ punti. E' importante precisare che entrambi i dataset non forniscono il valore di potenza contrattuale, ragione per cui si è deciso di adottare come fattore normalizzante la potenza massima raggiunta nella finestra temporale.

La Figura 5.1 rappresenta il tipo di CDF delle differenze medie ottenuta in entrambi i casi.

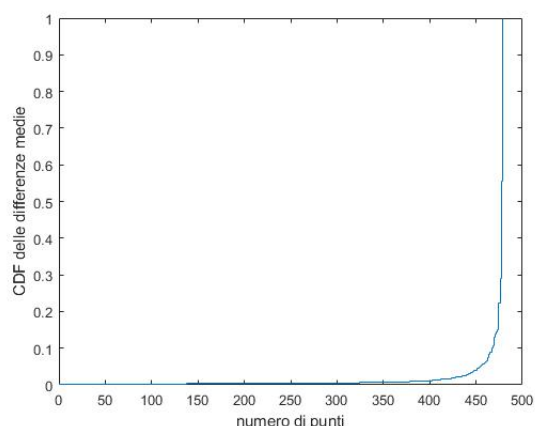


Figura 5.1: CDF delle differenze medie.

L'andamento della CDF con la concavità molto pronunciata come in questo caso può essere attribuito al fatto che la maggior parte degli utenti presenta dei consumi molto bassi durante la maggior parte della giornata e pochi picchi di ampiezza elevata. Purtroppo la CDF con questa forma non permette di estrarre dei decili in grado di rappresentare al meglio gli utenti (in questo caso i punti sarebbero tutti troppo vicini e con valori molto elevati). Considerando questi risultati, si è percorsa un'altra strada per definire la CDF. Data la natura degli utenti che compongono il dataset si è optato di considerare le differenze tra i valori adiacenti del vettore non ordinato comprendente tutti i consumi durante le due settimane. Il risultato viene mostrato in Figura 5.2.

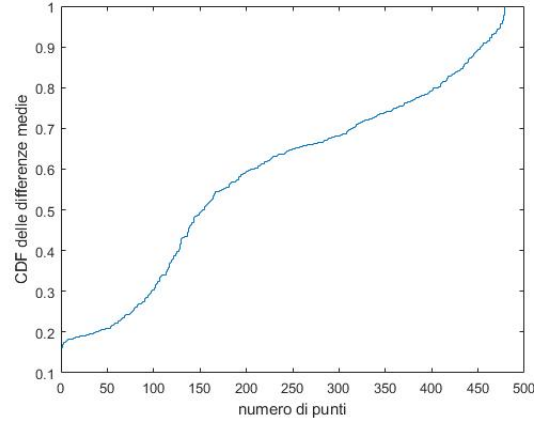


Figura 5.2: CDF delle differenze medie considerando le differenze tra valori adiacenti.

Questa nuova cumulativa mette in evidenza i salti di consumo energetico che si hanno durante la finestra temporale, mentre quella precedente metteva più in luce l'evoluzione durante la finestra temporale. I 9 decili ricavati dalla CDF rappresentati per entrambi i dataset sono riportati di seguito:

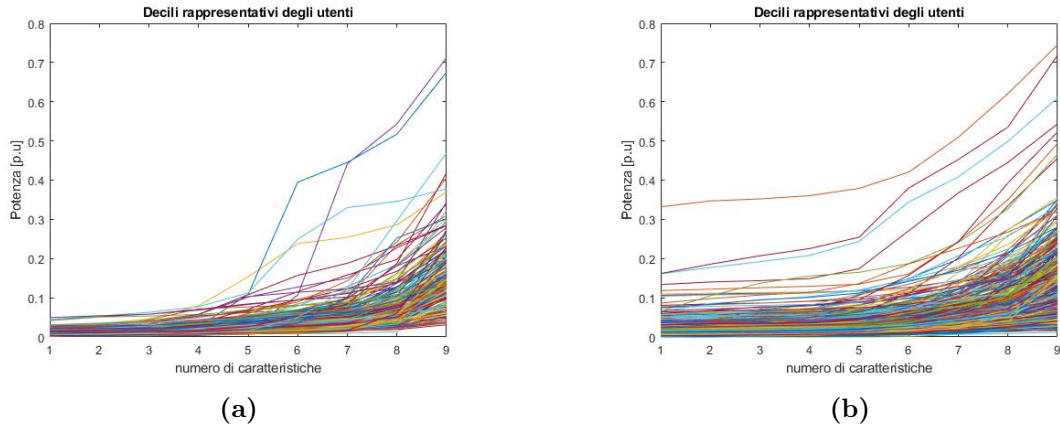


Figura 5.3: Decili rappresentativi degli utenti a) Thames Valley b) Low Carbon London.

I decili rappresentativi di ogni utenti sono i veri input degli algoritmi di cluster. Gli algoritmi con i quali sarà verificata l'efficacia dei decili per rappresentare i clienti sono:

- K-means
- Clustering gerarchico considerando i vari tipi di linkage.
- Spectral Clustering utilizzando due input diversi.

Per tutti i metodi la definizione di distanza è quella euclidea, che nel caso di confronto tra decili è appropriata.

5.2 Scelta del numero K di cluster

Gli algoritmi indicati in precedenza hanno in comune il fatto di richiedere il numero di cluster per poter eseguire la partizione dei dati. Per questo motivo per la deduzione del numero di cluster si è deciso di utilizzare l'indicatore esterno proposto da CONDUCTS, SSE. I risultati per i due dataset presi in considerazione sono riportati nella Figura 5.4.

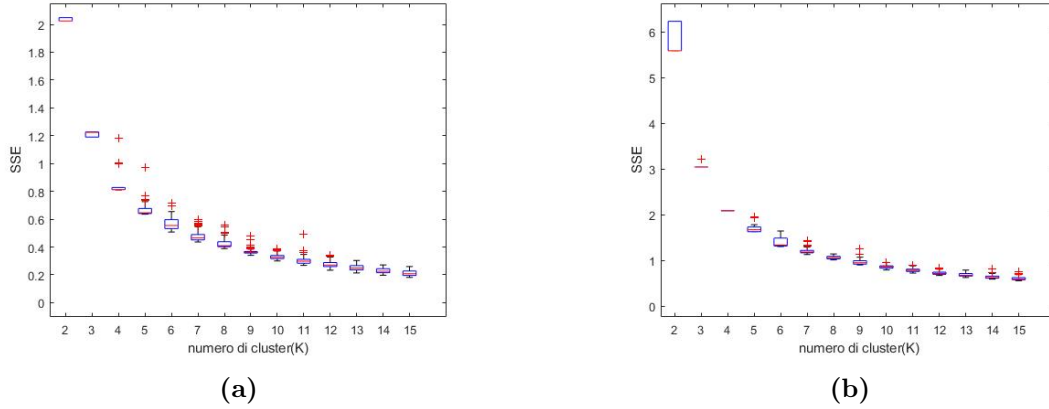


Figura 5.4: Variazione del SSE con il K-means a) Thames Valley Vision b) Low Carbon London.

Per quanto riguarda il K-means abbiamo riportato la variazione degli indicatori in base al numero di cluster all'interno di boxplot, Figura 5.4, in quanto essendo un metodo euristico i punti iniziali del cluster vengono scelti in modo casuale. Per essere certi che la scelta casuale abbia un impatto ridotto sul numero di cluster abbiamo eseguito l'algoritmo per ciascun valore di K per 100 volte, in modo da avere una evoluzione più corretta dell'indicatore. Lo stesso vale per lo spectral clustering, Figura 5.5 e 5.6, che si appoggia al k-means per eseguire la partizione finale dei dati.

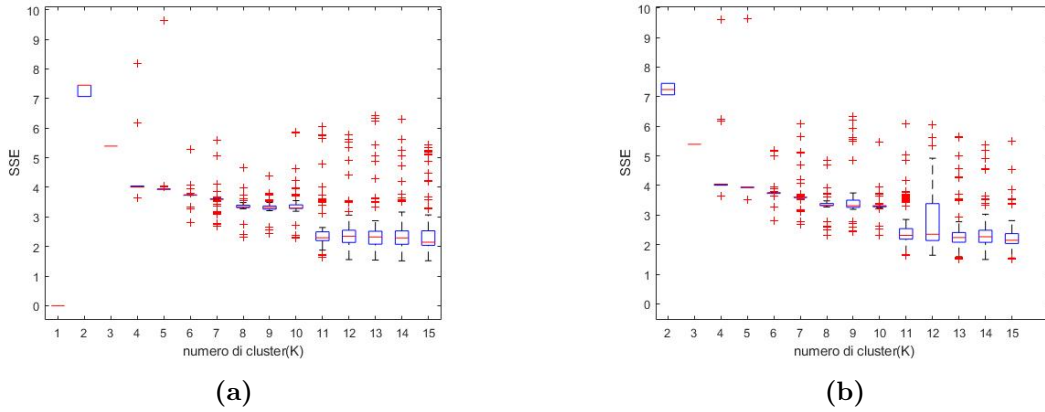


Figura 5.5: Variazione del SSE con lo spectral clustering (S) a) Thames Valley Vision b) Low Carbon London.

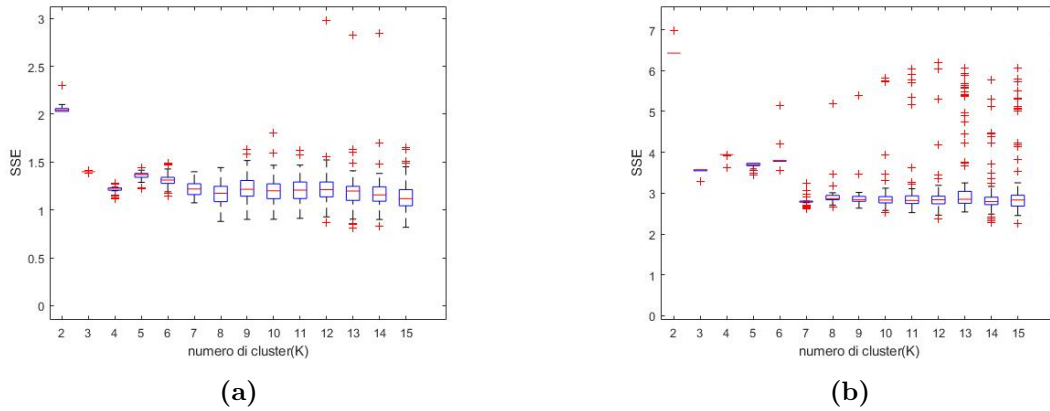


Figura 5.6: Variazione del SSE con lo spectral clustering (L) a) Thames Valley Vision b) Low Carbon London.

Invece per il clustering gerarchico, essendo un metodo deterministico non c'è stato bisogno di fare più cicli per cluster. Dalla Figura 5.7 si nota come per la maggior parte dei metodi, ad eccezione del clustering gerarchico con average linkage, il valore K che determina una minore variazione della compattezza è $K = 6$. Essendo che buona parte degli algoritmi presenta buoni valori di SSE già a K pari a 6 si è deciso di utilizzare questo numero ai fini del confronto.

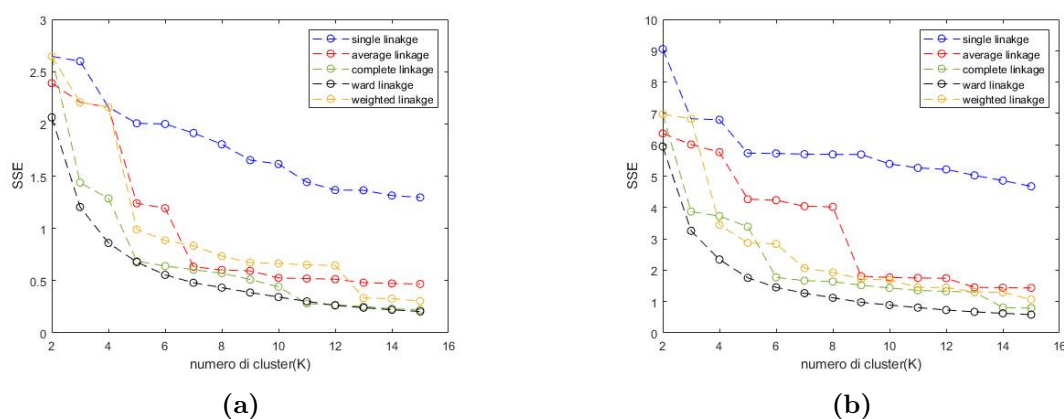


Figura 5.7: Variazione del SSE con il clustering gerarchico a) Thames Valley Vision b) Low Carbon London.

5.3 K-means

I risultati della partizione dei dati utilizzando il K-means sono quelli corrispondenti ai valori più bassi di SSE con $K=6$, trovati usando l'opzione "replicate" implementata da Matlab all'interno del K-means.

Tabella 5.1: Numero di utenti per cluster.

Thames valley dataset	
Cluster	numero di utenti
1	81
2	2
3	2
4	17
5	43
6	58

Low Carbon London dataset	
Cluster	numero di utenti
1	8
2	185
3	31
4	135
5	113
6	3

Tabella 5.2: GSI e ASI.

Thames valley dataset	
GSI	ASI
0.21	0.494

Low Carbon London dataset	
GSI	ASI
0.27	0.526

Dai valori in Tabella 5.1 si evince che la ripartizione degli utenti in entrambi i dataset non è proporzionale, tuttavia entrambi i casi presentano valori di GSI e ASI non

troppo bassi, assicurando la similarità di ciascun componente all'interno dei cluster, cosa che può essere notata anche dai centroidi, Figura 5.8 e 5.9 dove è possibile notare una differenziazione evidente in entrambi i casi.

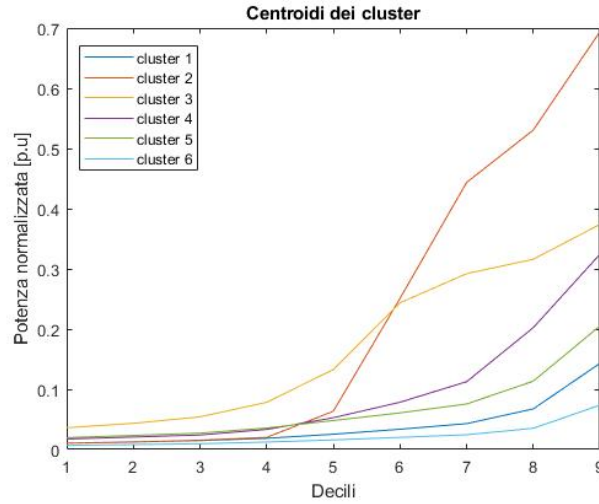


Figura 5.8: Centroidi del Thames Valley dataset.

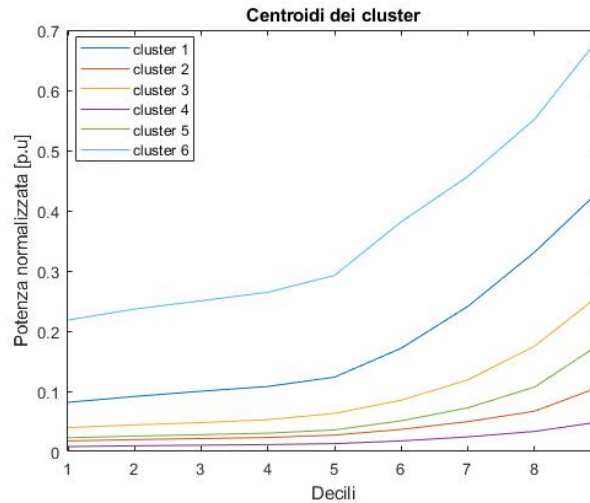


Figura 5.9: Centroidi del Low Carbon London dataset.

Riportando i boxplot dei decili appartenenti ai 6 cluster otteniamo una ulteriore conferma della buona distanza inter-cluster.

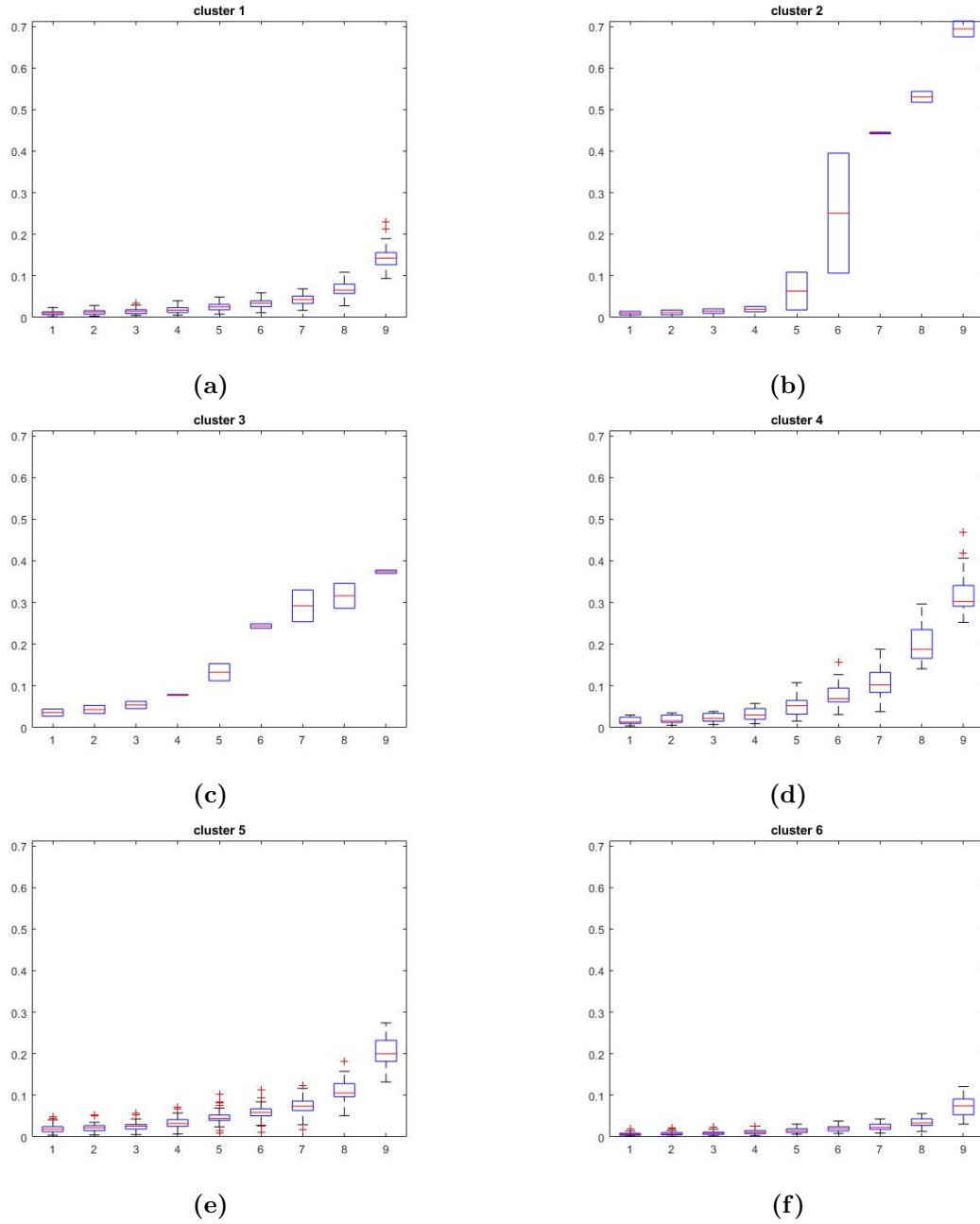


Figura 5.10: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

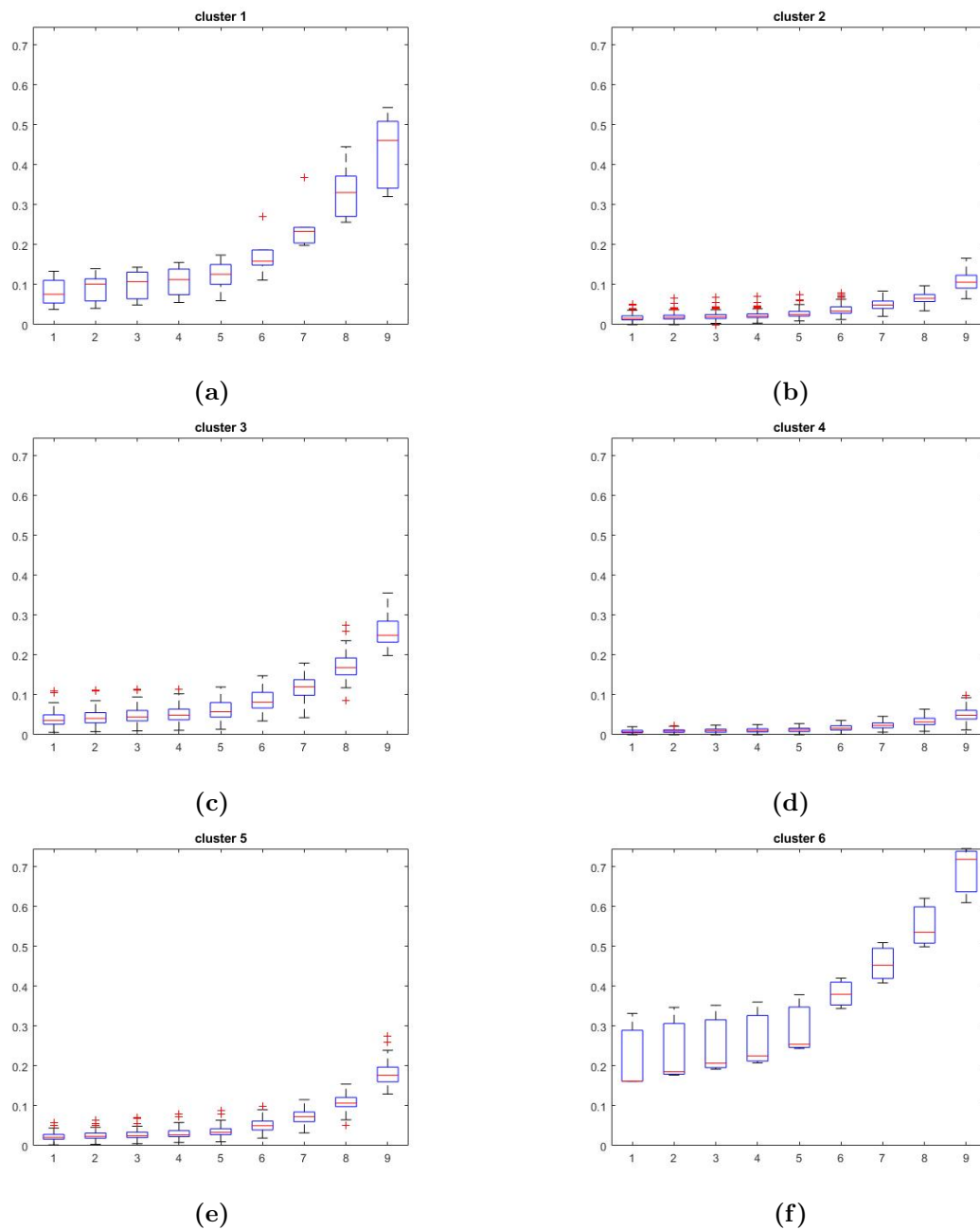


Figura 5.11: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

Nelle Figure 5.12 e 5.13 abbiamo riportato le curve di carico giornaliere appartenenti a utenti dei diversi cluster in modo da verificare come è avvenuta la distribuzione con i veri consumi.

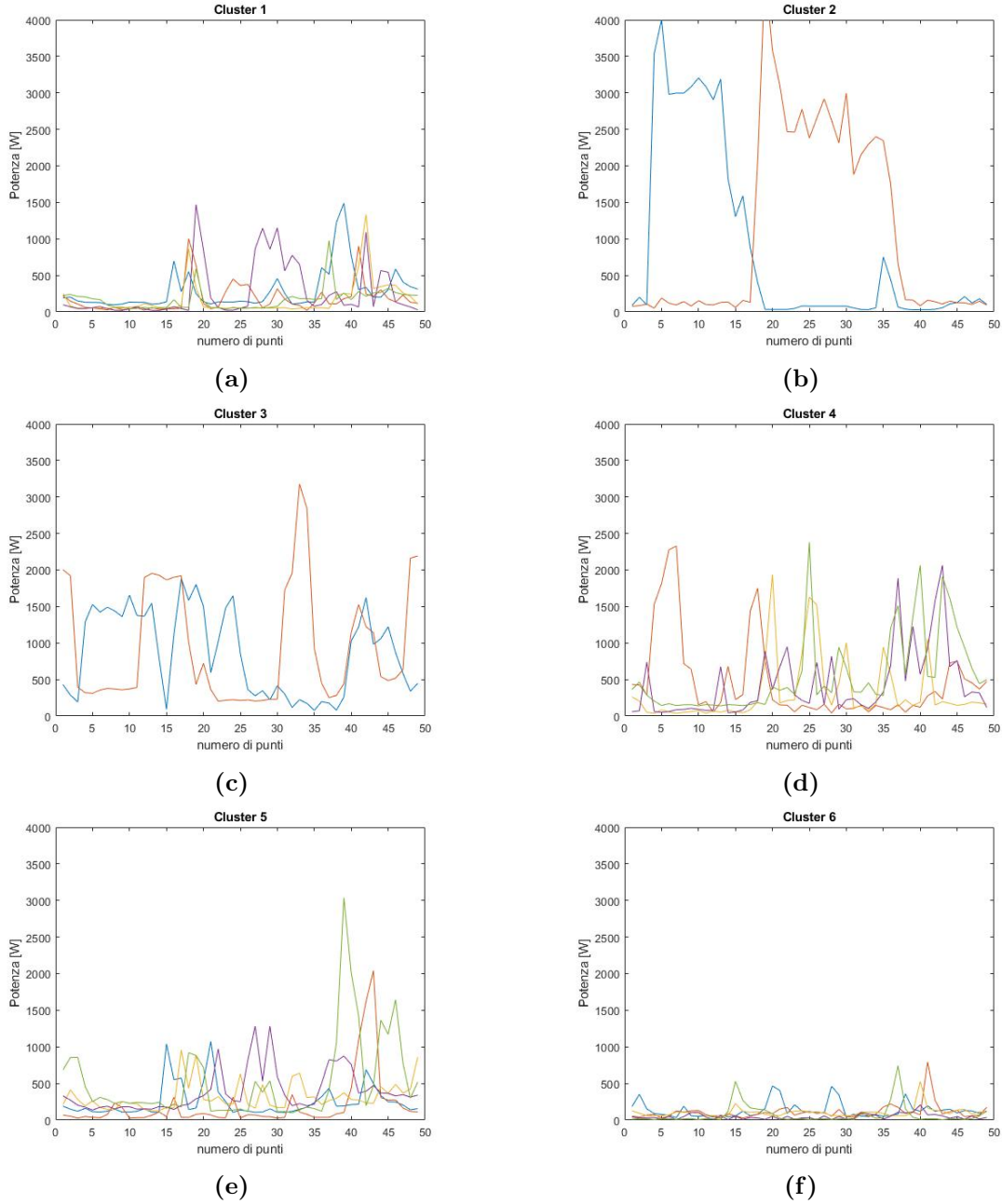


Figura 5.12: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Thames Valley dataset.

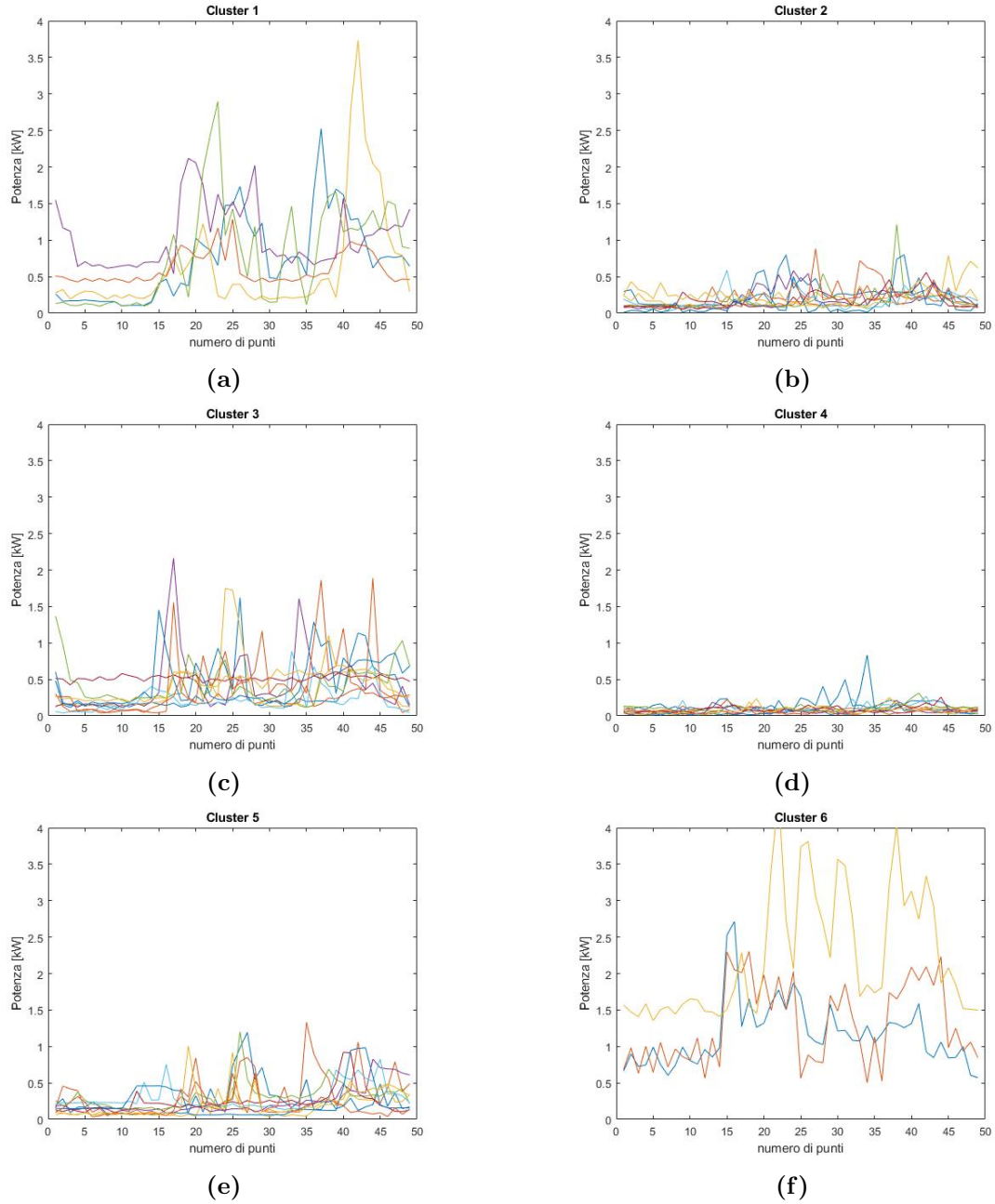


Figura 5.13: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.4 Clustering gerarchico

La tipologia di clustering gerarchico sulla quale si ci concentrerà in questo caso è quella di tipo agglomerativo, come accennato prima, uno dei punti principali di questo algoritmo è la creazione di un albero binario che racchiude tutti i componenti del dataset in una gerarchia di cluster, tuttavia a seconda della tipologia del linkage che si utilizza con questo algoritmo l'albero binario può formare i cluster diversamente.

5.4.1 Single linkage

In questo caso il risultato del clustering non è stato del tutto esaustivo, in quanto, come si può vedere anche dall'albero binario, Figura 5.13, la maggior parte degli elementi si aggregano in modo ordinato verso un solo lato, rendendo poco probabile una buona suddivisione degli utenti.

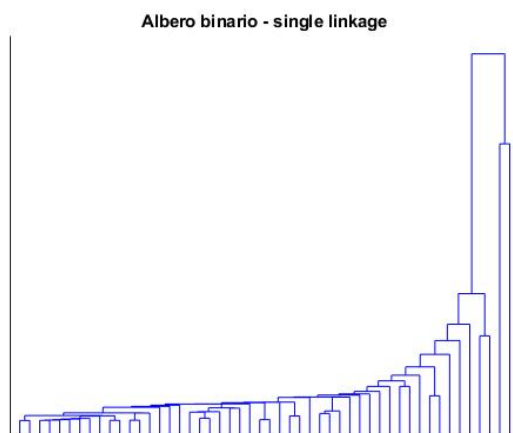


Figura 5.14: Albero binario - single linkage.

Tabella 5.3: Distribuzione degli utenti all'interno dei cluster.

Thames Valley dataset	
Cluster	numero di utenti
1	1
2	1
3	1
4	198
5	1
6	1

London Carbon London dataset	
Cluster	numero di utenti
1	1
2	1
3	497
4	4
5	1
6	1

La Tabella 5.3 riporta i componenti per ciascun cluster dove è possibile vedere un evidente squilibrio nella distribuzione degli utenti. Come rispecchiato anche dal GSI e dal ASI questa divisione degli utenti non è ottimale.

Tabella 5.4: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.035	0.193	0.011	0.210

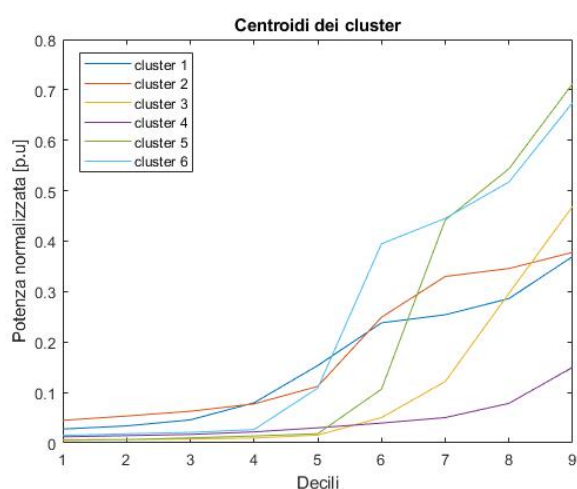


Figura 5.15: Centroidi del Thames Valley dataset.

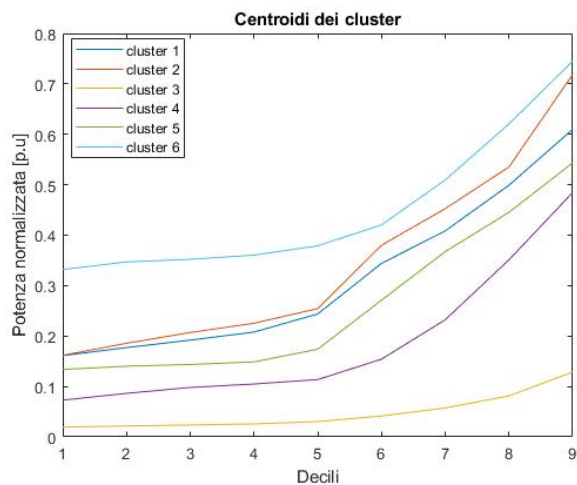


Figura 5.16: Centroidi del Low Carbon London dataset.

Per completezza si riportano i boxplot dei decili, Figure 5.17 e 5.18, e le curve giornaliere degli utenti appartenenti ai cluster, Figure 5.19 e 5.20, anche se non introducono informazioni aggiuntive in questo caso.

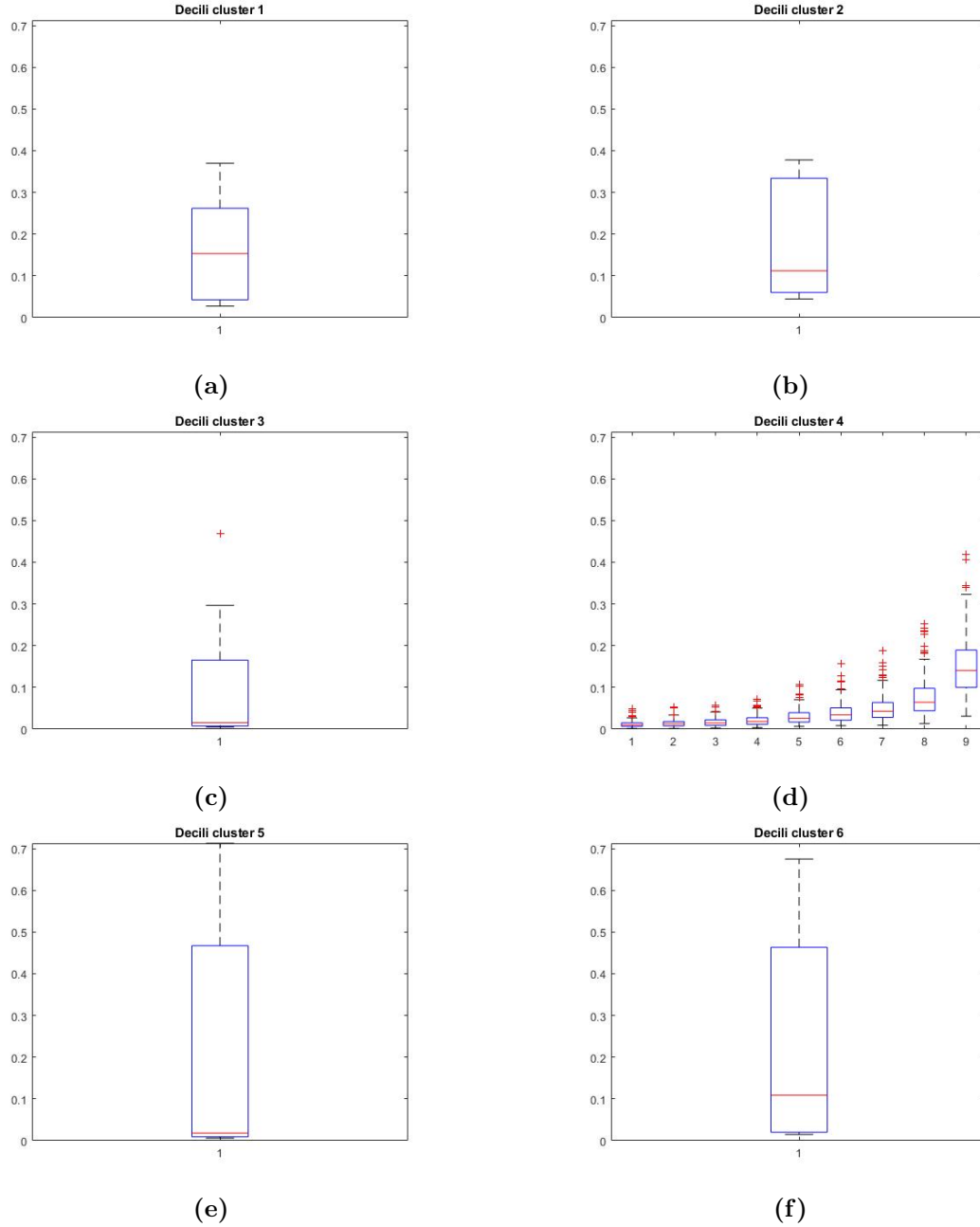


Figura 5.17: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

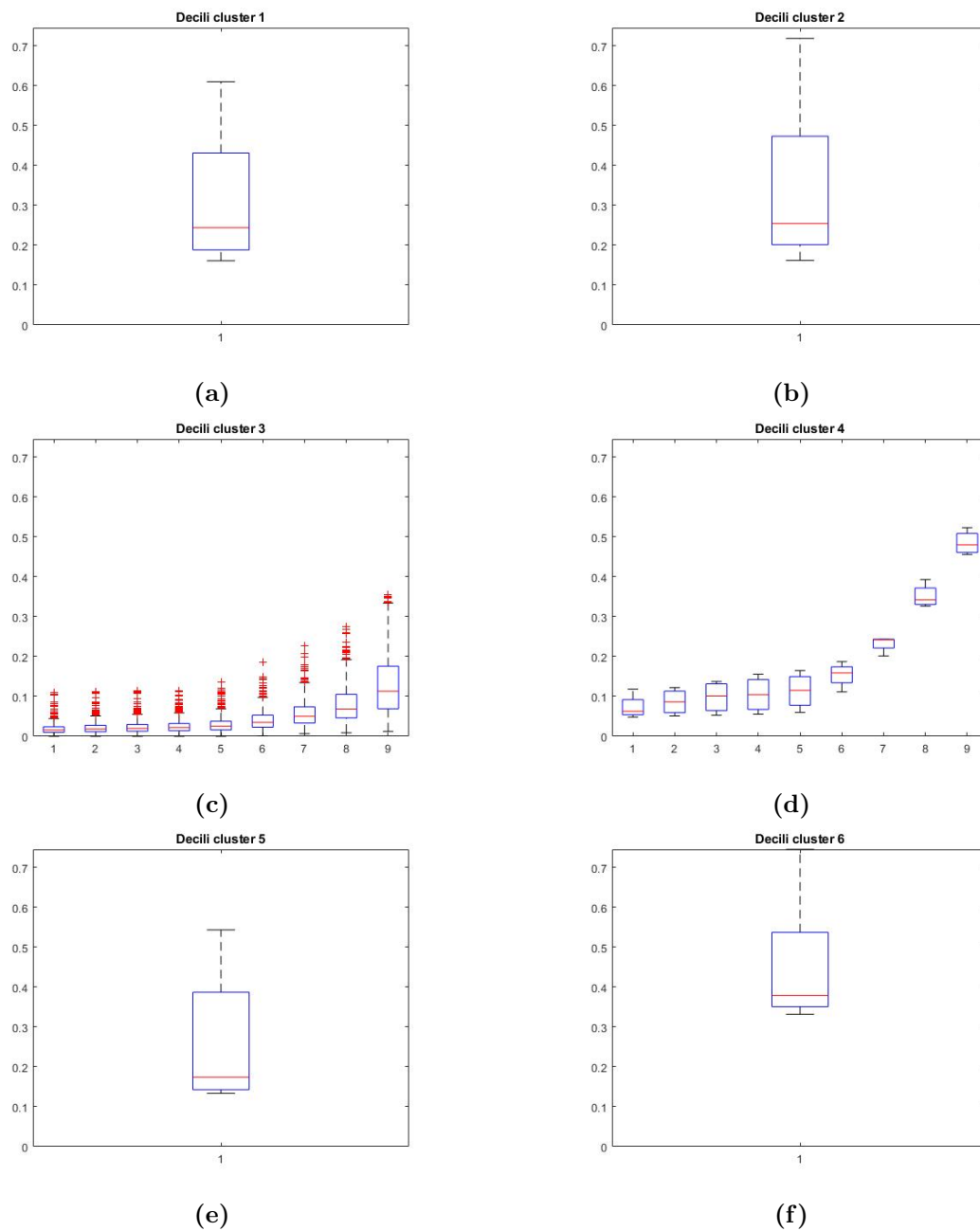


Figura 5.18: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

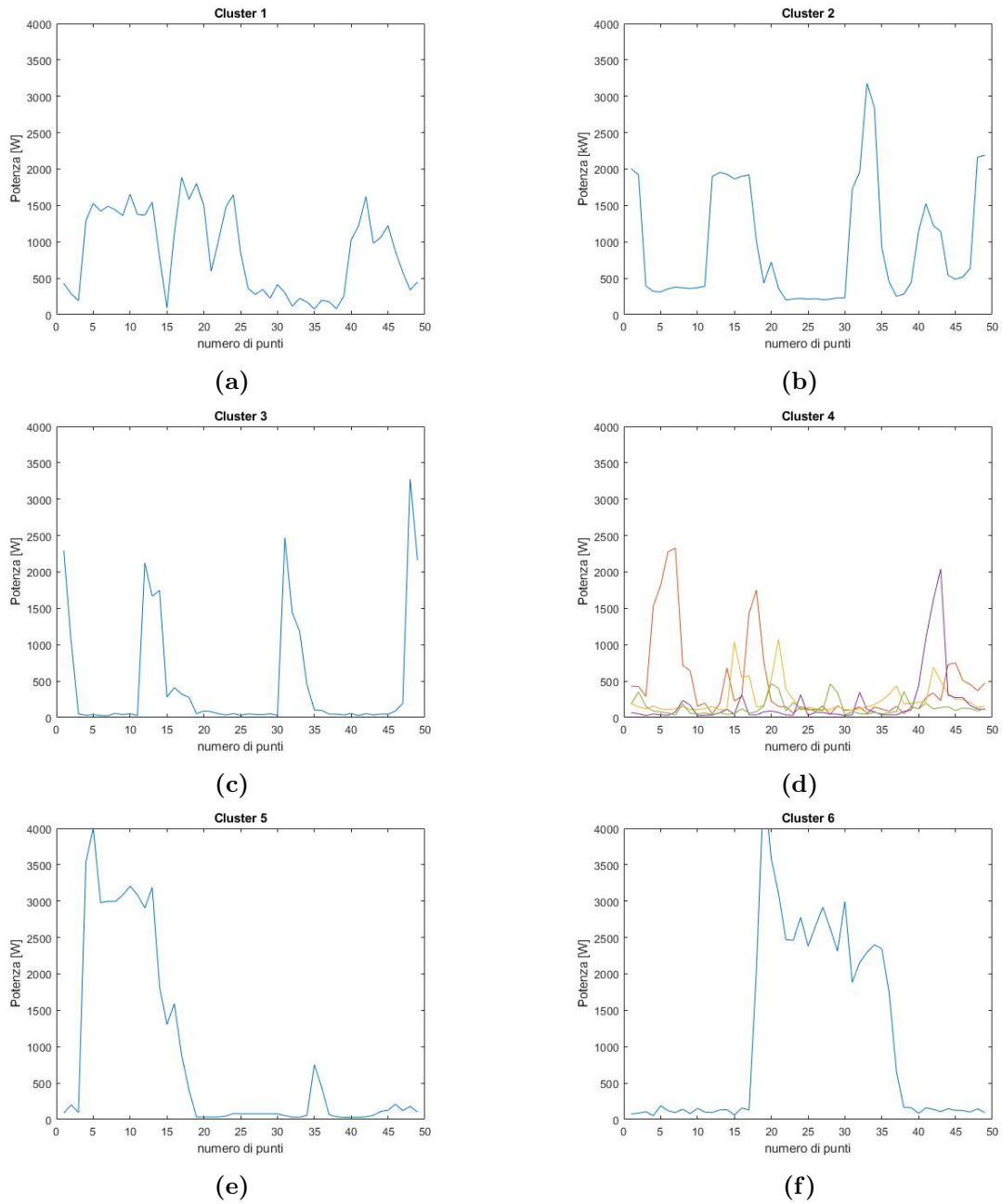


Figura 5.19: Curve di carico giornaliero per cluster appartenenti a utenti diversi nel caso del Thames Valley dataset.

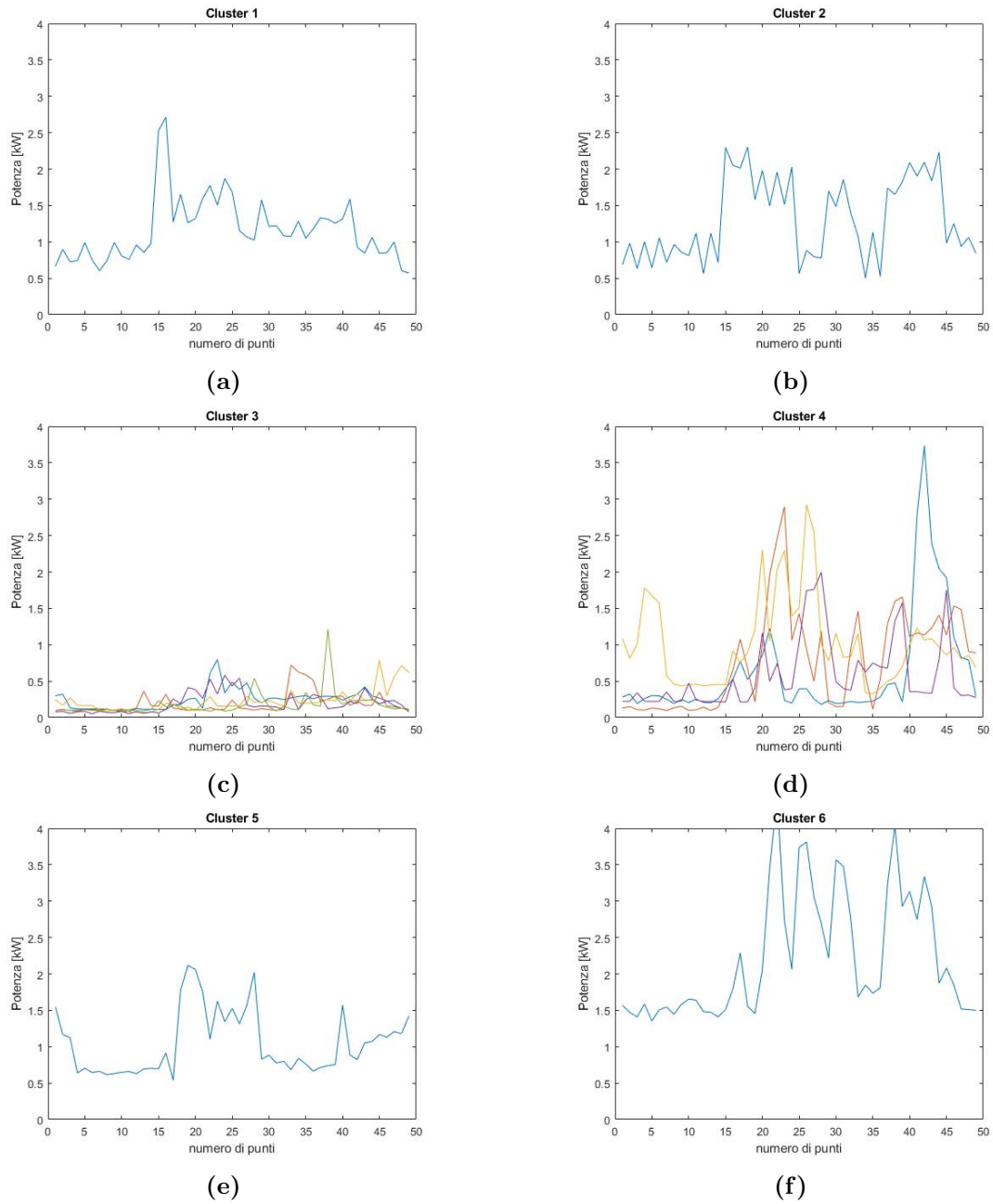


Figura 5.20: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.4.2 Complete linkage

L'albero binario in questo caso viene mostrato in Figura 5.21, il quale sembra avere una ramificazione migliore rispetto al Single linkage.

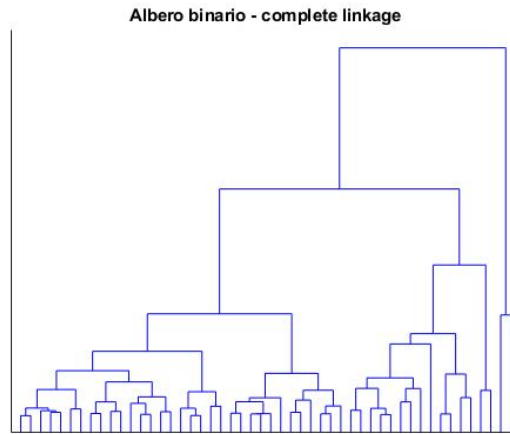


Figura 5.21: Albero binario - complete linkage.

Dalla Tabella 5.5 possiamo notare come per i due dataset ci sia la presenza di un cluster molto grande contenente oltre il 60 per cento degli utenti, mentre la distribuzione del resto degli utenti non è stata molto equa nei cluster rimanenti.

Tabella 5.5: Distribuzione degli utenti all'interno dei cluster.

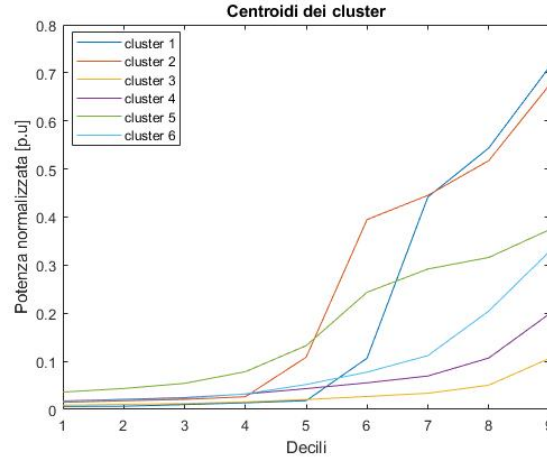
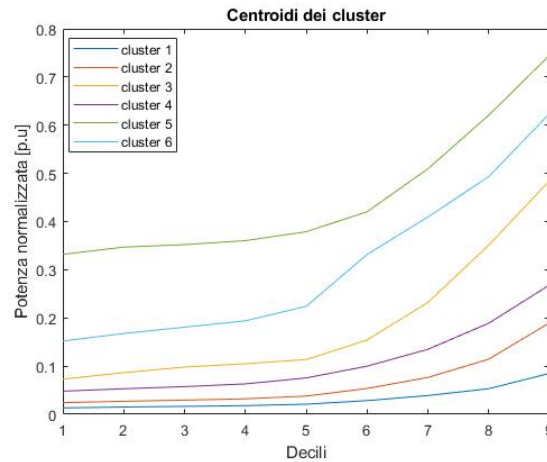
Thames Valley dataset	
Cluster	numero di utenti
1	1
2	1
3	124
4	59
5	2
6	16

Low Carbon London dataset	
Cluster	numero di utenti
1	324
2	127
3	4
4	46
5	1
6	3

Entrambi i dataset presentano valori di GSI e ASI migliori rispetto al single linkage, Tabella 5.6.

Tabella 5.6: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.085	0.349	0.094	0.382

**Figura 5.22:** Centroidi del Thames Valley dataset.**Figura 5.23:** Centroidi del Low Carbon London dataset.

Andando a vedere i centroidi si nota che nel caso del LCL la partizione risulta ancora molto buona, infatti i centroidi sono ben differenziati. Per quanto riguarda la TV invece i centroidi risultano meno differenziati, cosa che è possibile riscontrare anche dai boxplot dei decili, Figure 5.24 e 5.25.

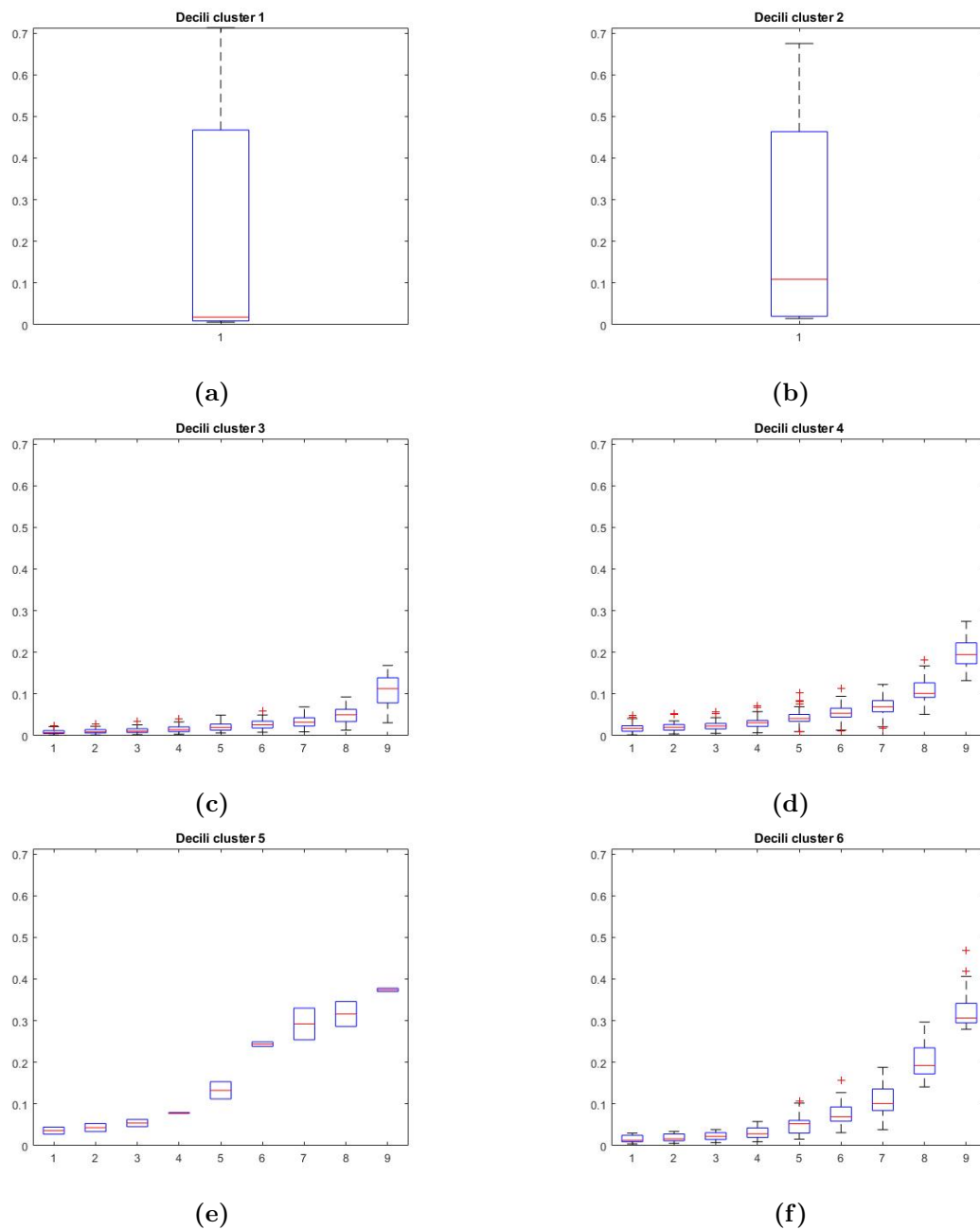


Figura 5.24: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

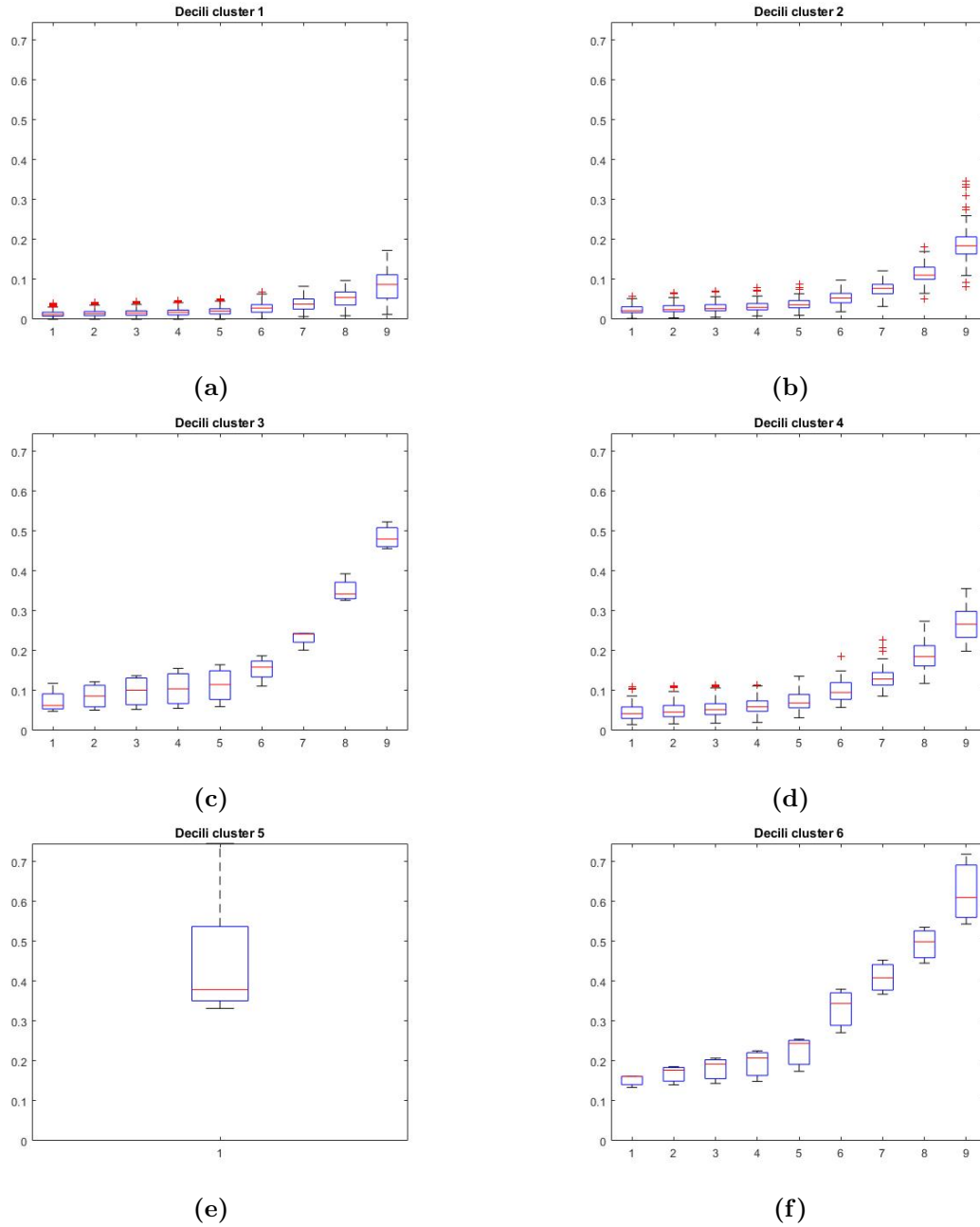


Figura 5.25: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

Guardando le curve di carico giornaliere, nel caso del Thames Valley dataset il cluster 1 e il cluster 2 potevano essere inseriti nello stesso cluster, mentre si sarebbe potuto ottenere una miglior partizione del cluster 4 e 6 in cui ci sono alcune curve con comportamenti e picchi simili.

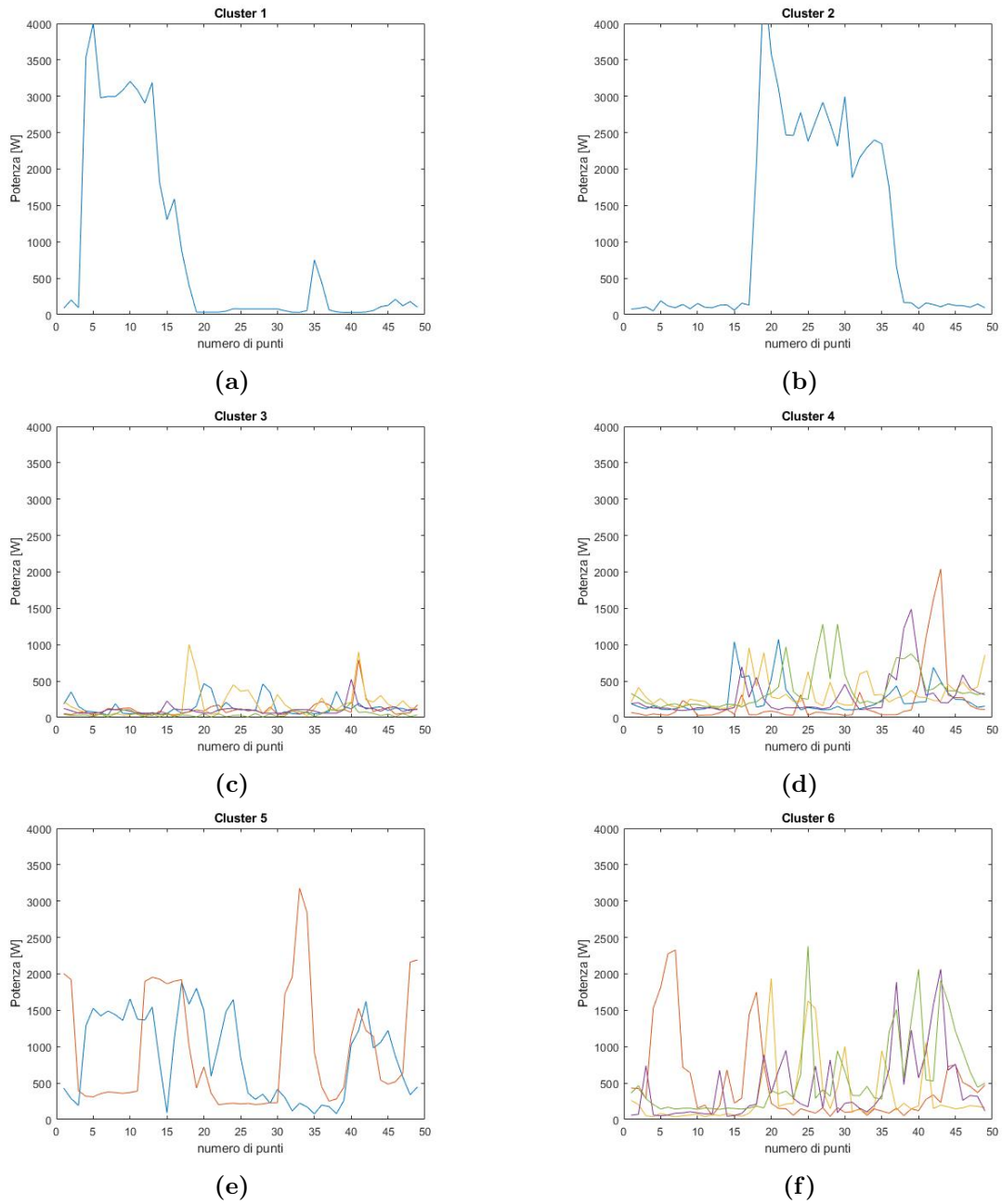


Figura 5.26: Curve di carico giornaliero per cluster appartenenti a utenti diversi nel caso del Thames Valley dataset.

Per quanto riguarda il dataset della Low Carbon London, la partizione dei dati risulta un po' migliore, ad eccezione di alcuni casi, come ad esempio la presenza del cluster 5 che ha un solo elemento.

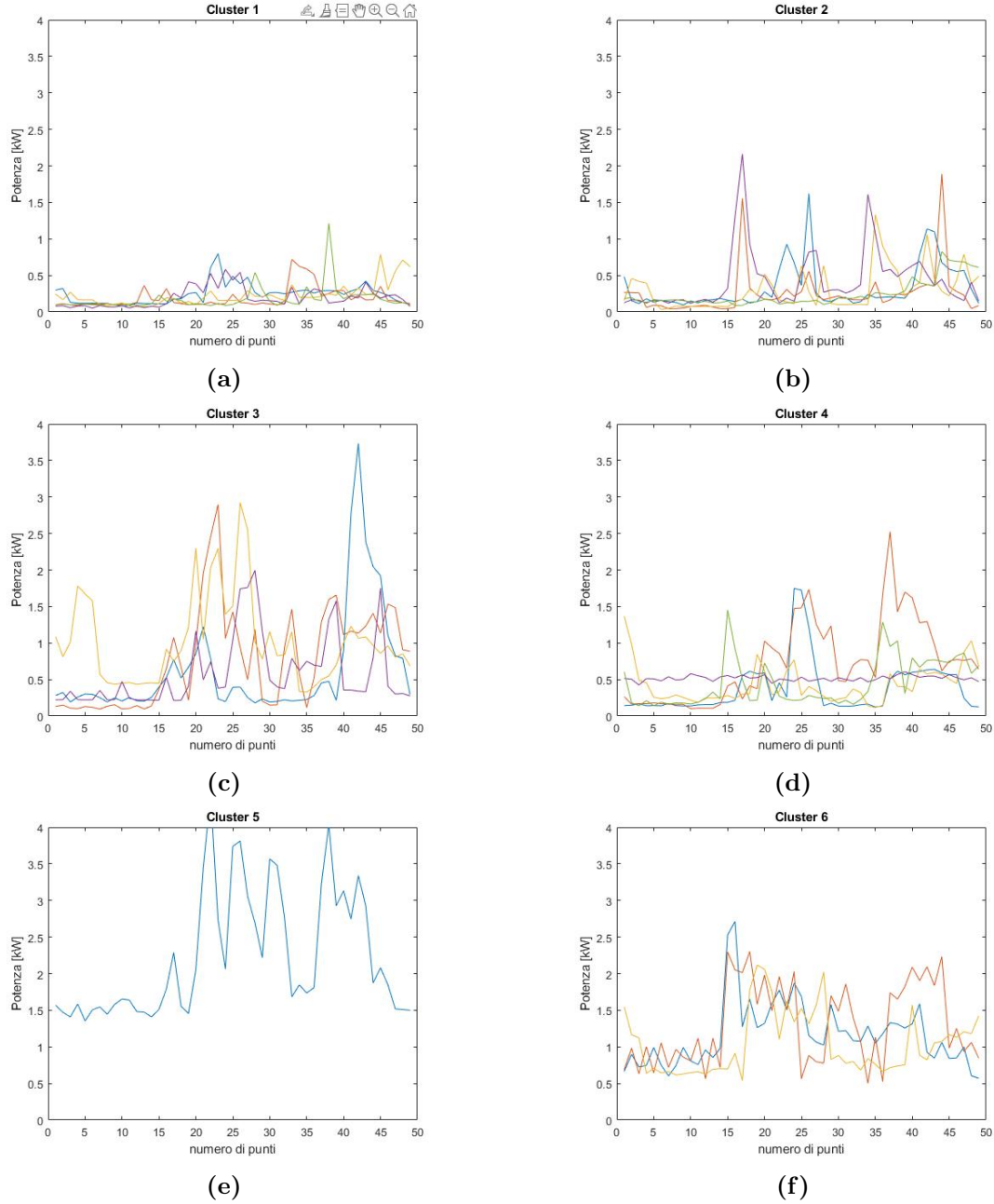


Figura 5.27: Curve di carico giornaliero per cluster appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.4.3 Average linkage

L'average linkage presenta un comportamento che è una via di mezzo tra il single e il complete linkage, infatti anche in questo caso notiamo la formazione di cluster molto squilibrati, in cui un cluster racchiude la maggior parte degli utenti. I centroidi, Figure 5.29 e 5.30, anche in questo caso sono per la TV poco differenziati, mentre per la LCL c'è una differenziazione notevole.

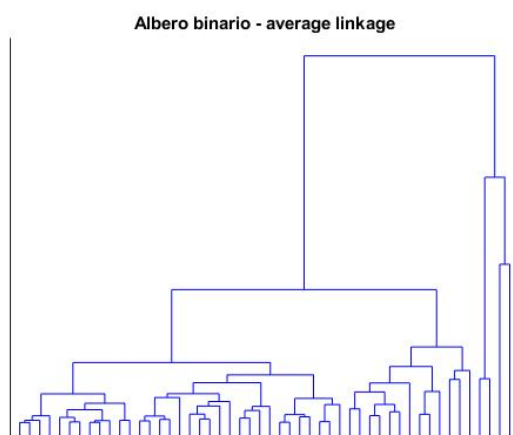


Figura 5.28: Albero binario - average linkage.

Tabella 5.7: Distribuzione degli utenti all'interno dei cluster.

Thames Valley dataset	
Cluster	numero di utenti
1	3
2	13
3	183
4	1
5	1
6	2

Low Carbon London dataset	
Cluster	numero di utenti
1	1
2	2
3	473
4	24
5	4
6	1

Tabella 5.8: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.061	0.313	0.056	0.384

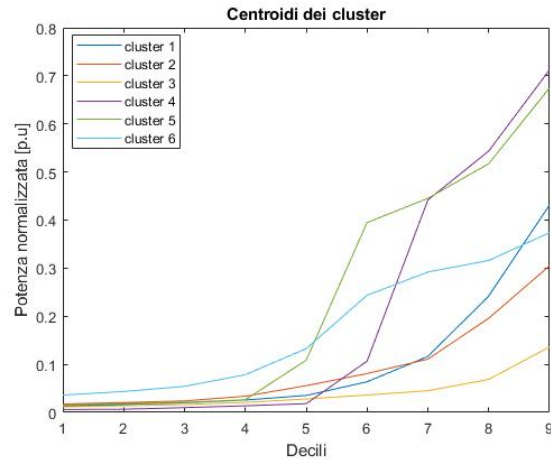


Figura 5.29: Centroidi del Thames Valley dataset.

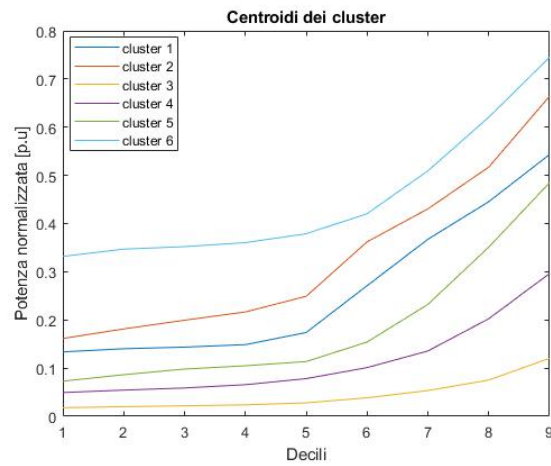


Figura 5.30: Centroidi del Low Carbon London dataset.

Dai boxplot di entrambi i dataset notiamo una discreta differenziazione nei decili, tuttavia si nota come non è così marcata come in altri casi.

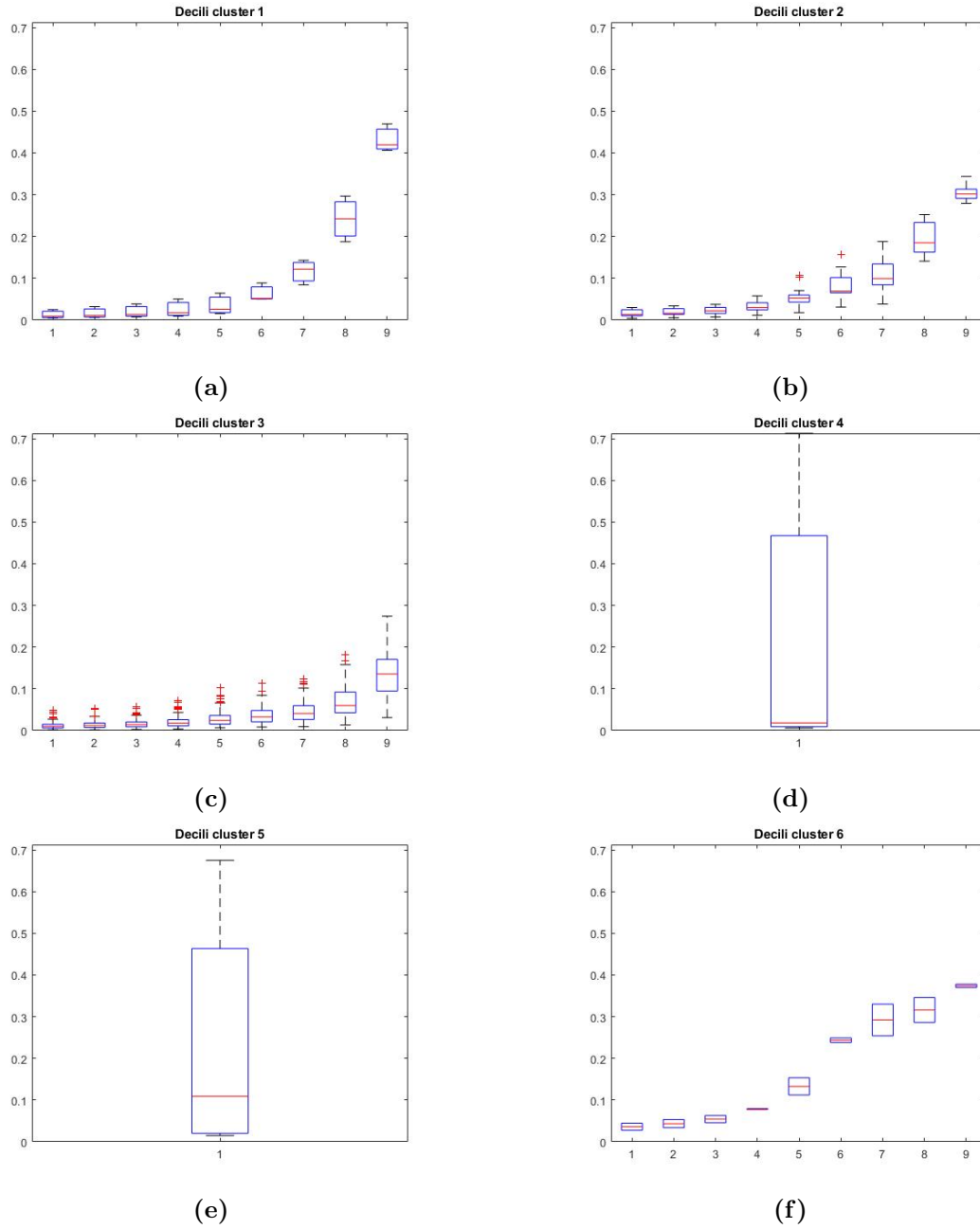


Figura 5.31: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

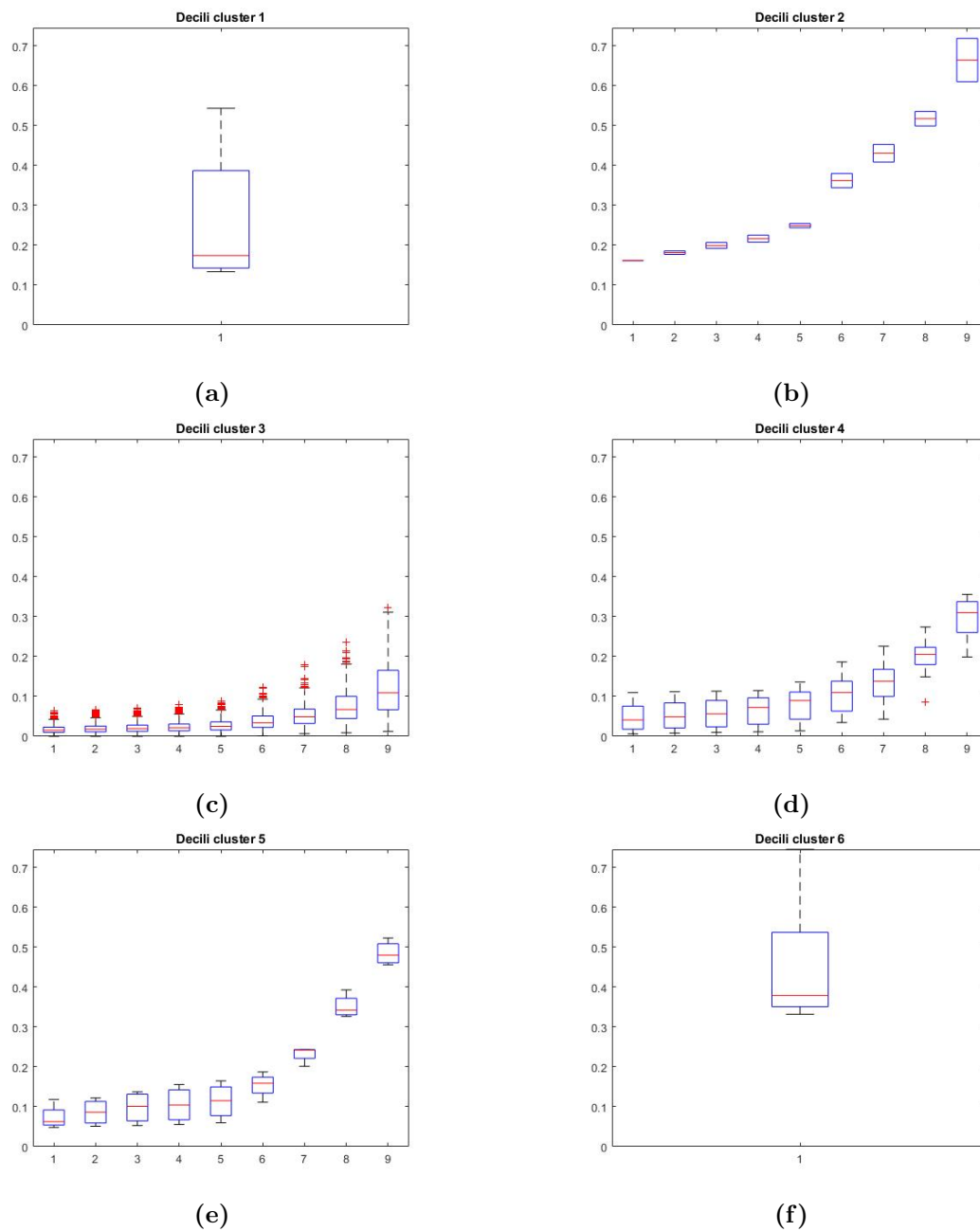


Figura 5.32: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

Le curve di carico giornaliere, Figure 5.33 e 5.34, dimostrano che la suddivisione pur non essendo tra le migliori ha un certo senso.

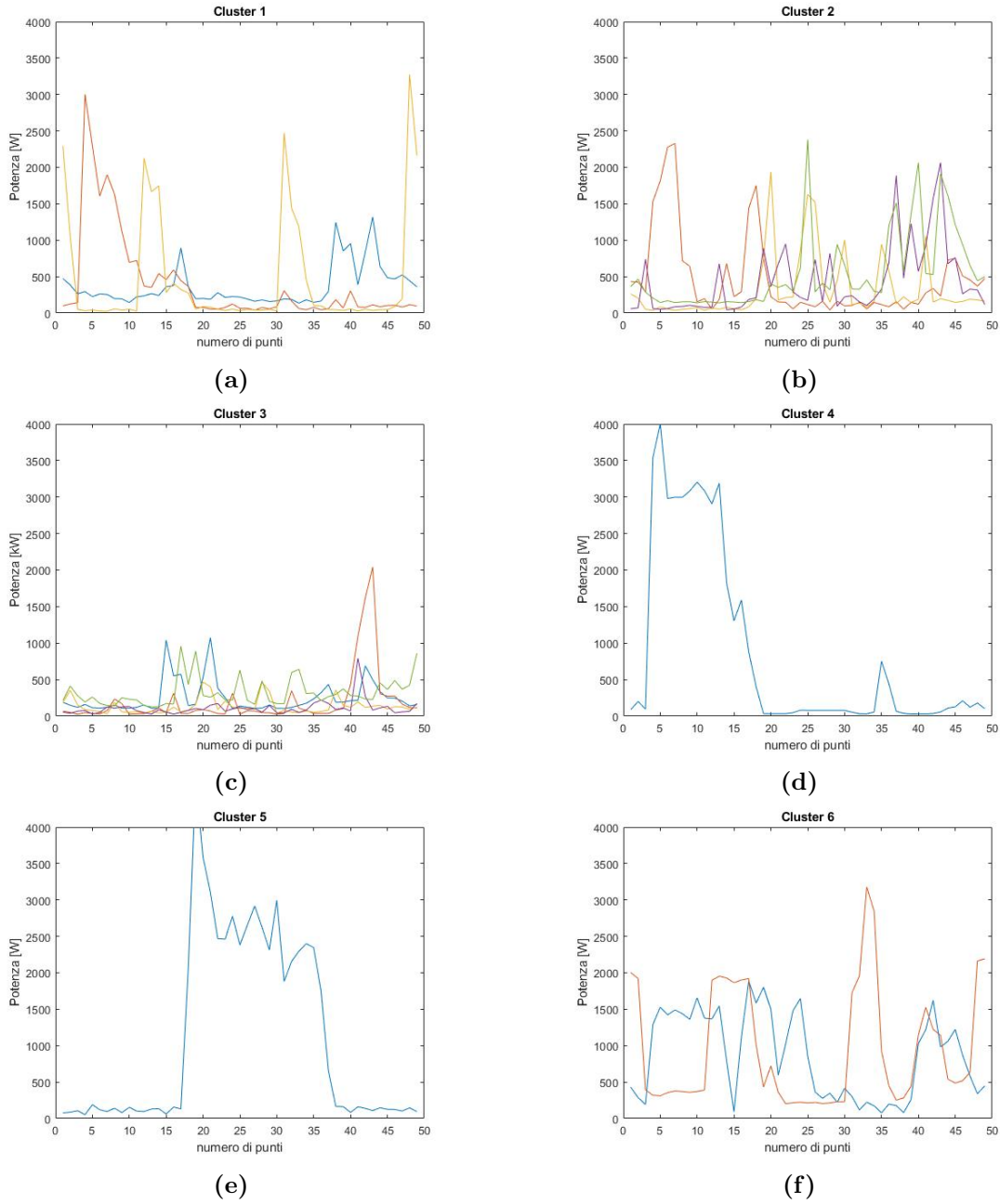


Figura 5.33: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Thames Valley dataset.

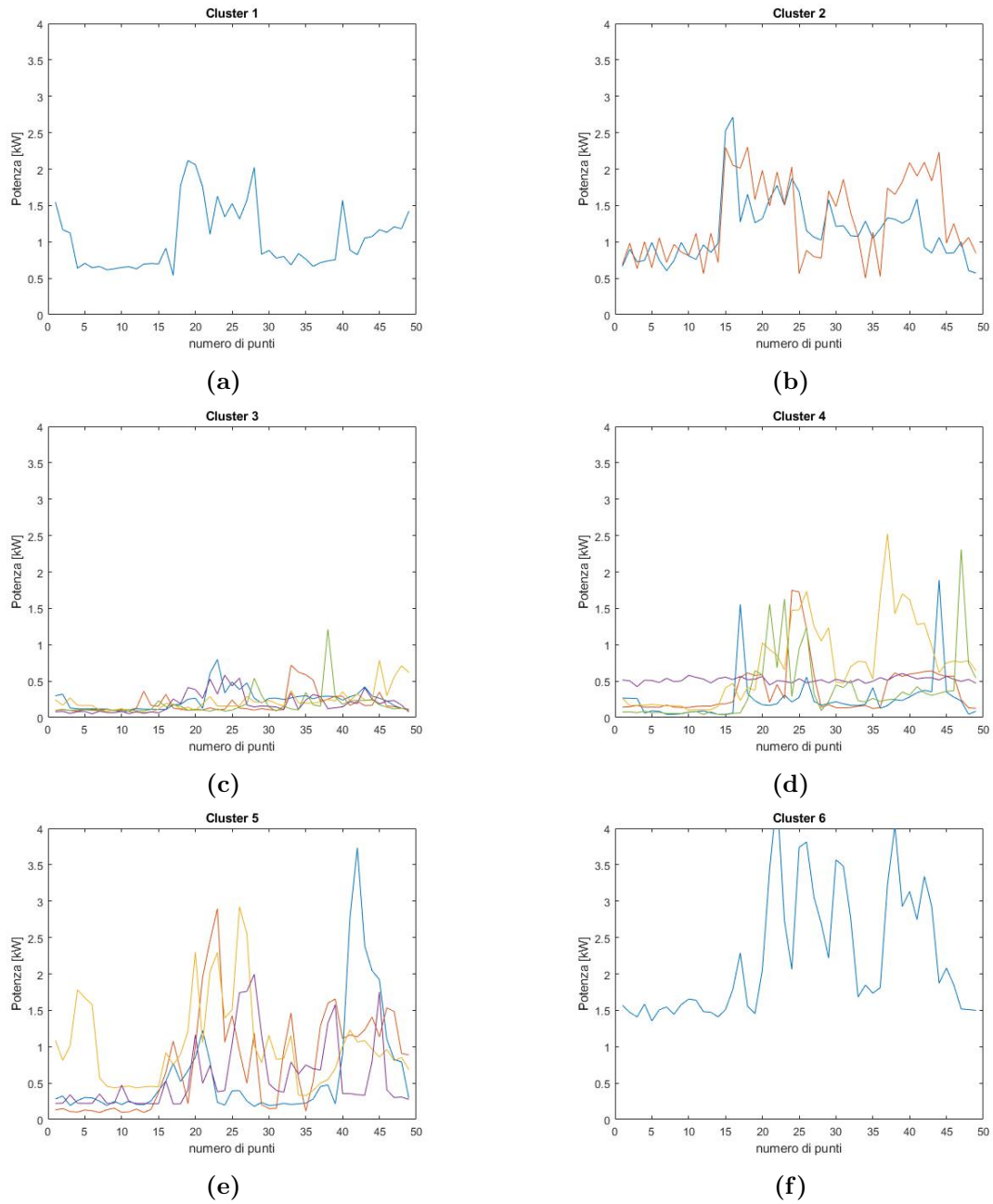


Figura 5.34: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.4.4 Centroid linkage

L'albero binario risultante da questo metodo risulta essere non monotono, quindi si deduce che questo criterio di linkage non è adatto in questo caso. La monotonicità dell'albero binario è fondamentale per il clustering gerarchico agglomerativo, questo vuol dire che mano a mano che si fondono i cluster i valori di similarità vanno a decrescere. Quando si presenta un albero non monotono alcuni nodi tendono a invertire questa tendenza, Figura 5.32.

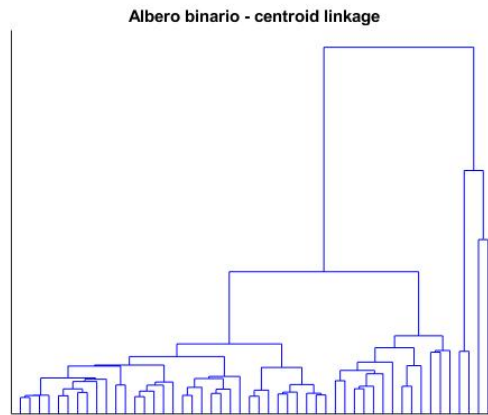


Figura 5.35: Albero binario - centroid linkage.

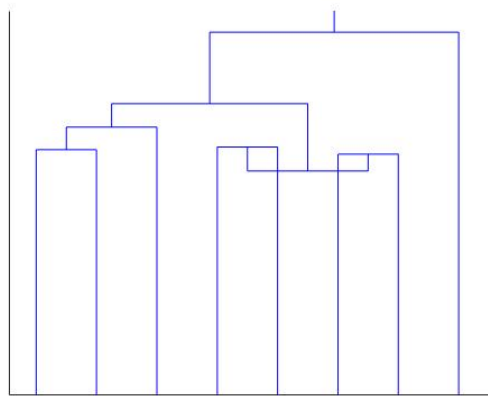


Figura 5.36: Prova che l'albero è non monotono.

Per questa ragione non abbiamo applicato questo criterio di linkage.

5.4.5 Ward linkage

I risultati ottenuti da questo criterio sono molto diversi dai linkage precedente, infatti in questo caso possiamo notare una distribuzione degli utenti all'interno dei cluster molto più equilibrata, la Tabella 5.9 riporta il numero di utenti per cluster. Ciò nonostante, i valori di silhouette medie sono alti, come riportato in Tabella 5.10.

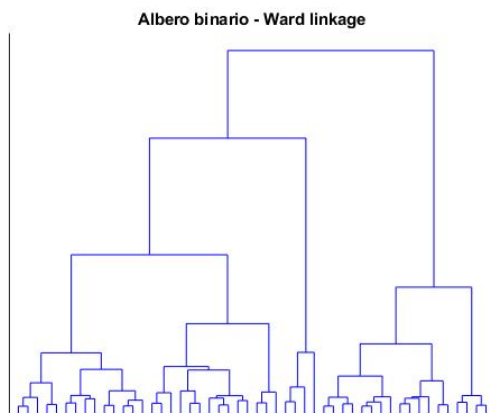


Figura 5.37: Albero binario - Ward linkage.

Tabella 5.9: Distribuzione degli utenti all'interno dei cluster.

Thames Valley dataset		Low Carbon London dataset	
Cluster	numero di utenti	Cluster	numero di utenti
1	19	1	4
2	17	2	37
3	2	3	109
4	2	4	240
5	72	5	111
6	91	6	4

Tabella 5.10: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.251	0.504	0.219	0.483

Pur avendo numeri di utenti abbastanza elevati per cluster i centroidi rappresentativi presentano una notevole diversità in entrambi i dataset, come è possibile vedere nelle Figura 5.38 e 5.39.

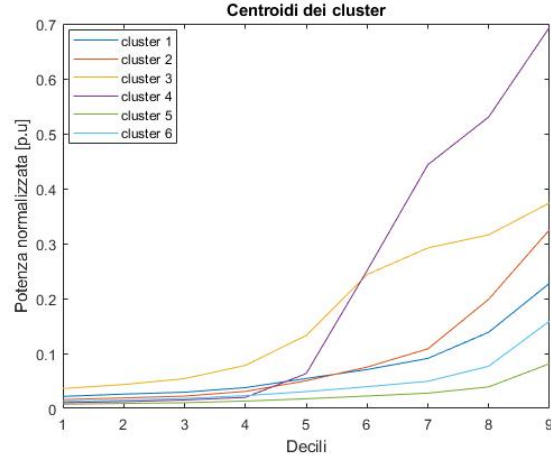


Figura 5.38: Centroidi del Thames Valley dataset.

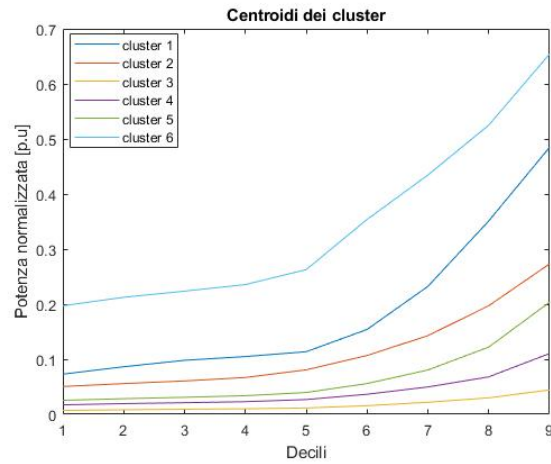


Figura 5.39: Centroidi del Low Carbon London dataset.

I boxplot confermano ciò che si deduce dai centroidi, Figure 5.41 e 5.42.

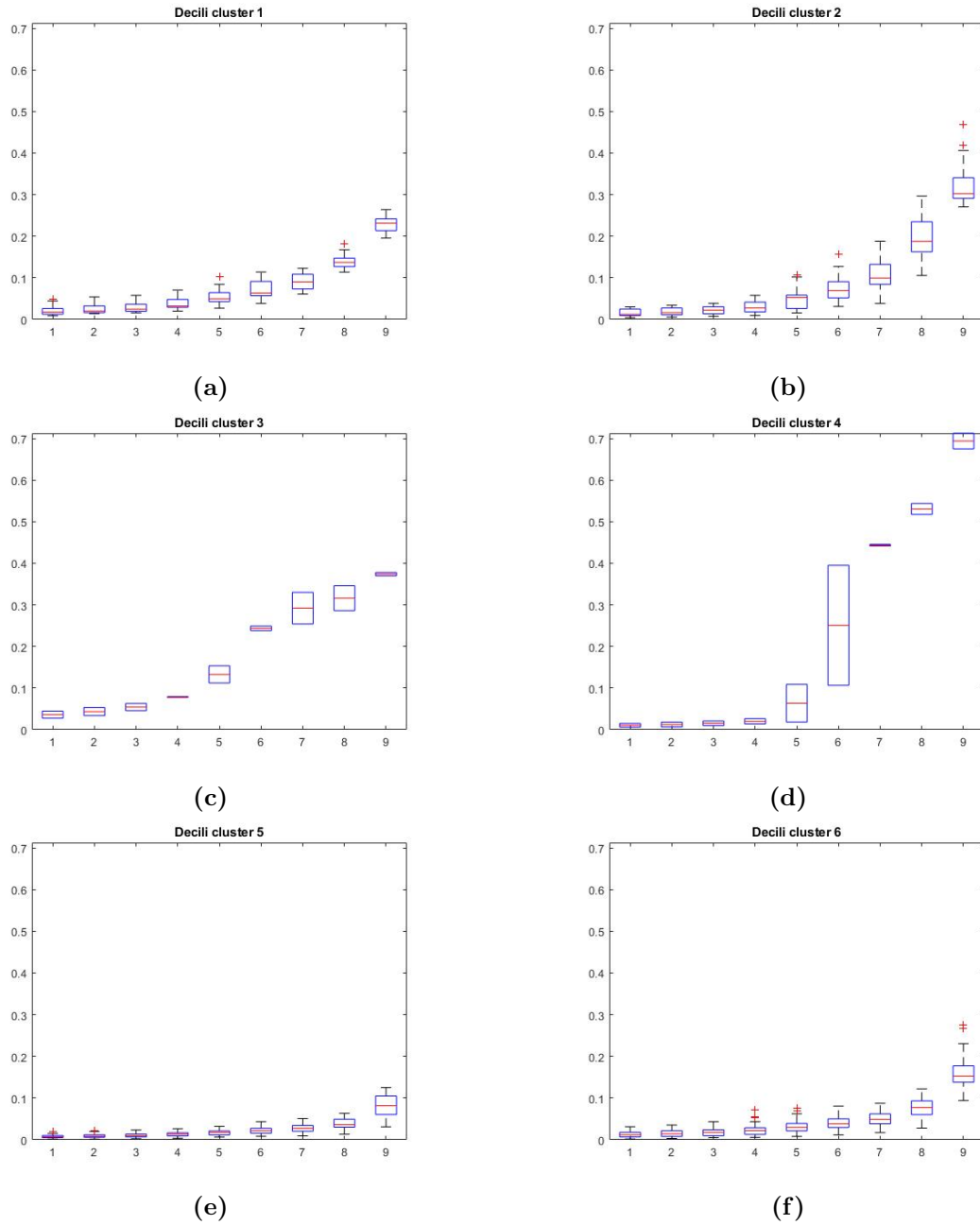


Figura 5.40: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

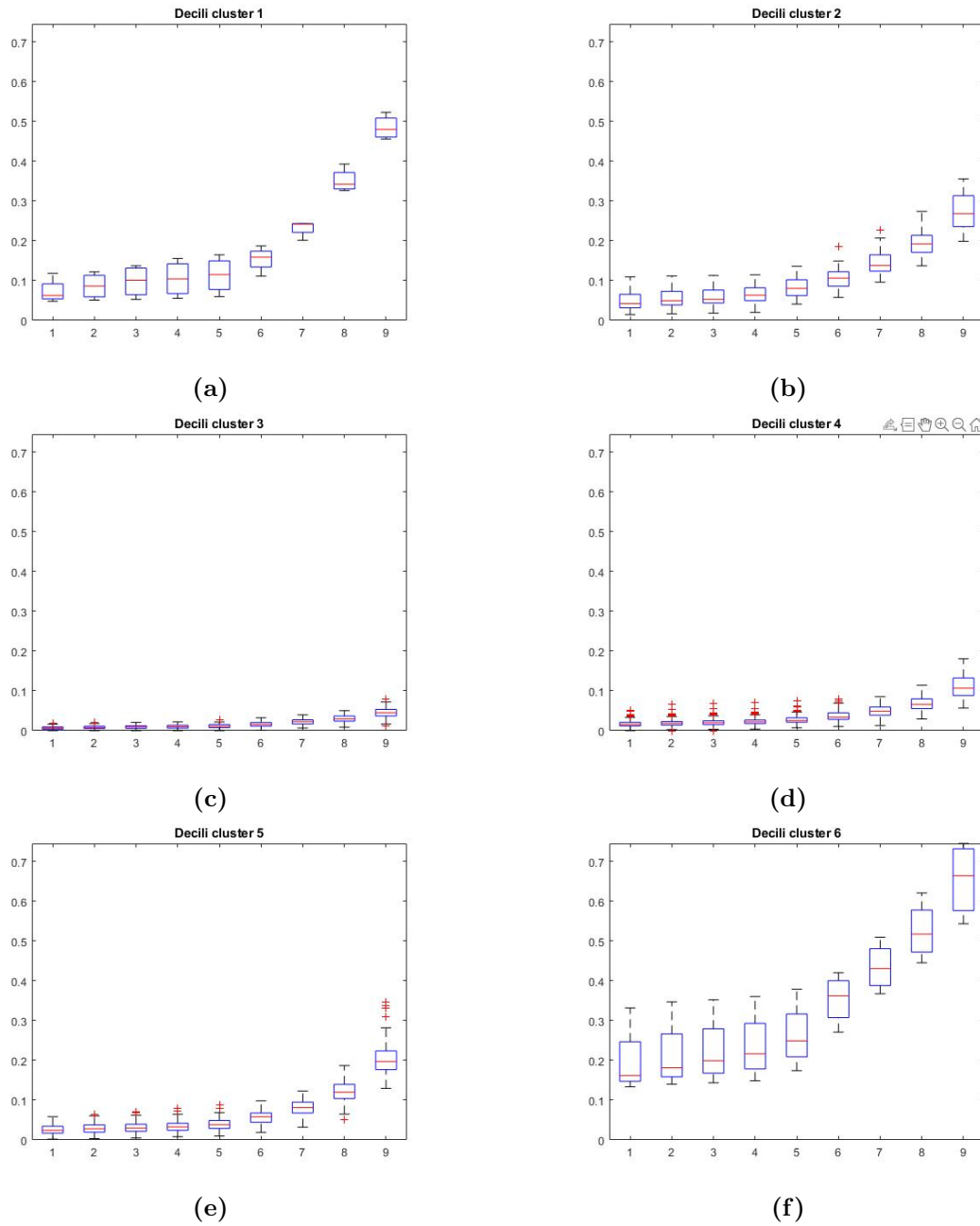


Figura 5.41: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

Guardando le curve giornaliere della Thames Valley possiamo notare come la suddivisione abbia messo insieme curve molto simili sia in termini di ampiezza che di forma, i cluster più piccoli (cluster 3 e 4) pur avendo pochi utenti sembrano aver identificato utenti molto simili, anche se con picchi spostati nel tempo che sono anche molto diversi dalle curve degli altri cluster.

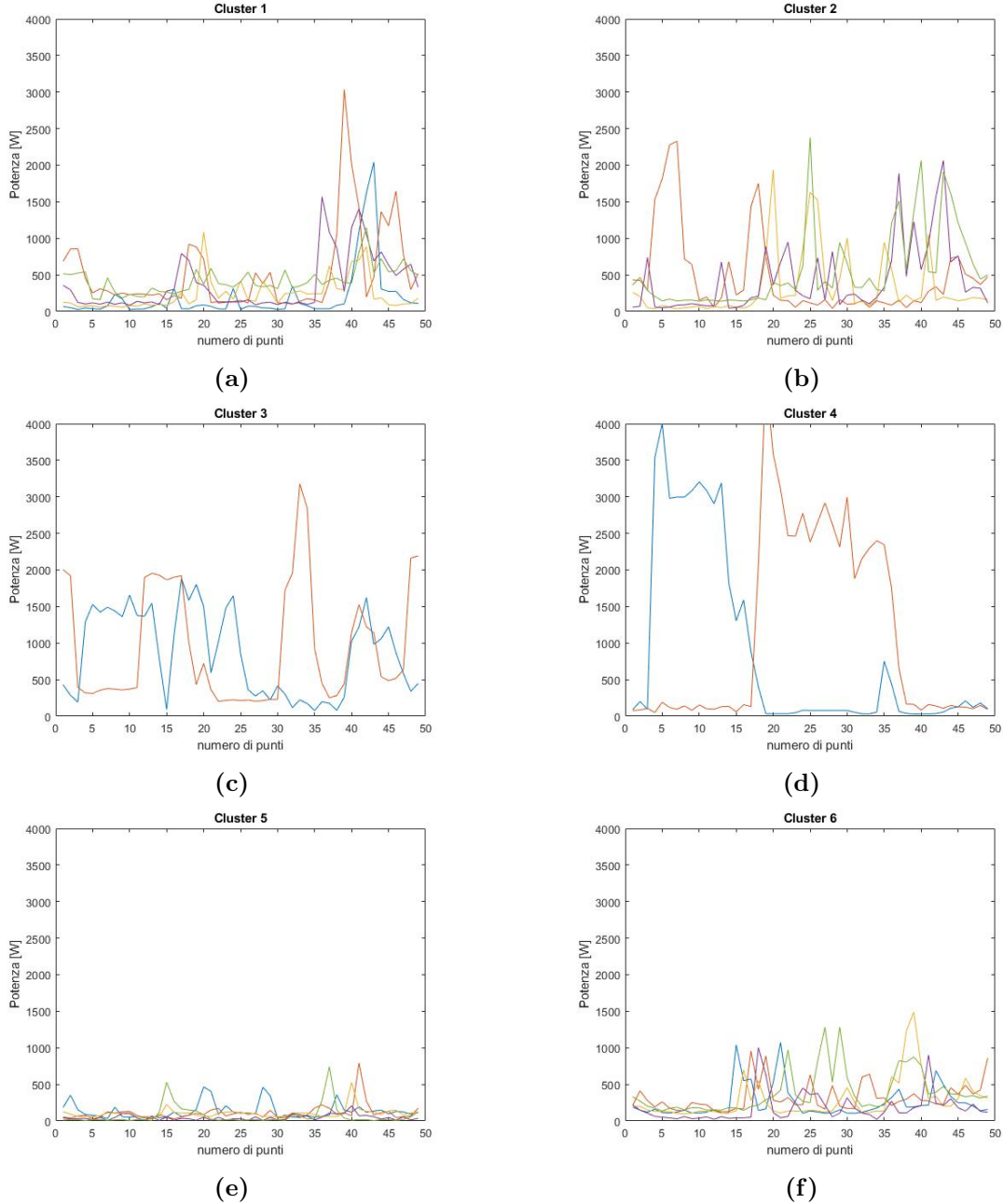


Figura 5.42: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Thames Valley dataset.

Lo stesso vale per il dataset della LCL.

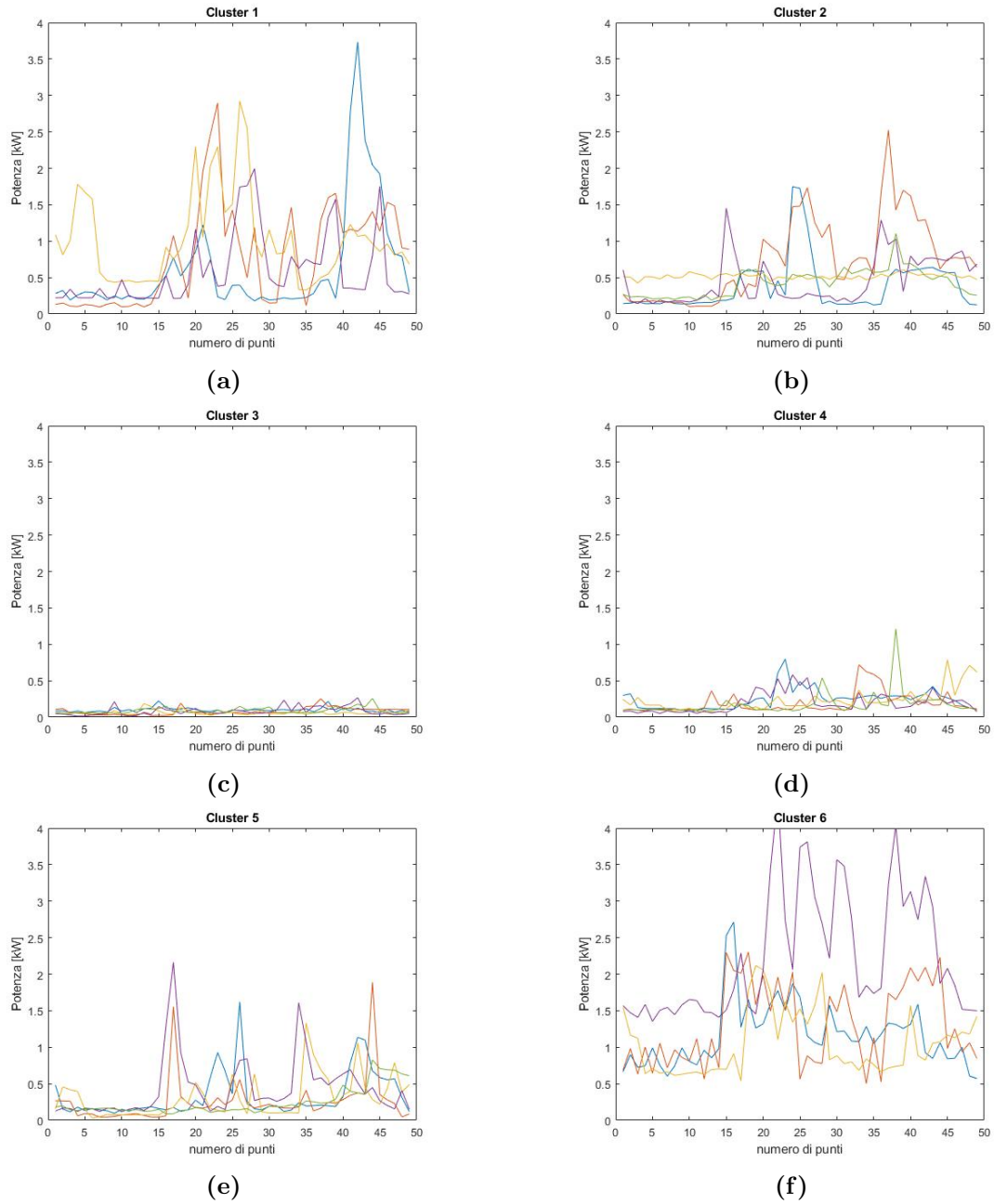


Figura 5.43: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Low Carbon London dataset.

Weighted average linkage

Con questo criterio di linkage abbiamo ottenuto una partizione dei dati molto simile a quella dell'average e del complete linkage infatti notiamo la presenza di 2 cluster molto corposi, mentre il resto ha meno di 5 elementi.

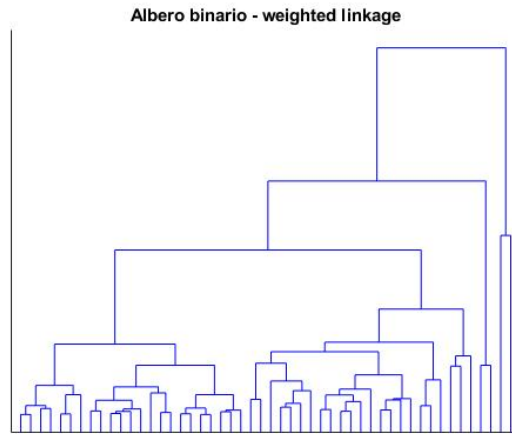


Figura 5.44: Albero binario - Weighted linkage.

Tabella 5.11: Distribuzione degli utenti all'interno dei cluster.

Thames Valley dataset		Low Carbon London dataset	
Cluster	numero di utenti	Cluster	numero di utenti
1	3	1	1
2	33	2	2
3	163	3	4
4	1	4	123
5	1	5	374
6	2	6	1

Tabella 5.12: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.092	0.37	0.081	0.34

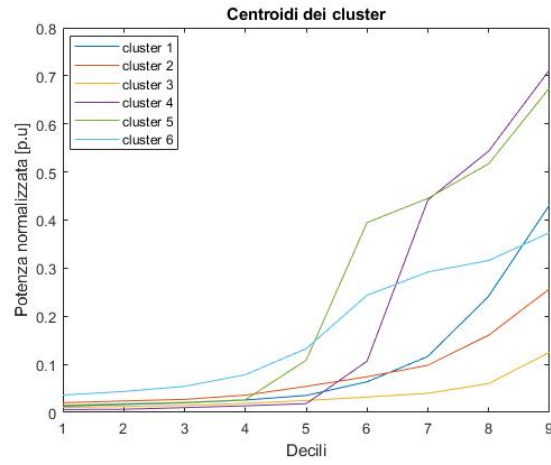


Figura 5.45: Centroidi del Thames Valley dataset.

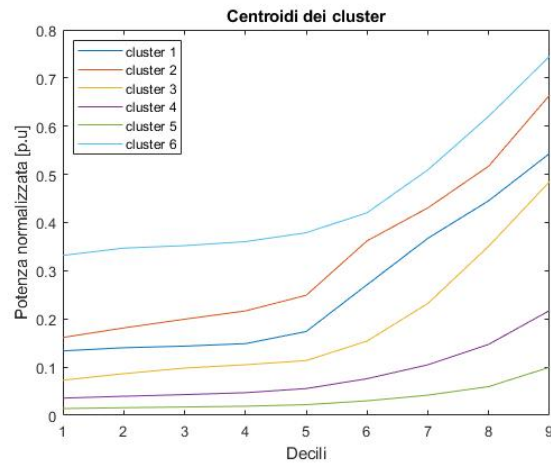


Figura 5.46: Centroidi del Low Carbon London dataset.

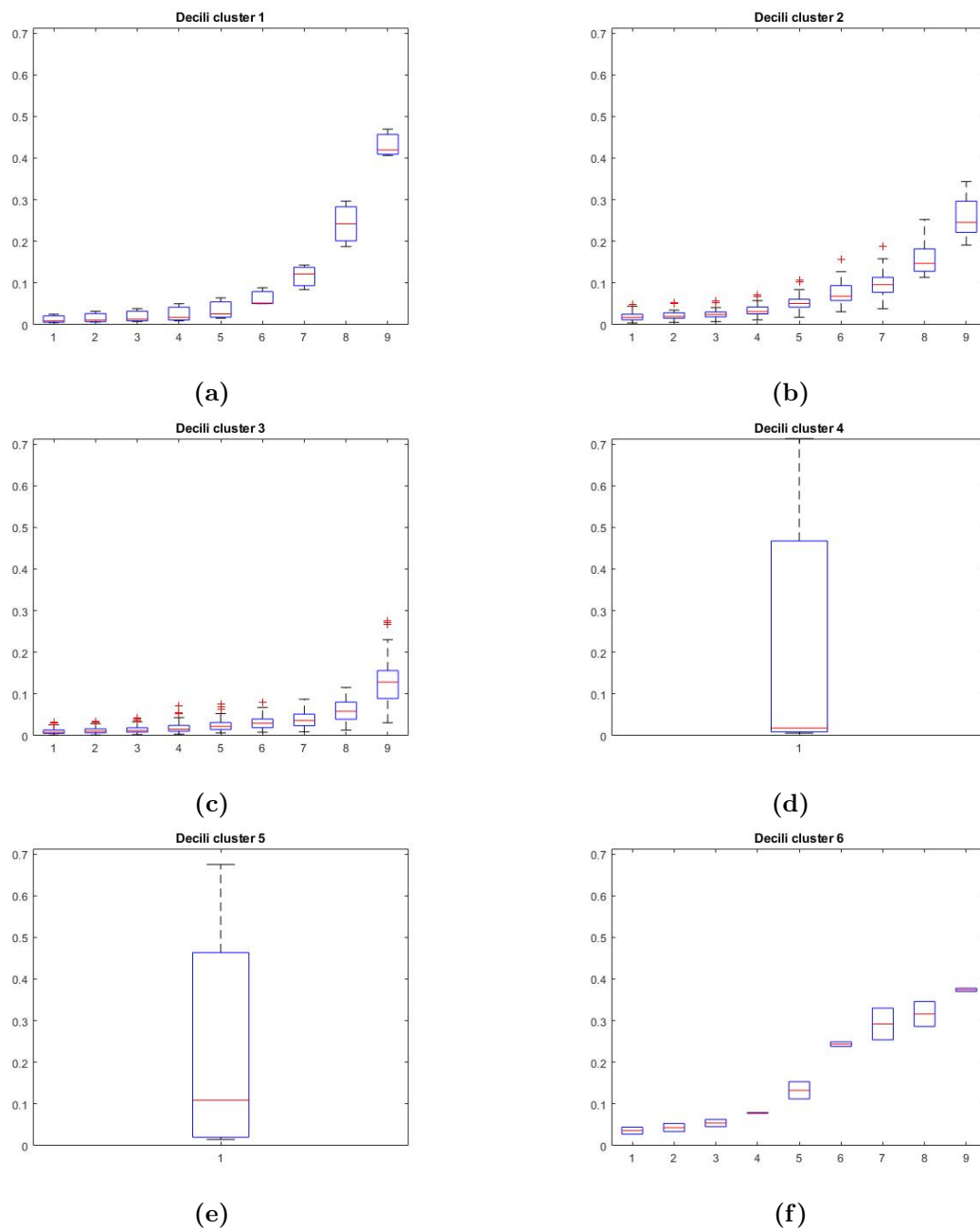


Figura 5.47: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

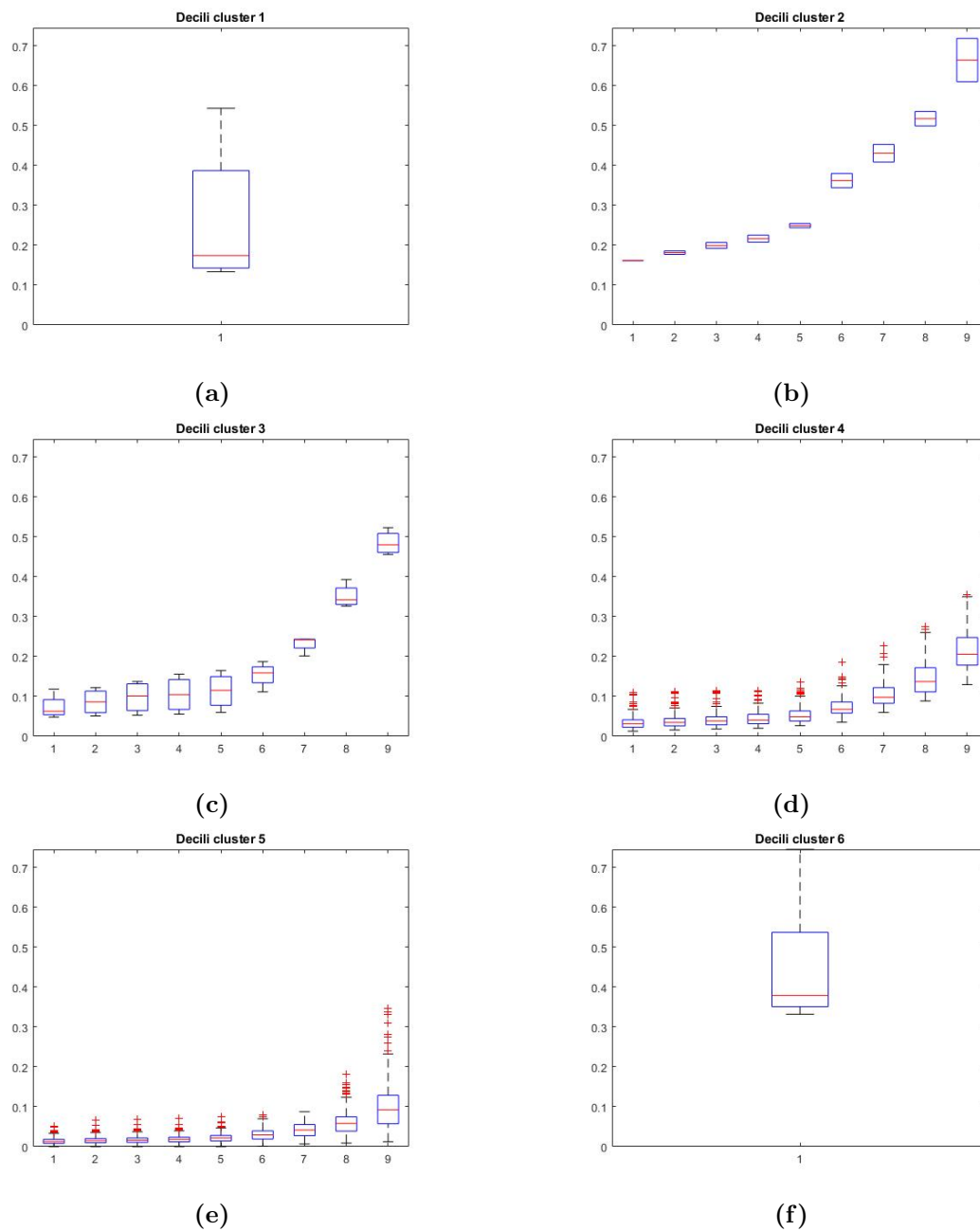


Figura 5.48: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

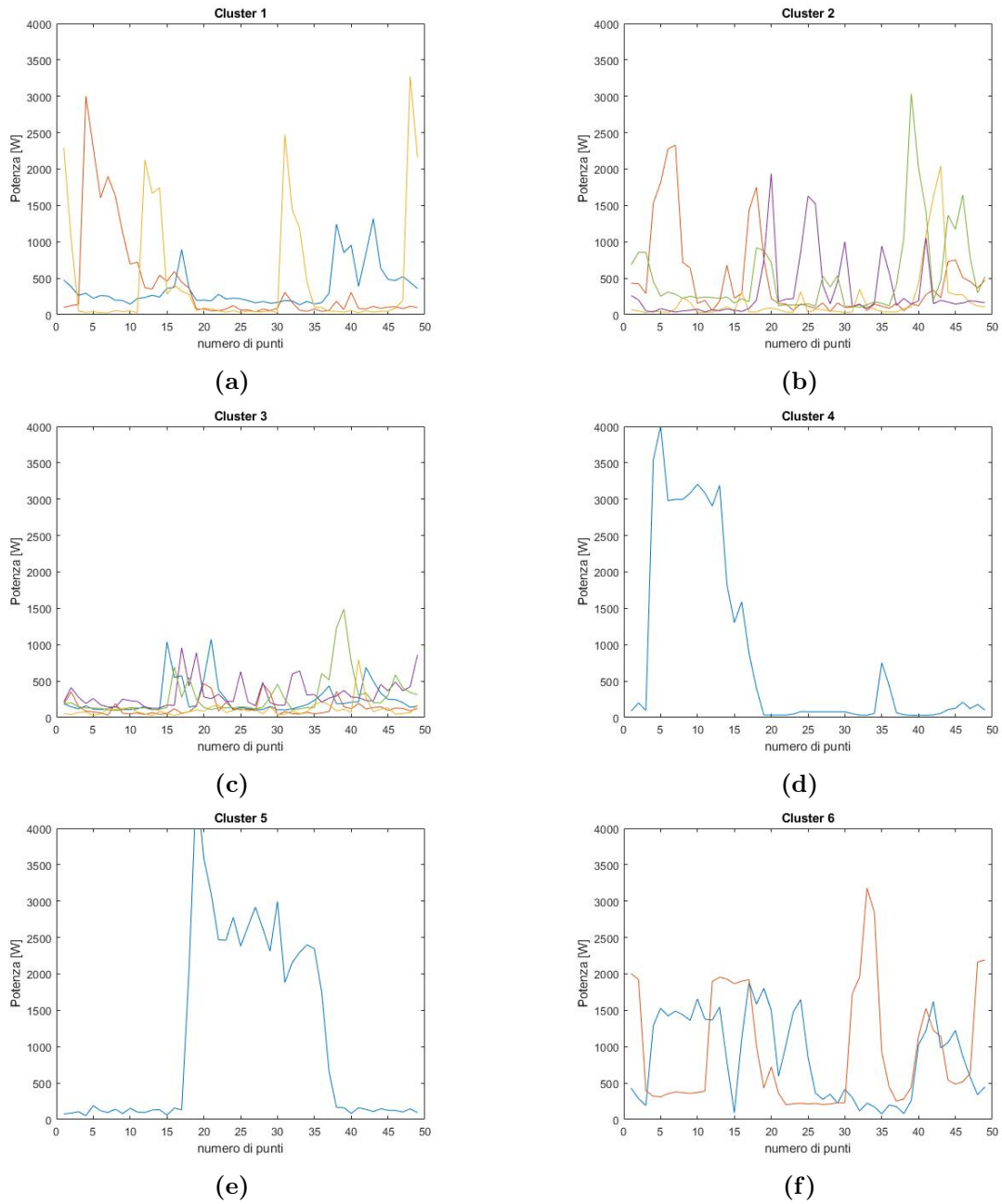


Figura 5.49: Curve di carico giornaliero per cluster, appartenenti a utenti diversi nel caso del Thames Valley dataset.

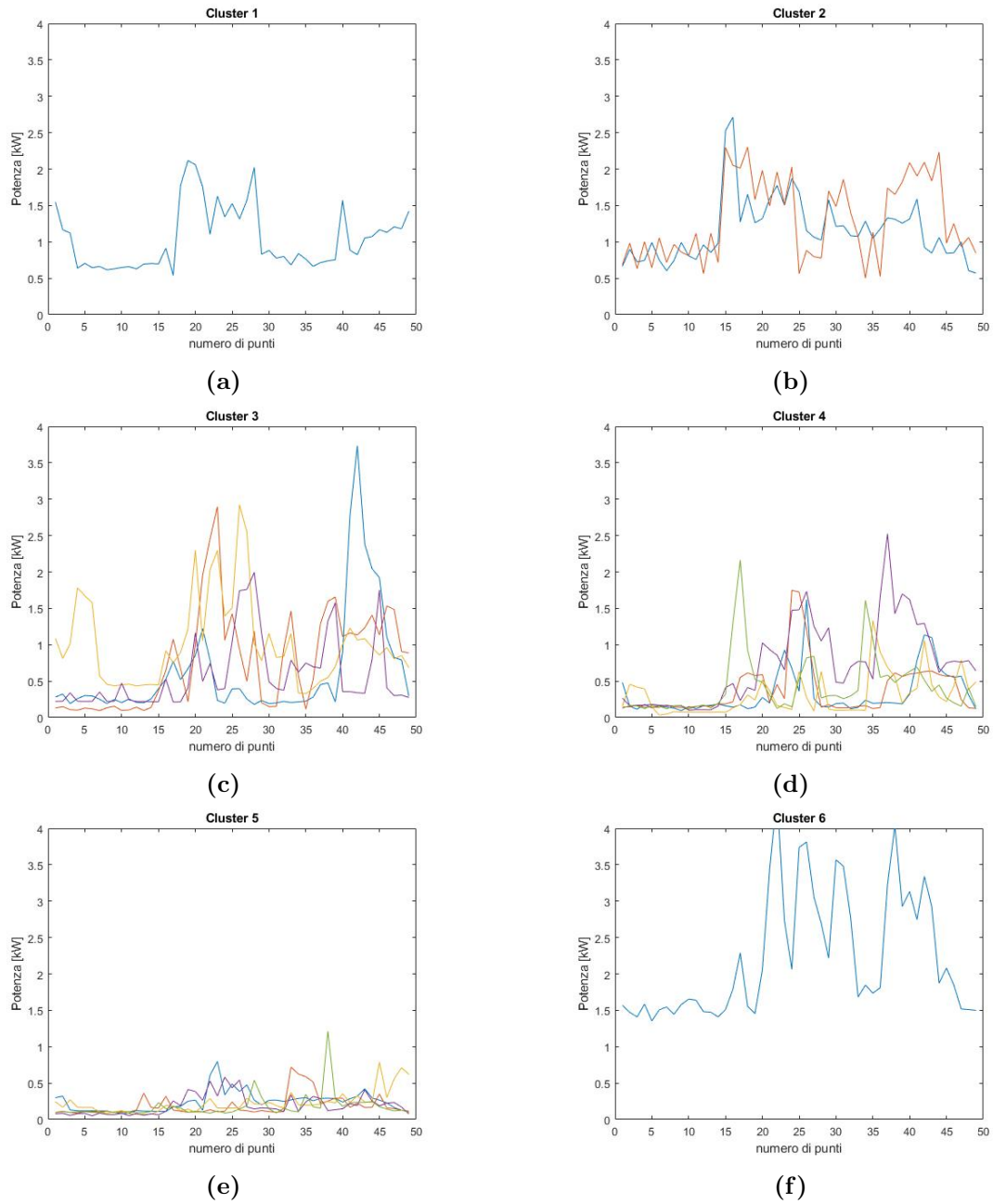


Figura 5.50: Curve di carico giornaliere per cluster appartenenti a utenti diversi nel caso del Low carbon London dataset.

5.5 Spectral clustering

L'ultimo algoritmo di clustering con il quale abbiamo deciso di verificare la filosofia di CONDUCTS è lo spectral clustering. In questo caso abbiamo voluto vedere come si comporta l'algoritmo in due casi:

1. L'input del clustering sono i 9 decili, da cui i rispettivi autovettori della matrice laplaciana sono quelli sottoposti al clustering - Spectral clustering (L).
2. L'input in questo caso sono gli indici di similarità ricavati dal calcolo della distanza euclidea tra gli utenti, usando i decili - Spectral clustering (S).

In entrambi i casi il tipo di grafo che si utilizza è il grafo del tipo k-neighbourhood, mentre la matrice laplaciana è quella normalizzata non simmetrica.

5.5.1 Spectral clustering (L)

Lo spectral clustering presenta la miglior suddivisione degli utenti all'interno dei cluster in termini di numero, come riportato in Tabella 5.8 la gran parte dei cluster presenta una quantità elevata di utenti. Tuttavia questo non è sempre indice di una allocazione degli utenti migliore. Come è possibile vedere nella Tabella 5.14, i valori di GSI e ASI sono bassi, indice che molti utenti potevano essere allocati in modo migliore.

Sia dai centroidi che dalle distribuzioni dei decili all'interno dei cluster, i valori dei decili che compongono cluster diversi non sono del tutto diversi, come è possibile notare per i cluster 1,4 e 5 nel caso della TV oppure nei cluster 2 e 4 nel caso della LCL. Guardando le curve di carico giornaliere si può notare come non ci sia una netta differenza dal punto di vista delle ampiezze in alcuni cluster.

Tabella 5.13: Distribuzione degli utenti all'interno dei cluster.

Low Carbon London dataset		Thames valley dataset	
Cluster	numero di utenti	Cluster	numero di utenti
1	79	1	76
2	27	2	106
3	20	3	120
4	24	4	75
5	39	5	59
6	14	6	69

I centroidi in Figure 5.52 e 5.23 mostrano come la scelta della partizione non abbia una separazione tra cluster marcata.

Tabella 5.14: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.13	0.25	0.12	0.31

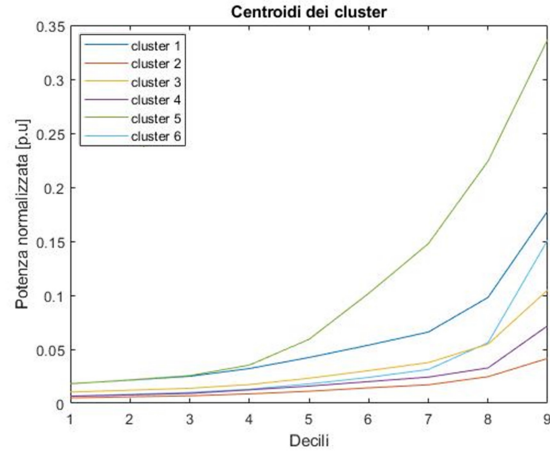


Figura 5.51: Centroidi del Thames Valley dataset.

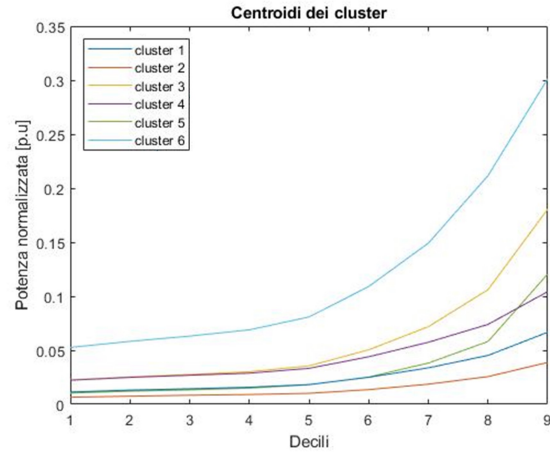


Figura 5.52: Centroidi del Low Carbon London dataset.

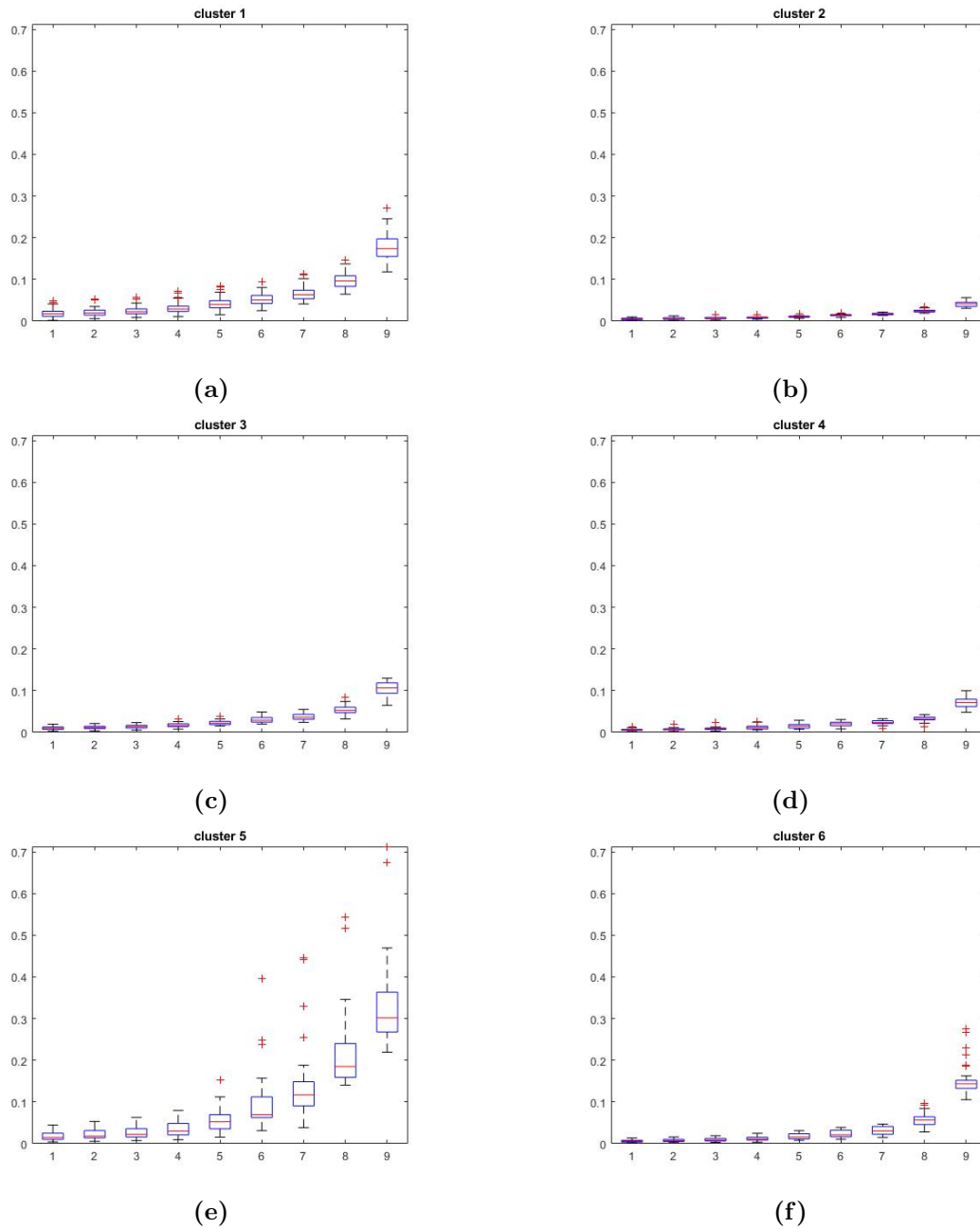


Figura 5.53: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

Guardando i decili del TV dataset notiamo la presenza di 3 cluster con decili molto simili (cluster 2, 3 e 4). Molto simile è anche la distribuzione dei decili del LCL dove i cluster 1 e 4 hanno dei decili poco differenziati.

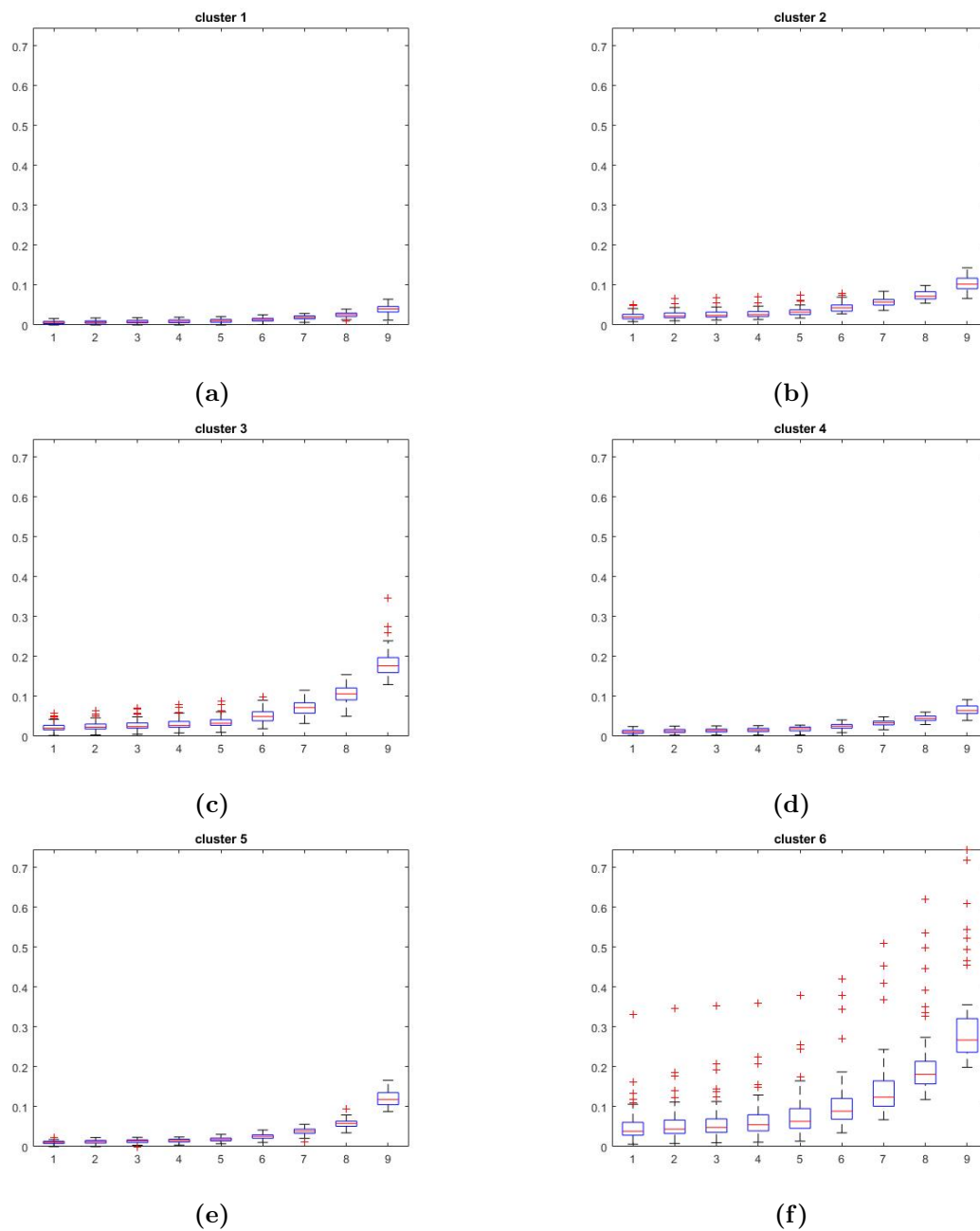


Figura 5.54: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

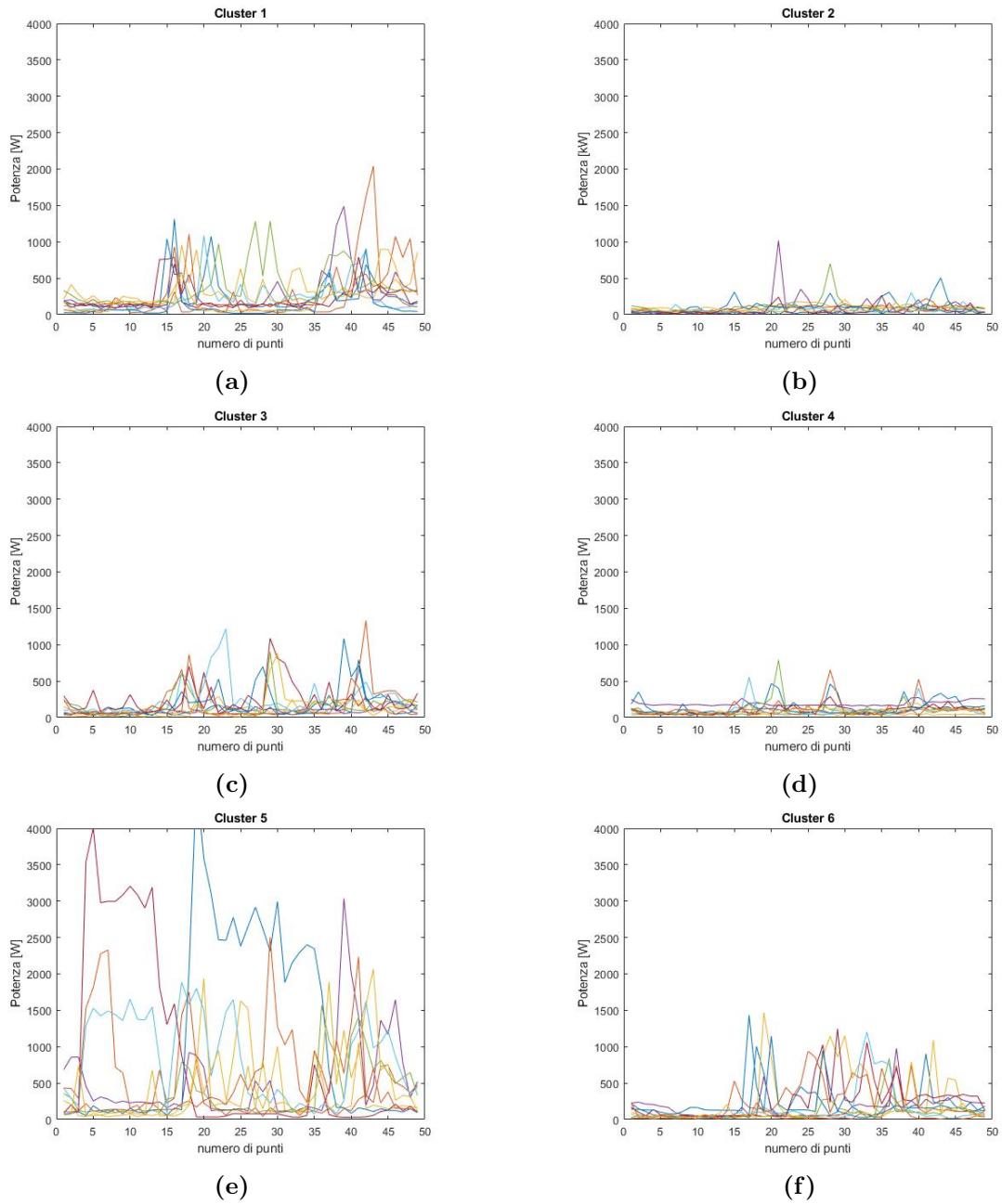


Figura 5.55: Curve di carico giornaliere per cluster, appartenenti a utenti diversi nel caso del Thames Valley dataset.

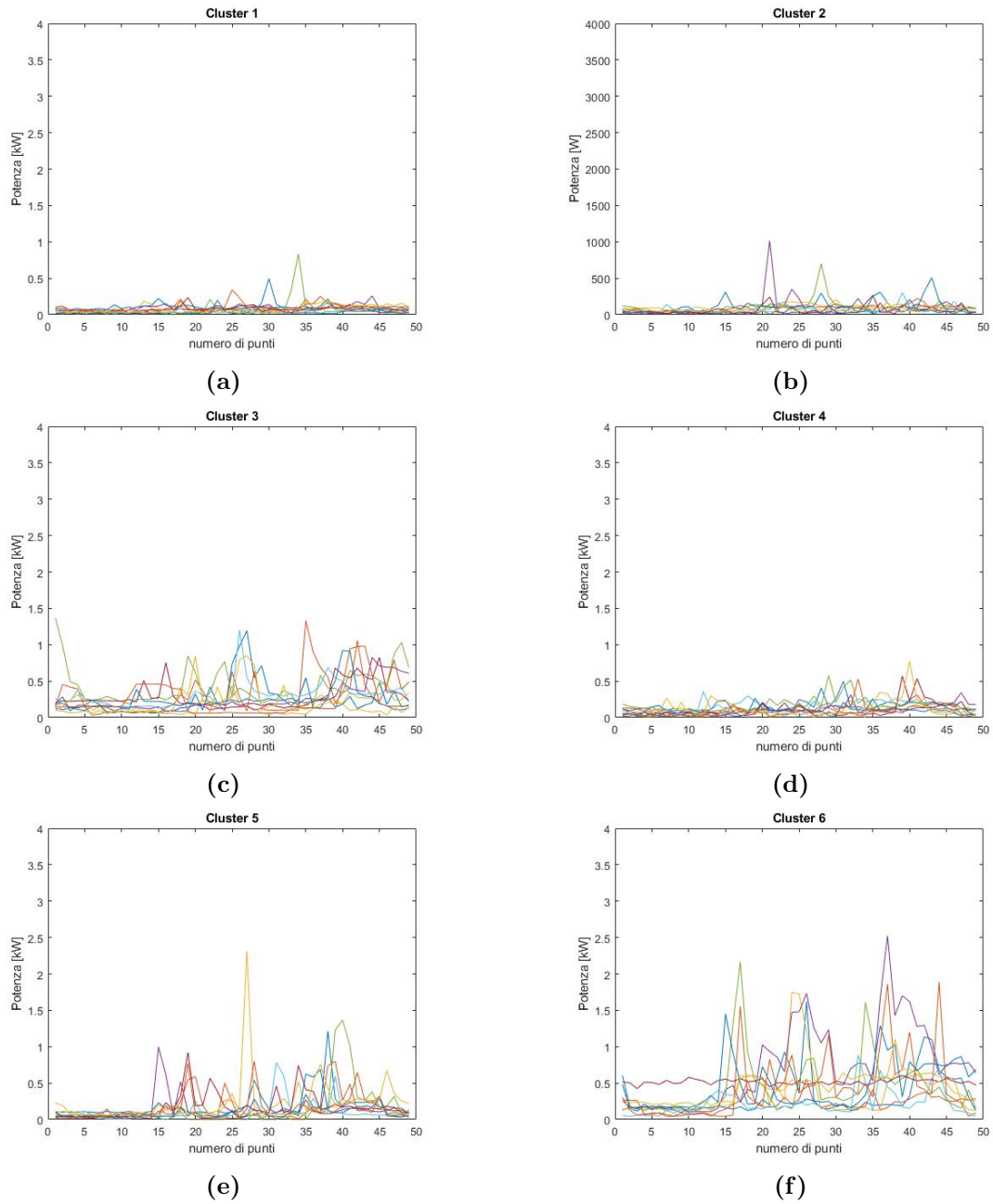


Figura 5.56: Curve di carico giornaliere per cluster, appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.5.2 Spectral clustering (S)

In questo caso abbiamo dato come input dell'algoritmo di clustering la distanza euclidea tra i decili rappresentativi di ogni utente, la quale poi è stata utilizzata per ricavare i pesi di similarità. Rispetto al caso precedente ritroviamo una situazione simile, in cui si ha una buona distribuzione degli utenti, tuttavia i valori degli indicatori di silhouette confermano una scarsa allocazione degli utenti, Tabella 5.16.

Tabella 5.15: Distribuzione degli utenti all'interno dei cluster.

Low Carbon London dataset		Thames valley dataset	
Cluster	numero di utenti	Cluster	numero di utenti
1	39	1	57
2	65	2	81
3	2	3	3
4	23	4	97
5	37	5	42
6	37	6	225

Tabella 5.16: GSI e ASI.

Thames valley dataset		Low Carbon London dataset	
GSI	ASI	GSI	ASI
0.14	0.33	0.15	0.22

Come per il caso precedente, i centroidi rappresentativi mostrano una elevata similarità nella maggior parte dei casi, Figure 5.57 e 5.58.

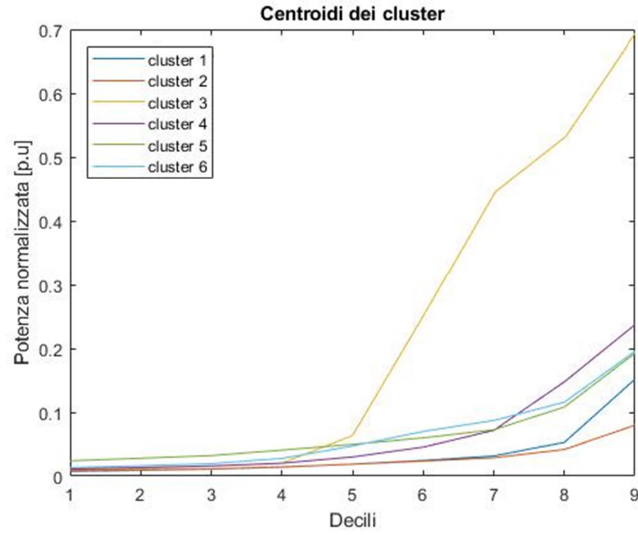


Figura 5.57: Centroidi del Thames Valley dataset.

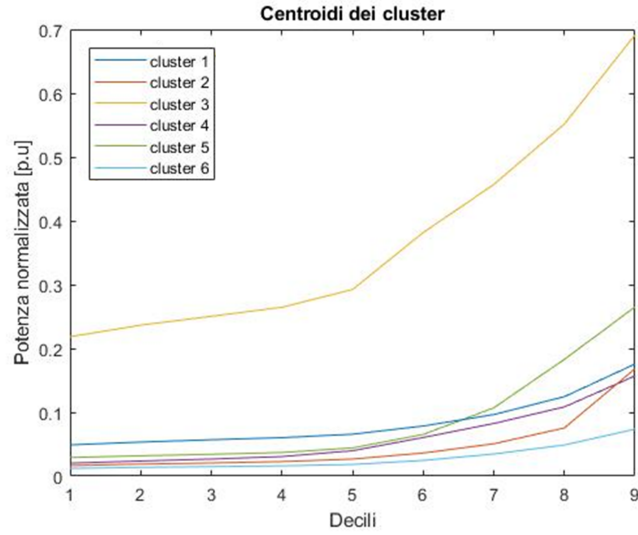


Figura 5.58: Centroidi del Low Carbon London dataset.

La somiglianza viene anche rispecchiata dalla distribuzione dei decili rappresentata dai boxplot, Figure 5.59 e 5.60.

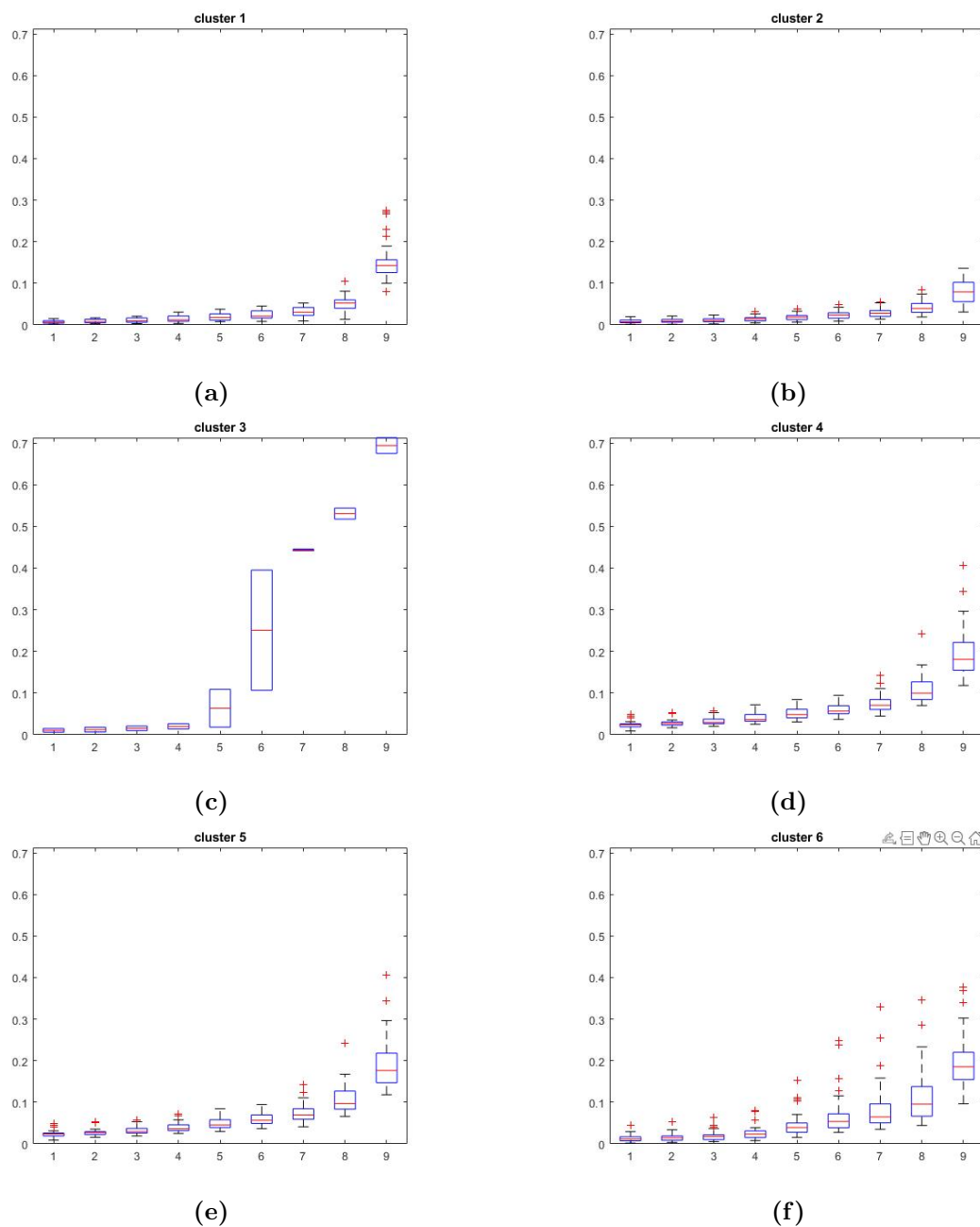


Figura 5.59: Boxplot dei 9 decili per i 6 cluster formati nel caso del Thames Valley dataset.

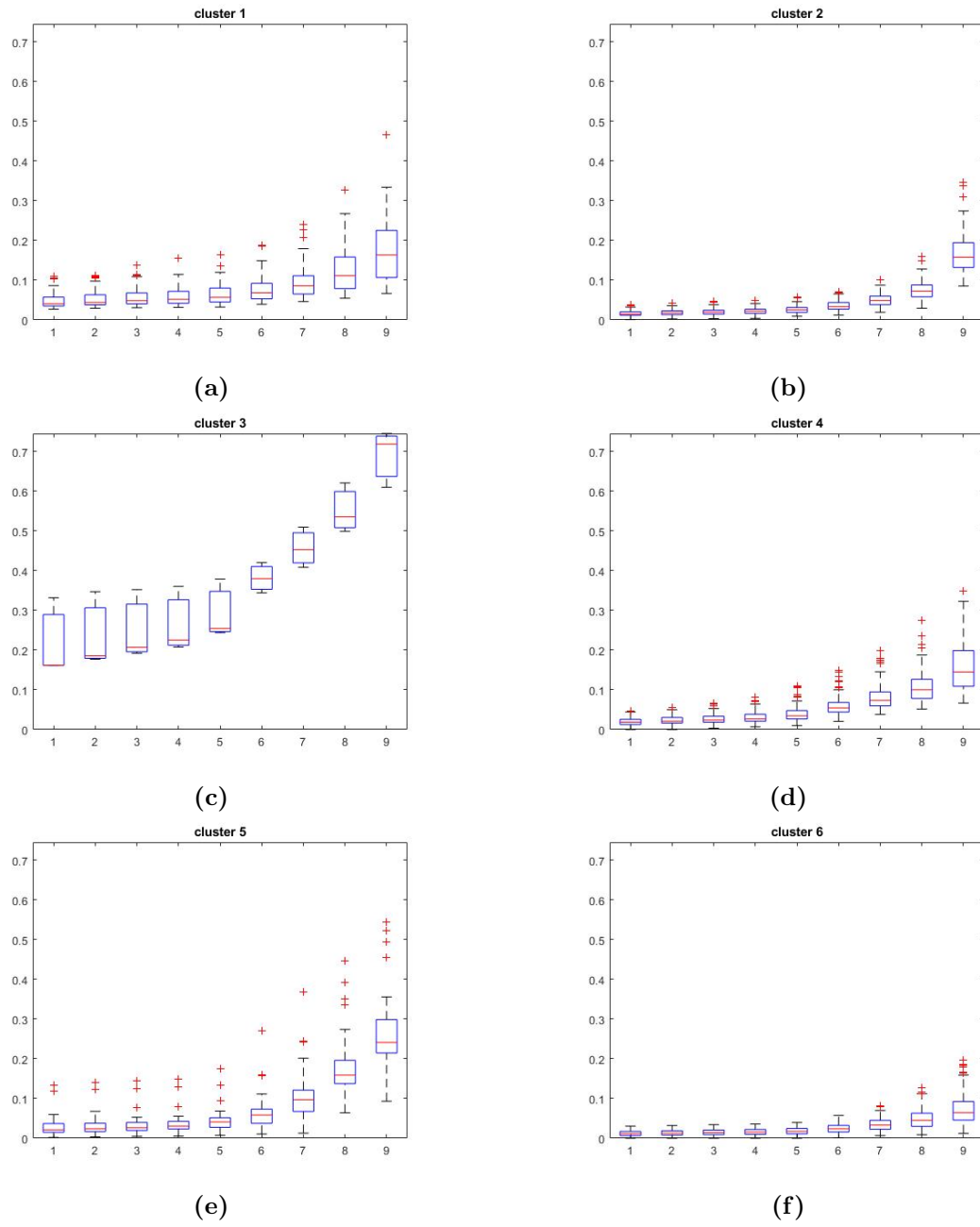


Figura 5.60: Boxplot dei 9 decili per i 6 cluster formati nel caso del Low Carbon London dataset.

Le curve di carico giornaliere, Figure 5.61 e 5.62 rispecchiano quanto detto prima.

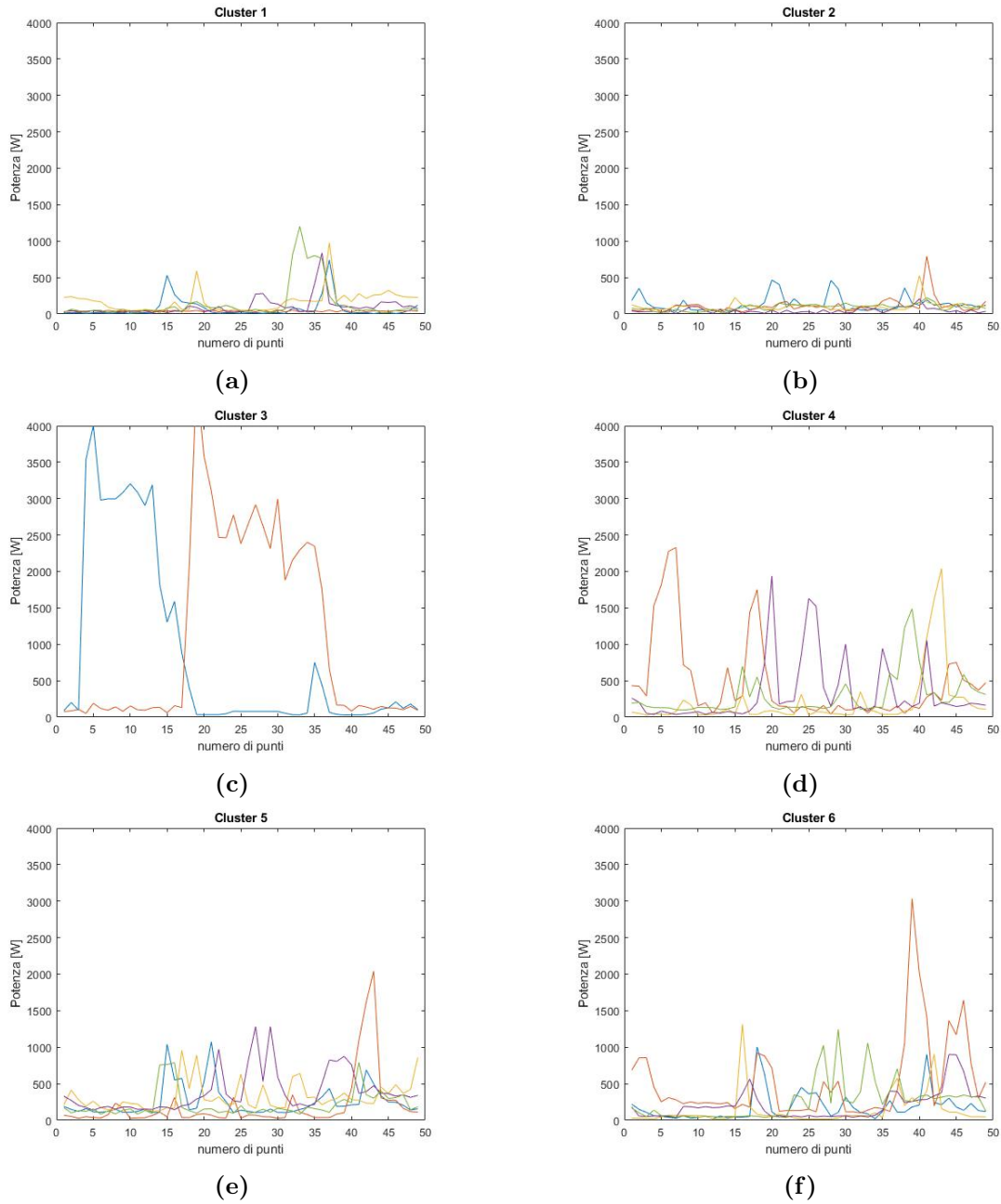


Figura 5.61: Curve di carico giornaliere per cluster, appartenenti a utenti diversi nel caso del Thames Valley dataset.

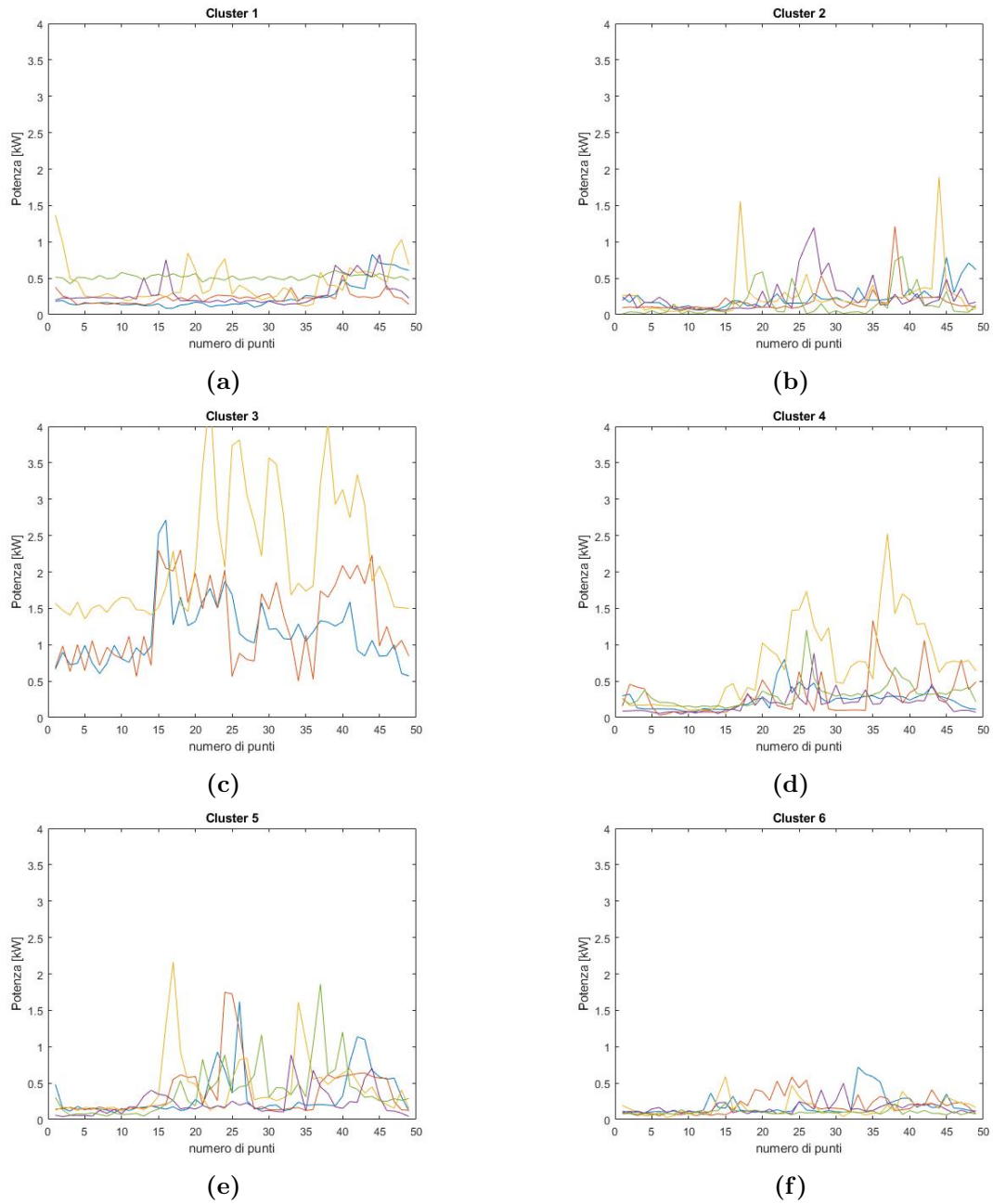


Figura 5.62: Curve di carico giornaliere per cluster, appartenenti a utenti diversi nel caso del Low Carbon London dataset.

5.6 Indici di validità

La suddivisione dei cluster vista precedentemente dà un'idea della qualità dei risultati, tuttavia è necessario utilizzare gli indici di validità, i quali danno una misura della compattezza e della separazione dei cluster. Gli indici utilizzati sono quelli riportati nel capitolo 1 e i cui valori per ciascun algoritmo di clustering sono riportati nella Tabella 5.17.

Tabella 5.17: Indici di validità.

Algoritmi	Thames valley dataset				Low Carbon London			
	MIA	CDI	SI	SMI2	MIA	CDI	SI	SMI2
K-means	0.0147	0.141	2.15	0.879	0.0218	0.144	2.94	0.891
Hs	0.0276	0.262	4.21	0.822	0.0332	0.292	8.64	0.810
Hc	0.0178	0.145	2.43	0.841	0.0258	0.166	4.11	0.853
Ha	0.0188	0.166	2.37	0.827	0.0253	0.176	3.33	0.869
Hwa	0.0136	0.127	2.03	0.887	0.0191	0.140	2.63	0.881
Hwe	0.0182	0.156	2.31	0.833	0.0261	0.149	3.33	0.847
Spec-L	0.0296	0.670	3,85	0.751	0.0249	0.688	5,21	0.803
Spec-S	0.0357	0.353	7.40	0.815	0.0416	0.320	9.60	0.792

Il K-means presenta dei valori degli indicatori molto buoni, infatti dalla suddivisione degli utenti si era vista una differenza abbastanza notevole tra i centroidi. Il clustering gerarchico con single linkage presenta i valori più alti in tabella per quasi tutti gli indicatori, in entrambi i dataset, confermando la scarsa qualità della suddivisione, infatti cinque su sei cluster presentavano solo un elemento.

Il clustering gerarchico con complete linkage, average linkage e weighted linkage avevano dimostrato un comportamento molto simile, infatti in tutti è tre i metodi c'è la prevalenza di un paio di cluster in termini di numero di utenti. I numeri degli indicatori confermano che oltre al numero di utenti anche la classificazione ha molte similitudini, infatti gli indicatori sono abbastanza vicini tra loro.

Lo spectral clustering con entrambi i dataset ha distribuito in modo abbastanza equilibrato gli utenti all'interno dei cluster, tuttavia come mostrato dagli indicatori di validità, l'equilibrio non rispecchia la qualità, infatti alcuni degli indicatori presentano i valori più alti ottenuti, come si era visto dai boxplot, i decili appartenenti agli utenti dei cluster non presentavano una differenziazione elevata.

Il clustering gerarchico con il Ward linkage, ha presentato i risultati migliori per la maggior parte degli indicatori, infatti i centroidi dei cluster erano molto diversi, in più è riuscito ad identificare utenti con consumi molto simili ma con picchi spostati nel tempo, che è uno degli obiettivi principali della metodologia CONDUCTS.

Capitolo 6

Conclusioni

In questa tesi sono stati analizzati i risultati di una metodologia innovativa di rappresentazione dei dati, detta CONDUCTS, applicata sugli algoritmi di clustering più tradizionali in modo da determinare quale tra questi algoritmi riporti una classificazione migliore degli utenti. Più in specifico gli algoritmi sottoposti all'analisi sono il *K-means*, il clustering gerarchico, e lo spectral clustering. I dataset utilizzati per l'analisi sono due, di dimensioni modeste, provenienti entrambi da progetti inglesi. Ogni dataset è stato prima sottoposto alla procedura di pre-clustering che CONDUCTS prevede, maggiore attenzione è stata data alla pulizia delle anomalie in entrambi i dataset in modo da avere dei dati più affidabili. I metodi utilizzati per questo compito hanno dato risultati abbastanza diversi, ma la scelta del metodo da utilizzare alla fine è ricaduta sulla analisi delle cumulative utilizzando il test statistico di Kolmogorov-Smirnov in quanto è stato molto costante nella identificazione degli outlier estremi, sia in caso di utenti con consumo molto variabile sia nel caso di utenti molto costanti.

Per quanto riguarda invece la scelta del numero di cluster abbiamo utilizzato l'indicatore proposto da CONDUCTS, somma degli errori al quadrato (SSE), per la maggior parte degli algoritmi questi indicatori presentavano una variazione minore attorno al numero di cluster uguale a sei, per cui ai fini della comparazione dei risultati abbiamo deciso di utilizzare quel numero di cluster per tutti gli algoritmi.

Dai risultati conseguiti, si può affermare che il clustering gerarchico con Ward linkage è quello che riporta i valori migliori per la maggior parte degli indicatori, questo può essere dovuto al modo in cui l'albero binario ha raggruppato gli utenti, con un ragionevole bilanciamento tra la creazione dei cluster uniformi e l'identificazione degli outlier.

Il metodo CONDUCTS considera i consumi degli utenti senza privilegiare la coincidenza nei tempi in cui i consumi avvengono. Per questo, può essere adatto all'identificazione di utenti con caratteristiche simili, ai quali possono essere proposte strategie di demand-response per lo spostamento dei consumi specifici. Con il metodo CONDUCT, anche spostando i consumi gli utenti non cambiano cluster.

Appendice

Le di distanze euclidee tra vettori e gruppi di vettori utilizzate sono riportate di seguito.

- Distanza due vettori, $\mathbf{x}$ e $\mathbf{y}$, aventi H elementi ciascuno.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{H} \sum_{h=1}^H (x_h - y_h)^2}$$

- Distanza tra un vettore $\mathbf{y}$ e un insieme $\mathbf{X}$, composto da M vettori.

$$d(\mathbf{y}, \mathbf{X}) = \sqrt{\frac{1}{M} \sum_{\mathbf{x} \in \mathbf{X}} d^2(\mathbf{y}, \mathbf{x})}$$

- Distanza tra gli M vettori membri di uno stesso insieme $\mathbf{X}$

$$d(\mathbf{X}) = \sqrt{\frac{1}{2M} \sum_{m=1}^M d^2(x^{(m)}, \mathbf{X})}$$

Dati due vettori $\mathbf{x}$ e $\mathbf{y}$, aventi n elementi ciascuno, le diverse definizioni di distanza, in alternativa alla distanza euclidea, possono essere:

- Distanza di Mahalanobis:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}$$

dove: $\mathbf{S}$ è la matrice di covarianza tra i due vettori.

- Distanza City block:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$$

- Distanza di Minkowski

$$d(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$$

- Distanza di Chebychev

$$d(\mathbf{x}, \mathbf{y}) = \max\{|x_i - y_i|\}$$

Bibliografia

- G. Chicco, R. Napoli and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," in *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 933-940, May 2006, doi: 10.1109/TPWRS.2006.873122.
- T. Xu, H. Chiang, G. Liu and C. Tan, "Hierarchical K-means Method for Clustering Large-Scale Advanced Metering Infrastructure Data," in *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 609-616, April 2017, doi: 10.1109/TPWRD.2015.2479941.
- T. Cerquitelli, G. Chicco, E. Di Corso, F. Ventura, G. Montesano, A. Del Pizzo, A. Mateo Gonzàles, E. Martin Sobrino, "Discovering electricity consumption over time for residential consumers through cluster analysis," 2018 International Conference on Development and Application Systems (DAS), 2018, pp. 164-169, doi: 10.1109/DAAS.2018.8396090.
- T. Cerquitelli, G. Chicco, E. Di Corso, F. Ventura, G. Montesano, M. Armiento, A. Mateo Gonzàles, A. Veiga Santiago, "Clustering-Based Assessment of Residential Consumers from Hourly-Metered Data," 2018 International Conference on Smart Energy Systems and Technologies (SEST), 2018, pp. 1-6, doi: 10.1109/SEST.2018.8495863.
- G. Chicco, "Overview and performance assessment of the clustering methods for electrical load pattern grouping", *Energy*, Volume 42, Issue 1, 2012, Pages 68-80, ISSN 0360-5442, <https://doi.org/10.1016/j.energy.2011.12.031>.
- K. Mets, F. Depuydt and C. Develder, "Two-Stage Load Pattern Clustering Using Fast Wavelet Transformation," in *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2250-2259, Sept. 2016, doi: 10.1109/TSG.2015.2446935.
- A. Notaristefano, G. Chicco, F. Piglione, "Data size reduction with symbolic aggregate approximation for electrical load pattern grouping," *IET generation, transmission and distribution*. 2013;7(2):108-117. doi:10.1049/iet-gtd.2012.0383.
- G. Chicco, R. Napoli, P. Postolache, M. Scutariu and C. Toader, "Customer characterization options for improving the tariff offer," in *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 381-387, Feb. 2003, doi: 10.1109/TPWRS.2002.807085.

- G. Muruti, F. A. Rahim and Z. bin Ibrahim, "A Survey on Anomalies Detection Techniques and Measurement Methods," 2018 IEEE Conference on Application, Information and Network Security (AINS), 2018, pp. 81-86, doi: 10.1109/AINS.2018.8631436.
- M. Sun, I. Konstantelos and G. Strbac, "C-Vine Copula Mixture Model for Clustering of Residential Electrical Load Pattern Data," in IEEE Transactions on Power Systems, vol. 32, no. 3, pp. 2382-2393, May 2017, doi: 10.1109/TPWRS.2016.2614366.

Siti web

<http://www.innovation.ukpowernetworks.co.uk>

<http://www.towardsdatascience.com>

<http://www.it.mathworks.com>

<http://statisticsbyjim.com>

Ringraziamenti

Ritengo sia doveroso dedicare le ultime righe di questo elaborato per ringraziare tutte le persone che mi sono state vicine e che in modi diversi mi hanno permesso di concludere il mio percorso universitario, soprattutto considerando gli avvenimenti degli ultimi due anni, i quali sono stati molto duri per tutti quanti. Ringrazio innanzitutto il mio relatore il professore Gianfranco Chicco per tutto il supporto e i consigli che mi ha fornito durante la stesura della tesi. Un ringraziamento speciale va ai miei più cari, che grazie ai loro incoraggiamenti, mi hanno permesso di superare molti ostacoli.

Infine vorrei esprimere tutta la mia gratitudine alla mia famiglia, senza la quale non sarei arrivato a questo traguardo importantissimo della mia vita.